

Statement and Confirmation of Own Work

Programme/Qualification:
BSc. (Hons) Business Computing and Information Systems

All NCC Education/University of Lancashire assessed assignments submitted by students must have this statement as the cover page or it will not be accepted for marking. Please ensure that this statement is either firmly attached to the cover of the assignment or electronically inserted into the front of the assignment.

Student declaration

I have read and understood both [University of Lancashire's Assessment Handbook](#) and **NCC Education's Referencing and Bibliographies** document (available on the [main course page of the VLE](#)). To the best of my knowledge my work has been accurately referenced and all sources cited correctly.

I can confirm the following details:

Student ID/Registration number: 200495 / G21344079

Name: Phone Myint Maw

Centre Name: Strategy First International College

Module Name: Information Systems Project (Final Report)

Module Leader: Steve Wade (steve.wade@nccedu.com)

Number of words: 7019

I confirm that this is my own work and that I have not colluded or plagiarized any part of it.

Due Date: 30.11.2025

Student Signature:



Submitted Date: 30.11.2025

Project Title: Job Application System for “Oshigoto”, a Job Agency from Japan

Abstract

The Oshigoto Japan System is a web-based job application system that is developed using a combination of Laravel, Tailwind CSS, and MySQL. The system provides a secure role-based access for admin and partner users, along with core features such as CV creation, CV management, job creation, job management, application processing, keyword-based filtering, PDF generation, and dashboard visualization. This system replaces the company’s manual processes with an organized, user-friendly workflow, which addresses the company’s requirements and provides a foundation for future digital transformation and enhancements.

Contents

Project Title: Job Application System for “Oshigoto”, a Job Agency from Japan	2
Abstract.....	2
Chapter 1 – Introduction, Aims and Objectives.....	8
1.1. Introduction.....	8
Company Background	8
Project Background	8
Justification	8
1.2. Aims and Objectives.....	8
1.3. Content Structure of Final Report.....	9
Chapter 2 - Research.....	11
2.1. Introduction.....	11
2.2. Literature Review	11
Recruitment Management System (RMS)	11
Comparison Between RMS and Oshigoto Japan System, and Evaluation	11
Conclusion	13
2.3 Competitor Analysis	14
Business-to-Business (B2B) Recruitment Management System.....	14
Comparison and Evaluation of Noborder, Manatal, and Oshigoto Japan Systems.....	16
Conclusion	17
2.4. Development Methodology.....	17
Introduction	17
The Agile Methodology	17
2.5 Technology Review of Chosen Technologies	18
Introduction.....	18
Backend Technologies – Laravel.....	18
Frontend Technologies – HTML, CSS, and Tailwind CSS	18
Third Party Integration	18
Database Management – MySQL	19
Development Environment and Tools.....	19
2.6. Conclusion.....	19
Chapter 3 – Analysis.....	20
3.1. Introduction.....	20
3.2. System Analysis	20
Intorduction	20
CATWOE Analysis.....	20
Functional and Non-Functional Requirement Analysis with MoSCoW Prioritization	20
Conclusion	21
3.3. System Design	22
Overview.....	22

Use Case Diagram	22
Sequence Diagram	27
3.4. Conclusion.....	28
Chapter 4 - Design.....	29
4.1. Introduction.....	29
4.2. System Architecture	29
4.3. Entity Relationship Diagram	30
4.4. Class Diagram.....	30
4.5. Data Design.....	31
Introduction	31
Users Table	31
Partners Table	31
C_V_Forms Table.....	32
Joblistings Table	34
Applications Table	35
4.6. Flowchart.....	36
Introduction	36
Partner Side Flowcharts	36
Admin Side Flowchart.....	38
4.7. Graphical User Interface	41
Introduction	41
Main Log in Screen Interface.....	41
Partner Side Interface.....	41
Admin Side Graphical User Interface	47
PDF File View User Interface	57
4.8. Conclusion.....	58
Chapter 5 – Implementation.....	59
5.1. Introduction.....	59
5.2. Page Navigation Map	59
5.3. Database Structure and Data Population	60
Database Structure using Migration and Model.....	60
Data Population with Factories and Database Seeder	65
5.4. Coding and Implementation	67
Authentication and Authorization	67
Blade Templates and components	73
CRUD Operations (Partner and Admin Users).....	80
Application Process	112
File Upload Functionality	118
Dashboard (Chart.js and Google Calendar)	119
Filtering (Keyword/Search)	122
PDF View Rendering	123

5.5. Risk Management	125
5.6. Problems Encountered	126
Database Configuration	126
Authentication Guard Conflicts	126
Migration and Seeding Errors	126
Create and Edit Errors	126
File Upload Storage Issue	126
PDF Generation Error	126
5.7. Data Migration	127
Creating Initial System Users	127
Generating Test and Demo Data	127
Manual Data Entry Through the System	127
5.8. User Manual	128
Partner Side Manual	128
Admin Side Manual	135
5.9. Project Management	145
Work Breakdown Structure	145
Gantt Chart	146
5.10. Training Schedule	147
Admin Side Schedule	147
Partner Side Schedule	148
5.11. Conclusion	149
Chapter 6 – Testing and Results	150
6.1. Unit Testing	150
Log In Error Test	150
Log In Test	151
Partner Create CV Error	152
Partner Create CV	153
Partner Edit CV	154
Admin Job Create Error	155
Admin Job Create	156
Admin Job Edit	157
Admin Partner Create Error	158
Admin Partner Create	159
Admin Partner Edit	160
Admin Dashboard Date Filter	161
Filteration Feature	162
6.2. Integrated Testing	163
Logging Out and Going Back	163
Partner Job Application and View Applied Job	164
Admin Applicant Evaluation	166

Chapter 7 – Conclusions and Future Improvements	168
7.1. Conclusion.....	168
7.2. Future improvements	168
Automated Email Notifications	168
Fully Integrated Scheduling	168
Partner-Based Application Record	168
Multi-language support	168
Cloud Storage.....	169
Admin Account Creation	169
References.....	170
Appendix	171
1. Development Flow using Agile Methodology	171
Planning.....	171
Requirement Analysis.....	171
Designing.....	171
Implementation	171
Testing	172
Deployment.....	172
2. CATWOE Analysis	173
3. Use Case Diagram Descriptions and Explanations	174
Overall Use Case Diagram	174
Detailed Use Case Diagram	174
4. Sequence Diagram Explanations	179
Partner Sequence Diagram	179
Admin Sequence Diagram.....	179
5. Entity Relationship Diagram Explanation	179
6. Class Diagram.....	179
7. System Architecture Explanation.....	180
8. CRUD Operations of Oshigoto Japan System Users.....	180
Create	180
Update	180
Retrieve	181
Delete	181
9. Application Operations of Oshigoto System Users	181
Partner Side.....	181
Admin Side	181
10. All Folders and Files.....	182
11. Source Codes.....	184
11.1. Migration Files	184
11.2. Factory Files for Defining Test Data Structure for Seeding	187
11.3. Database Seeder file for Seeding Test Data	189

11.4. Model files.....	190
11.5. Configurations.....	194
11.6. Components	195
11.7. Login	200
11.8. Partner	202
11.9. Admin.....	215
11.10. Routes	232
11.11. Controllers	233
11.12. PDF View.....	242

Chapter 1 – Introduction, Aims and Objectives

1.1. Introduction

Company Background

Oshigoto Japan is a small but rapidly growing Job Agency based in Sendai, Japan, that provides employment opportunities to workers from developing countries who wish to make a living in Japan. As Japan continues to face labour shortages in multiple sectors, the recruitment of skilled workers from foreign countries has become increasingly important for many industries. To respond to this demand, more and more Japanese language schools from different countries are willing to partner with Oshigoto Japan to find suitable employment positions for their students.

Managing jobs, CVs, and application records has become more difficult and complex to handle manually as the Oshigoto Japan continues to grow. The existing workflow relies on separate communication channels, manual document handling, and repetitive tasks. These create challenges in maintaining accuracy and efficiency.

Project Background

To address the challenges and keep up with the digital transformation, Oshigoto Japan has made a strategic decision to implement a web-based job application platform, which connects the international school partners with the agency for enhancing the flow of the entire recruitment process through a single online portal.

Justification

The implementation of the customized job application system will significantly improve Oshigoto Japan's data management and recruitment workflow. This digital solution will also enable the agency to maintain competitiveness in the recruitment sector, enhance partner communication and management of increasing volumes of applicants, and track all the application records, while also ensuring that it is not left behind by other companies, which have embraced similar technologies.

1.2. Aims and Objectives

The primary aim of the Oshigoto Japan System is to improve CV creation, job application, posting jobs, and keeping track of applications from the partners. The Partner side will be able to create CV forms and submit them to jobs, while the admin (agency) side will be able to create jobs, partners, and see and evaluate CVs from application records.

For the objectives, there are three main categories: business, technical, and personal objectives, which were previously outlined in the interim report.

Business Objective	<ul style="list-style-type: none"> - To save time in CV, partner, and Job listing management - To minimize repetitive manual tasks - To establish a consistent and centralized method for tracking Job Application Records
Technical Objectives	<ul style="list-style-type: none"> - To develop a user-friendly job application system - To implement features that allow users to create and edit CV forms, job listings, and partner accounts, and make CV applications for jobs
Personal Objectives	<ul style="list-style-type: none"> - To develop a professional IT project for the career portfolio - To learn new technologies and skills for future career development - To integrate knowledge and skills gained from other learning modules

Table: 1.2.1. Objectives of Oshigoto Japan System Project

1.3. Content Structure of Final Report

The final report is organized into 7 chapters, each covering key stages of the project's development.

Chapter 1 – Introduces the company background, the challenges, the project justification, and the aims and objectives that guide the system development.

Chapter 2 – Presents the research on the project, including an analysis of similar systems, a literature review, and the evaluations comparing the developed system and the similar systems.

Chapter 3 – Provides a detailed analysis of System Requirements and outlines system analysis methods, and Relevant Diagrams such as Use Case Diagrams, and data requirement models, made for the system development.

Chapter 4 – Focuses on GUI designs, Database Design, Flow Charts, System Architecture, and explanations of design decisions to address the identified problems

Chapter 5 – Explains the implementation of the Oshigoto Job Application System and describes how the system was developed, how various components were integrated, and how the chosen technologies are applied.

Chapter 6 – Presents the testing phase of the project, which describes the testing methods and the results.

Chapter 7 – Concludes the final report. The system is analyzed and evaluated to determine whether it meets Oshigoto Japan's needs and identifies the areas that can be implemented and improved in the future.

Chapter 2 - Research

2.1. Introduction

Chapter 2 presents the research undertaken for the development of the Oshigoto Job Application System. This includes the literature review on recruitment management systems, analysis of the existing job application platforms with functionalities similar to the proposed system, and evaluation of these solutions by comparing with the developed system. Additionally, this chapter includes the chosen project development methodology, along with the technologies chosen.

2.2. Literature Review

Recruitment Management System (RMS)

Gordon (2025) argues that a Recruitment Management System (RMS) is a type of software that is made for automating and standardizing the entire recruitment process. It allows recruiters to manage, handle, and simplify job listings, CV submissions, and communications. As the recruitment lifecycle has evolved from manual to modern digitalized and automated practices, having an RMS has become a strategic necessity (RecruitmentTraits, 2025).

According to Gordon (2025), an RMS solution has features such as Applicant Tracking System (ATS), automated job posting, automated CV submission, job and CV Search and filtering, interview scheduling, dashboards, reporting analytics, and secure personal information handling.

For a small job agency like Oshigoto Japan, RMS's functions provide significant benefits as they help manage partners, CV submissions, applications, and job listings, which are increasing day by day. Therefore, implementing RMS-inspired features to the Oshigoto Japan System addresses the organization's challenges.

Comparison Between RMS and Oshigoto Japan System, and Evaluation

Recruitment Management System (RMS) has a wide range of advanced functionalities used for automating, centralizing, and optimizing the entire recruitment process. However, the system's scopes and complexity can vary depending on the organizational scale. As the Oshigoto Japan System focuses on implementing the functions that align with Oshigoto Japan's operational needs, it is not required to fully replicate RMS's functions. The table below compares the features commonly found in a standard RMS platform with the implemented ones in the Oshigoto Job Application System.

Feature Comparison Between Typical RMS and Oshigoto Japan Job Application System

Features	Typical RMS Platforms	Oshigott Japan System	Implementation in the Oshigoto Japan System
Job Posting Management	Create, Update, and Automate posting to multiple platforms	Admin creates and manages job listings	Implemented
CV Submission and Job Application	CV Upload with required documents and certificates directly to job posts	Partners upload CV information and required files to the system and apply for the job	Implemented
Applicant Tracking System (ATS)	Full tracking through multiple stages (Screening, Interview, and Recruitment)	Basic Tracking of submitted CVs under Job posts with CV/Application status	Partially Implemented
Multi-User Role Assignment	Multiple Roles with Custom Permissions (HR, Recruiters, Managers, and Applicants)	Role-based Permissions (Admin and Partner)	Implemented
Interview Scheduling	Automated Schedules in Calendars	Not Implemented	Partially Implemented
Reporting and Analytics	Highly Detailed Dashboards, hiring metrics, and Performance Charts	Basic Dashboard showing Job, Partner, CV, and Application counts with a bar chart, pie charts, and simple monthly and yearly statistics	Simplified Implementation
Notification	Email alerts and reminders	Not Implemented	Not Implemented
Integration with External Platform	Linked to multiple channels such as LinkedIn and Job websites	Not Implemented	Not Implemented

CV Screening and Job Filtration	AI-based CV reading, matching, and keyword scoring	Basic Keyword-based Filtration	Partially Implemented
Data Security Features	Enterprise Level Encryption and Privacy Law Compliance	Basic Laravel Authentication and Validation	Basic Level Implementation

Table 2.2.1: Comparison Table for Typical RMS and Oshigoto Japan Job Application System

Implemented, Simplified, and Excluded Features

The Oshigoto Japan System is implemented with most of the features that are included in a Recruitment Management System. CV creation, file upload, job application, job posting, partner account management, search functions for keyword-based filtering, and a basic admin dashboard for displaying basic statistics are implemented. The system is also implemented with a simplified applicant tracking system (ATS), which allows administrators to view and evaluate submitted CVs. A role-based authentication is also implemented to restrict user access.

For the interview scheduling, the Google Calendar embed is implemented in the admin dashboard page. The admin can view the existing and upcoming schedules to check availability for adding new interviews. The link for the calendar can be changed to the admin's account, as the one added on the website is an example calendar, and the admin will have to use their own Google account to add interview schedules manually.

In contrast, several remaining advanced features, such as automated email notification, AI-based Screening, multi-platform integration, and deep analytics, were excluded from the current proposed system as these functions are highly complex and require extensive investment in time and research, and additional infrastructure support. More importantly, these exceed the organization's operational requirements, as Oshigoto Japan primarily manages applications and CVs from partnered schools rather than the general public.

Conclusion

The Oshigoto Japan System's core functions are closely aligned with the essential features of a standard Recruitment Management System. The comparison demonstrates that although the proposed system does not offer full capabilities as a commercial-use RMS, the exclusion of the advanced capabilities ensures that the system remains maintainable and well-suited for the organization's current scale. For the excluded features, they could be considered as future improvements for the system.

2.3 Competitor Analysis

Business-to-Business (B2B) Recruitment Management System

Introduction

For the development of the Oshigoto Japan System, an analysis of existing systems similar to Oshigoto Japan has been conducted. The Oshigoto Japan system is designed for a highly specific workflow, which involves partner language schools submitting applicants directly to a job agency. There are very few existing systems that possess similar functions. As these systems are only available for business use, their features are not fully shown to the public, and there is a limitation on fully analyzing these systems. In this section of the report, the two systems, Noborder Tokutei Management System, and Manatal staffing portal are examined.

Noborder Tokuteiginou (Specific Skilled Worker) Management System

Noborder is a cloud-based software for visa application workflow and HR management aimed to foreign workers under the Specific Skilled Worker program, which allows agencies to upload applicant information, required files, and documents, and track document compliance and visa application status (Noborder, n.d). Additionally, as monitoring of the well-being of workers and conducting periodic interviews with them to address any issues that arise, Noborder also provides tools for scheduling and recording interview activities.

Although Noborder's functions support some parts of the recruitment lifecycle, particularly documentation handling and compliance monitoring, it does not manage partner language schools, allow CV creation/partner submission, or provide posting job listing functions. the same way as the Oshigoto Japan System. So, while Noborder is aimed at documentation and visa application flow tracking and compliance, it is a relevant comparator for features such as profile uploading and tracking.



Figure 2.3.1: Home Page and its Functions of Noborder

Manatal Staffing Portal

Manatal is a cloud-based all-in-one solution designed for staffing agencies, which provides tools that support staffing professionals with various recruitment lifecycle, from sourcing candidates to final placement (Manatal, n.d). According to Manatal (n.d), it can source qualified candidates from third-party platforms such as LinkedIn and Indeed, automate job-matching and candidate scoring, track candidates through different stages, and monitor and manage client activities.

Manatal possesses many of the advanced features mentioned in the Recruitment Management System section above. As the functionalities extend far beyond Oshigoto Japan system's scope and requirement, only some of the features in Manatal are implemented and simplified in Oshigoto Japan. The functions include custom CV creation, basic applicant tracking, data-driven job matching, and manual candidate evaluation.

The image displays two screenshots of the Manatal website. The top screenshot shows the main landing page with the headline "Optimizing Staffing with Manatal". The page features a navigation bar with the Manatal logo, language selector (EN), and menu items: Features, Solutions, Resources, Integrations, Pricing, Sign In, Book a Demo, and Start for Free. The main content area includes a testimonial from Eleanor Pena, Lead Recruiter at Talent Match Staffing, and a call to action to "Start 14-day Free Trial" and "Book a Demo". A small chart shows the "No. of placements" for Talent Match Staffing. The bottom screenshot shows a section titled "The All-in-One Solution for Staffing Agencies" with buttons for Sourcing Center, AI Recommendations, Applicant Tracking, and Recruitment CRM. Below this, there is a section for "Maximize your sourcing efforts" with a Chrome extension interface showing job listings from LinkedIn, Indeed, and Manatal.

Figure 2.3.2: Home Page and its Functions of Manatal

Comparison and Evaluation of Noborder, Manatal, and Oshigoto Japan Systems

Both Noborder and Manatal offer features related to recruitment management systems and Oshigoto Japan’s industry, but neither of them fully aligns with Oshigoto Japan’s. Noborder offers information and documentation tracking, which specializes in Visa processing and compliance, making the features unsuitable for the Oshigoto Japan System to fully implement. For Manatal, as it provides Application Tracking System (ATS) features such as automation, AI usage, and third-party integration, this exceeds the project scope. The table below compares the implemented features and functionalities of each of the three systems.

Feature Comparison between Noborder, Manatal, and the Oshigoto Japan System

Feature / Functionality	Noborder	Manatal	Oshigoto Japan System
Primary Purpose	Visa Documentation, Compliance Tracking, Worker Monitoring	Full Recruitment Management System support for Staffing Agencies	Partner CV Submission and Job Management
Targeted Users	Agencies managing Specified Skilled Workers	Staffing Agencies, HR Professionals	Oshigoto Japan Staff and partner language schools
CV / Profile Upload	Implemented. Applicant Profile and Document Upload	Implemented. CV parsing and candidate profiles	Implemented. CV form creation and file upload from partners
Job Posting	Not Implemented, Not focused on job listings	Implemented. Multi-channel job posting	Implemented, Admin creates job listings
Application Tracking	Implemented, Document / Compliance/ Visa progress tracking	Implemented. Full candidate tracking through the recruitment lifecycle	Implemented. Basic application tracking
Document Management	Implemented. Strong compliance documentation workflow	Implemented, File attachments supported	Implemented. Basic CV upload with files attached by partners
Interview Scheduling	Implemented. Worker monitoring and interviews	Implemented, Automated Scheduling	Implemented. Manual Scheduling via Google Calendar

Automation / AI Features	Not Implemented, Focused on Compliance	Implemented. Auto Job matching and candidate scoring	Not Implemented, Out of Scope
System Complexity	Medium, Specialized Compliance Workflow	High, Full ATS capabilities	Low, tailored to specific needs only
External Users	None, only for Agency use	None, only for Agency use	Partner School Accounts with restricted access
Alignment with Proposed System	Partially Aligned (Document Uploading and Tracking)	Partially Aligned (CV and job posting, and applicant tracking)	Aligned, built specifically for Oshigoto Japan

Table 2.3.1: Function Comparison of Noborder, Manatal, and Oshigoto System

Conclusion

The comparison shows that while Noborder and Manatal provide relevant features, none of them fully meet Oshigoto Japan's specialized needs. Noborder is fully specialized in compliance tracking, and Manatal offers advanced ATS functionalities. In contrast, Oshigoto Japan System combines and simplifies CV creation and submission, job posting, and application tracking from these two systems to create a custom solution tailored to Oshigoto Japan's requirements.

2.4. Development Methodology

Introduction

As stated in the interim report, the Agile Methodology is used for the development of the Oshigoto Japan System. It ensures flexibility, iterative development, and allows continuous upgrade and refinement throughout development.

The Agile Methodology

The Agile Methodology is made up of six stages: Planning, Requirement Analysis, Design, Implementation, Testing, and Deployment, which go in a circle (GeeksForGeeks, 2025). As the project is managed and developed by one person, some stages were simplified to suit the workflow, while maintaining the key principles.

Agile Methodology allows building and refining each feature step-by-step. As the system requires continuous testing and refining for the UI and multiple features, and the requirements for the system were uncertain from the beginning, the development needed to be carried out iteratively.

The core six stages mentioned above are followed continuously throughout the project based on evolving requirements. The details of the step-by-step stages are described in the appendix section.

2.5 Technology Review of Chosen Technologies

Introduction

The Oshigoto Japan System was developed by using a combination of different web development technologies, frameworks, and tools. This section discusses all the key technologies and tools that were used for the development of the project.

Backend Technologies – Laravel

The Oshigoto Japan System was developed using the Laravel framework, which is built on PHP. Laravel follows the Model-View-Controller (MVC) architecture, which separates application logic into three components: Model, View, and Controller. It promotes efficient and well-organized coding by allowing each component to be developed independently. Laravel also simplifies backend tasks such as routing, database migration, session management, and user authentication. Additionally, its blade templating and eloquent Object-Relational Mapping improve database interactions and UI renderings, making it useful for both frontend and backend integration.

Frontend Technologies – HTML, CSS, and Tailwind CSS

HTML, CSS, and Tailwind CSS were used for the frontend development of the Oshigoto Japan System. As HTML is the standard language for structuring web pages, it is used as the main language for laying out the pages.

For styling, Tailwind is the styling framework. In contrast to Bootstrap, which only offers pre-built UI components, and traditional CSS, which requires manual coding, Tailwind allows low-level styling for HTML pages just by using utility classes directly in the HTML elements. Due to simplicity, flexibility, and responsiveness, Tailwind was chosen as the primary styling framework for Oshigoto Japan System.

For traditional CSS, it is used for rendering webpages like CVs and jobs into a viewable PDF format, as Tailwind does not support styling in PDF-rendered webpages.

Third Party Integration

For data visualization, Chart.js, an open-source JavaScript library that allows interactive and responsive graph integration, was used for displaying statistics on the admin dashboard page. In the Oshigoto Japan System, Chart.js was used for displaying statistics such as the total number of CVs, jobs, CV submissions, registered partners, and passed CVs in bar charts and pie charts.

For managing schedules and tracking important events, the Google Calendar was integrated as it provides a visual and convenient way to monitor and manage upcoming interviews and schedules. The calendar was implemented through an embedded Google Calendar component.

While the Google Calendar embed does not allow direct control of schedules, it allows a real-time synchronized view of the monthly view of events within the calendar.

Database Management – MySQL

For the database management system, MySQL was used for data storage and management. It is a widely used open-source relational database, known for its reliability and performance in handling data. Additionally, MySQL is compatible with the Laravel framework through the Eloquent ORM, which allows seamless database interaction and simplifies query operations.

Oshigoto Japan System's database is designed to store only the essential data of the system, such as CV records, job listings, partner information, and application details. MySQL's relational model ensured that the data is well-maintained and ensured integrity through defining relationships between tables, which reduces redundancy and ensures accuracy.

Development Environment and Tools

Visual Studio Code, XAMPP, and Google Chrome were the primary tools that were used throughout the development of the Oshigoto Japan System, each serving different purposes in the project.

Visual Studio Code is an integrated development environment (IDE) tool that is widely used for software development. Due to its extensive plugin support and compatibility with multiple programming languages, it was chosen as the main tool for the coding part of the project.

XAMPP was used for hosting the system on a local server. It provides an Apache server and MySQL integration, which allows the system to be developed and tested on the browser in the local device before deployment.

For testing the functionality and responsiveness of the system, Google Chrome was used as a primary browser. It was used for testing features, user interfaces, and layout consistency across different screen sizes.

2.6. Conclusion

Chapter 2 explored the research that is necessary for the development Oshigoto Japan System. The literature examined the Recruitment Management System concepts, and the competitor analysis explored the real-world solutions for the recruitment companies. The evaluation made by comparing the competitors and the Recruitment Management System demonstrated that while commercial RMS offered advanced features, the Oshigoto Japan System required only focused features tailored to its school-agency workflow. The review on development methodology justified the selection of the Agile approach, and the technology review ensured the suitability of the chosen tools and technologies.

Chapter 3 – Analysis

3.1. Introduction

Chapter 3 presents the system analysis and design elements that are used in the development of the Oshigoto Job Application System. In this chapter, the system analysis, along with the initial system diagrams and data designs that form the foundation of the system development, is outlined with initially designed key artefacts such as use case diagrams, sequence diagrams, data models, and user interface structures. As most of these artefacts were originally developed in the interim report, some were later refined and expanded during the implementation phase.

3.2. System Analysis

Introduction

To identify and understand the information and system requirements of the company, the CATWOE analysis and the functional and nonfunctional analysis were conducted, as initially presented in the interim report.

CATWOE Analysis

The CATWOE analysis was used to identify different viewpoints of the parties and stakeholders who are involved with Oshigoto Japan. As described in the interim report, the stakeholders include the local school partners, the agency staff, the owner/founder of Oshigoto Japan, and the local government. The details will be provided in the Appendix section.

Functional and Non-Functional Requirement Analysis with MoSCoW Prioritization

For functional and non-functional requirement analysis, the detailed project and system requirements for the end users were examined. To determine their importance in the system, each requirement is ranked using the MOSCOW prioritization method, categorizing requirements as Must Have, Should Have, Could Have, and Won't Have. The following lists are the shortened versions of all the requirements with their prioritization levels.

MoSCoW Prioritization with Functional Requirements

<p>Must Have</p>	<ul style="list-style-type: none"> - CV Creation and Management (Partner) - Job Creation and Management (Admin) - CV Submission to Job Listings (Partner) - CV, Job, and Application view (Admin and Partner) - User Authentication (Admin and Partner) - Role-based Access Control (Admin and Partner) - CV (Applied) Evaluation (Admin) - Dashboard Statistics (Admin)
<p>Should Have</p>	<ul style="list-style-type: none"> - Search and Filtering Feature (Admin and Partner) - PDF Viewing for CVs and Jobs (Admin and Partner)
<p>Could Have</p>	<ul style="list-style-type: none"> - Automated Email Notification - Interview Scheduling Features - More detailed Analytics and Reporting
<p>Wouldn't Have</p>	<ul style="list-style-type: none"> - AI-based CV Screening - Integration with Third-party platforms - Mobile Application Version

Table: 3.2.1. Functional Requirement Ranked with MoSCoW Prioritization Method

Non-Functional Requirements (All Considered Must-Haves)

- User-Friendliness
- Reliability
- Compatibility
- Security
- Performance
- Maintainability

Conclusion

Both functional and non-functional requirement analysis, and the CATWOE analysis were made in the interim report and as some requirements were refined and changed throughout development of the project. The updated versions of the analyses and their details will be presented in the appendix section.

3.3. System Design

Overview

In the System Design section, the key design artefacts, including use case modelling, sequence diagrams, flow charts, entity relationship diagrams, and data design, are presented. These elements were created before implementation, and they define how the users interact with the system, how the system behaves, and how the underlying database is structured.

Use Case Diagram

A Use Case Diagram is a visual representation that illustrates the functional requirements of the system by showing the relationships between different actors (users) and the system's different use cases (Visual Paradigm, n.d). Although it does not show the sequence or order of how the system works, it clearly defines the expected behaviour of the system and the roles of different users within it.

For the Oshigoto Japan System, as previously mentioned in the interim report, there are 2 main actors who use the system: the language school side Partner actor and the agency side Admin actor. The initial use case diagrams were presented in the interim report. As some of the use cases were refined to meet the changing requirements, the updated version of the overall and the detailed use case diagrams is presented below.

The explanations and descriptions of the diagrams will be presented in the appendix section.

Overall Use Case Diagram

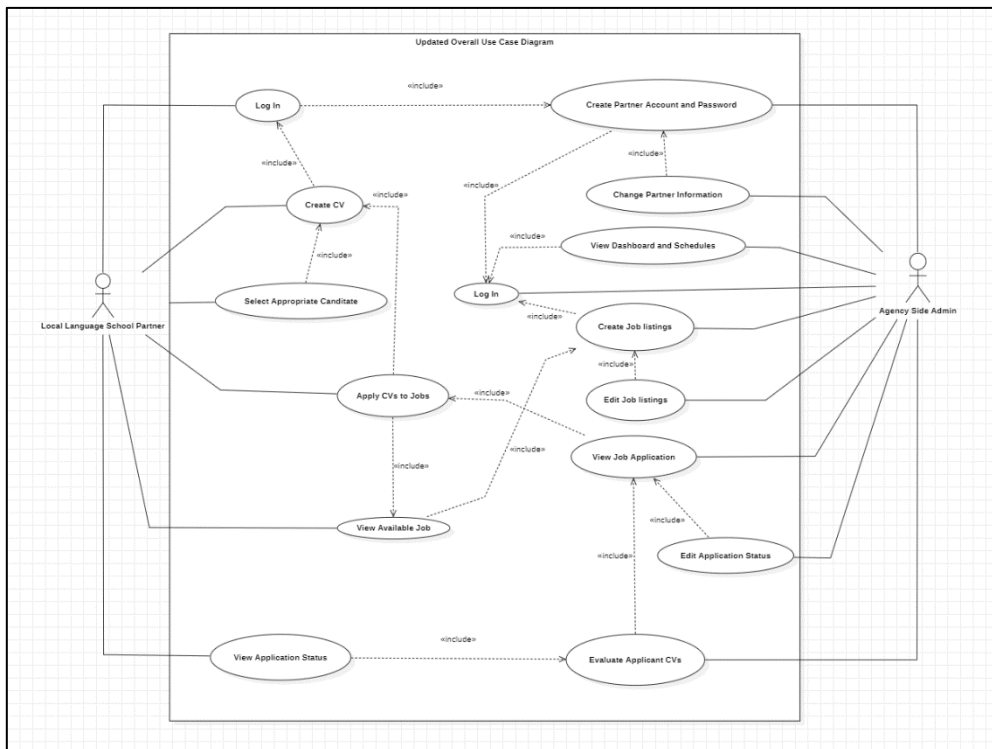


Figure: 3.3.1. Overall Use Case Diagram of Oshigoto Japan Job Application System

Detailed Use Case Diagram – Partner CV Creation

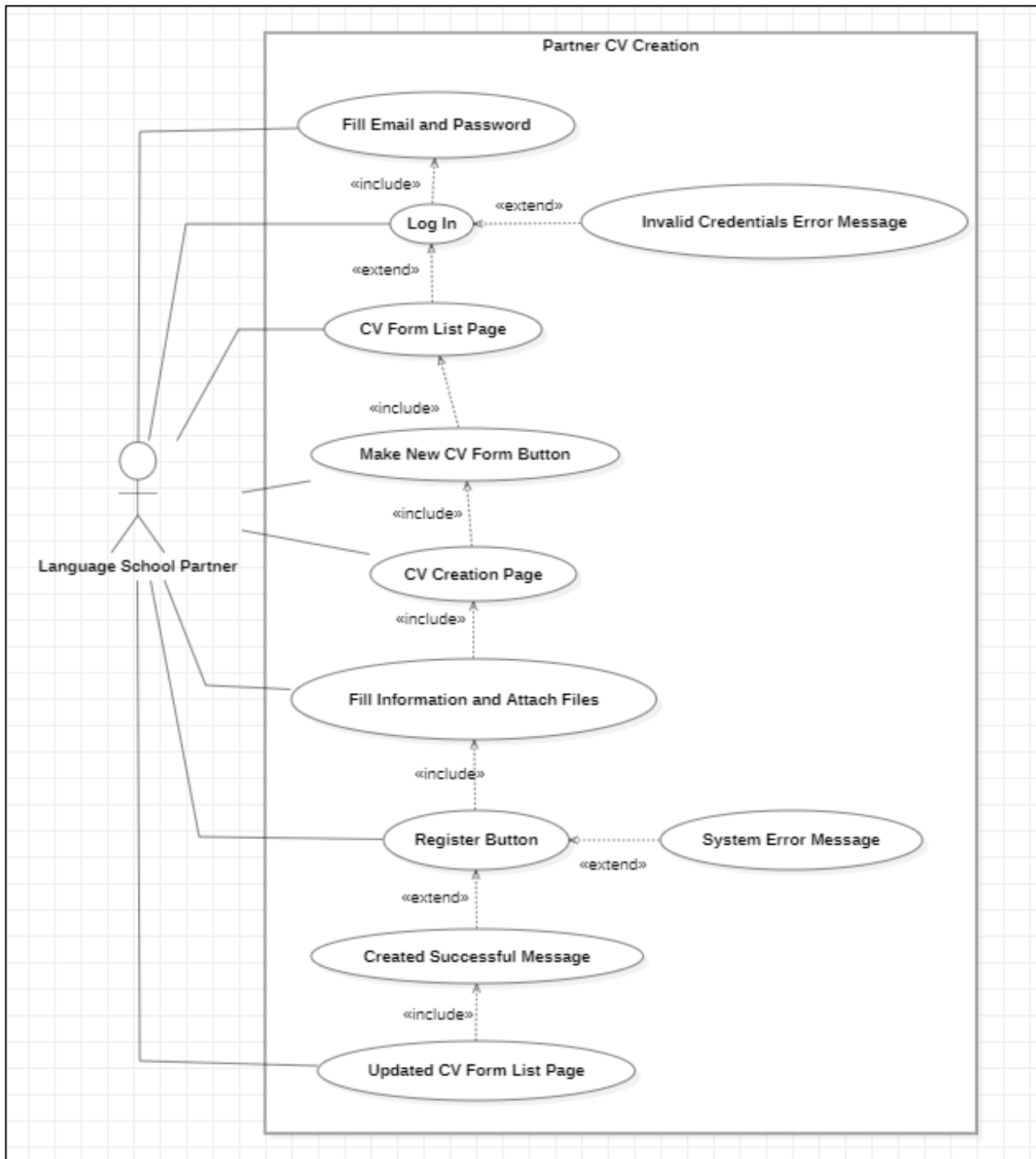


Figure: 3.3.2. Detailed Use Case Diagram of Partner Side CV Creation

Detailed Use Case Diagram – Partner Job Application

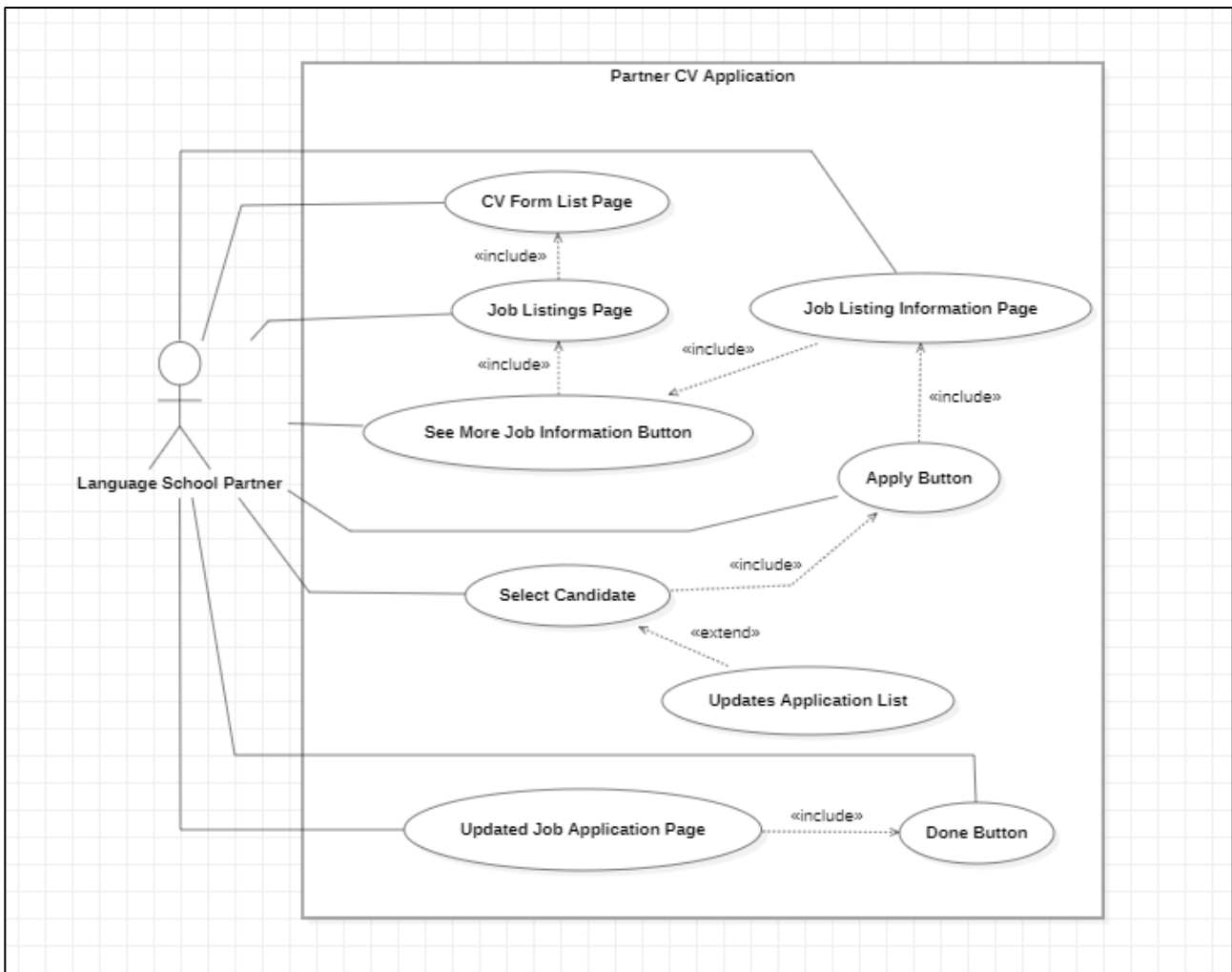


Figure: 3.3.3. Detailed Use Case Diagram of Partner Side Job Application

Detailed Use Case Diagram – Admin Job Creation

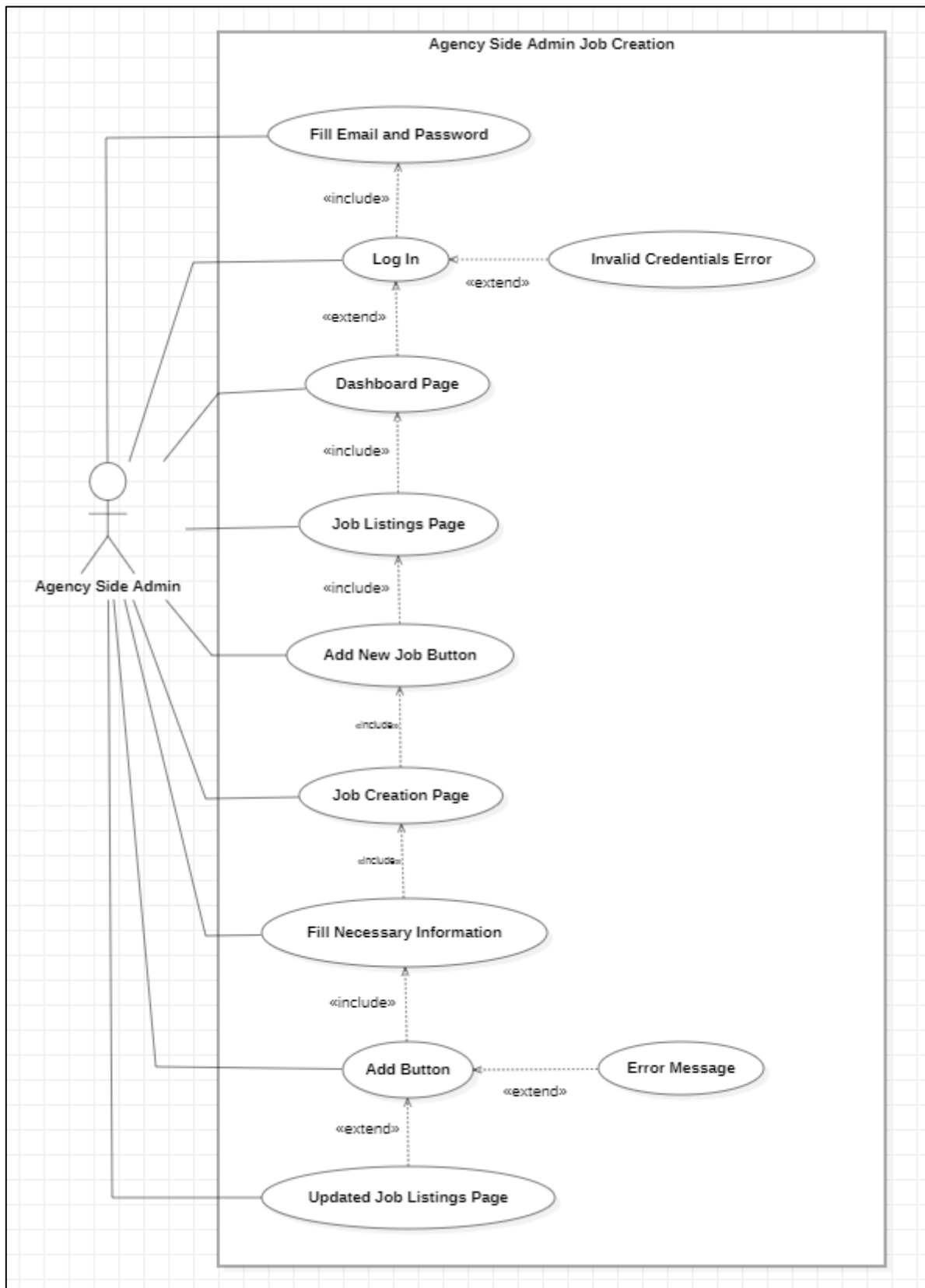


Figure: 3.3.4. Detailed Use Case Diagram of Admin Side Job Creation

Detailed Use Case Diagram – Admin Partner Creation

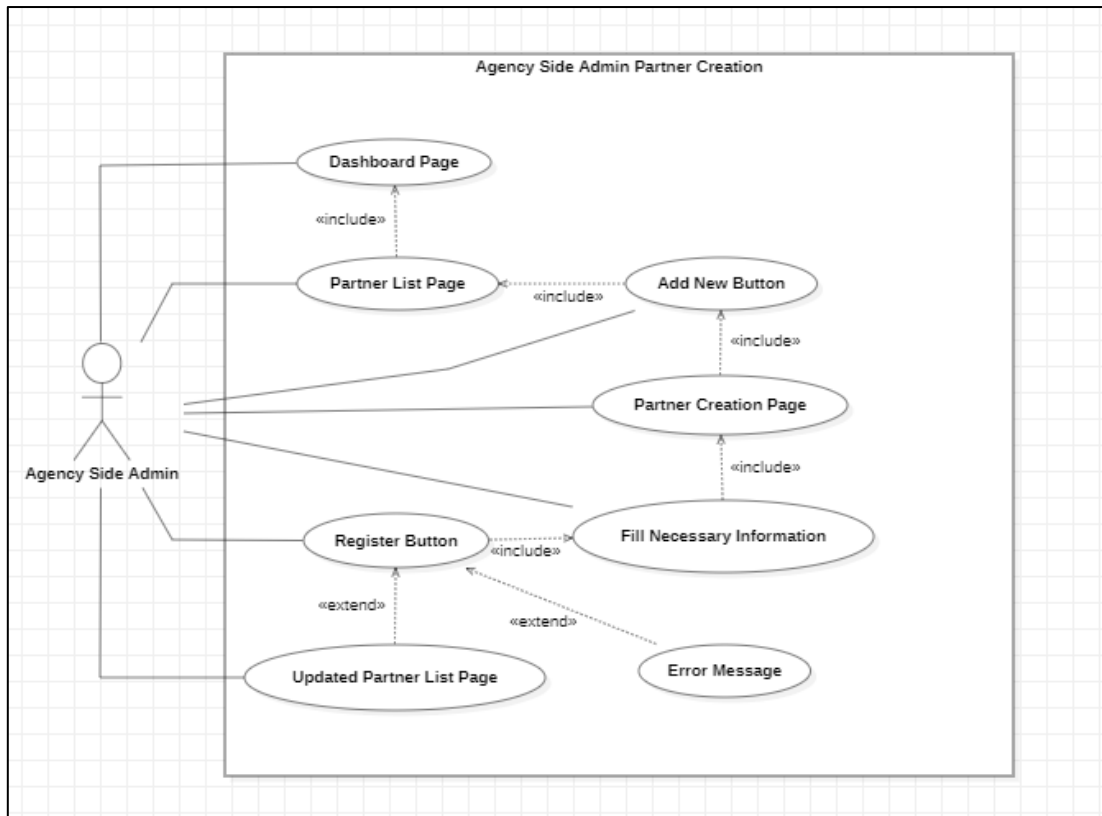


Figure: 3.3.5. Detailed Use Case Diagram of Admin Side Partner Creation

Detailed Use Case Diagram – Admin Applicant CV Evaluation

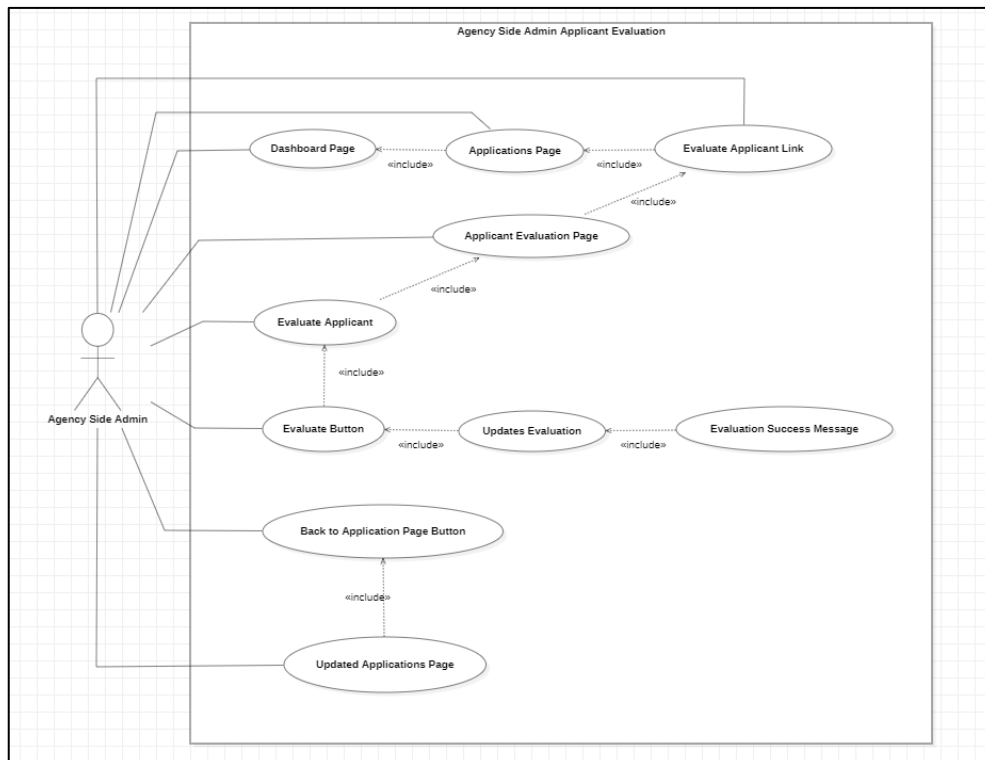


Figure: 3.3.6. Detailed Use Case Diagram of Admin Side Application Evaluation

Sequence Diagram

A sequence diagram is a type of Unified Modelling Language (UML) diagram that represents the details of how the operations in the system are carried out by the user by capturing interactions between different objects (Visual Paradigm, n.d). In this section, the 2 sequence diagrams, which illustrate the overall main workflows of 2 users, the partner and the admin, are presented. These are the refined versions of previously presented sequence diagrams, which were described in the interim report. The rest of the explanations will be presented in the Appendix section.

Partner Workflow Sequence Diagram

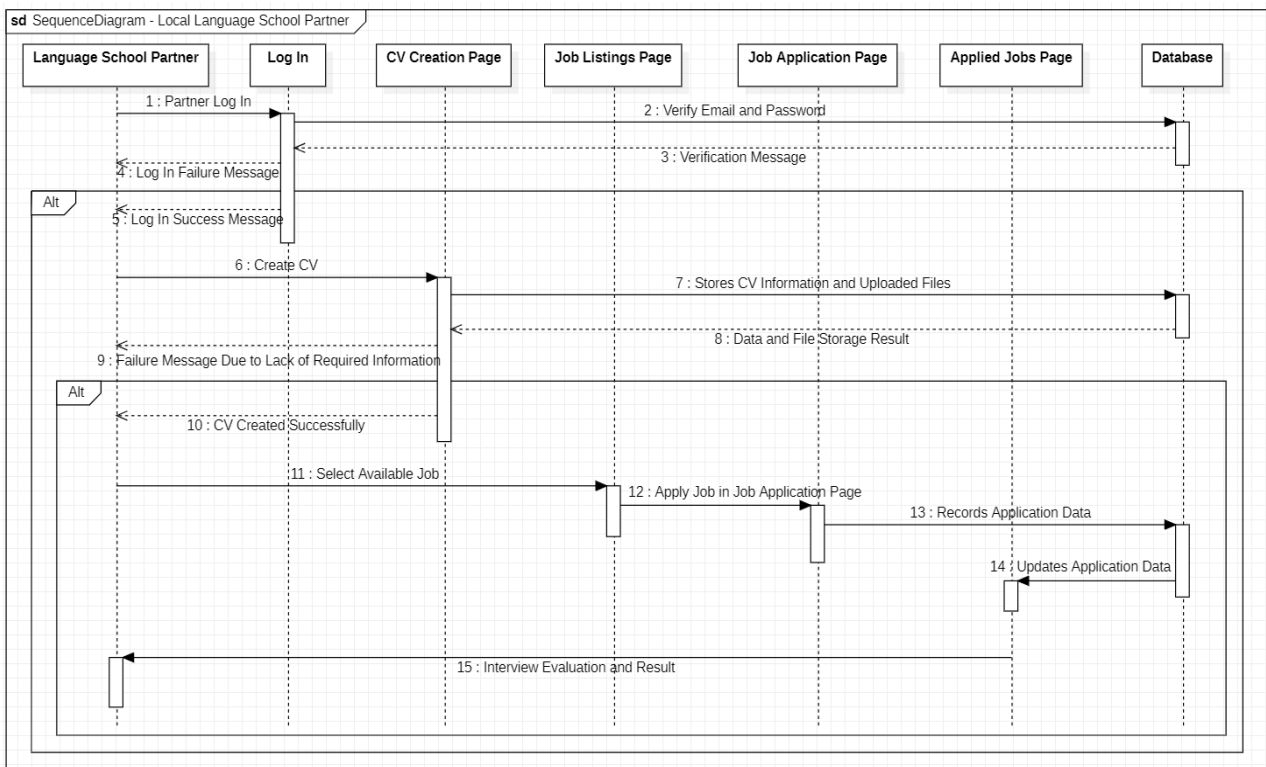


Figure: 3.3.7. Sequence Diagram of Partner Workflow

Admin Workflow Sequence Diagram

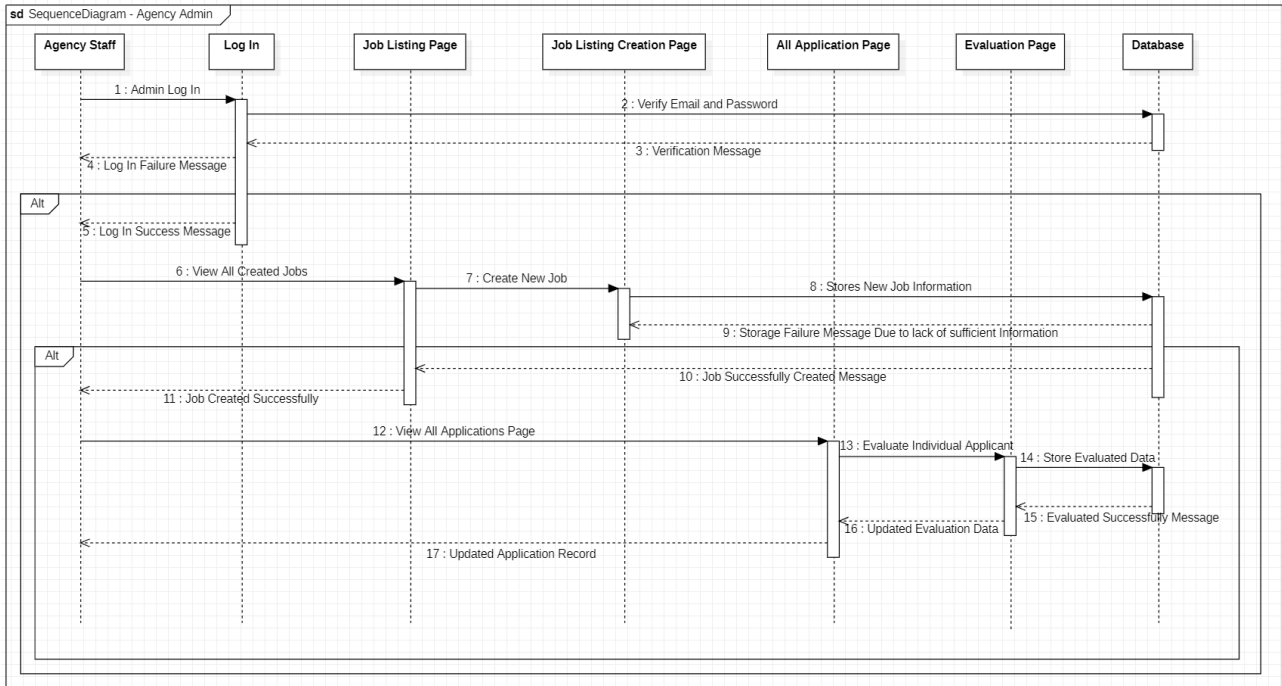


Figure: 3.3.8. Sequence Diagram of Admin Workflow

3.4. Conclusion

Chapter 3 presented the system analysis and fundamental design artefacts. The CATWOE analysis, and the functional and non-functional requirements helped clarify the system’s purposes, the stakeholders and their needs, and the essential capabilities to solve the challenges of Oshigoto Japan. The design artefacts, including use case diagrams and sequence diagrams, established the structure to understand system behaviour and user interactions. Together, both the analyses and designs served as the foundation for the system development, which aligns with the project’s objectives.

Chapter 4 - Design

4.1. Introduction

The Design Chapter presents the system designs of the Oshigoto Japan System. They include system architecture, user interface design, data design, flowcharts, and a navigation map of the system. These were developed before the implementation, and some designs were refined during the development process through iterative development.

4.2. System Architecture

The Oshigoto Japan System is built upon three layers: Presentation, Application, and Data Layer. The explanation will be presented in the Appendix section.

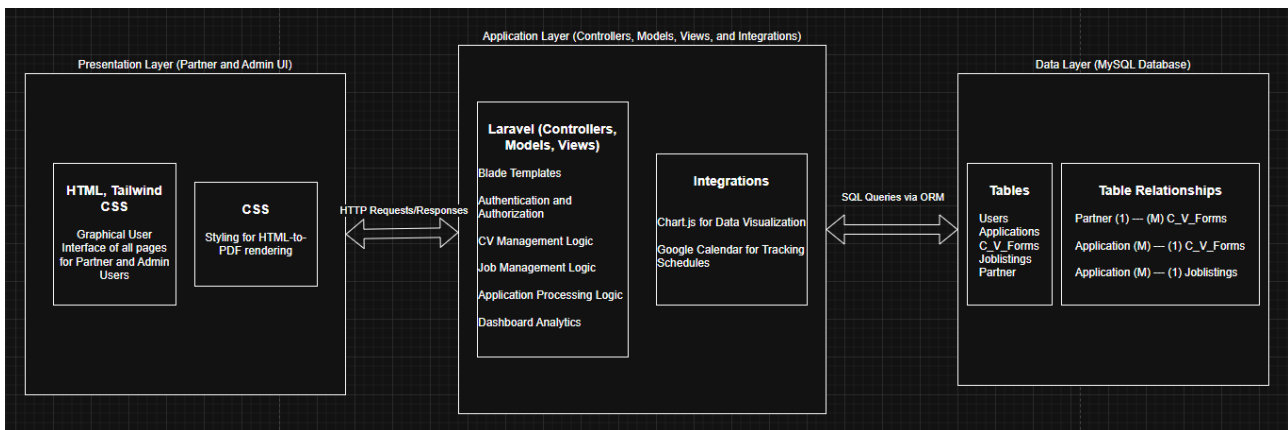


Figure: 4.2.1. System Architecture of Oshigoto Japan System

4.3. Entity Relationship Diagram

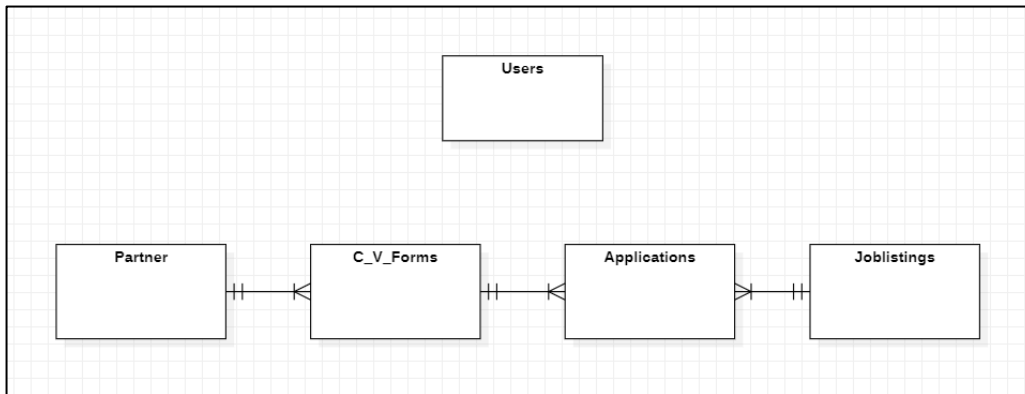


Figure: 4.3.1. Entity Relationship Diagram of Oshigoto Japan

4.4. Class Diagram

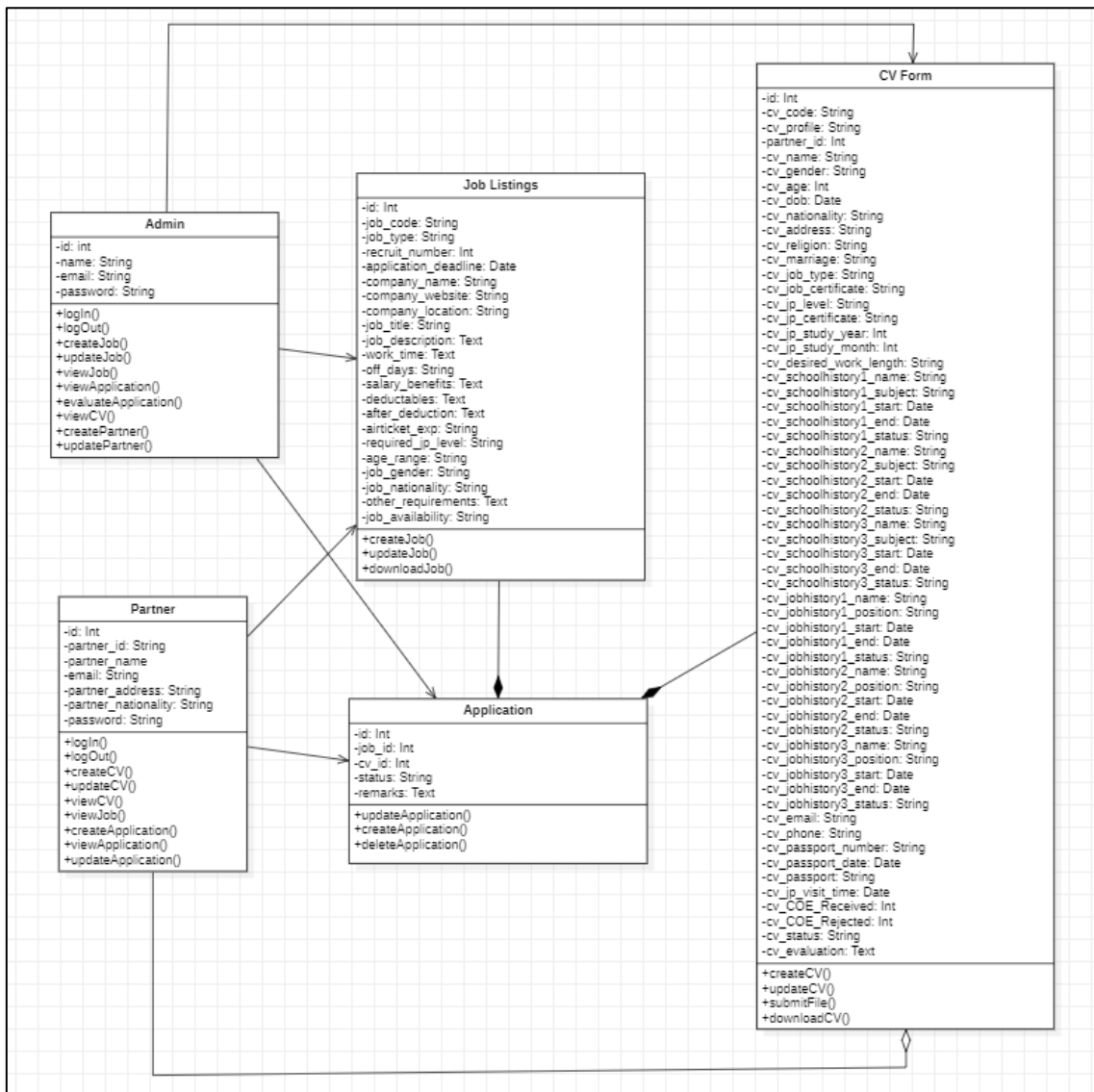


Figure: 4.4.1. Class Diagram of Oshigoto Japan System

4.5. Data Design

Introduction

Data design defines how the essential data and information of the system are organized and stored in the database. The Oshigoto Japan system uses a MySQL relational database to provide a clear structure for managing different entities.

The database is designed with several main tables. These include partner, users, applications, joblistings, and c_v_forms, each containing necessary attributes for their own functions, such as user information for partner, job information for joblistings, admin information for users, CV information for c_v_forms, and submitted CV data and job information for applications.

Users Table

Field Name	Data Type	NULL	Key	Remarks
Id	bigint (20)	NO	PRIMARY KEY	Auto Increment
Name	Varchar (255)	NO		
email	Varchar (255)	NO	UNIQUE	
email_verified_at	Timestamp	YES		
password	Varchar (255)	NO		
Remember token	Varchar (100)	YES		
Created_at	Timestamp	YES		
Updated_at	Timestamp	YES		

Table: 4.5.1. User Table for Oshigoto Japan System's Database

Partners Table

Field Name	Data Type	NULL	Key	Remarks
Id	bigint (20)	NO	PRIMARY KEY	Auto Increment
Partner_id	Varchar (255)	NO	UNIQUE	
Partner_name	Varchar (255)	NO		
Email	Varchar (255)	NO	UNIQUE	
Partner_address	Varchar (255)	NO		
Partner_nationality	Varchar (255)	NO		
Password	Varchar (255)	NO		
Remember token	Varchar (100)	YES		
Created_at	Timestamp	YES		
Updated_at	Timestamp	YES		

Table: 4.5.2. Partner Table for Oshigoto Japan System's

C_V_Forms Table

Field Name	Data Type	NULL	Key	Description
Id	bigint (20)	NO	PRIMARY KEY	Auto Increment
Cv_code	Varchar (255)	NO	UNIQUE	
Cv_profile	Varchar (255)	YES	FOREIGN KEY	
Partner_id	bigint (20)	YES		
Cv_name	Varchar (255)	NO		
Cv_gender	Varchar (255)	NO		
Cv_age	Int (11)	NO		
Cv_dob	Date	NO		
Cv_nationality	Varchar (255)	NO		
Cv_address	Varchar (255)	NO		
Cv_religion	Varchar (255)	NO		
Cv_marriage	Varchar (255)	NO		
Cv_job_type	Varchar (255)	NO		
Cv_job_certificate	Varchar (255)	YES		
Cv_jp_level	Varchar (255)	NO		
Cv_jp_certificate	Varchar (255)	YES		
Cv_jp_study_year	Int (11)	NO		
Cv_jp_study_month	Int (11)	NO		
Cv_desired_work_length	Varchar (255)	NO		
Cv_schoolhistory1_name	Varchar (255)	NO		
Cv_schoolhistroy1_subject	Varchar (255)	NO		
Cv_schoolhistory1_start	Date	NO		
Cv_schoolhistory1_end	Date	NO		
Cv_schoolhistory1_status	Varchar (255)	NO		
Cv_schoolhistory2_name	Varchar (255)	YES		
Cv_schoolhistroy2_subject	Varchar (255)	YES		
Cv_schoolhistory2_start	Date	YES		
Cv_schoolhistory2_end	Date	YES		
Cv_schoolhistory2_status	Varchar (255)	YES		
Cv_schoolhistory3_name	Varchar (255)	YES		
Cv_schoolhistroy3_subject	Varchar (255)	YES		
Cv_schoolhistory3_start	Date	YES		
Cv_schoolhistory3_end	Date	YES		
Cv_schoolhistory3_status	Varchar (255)	YES		

Cv_jobhistory1_name	Varchar (255)	YES		
Cv_jobhistory1_position	Varchar (255)	YES		
Cv_jobhistory1_start	Date	YES		
Cv_jobhistory1_end	Date	YES		
Cv_jobhistory1_status	Varchar (255)	YES		
Cv_jobhistory2_name	Varchar (255)	YES		
Cv_jobhistory2_position	Varchar (255)	YES		
Cv_jobhistory2_start	Date	YES		
Cv_jobhistory2_end	Date	YES		
Cv_jobhistory2_status	Varchar (255)	YES		
Cv_jobhistory3_name	Varchar (255)	YES		
Cv_jobhistory3_position	Varchar (255)	YES		
Cv_jobhistory3_start	Date	YES		
Cv_jobhistory3_end	Date	YES		
Cv_jobhistory3_status	Varchar (255)	YES		
Cv_email	Varchar (255)	YES		
Cv_phone	Varchar (255)	YES		
Cv_passport_number	Varchar (255)	YES		
Cv_passport_date	Date	YES		
Cv_passport	Varchar (255)	YES		
CV_jp_visit_time	Int (11)	YES		
Cv_COE_Received	Int (11)	YES		
Cv_COE_Rejected	Int (11)	YES		
Cv_Status	Varchar (255)	NO		
Cv_evaluation	Text	YES		
Created_at		YES		
Updated_at		YES		

Table: 4.5.3. C_v_Forms Table for Oshigoto Japan System's Database

Joblistings Table

Field Name	Data Type	NULL	Key	Remarks
Id	bigint (20)	NO	PRIMARY KEY	Auto Increment
Job_code	Varchar (255)	NO	UNIQUE	
Job_type	Varchar (255)	NO		
Recruit_number	Int (11)	NO		
Application_deadline	Date	NO		
Company_name	Varchar (255)	NO		
Company_website	Varchar (255)	NO		
Company_location	Varchar (255)	NO		
Job_title	Varchar (255)	NO		
Job_description	Text	NO		
Work_time	Text	NO		
Off_days	Varchar (255)	NO		
Salary_benefits	Text	NO		
Deductables	Text	NO		
After_deduction	Text	NO		
Airticket_exp	Varchar (255)	NO		
Required_jp_level	Varchar (255)	NO		
Age_range	Varchar (255)	NO		
Job_gender	Varchar (255)	NO		
Job_nationality	Varchar (255)	NO		
Other_requirements	Text	YES		
Job_availability	Varchar (255)	NO		
Created_at	Timestamp	YES		
Updated_at	Timestamp	YES		

Table: 4.5.4. Joblistings Table for Oshigoto Japan System's Database

Applications Table

Field Name	Data Type	NULL	Key	Remarks
Id	bigint (20)	NO	PRIMARY KEY	Auto Increment
Job_id	bigint (20)	NO	FOREIGN KEY	
Cv_id	bigint (20)	NO	FOREIGN KEY	
Status	Varchar (255)	NO		
Remarks	Text	YES		
Created_at	Timestamp	YES		
Updated_at	Timestamp	YES		

Table: 4.5.5. Applications Table for Oshigoto Japan System's Database

4.6. Flowchart

Introduction

The flowcharts are visual tools that are used for understanding the processes and algorithms of the system (GeeksForGeeks, 2025). For the Oshigoto Japan System, the partner side flowcharts and admin side flowcharts are illustrated and presented below.

Partner Side Flowcharts

Partner Login Flowchart

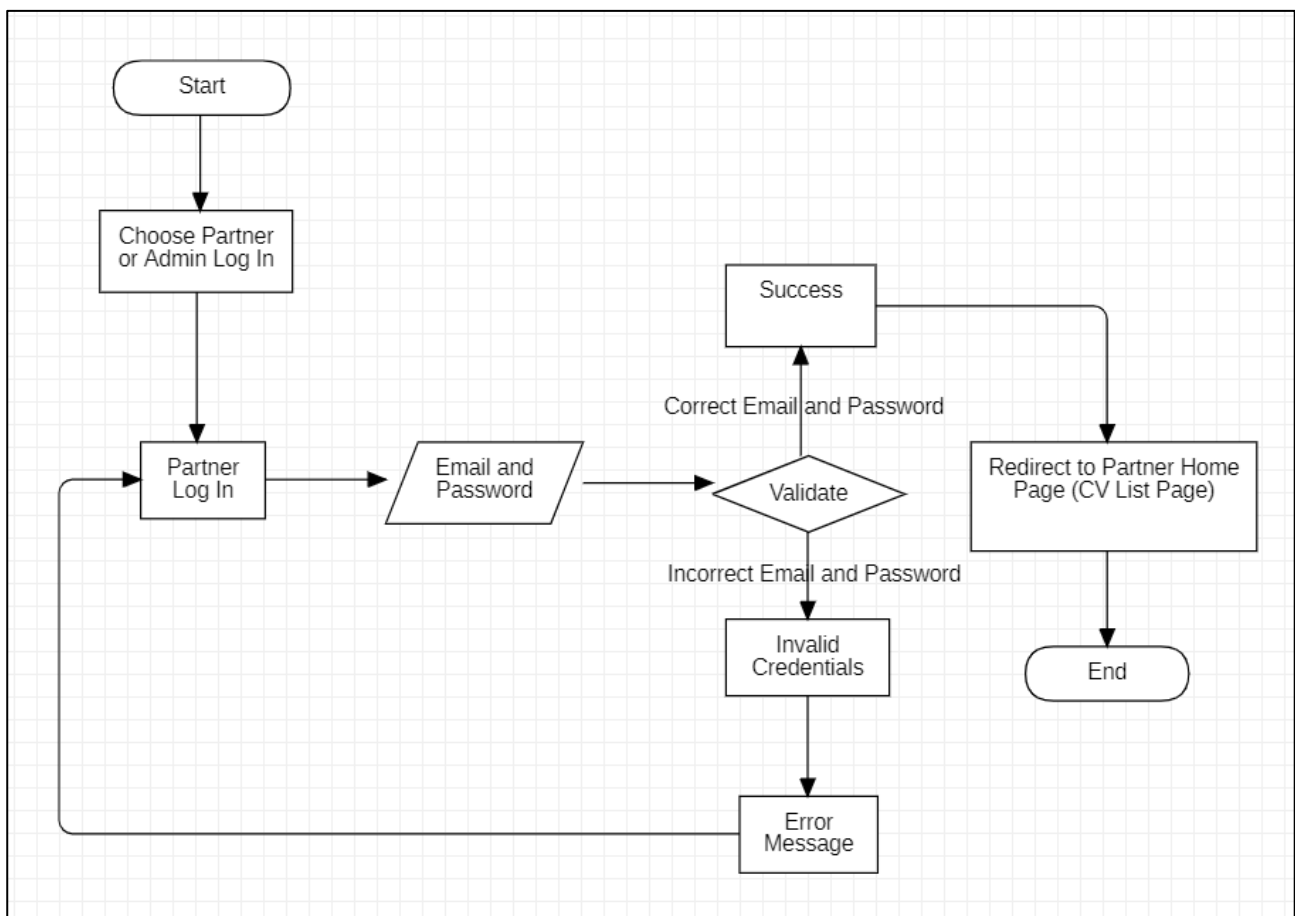


Figure: 4.6.1. Partner Login Flowchart

Partner CV Application Flowchart

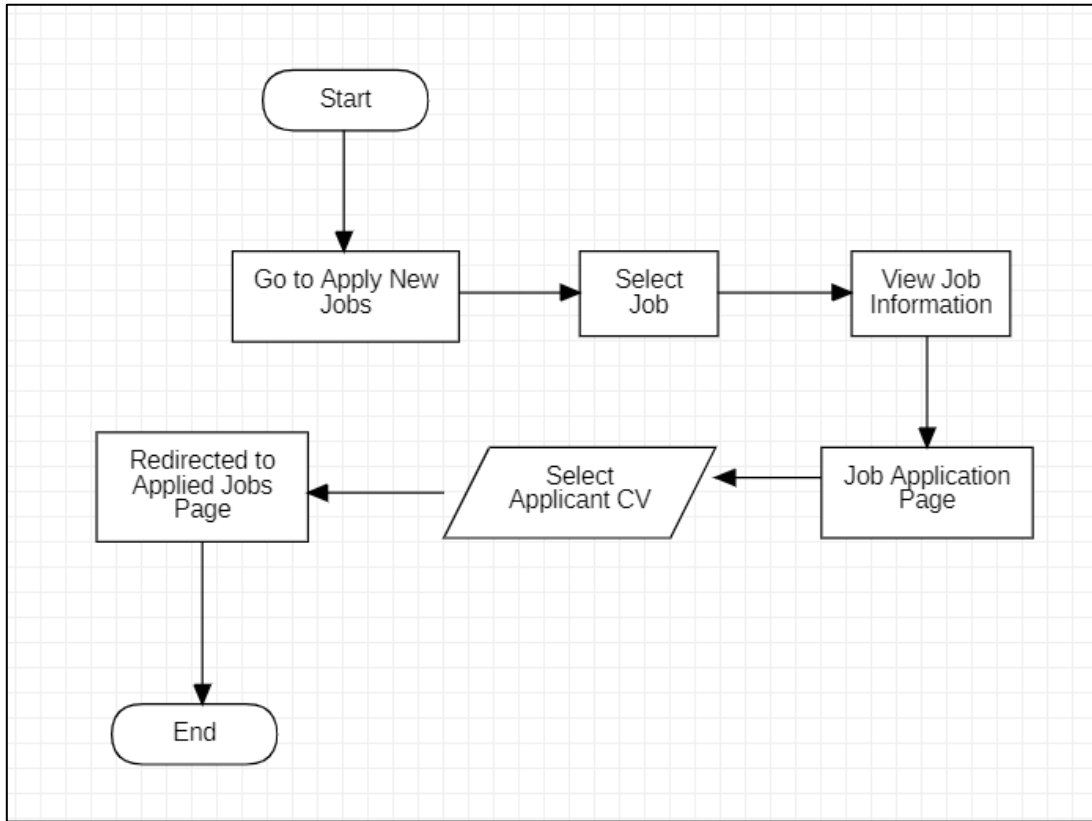


Figure: 4.6.2. Partner CV Application Flowchart

Partner CV Creation Flowchart

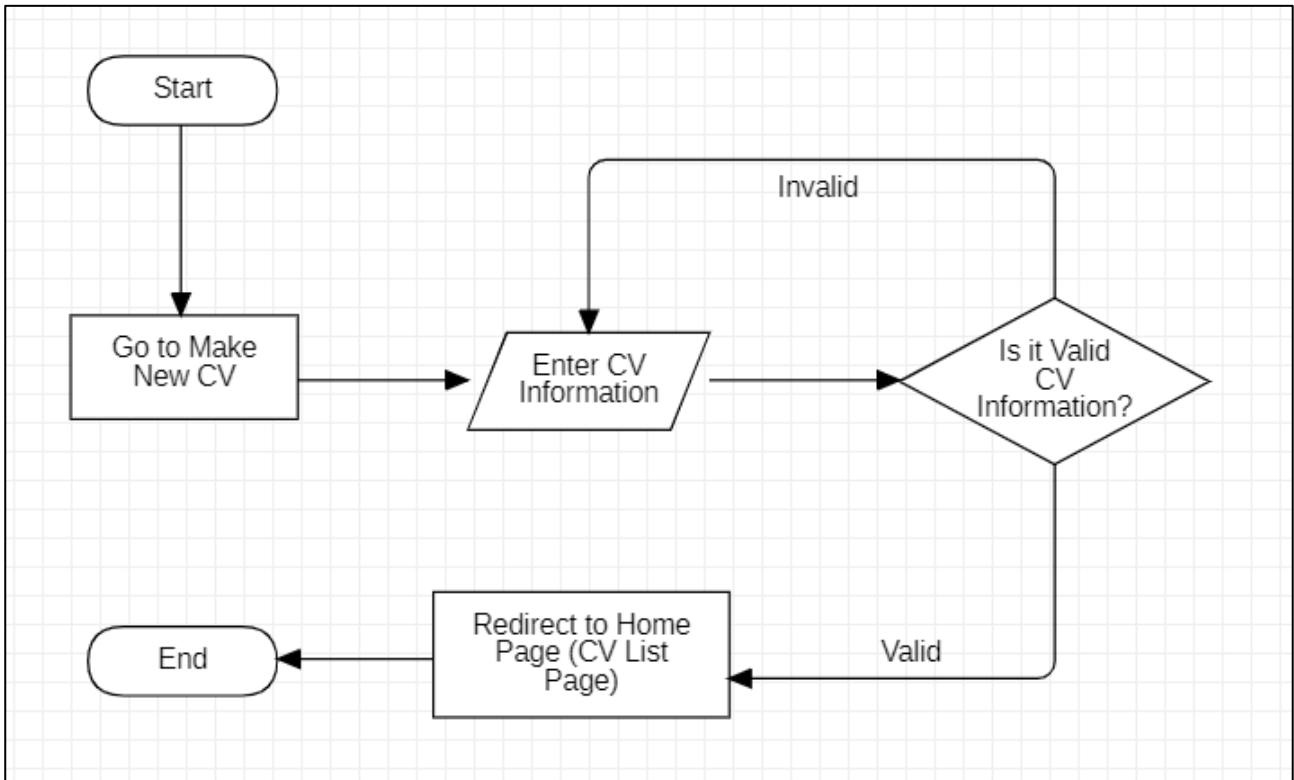


Figure: 4.6.3. Partner CV Creation Flowchart

Partner View and Edit CV

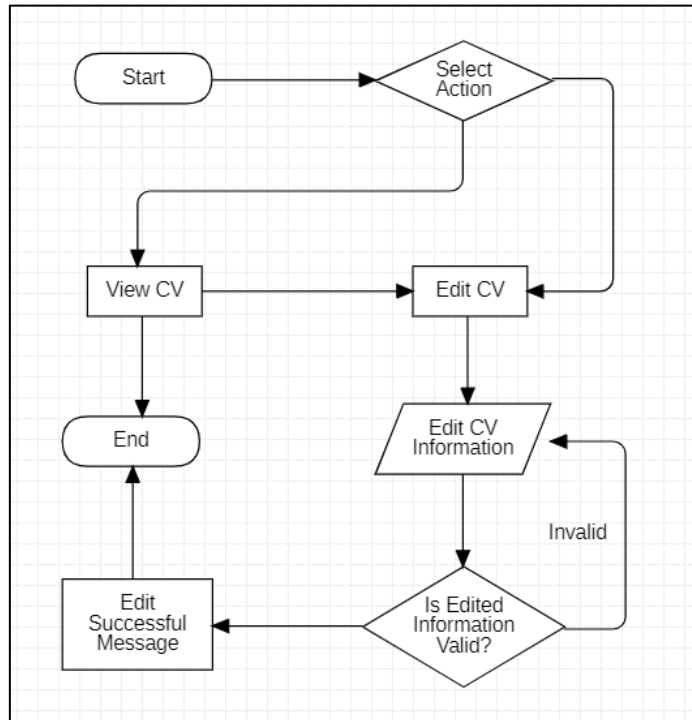


Figure: 4.6.4. Partner CV Creation Flowchart

Admin Side Flowchart

Admin Login Flowchart

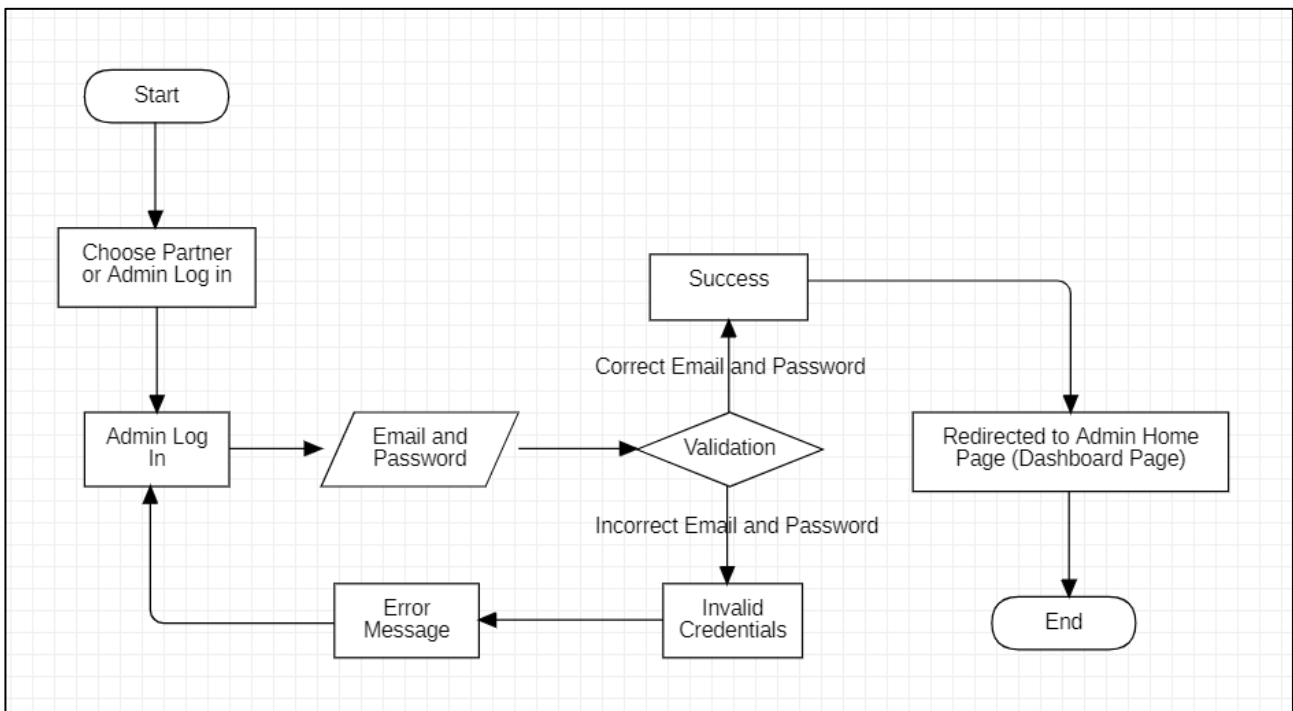


Figure: 4.6.5. Admin Login Flowchart

Admin Partner Creation and Edit Flowchart

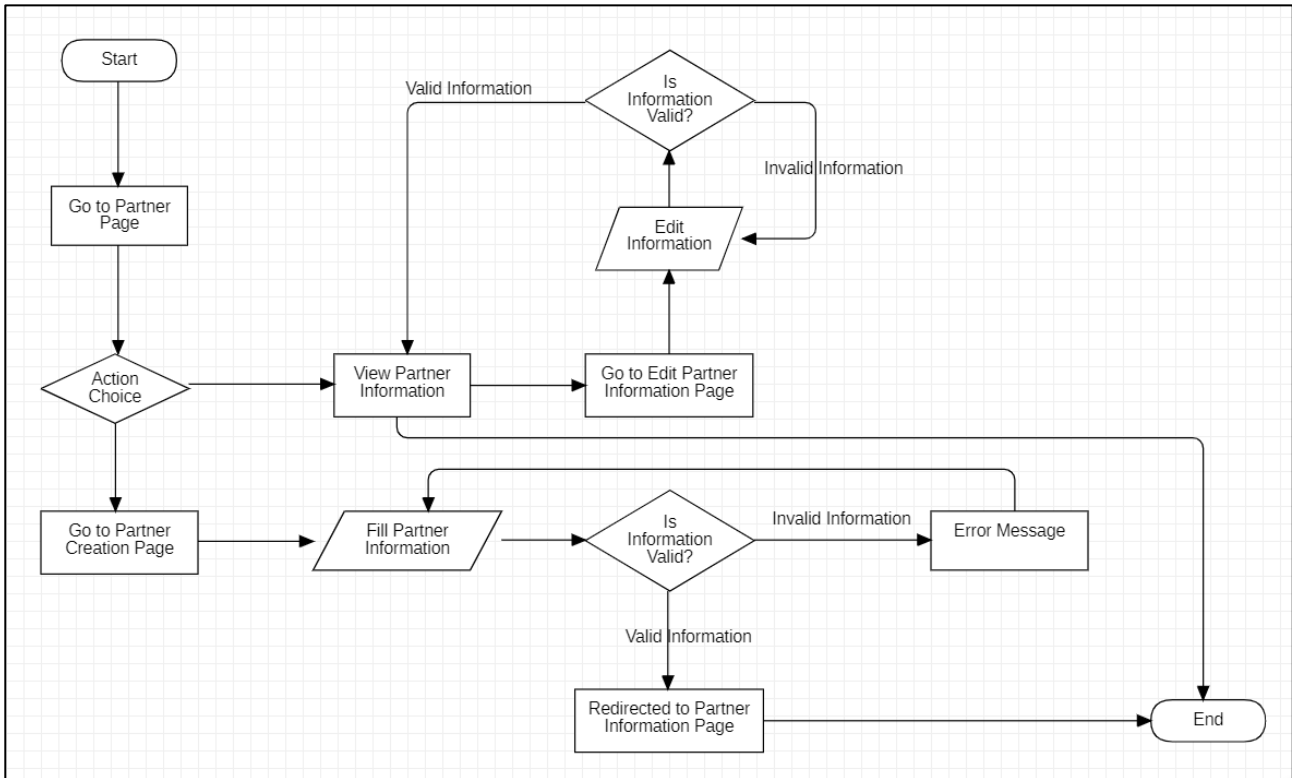


Figure: 4.6.6. Admin Partner Creation and Edit Flowchart

Admin Partner View, Edit, and Create Job

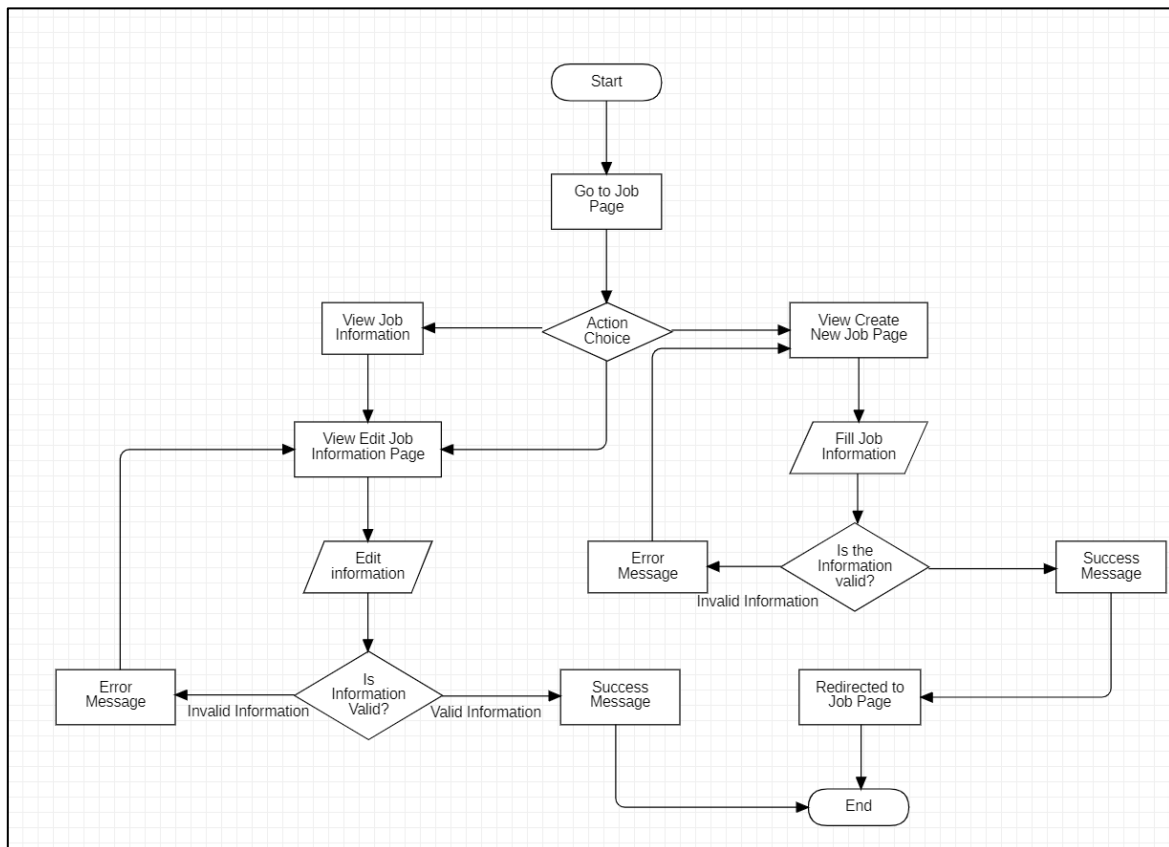


Figure: 4.6.7. Admin Partner View, Edit, and Create Job

Admin Partner View and Edit Job, View Application, and Evaluate Applicants

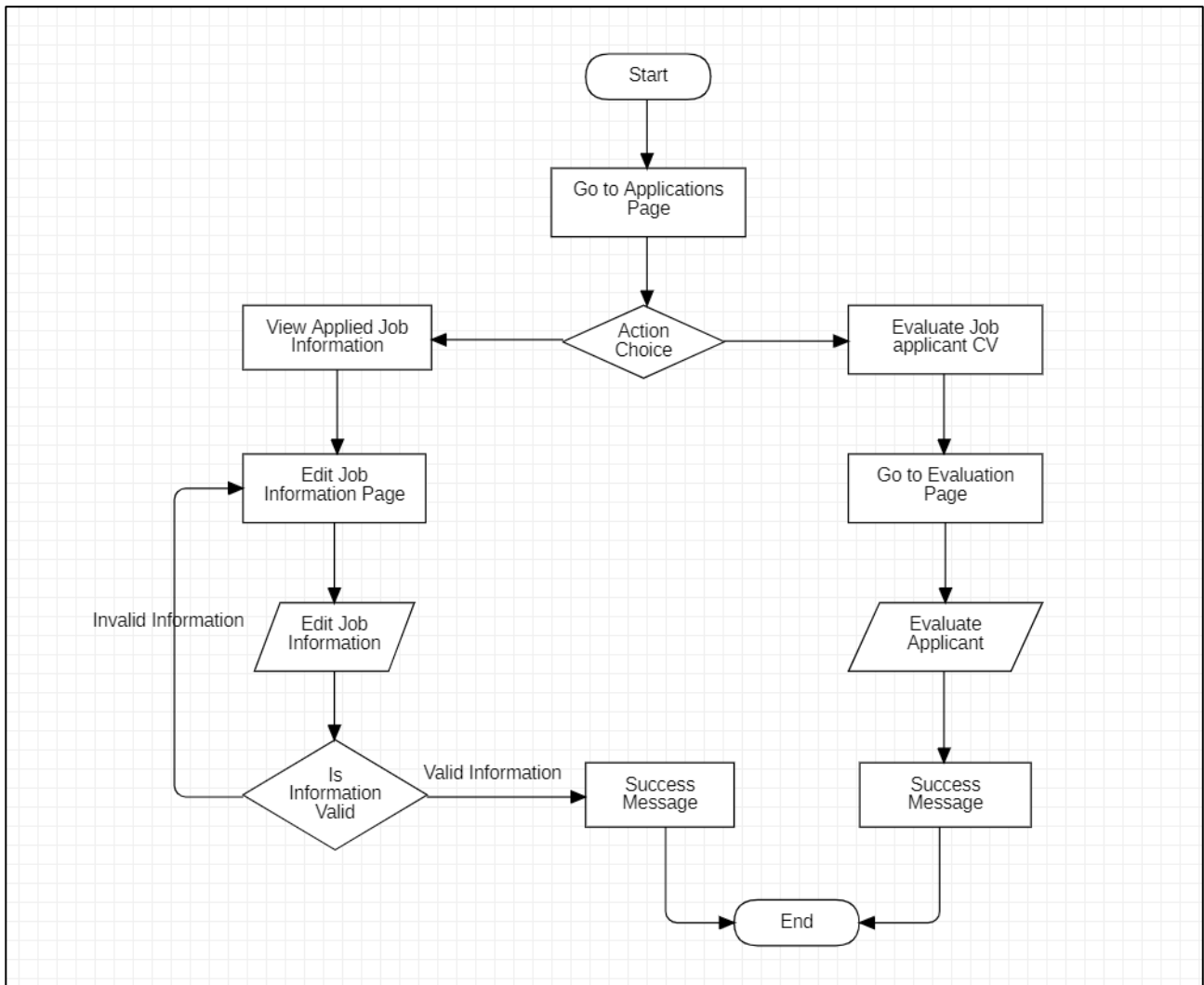


Figure: 4.6.8. Admin Partner View and Edit Job, View Application, and Evaluate Applicants

4.7. Graphical User Interface

Introduction

In the Graphical User Interface section, the screenshots of the web page designs from the completed Oshigoto Japan System are presented. As mentioned in the Technology Review in Chapter 2, HTML, Tailwind CSS, Traditional CSS, and Blade templates were used for user interface development. Some adjustments and refinements were made to ensure the system is simple, functional, professional, and user-friendly for both parties.

Main Log in Screen Interface

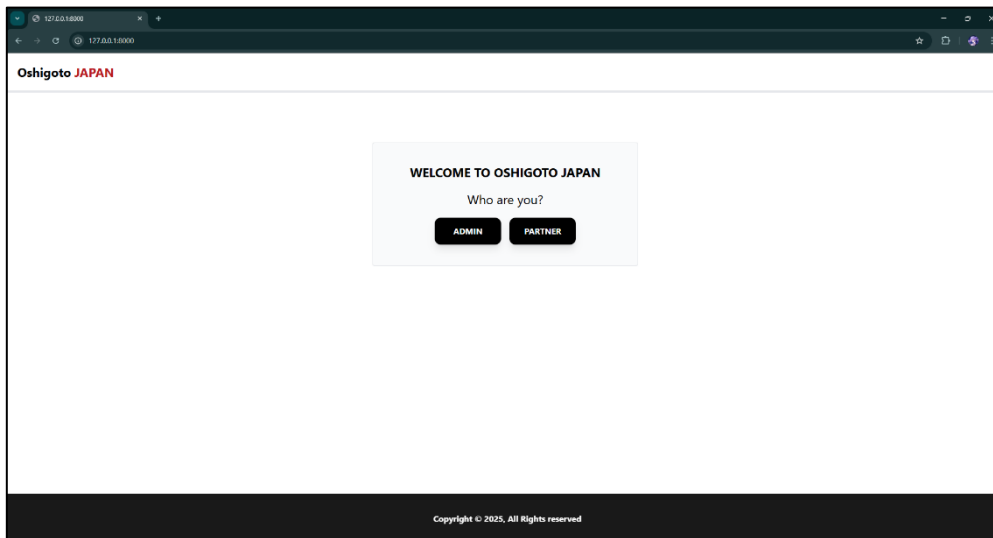


Figure: 4.7.1. Main Log in User Interface

Partner Side Interface

Partner Log In and Home (CV List Page)

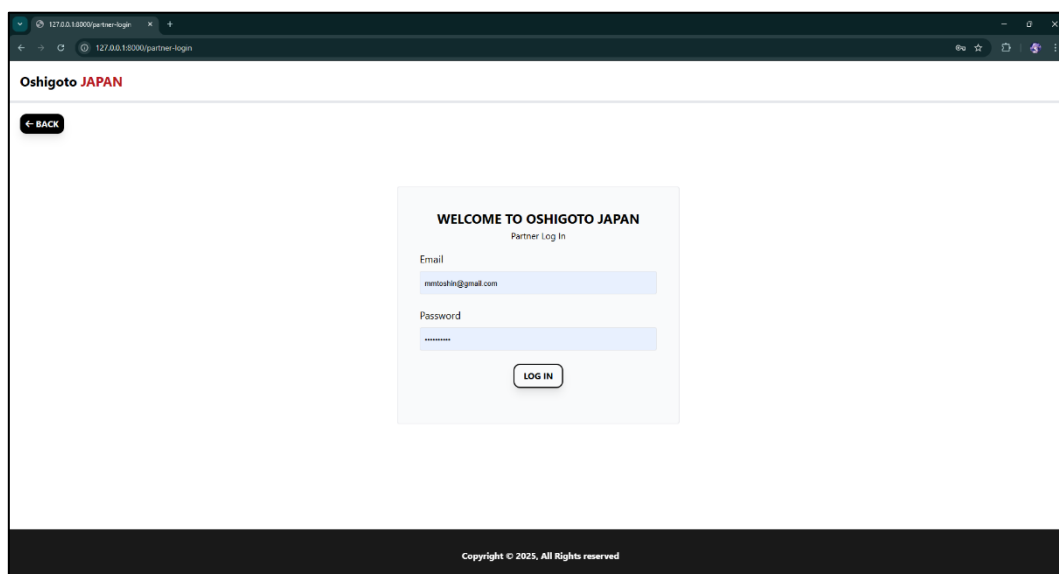


Figure: 4.7.2. Partner Log in User Interface

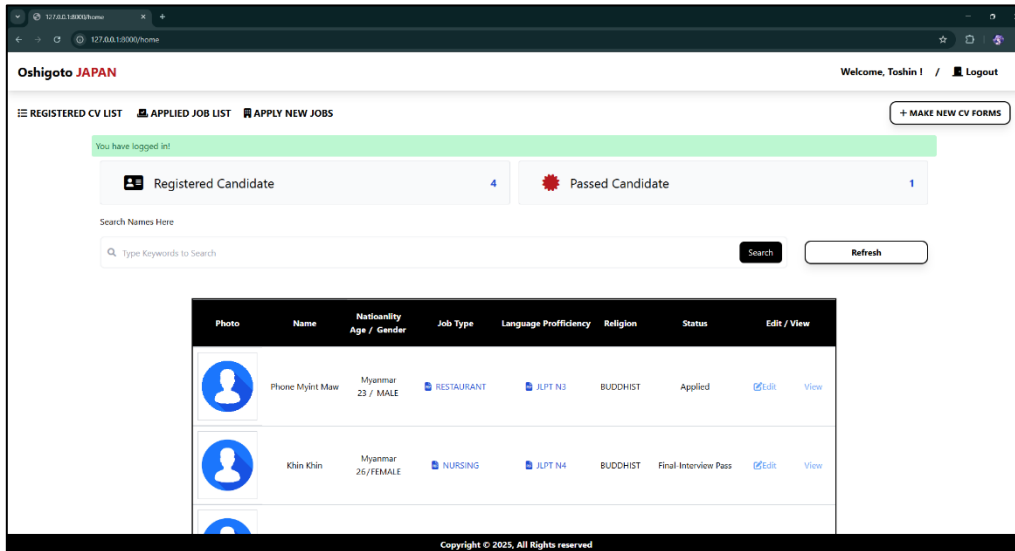


Figure: 4.7.3. Partner Log in Successful and Home Page User Interface

Partner CV View

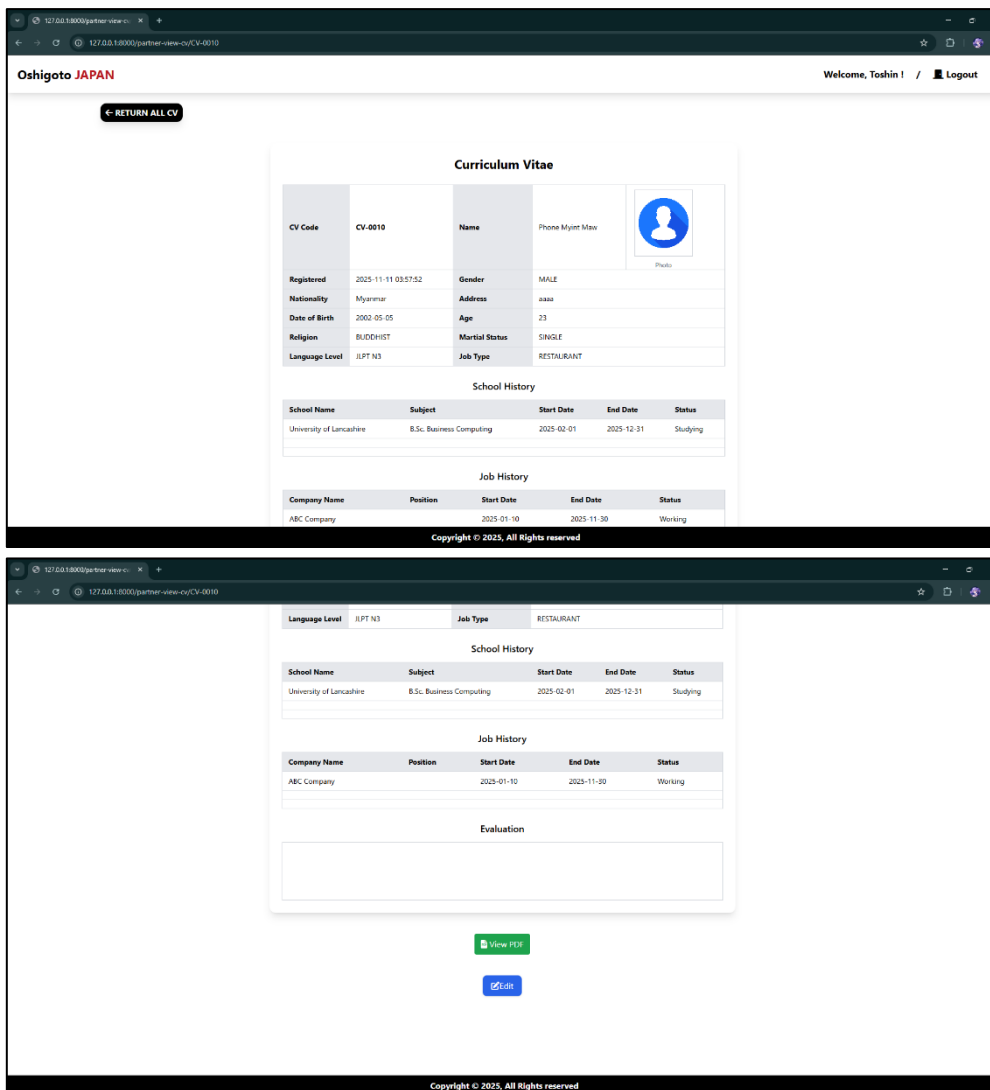


Figure: 4.7.4. Partner CV View User Interface

Partner CV Edit

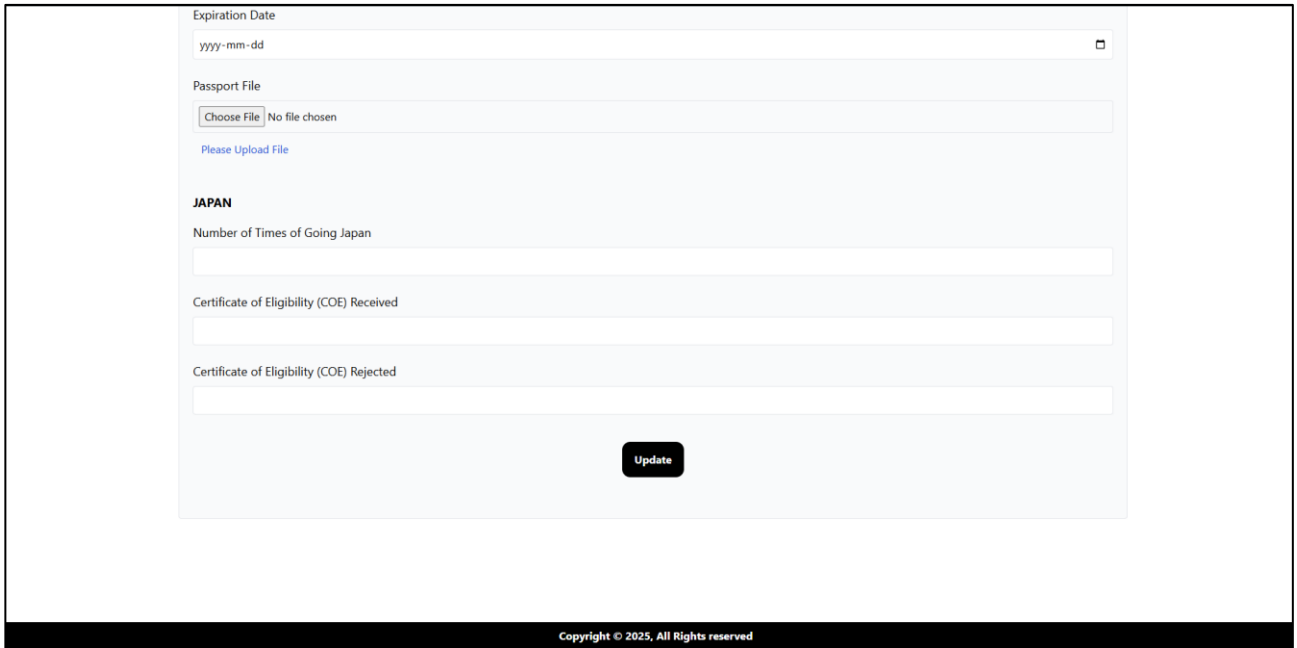
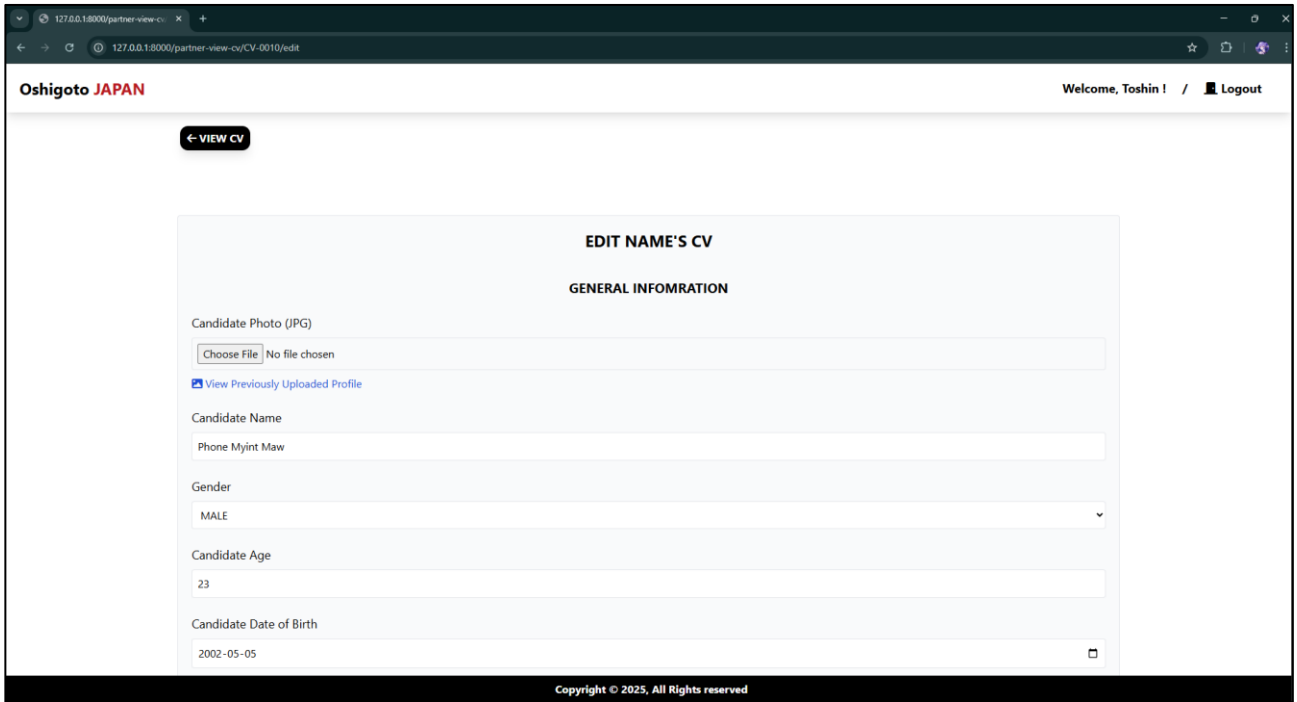


Figure: 4.7.5. Partner CV Edit User Interface

Partner Apply New Job

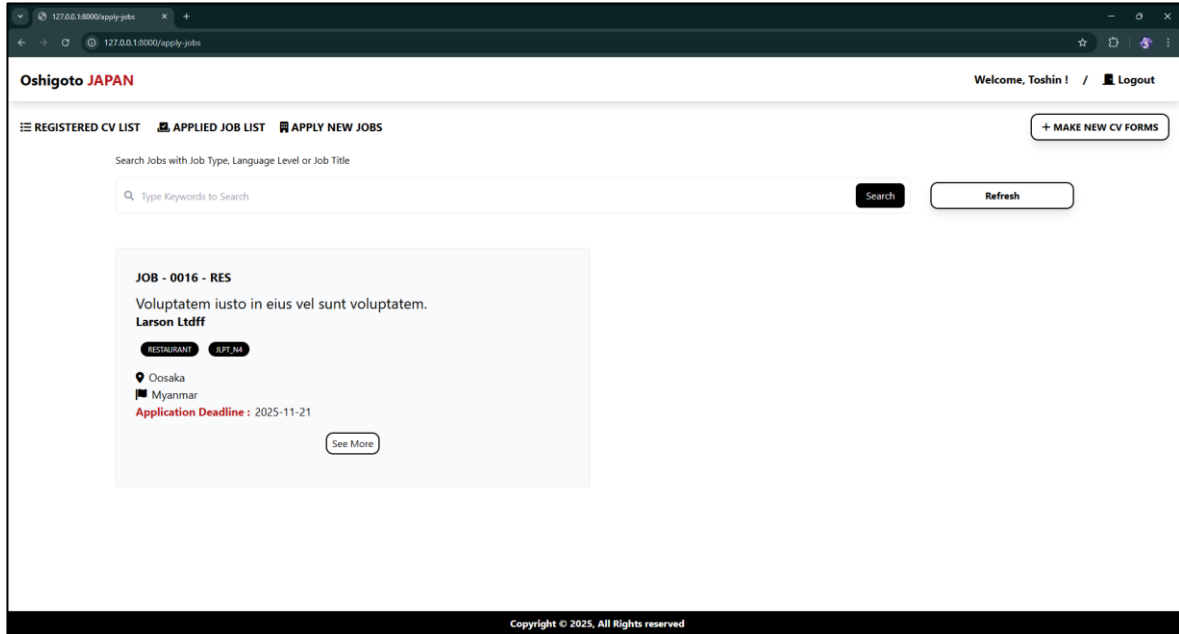


Figure: 4.7.6. Partner Apply New Job User Interface

Partner Job View

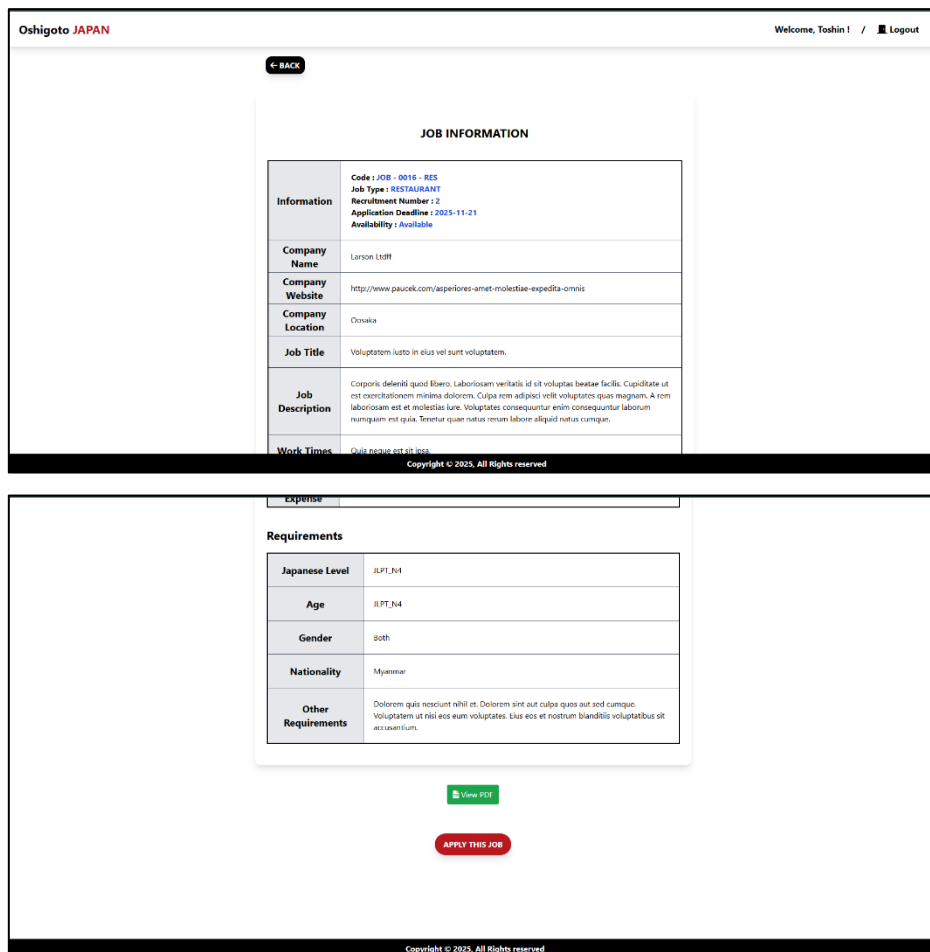


Figure: 4.7.7. Partner Job View User Interface

Partner Job Application

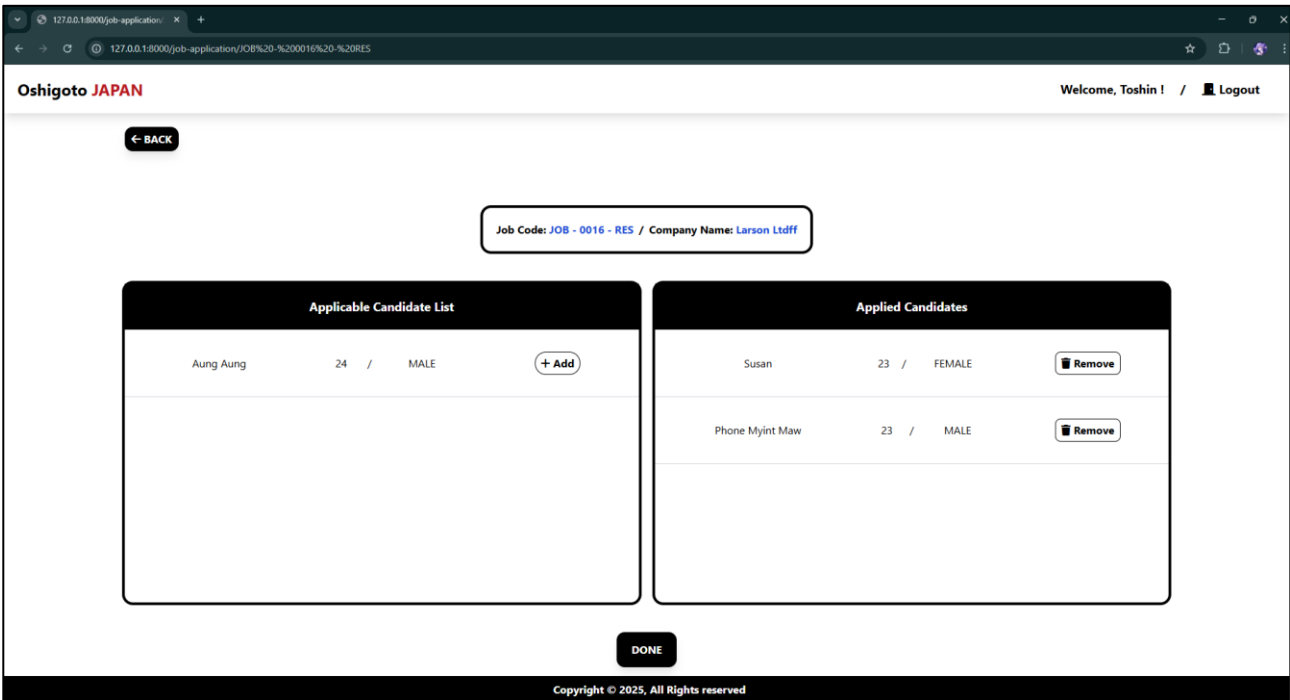


Figure: 4.7.8. Partner Job Application User Interface

Partner Applied Job List

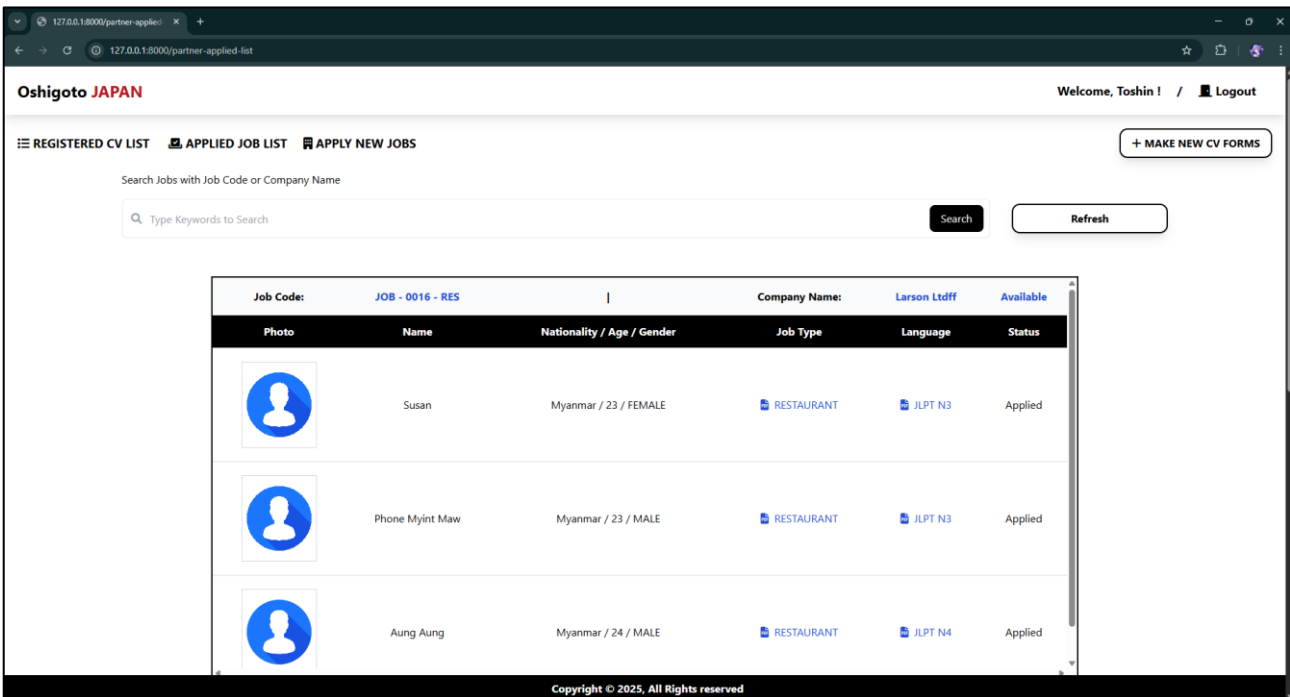


Figure: 4.7.9. Partner Applied Job List User Interface

Partner Navigation Bar

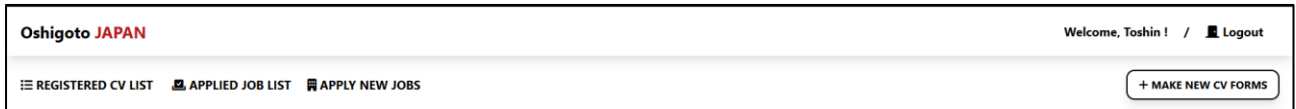


Figure: 4.7.10. Partner Navigation Bar User Interface

Partner After Log Out

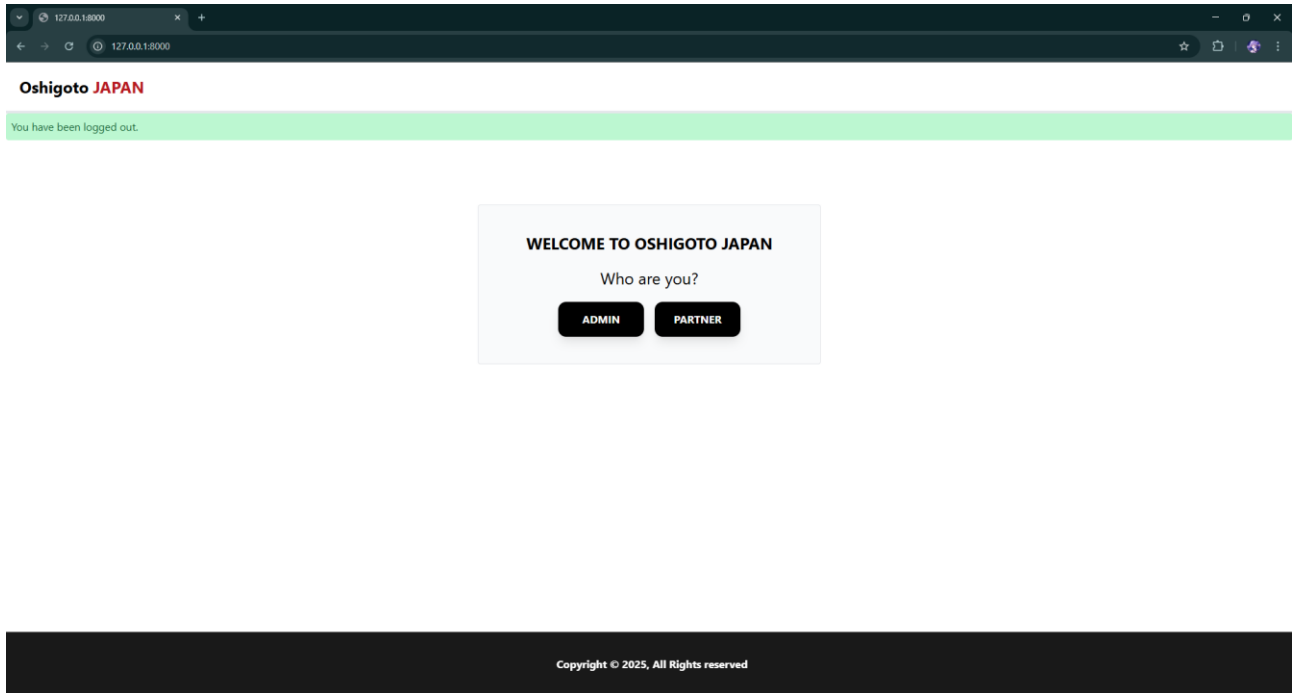


Figure: 4.7.11. Partner After Log Out User Interface

Admin Side Graphical User Interface

Admin Log in

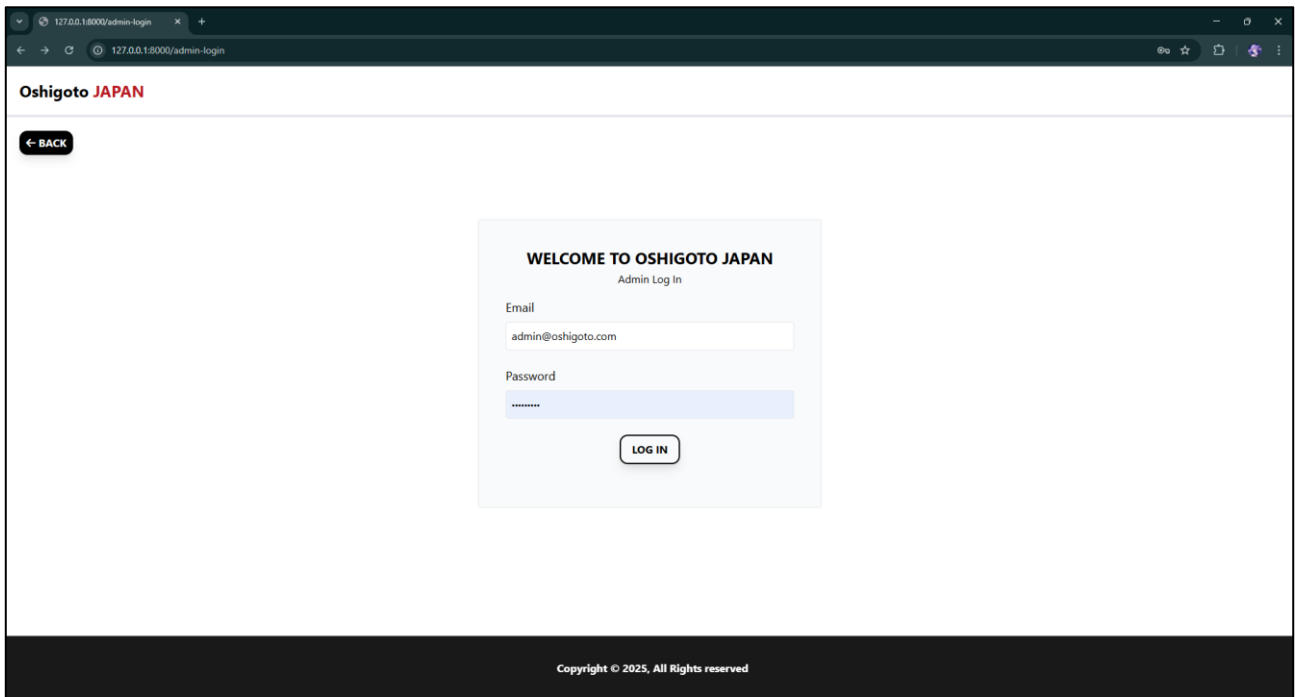


Figure: 4.7.12. Admin Log in User Interface

Admin Side Navigation Bar

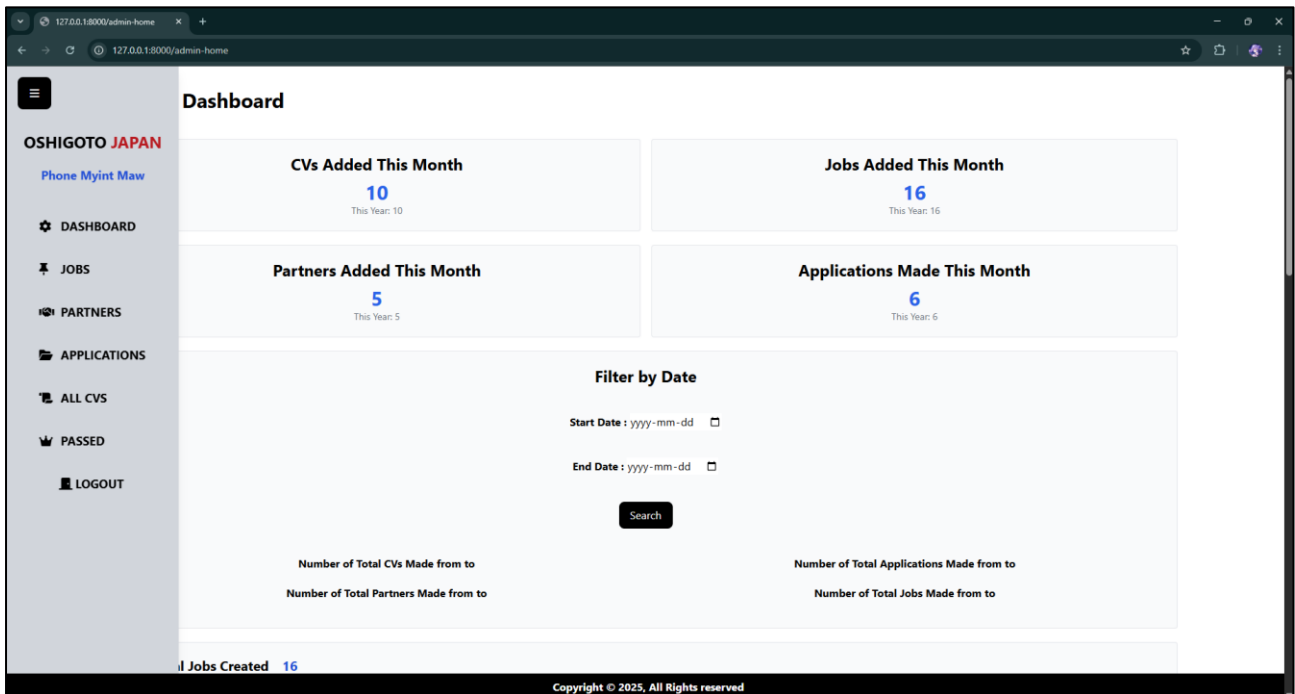


Figure: 4.7.13. Admin Side Navigation Bar User Interface

Admin Dashboard

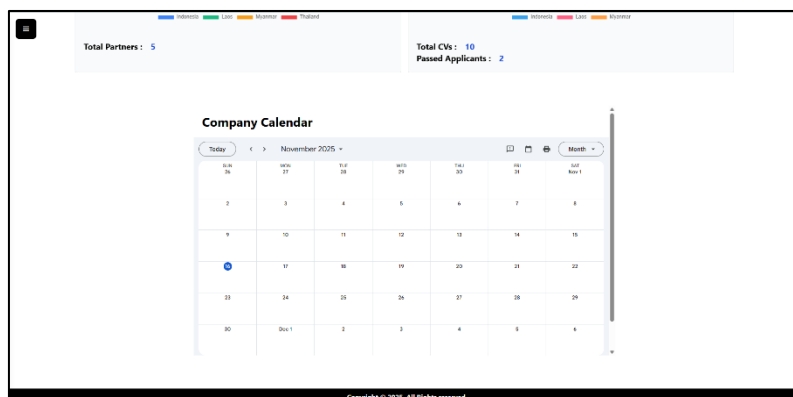
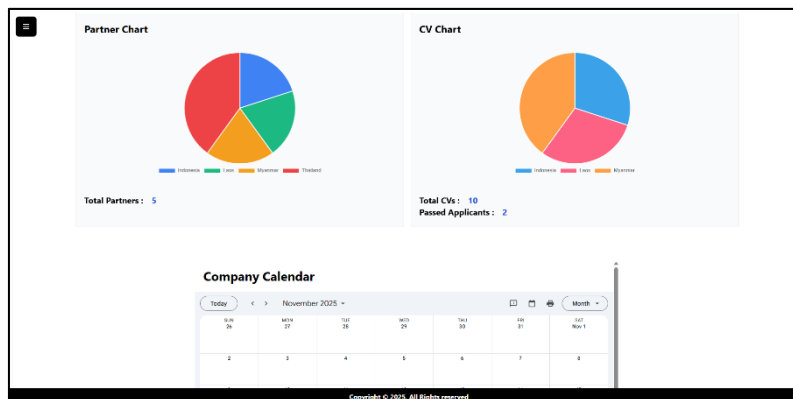
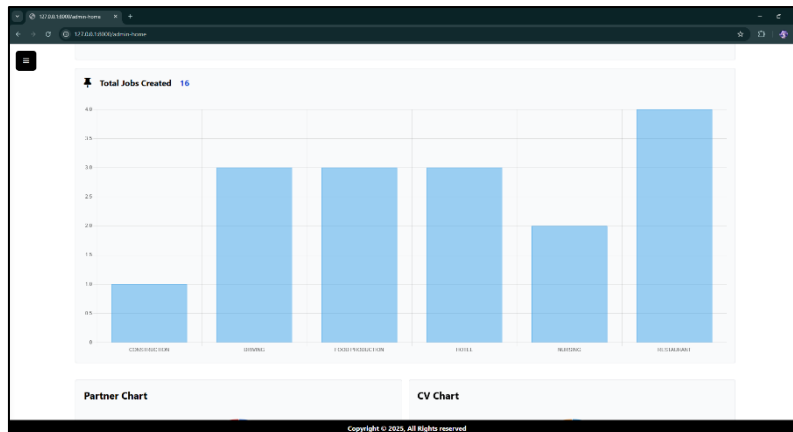
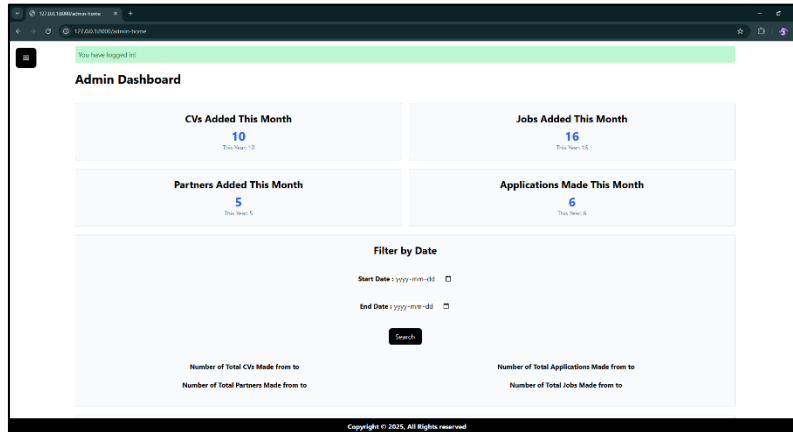


Figure: 4.7.14. Admin Dashboard User Interface

Admin Jobs

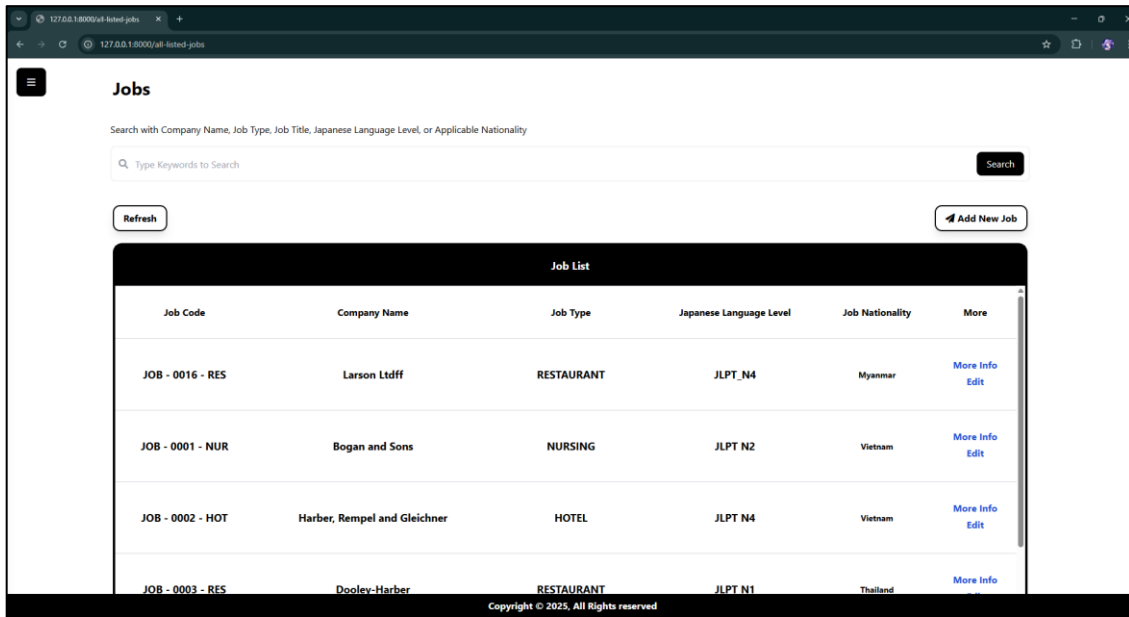


Figure: 4.7.15. Admin Jobs Page User Interface

Admin View Job

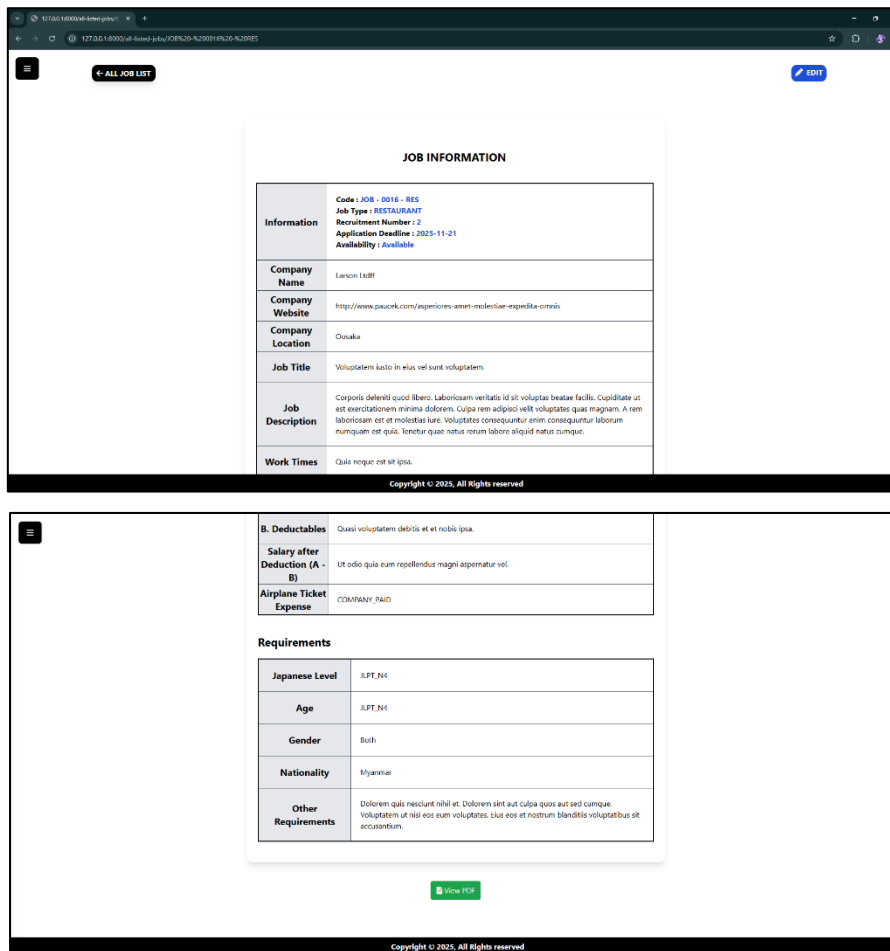


Figure: 4.7.16. Admin View Job User Interface

Admin Edit Job

The image displays two screenshots of a web application interface for editing job information. The top screenshot shows the 'EDIT JOB INFORMATION' page with a '← VIEW JOB' button and a form containing the following fields:

- Job Code: JOB - 0016 - RES
- Job Type: RESTAURANT
- Recruitment Number: 2
- Application Deadline: 2025-11-21
- Company Name: Larson Ltdff
- Company Website: http://www.paucek.com/asperiores-amet-molestiae-expedita-omnis
- Company Location: Osaka

The bottom screenshot shows a detailed form with the following fields:

- Airplane Ticket Expense: COMPANY_PAID
- Required Japanese Level: JLPT_N4
- Age: JLPT_N4
- Gender: Both
- Nationality: Myanmar
- Other Requirements: Dolorem quis nesciunt nihil et. Dolorem sint aut culpa quos aut sed cumque. Voluptatem ut nisi eos eum voluptates. Eius eos et nostrum
- Availability: Available

Both screenshots include a copyright notice at the bottom: Copyright © 2025, All Rights reserved.

Figure: 4.7.17. Admin Edit Job User Interface

Admin Add New Job

The screenshot shows a web browser window with the URL `127.0.0.1:8000/add-new-job`. The page has a dark header with a hamburger menu icon and a button labeled `← BACK TO ALL JOBS`. The main content area is titled **JOB INFORMATION** and contains a form with the following fields:

- Job Type: A dropdown menu with the placeholder text `-- Select Job Type --`.
- Recruitment Number: A text input field.
- Application Deadline: A date input field with the placeholder `yyyy-mm-dd` and a calendar icon.
- Company Name: A text input field.
- Company Website: A text input field.
- Company Location: A text input field.
- Job Title: A text input field.

At the bottom of the form area, there is a copyright notice: `Copyright © 2025, All Rights reserved`.

This screenshot shows the lower portion of the form, containing the following fields:

- Salary After Deduction (A-B): A text input field.
- Airplane Ticket Expense: A dropdown menu with the placeholder text `-- Select One --`.
- Required Japanese Level: A dropdown menu with the placeholder text `-- Select Japanese Level --`.
- Age Range: A dropdown menu with the placeholder text `-- Select Age Range --`.
- Gender: A dropdown menu with the placeholder text `-- Select Gender --`.
- Nationality: A dropdown menu with the placeholder text `-- Select Nationality --`.
- Other Requirements: A text input field.

Below these fields is a green button with a pencil icon and the text `Add`. At the bottom of the page, there is a copyright notice: `Copyright © 2025, All Rights reserved`.

Figure: 4.7.18. Admin Add New Job User Interface

Admin Partners

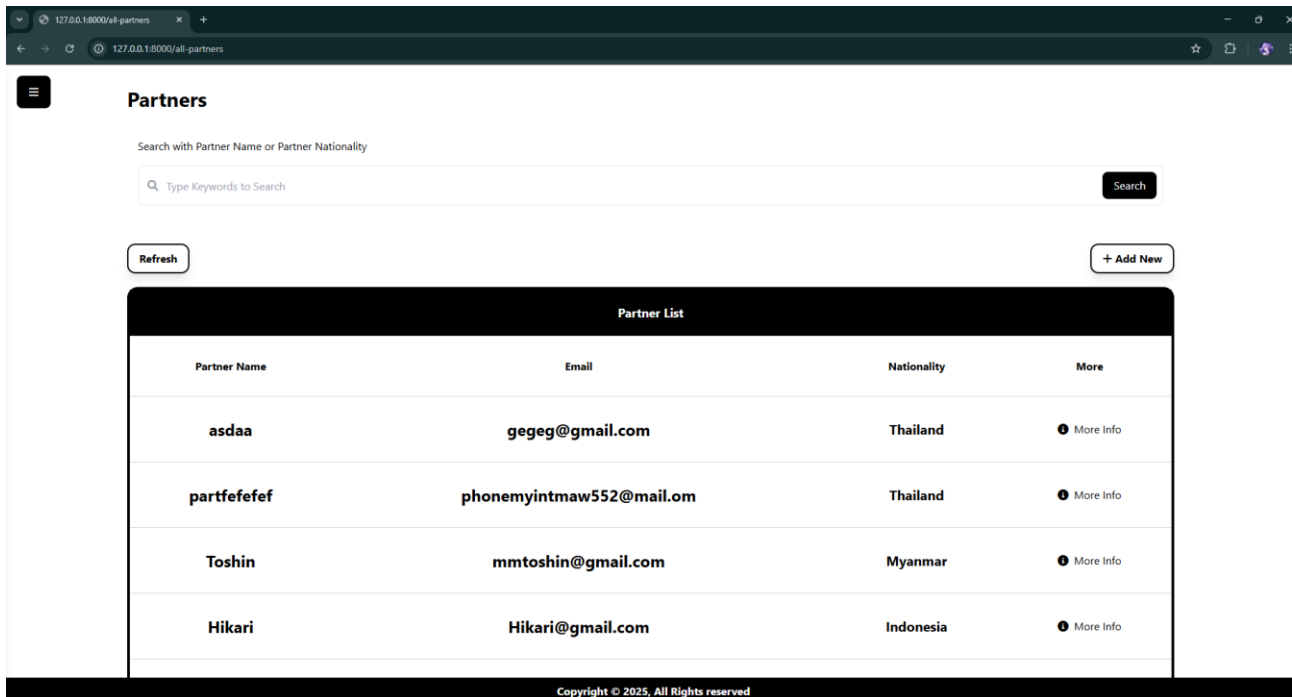


Figure: 4.7.19. Admin Partners Page User Interface

Admin View Partner

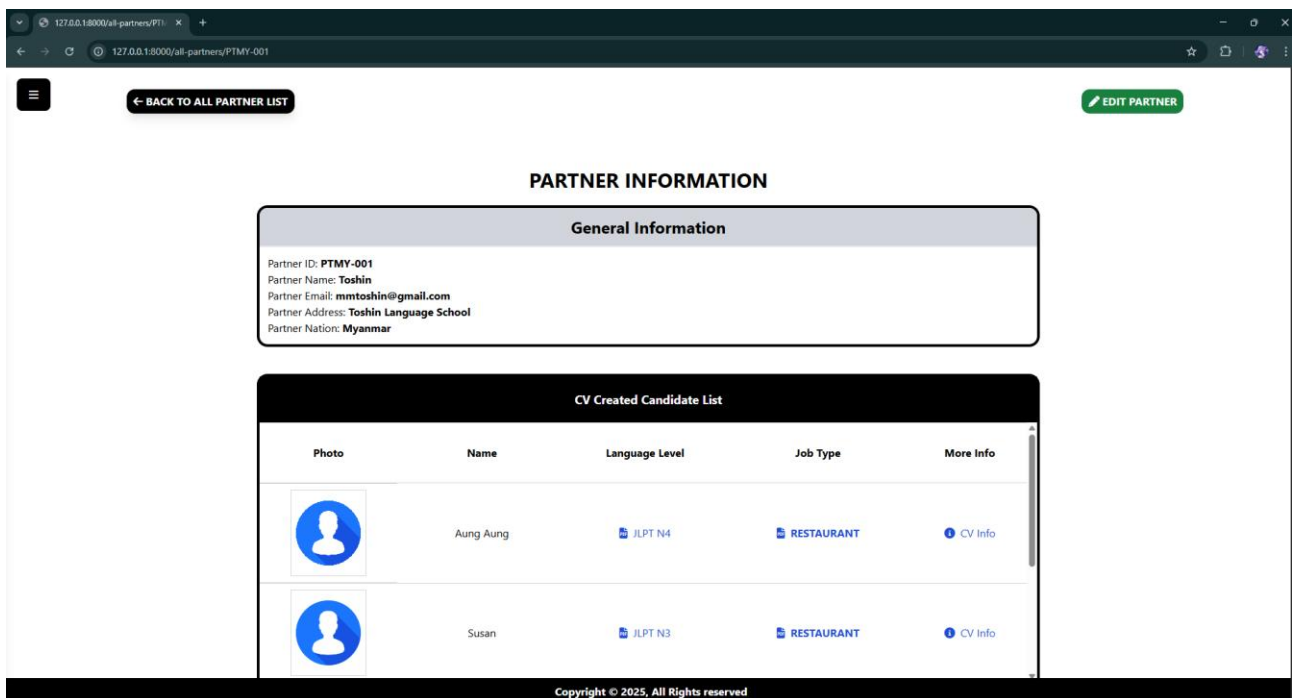


Figure: 4.7.20. Admin View Partner Page User Interface

Admin Edit Partner

The screenshot shows a web browser window with the URL `127.0.0.1:8000/all-partners/PTMY-001/edit`. The page title is "EDIT PARTNER INFORMATION". At the top left, there is a hamburger menu icon and a "← BACK" button. The main content area contains a form with the following fields:

- Partner ID:** PTMY-001
- Partner Name:** Toshin
- Partner Email:** mmtoshin@gmail.com
- Partner Address:** Toshin Language School
- Partner Password (Leave if No Update is Needed):** (empty field)

At the bottom of the form is a green "Edit" button with a pencil icon. The footer of the page reads "Copyright © 2025, All Rights reserved".

Figure: 4.7.21. Admin Edit Partner Page User Interface

Admin Add Partner

The screenshot shows a web browser window with the URL `127.0.0.1:8000/register-new-partner`. The page title is "PARTNER INFORMATION". At the top left, there is a hamburger menu icon and a "← BACK TO ALL PARTNERS" button. The main content area contains a form titled "General Information" with the following fields:

- Partner Name:** (empty text input)
- Partner Email:** (empty text input)
- Partner Address:** (empty text input)
- Partner Location:** -- Select Location -- (dropdown menu)
- Partner Password:** (empty text input)

At the bottom of the form is a blue "Register" button. The footer of the page reads "Copyright © 2025, All Rights reserved".

Figure: 4.7.22. Admin Add Partner Page User Interface

Admin Applications

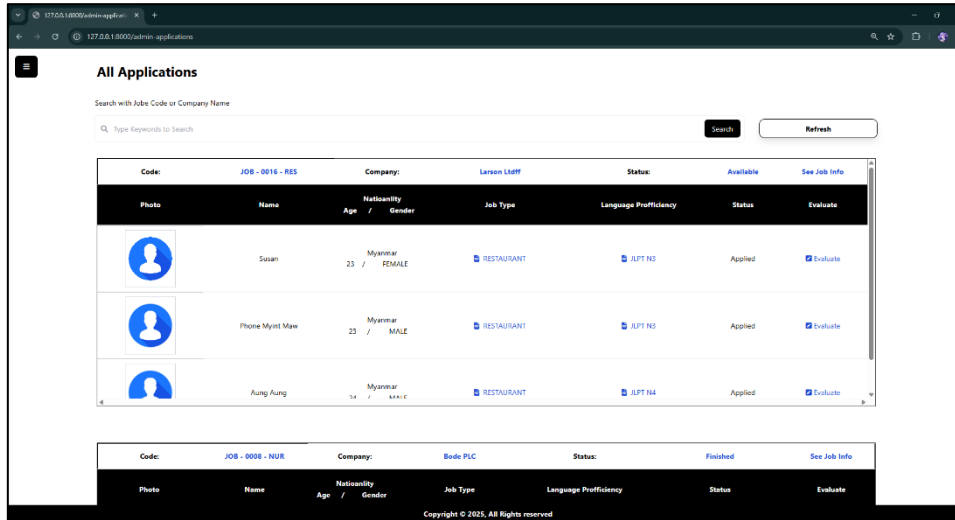


Figure: 4.7.23. Admin Applications Page User Interface

Admin Applicant Evaluation

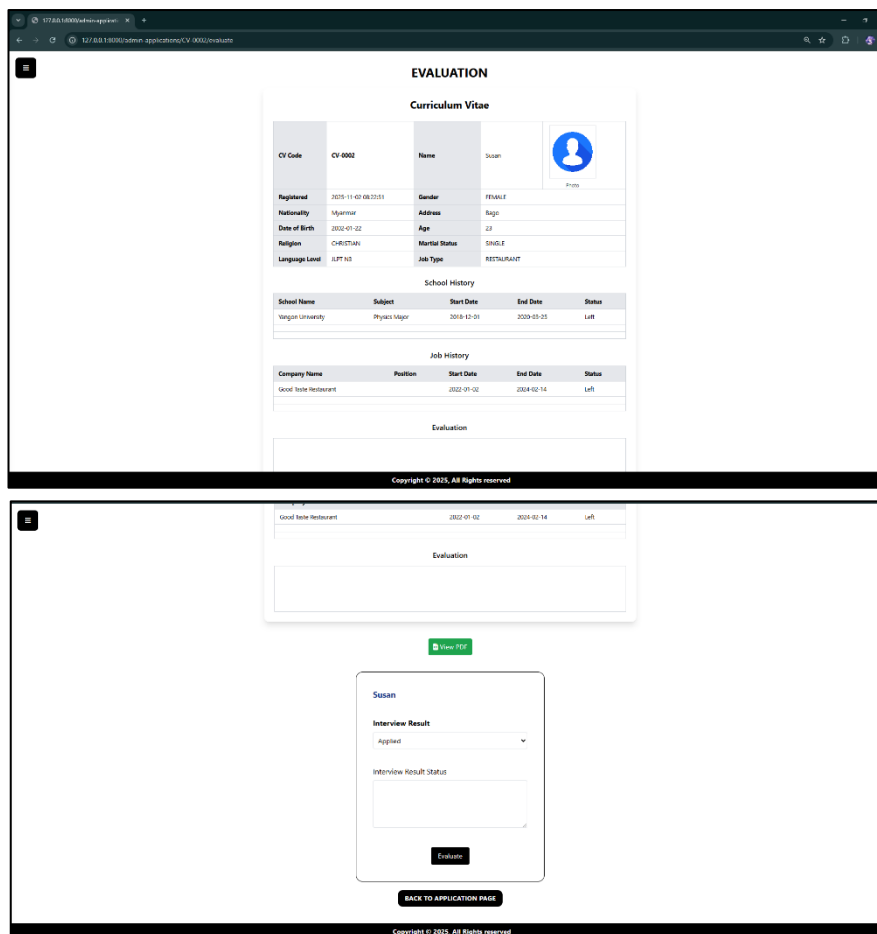


Figure: 4.7.24. Admin Applicant Evaluation User Interface

Admin All Registered CVs

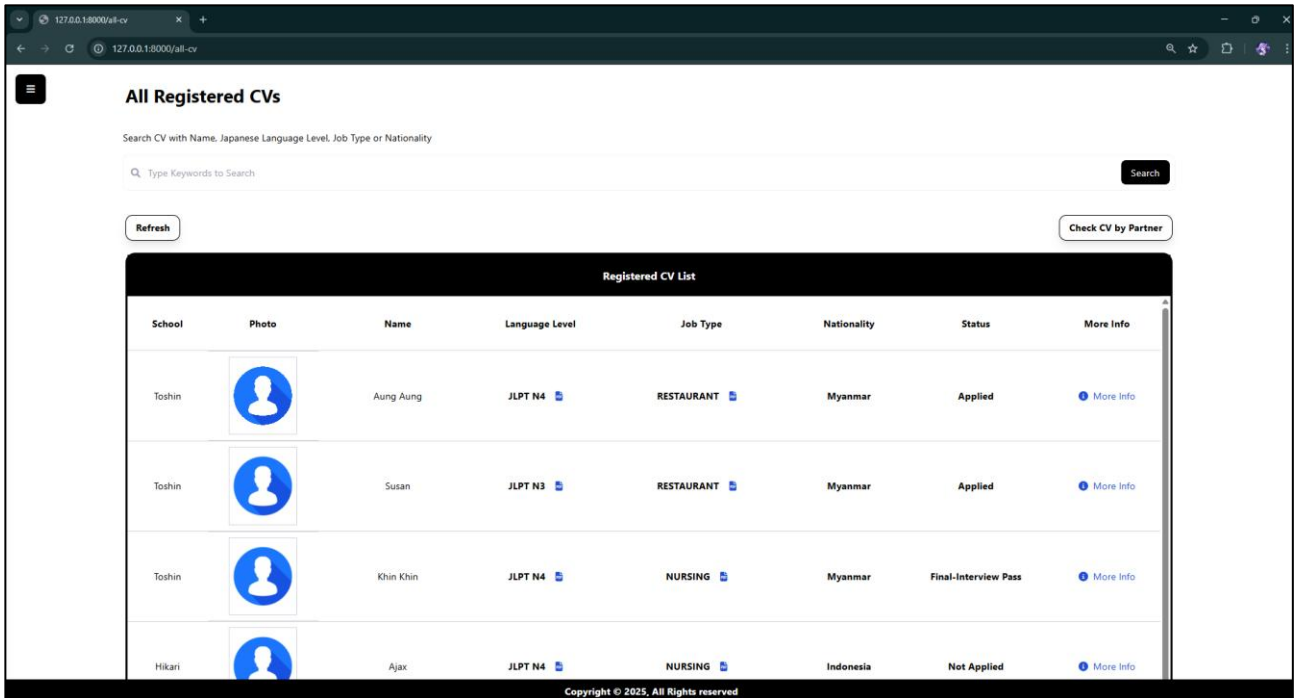


Figure: 4.7.25. Admin All Registered CVs Page User Interface

Admin View CV

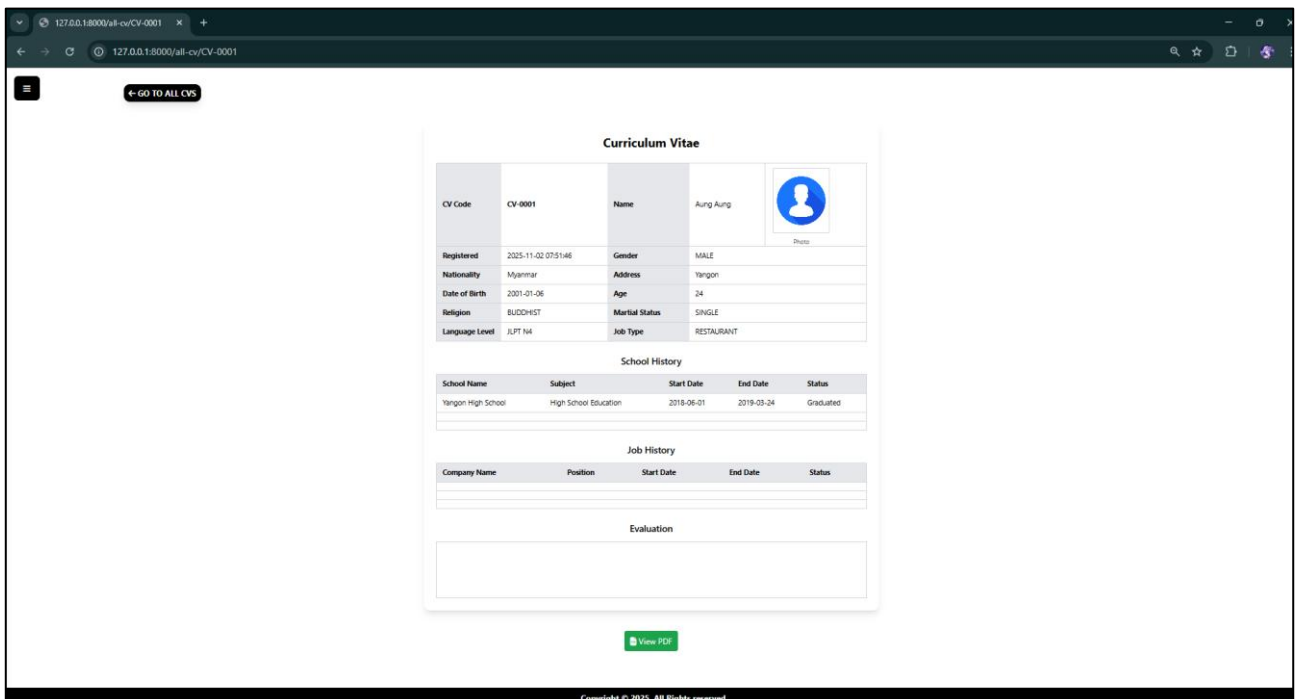


Figure: 4.7.26. Admin View CV Page User Interface

Admin Passed CVs

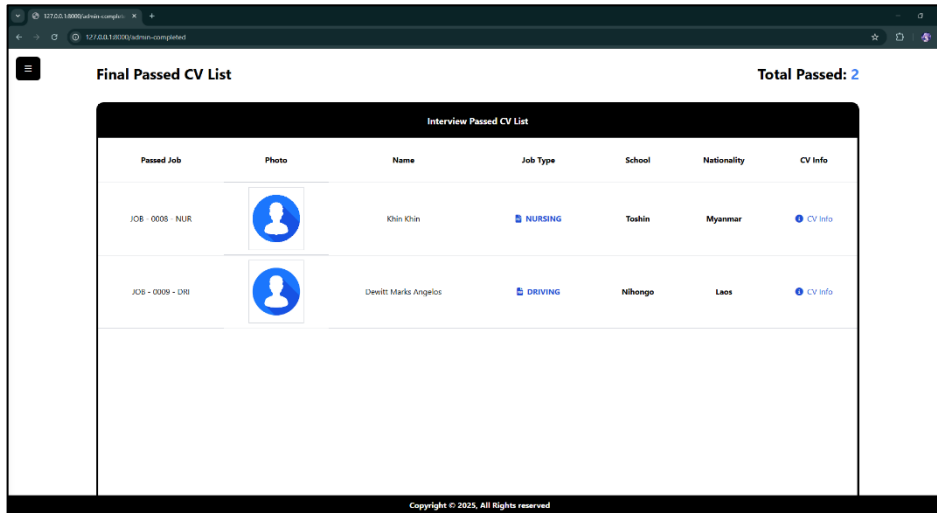


Figure: 4.7.27. Admin Passed CVs Page User Interface

Admin Log out

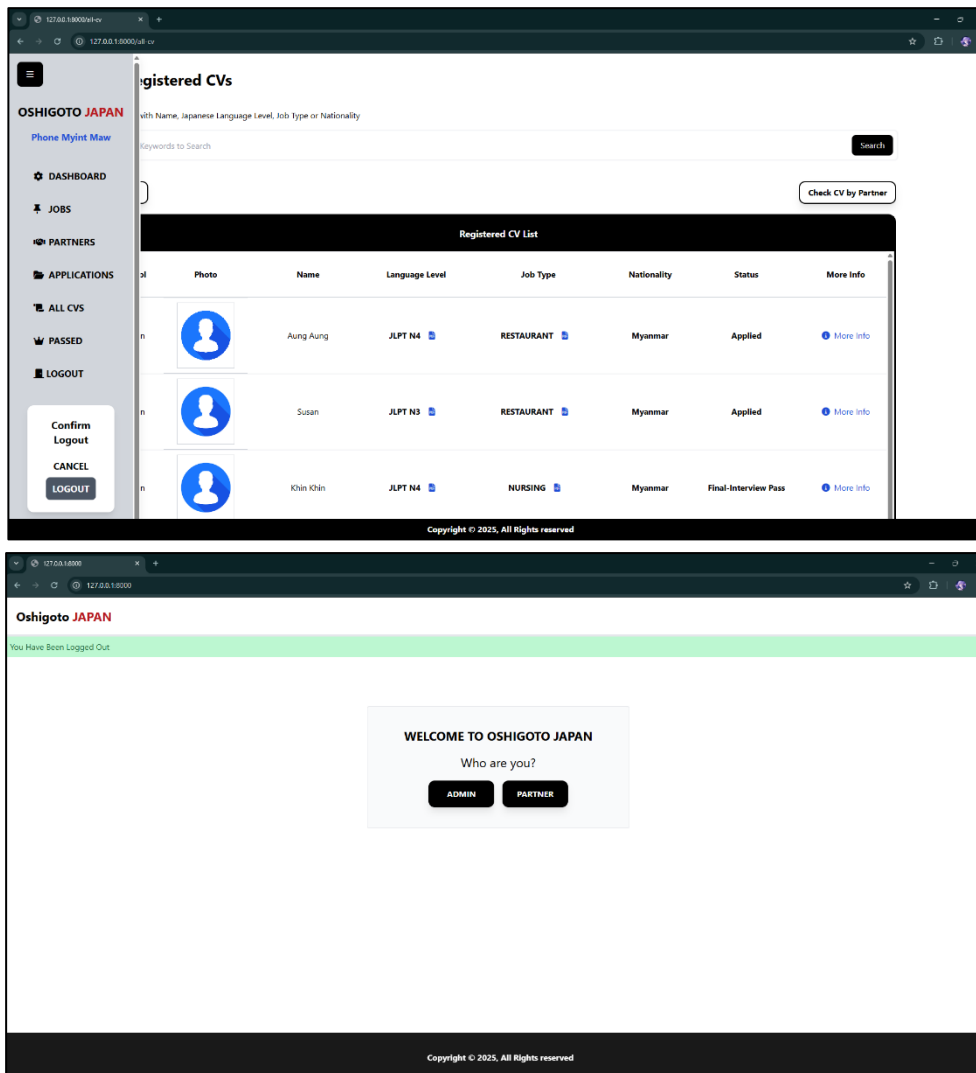


Figure: 4.7.28. Admin Log out Page User Interface

PDF File View User Interface

PDF view for Job

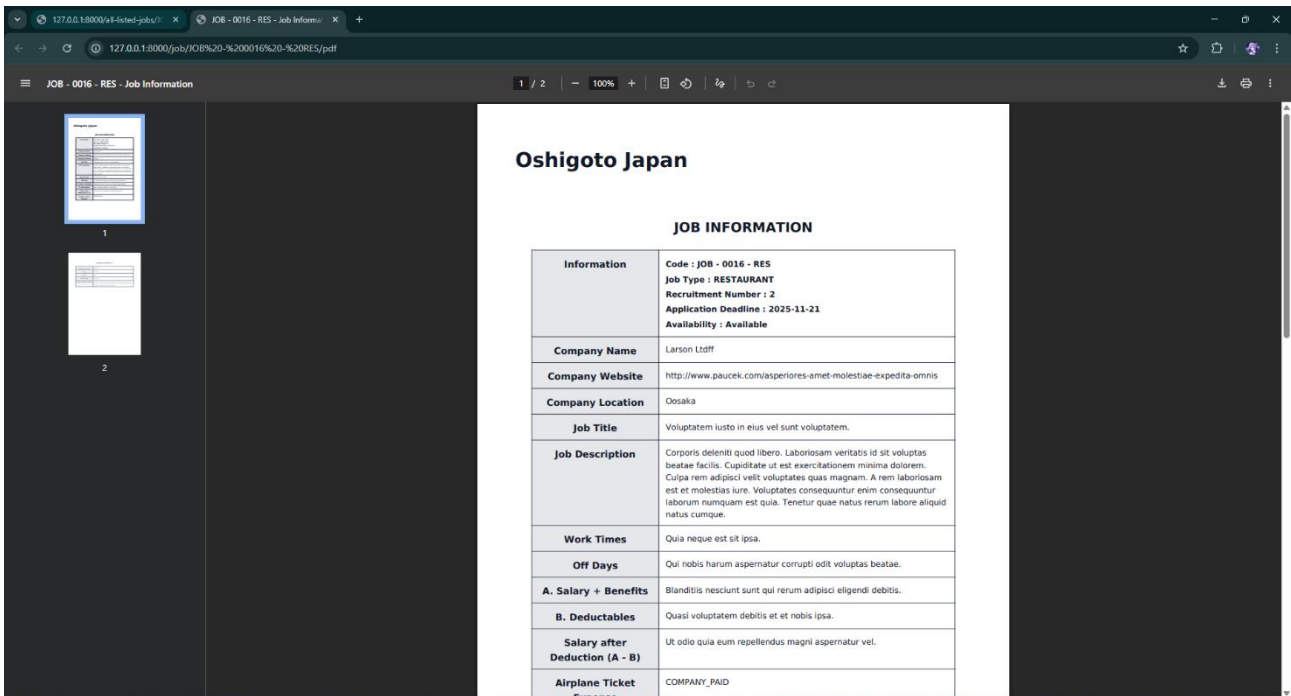


Figure: 4.7.29. PDF View for Job

PDF view for CV

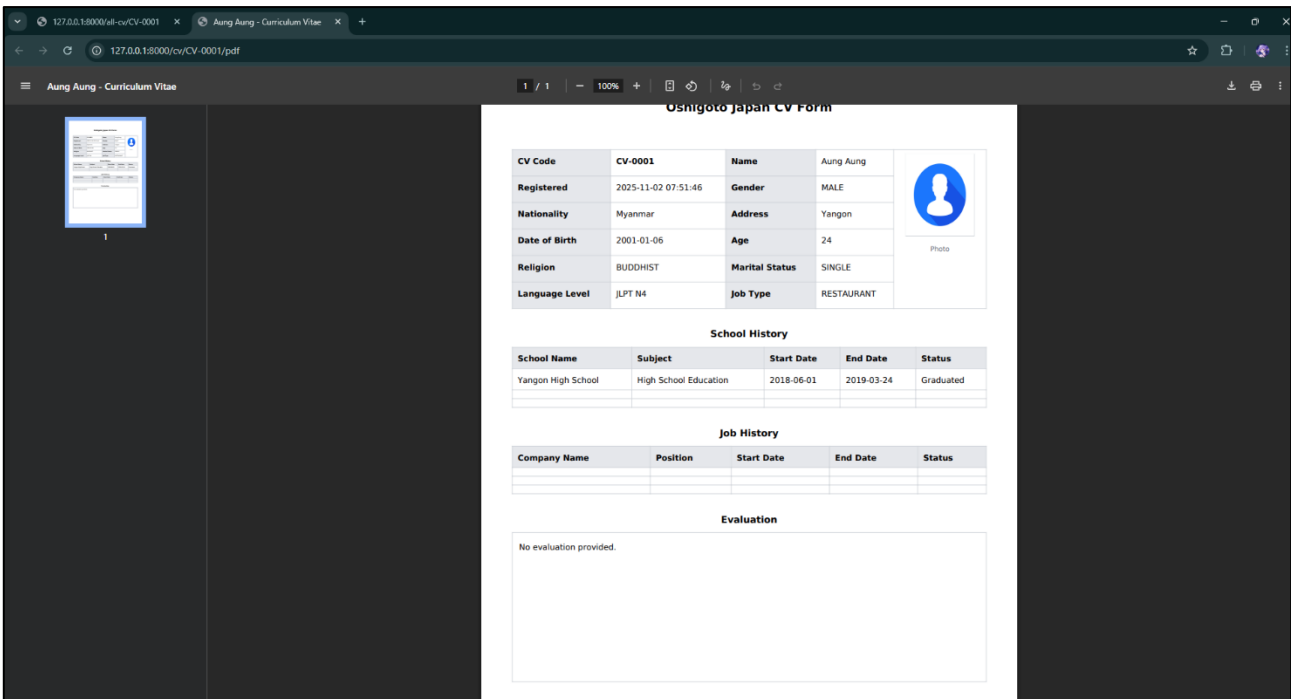


Figure: 4.7.30. PDF View for CV

4.8. Conclusion

Chapter 4 presented the system architecture, class design, data design, entity relationship diagram, and flowcharts, which served as the basis for implementing the Oshigoto Japan System. The graphical user interfaces represented the visual design of the system. The detailed implementation and the coding parts will be presented in Chapter 5.

Chapter 5 – Implementation

5.1. Introduction

The Implementation section presents the overall system web page site navigation, along with database population, coding parts, web page implementation, risk management, project management, and the schedule for training the Oshigoto Japan staff.

5.2. Page Navigation Map

The page navigation map describes how users move through the system. It provides an overview of the structure of the system, showing the connections between all the pages for the partners and the admin users. The illustration below describes the full structure of the Oshigoto Japan System, showing Partner Side and Admin Side Navigation Maps.



Figure: 5.2.1. Partner Side Navigation Map

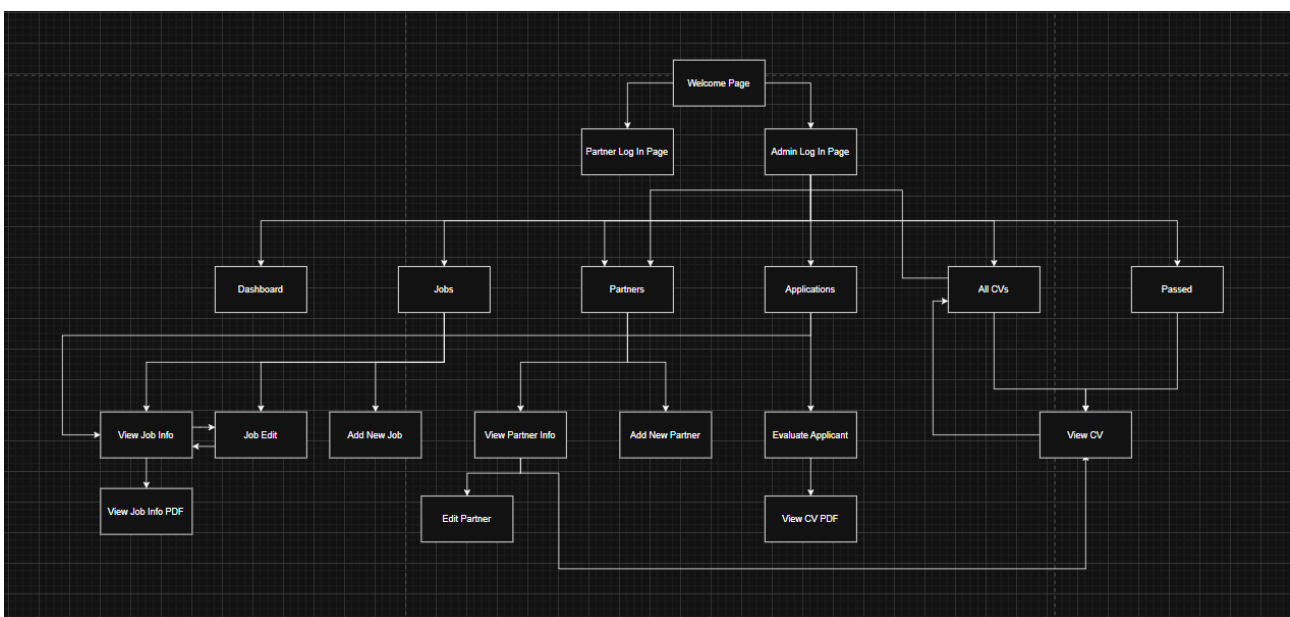


Figure: 5.2.2. Admin Side Navigation Map

5.3. Database Structure and Data Population

Structuring the database and populating data were made using Laravel's migration, models, factory, and seeding functionalities, along with manual data entry through the developed system.

Laravel migrations were used to define and construct the database schema, which includes tables, attributes, and relationships. Each database table has a corresponding Eloquent model that represents the table within the application layer. The models were used to define relationships between tables, such as `hasMany`, `belongsTo`, `belongsToMany`, and `hasOne`, to enable data interaction.

Factories, seeders, and manual data entry were selected as alternatives to MySQL-client-based traditional data population as they provide a faster, more efficient, and more consistent method for generating test data during development.

Database Structure using Migration and Model

User Table Structure

```
0001_01_01_000000_create_users_table.php X
database > migrations > 0001_01_01_000000_create_users_table.php > class > up > Closure
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->rememberToken();
21             $table->timestamps();
22         });
23
24         Schema::create('password_reset_tokens', function (Blueprint $table) {
25             $table->string('email')->primary();
26             $table->string('token');
27             $table->timestamp('created_at')->nullable();
28         });
29
30         Schema::create('sessions', function (Blueprint $table) {
31             $table->string('id')->primary();
32             $table->foreignId('user_id')->nullable()->index();
33             $table->string('ip_address', 45)->nullable();
34             $table->text('user_agent')->nullable();
35             $table->longText('payload');
36             $table->integer('last_activity')->index();
37         });
38     }
39
40     /**
41     * Reverse the migrations.
42     */
43     public function down(): void
44     {
45         Schema::dropIfExists('users');
46         Schema::dropIfExists('password_reset_tokens');
47         Schema::dropIfExists('sessions');
48     }
49 };
50
```

Figure: 5.3.1. Laravel Migration for creating the *Users* Table

Partner Table Structure and Model

```
2025_09_10_094451_create_partner_table.php X
database > migrations > 2025_09_10_094451_create_partner_table.php > ...
1  k?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('partners', function (Blueprint $table) {
15             $table->id();
16             $table->string('partner_id')->unique();
17             $table->string('partner_name');
18             $table->string('email')->unique();
19             $table->string('partner_address');
20             $table->string('partner_nationality');
21             $table->string('password');
22             $table->rememberToken();
23             $table->timestamps();
24         });
25     }
26
27     /**
28     * Reverse the migrations.
29     */
30     public function down(): void
31     {
32         Schema::dropIfExists('partners');
33     }
34 };
35
```

Figure: 5.3.2. Laravel Migration for creating the *partners* Table

```
2025_09_10_094451_create_partner_table.php Partner.php X
app > Models > Partner.php > Partner
9  class Partner extends Authenticatable
27  ];
28
29  public function cvs()
30  {
31      return $this->hasMany(CVForms::class, 'partner_id');
32  }
33
34
```

Figure: 5.3.3. Partner Model for creating relationship between the *partners* Table and the *c_v_forms* table

The Partner model defines a one-to-many relationship with the *c_v_forms* table, allowing a single partner to create multiple CVs.

C_v_forms Table Structure and Model

The figure consists of two side-by-side screenshots of code editors. The left screenshot shows a Laravel migration file named '2025_09_12_102439_create_c_v_forms_table.php'. It defines a migration class that creates a table named 'c_v_forms' with various columns including personal information, education history, and job history. The right screenshot shows the 'CVForms.php' model file, which defines the 'CVForms' model class with relationships to 'Partner', 'Application', and 'JobListing' models.

```

2025_09_12_102439_create_c_v_forms_table.php X CVForms.php
database > migrations > 2025_09_12_102439_create_c_v_forms_table.php > class > up > Closure
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('c_v_forms', function (Blueprint $table) {
15             $table->id();
16             $table->string('cv_code')->unique();
17             $table->string('cv_profile')->nullable();//File
18             $table->foreignId('partner_id')->nullable()->constrained('partners')
19                 ->onDelete('cascade');
20
21             $table->string('cv_name');
22             $table->string('cv_gender');
23             $table->integer('cv_age');
24             $table->date('cv_dob');
25             $table->string('cv_nationality');
26             $table->string('cv_address');
27             $table->string('cv_religion');
28             $table->string('cv_marriage');
29             $table->string('cv_job_type');
30
31             $table->string('cv_job_certificate')->nullable();//File
32
33             $table->string('cv_ip_level');
34
35             $table->string('cv_ip_certificate')->nullable();//File
36
37             $table->integer('cv_ip_study_year');
38             $table->integer('cv_ip_study_month');
39             $table->string('cv_desired_work_length');
40
41             //SCHOOL HISTORY
42             $table->string('cv_schoolhistory1_name');
43             $table->string('cv_schoolhistory1_subject');
44             $table->date('cv_schoolhistory1_start');
45             $table->date('cv_schoolhistory1_end');
46             $table->string('cv_schoolhistory1_status');
47
48             $table->string('cv_schoolhistory2_name')->nullable();
49             $table->string('cv_schoolhistory2_subject')->nullable();
50             $table->date('cv_schoolhistory2_start')->nullable();
51             $table->date('cv_schoolhistory2_end')->nullable();
52             $table->string('cv_schoolhistory2_status')->nullable();
53
54             $table->string('cv_schoolhistory3_name')->nullable();
55             $table->string('cv_schoolhistory3_subject')->nullable();
56             $table->date('cv_schoolhistory3_start')->nullable();
57             $table->date('cv_schoolhistory3_end')->nullable();
58             $table->string('cv_schoolhistory3_status')->nullable();
59
60             //JOB HISTORY
61             $table->string('cv_jobhistory1_name')->nullable();
62             $table->string('cv_jobhistory1_position')->nullable();
63             $table->date('cv_jobhistory1_start')->nullable();
64             $table->date('cv_jobhistory1_end')->nullable();
65             $table->string('cv_jobhistory1_status')->nullable();
66
67             $table->string('cv_jobhistory2_name')->nullable();
68             $table->string('cv_jobhistory2_position')->nullable();
69             $table->date('cv_jobhistory2_start')->nullable();
70             $table->date('cv_jobhistory2_end')->nullable();
71             $table->string('cv_jobhistory2_status')->nullable();
72
73             $table->string('cv_jobhistory3_name')->nullable();
74             $table->string('cv_jobhistory3_position')->nullable();
75             $table->date('cv_jobhistory3_start')->nullable();
76             $table->date('cv_jobhistory3_end')->nullable();
77             $table->string('cv_jobhistory3_status')->nullable();
78
79             //OTHERS
80             $table->string('cv_email')->nullable();
81             $table->string('cv_phone')->nullable();
82
83
84             $table->string('cv_passport_number')->nullable();
85             $table->date('cv_passport_date')->nullable();
86
87             $table->string('cv_passport')->nullable();//File
88
89             $table->integer('cv_ip_visit_time')->nullable();
90             $table->integer('cv_COE_Received')->nullable();
91             $table->integer('cv_COE_Rejected')->nullable();
92
93             $table->string('cv_status')->default('Not Applied');//For Admins Only
94             $table->text('cv_evaluation')->nullable();//For Admins Only
95
96             $table->timestamps();
97
98         });
99     }
100
101     /**
102      * Reverse the migrations.
103      */
104     public function down(): void
105     {
106         Schema::dropIfExists('c_v_forms');
107     }
108 };
109
2025_09_12_102439_create_c_v_forms_table.php X CVForms.php
database > migrations > 2025_09_12_102439_create_c_v_forms_table.php > class > up > Closure
7 return new class extends Migration
8 {
9     public function up(): void
10     {
11         Schema::create('c_v_forms', function (Blueprint $table) {
12             $table->date('cv_schoolhistory2_end')->nullable();
13
14             $table->string('cv_jobhistory2_status')->nullable();
15
16             $table->string('cv_jobhistory3_name')->nullable();
17             $table->string('cv_jobhistory3_position')->nullable();
18             $table->date('cv_jobhistory3_start')->nullable();
19             $table->date('cv_jobhistory3_end')->nullable();
20             $table->string('cv_jobhistory3_status')->nullable();
21
22             //OTHERS
23             $table->string('cv_email')->nullable();
24             $table->string('cv_phone')->nullable();
25
26             $table->string('cv_passport_number')->nullable();
27             $table->date('cv_passport_date')->nullable();
28
29             $table->string('cv_passport')->nullable();//File
30
31             $table->integer('cv_ip_visit_time')->nullable();
32             $table->integer('cv_COE_Received')->nullable();
33             $table->integer('cv_COE_Rejected')->nullable();
34
35             $table->string('cv_status')->default('Not Applied');//For Admins Only
36             $table->text('cv_evaluation')->nullable();//For Admins Only
37
38             $table->timestamps();
39
40         });
41     }
42
43     /**
44      * Reverse the migrations.
45      */
46     public function down(): void
47     {
48         Schema::dropIfExists('c_v_forms');
49     }
50 };

```

Figure: 5.3.4. Laravel Migration for creating the c v forms

The screenshot shows the 'CVForms.php' model file. It defines the 'CVForms' model class which extends 'Model'. It includes three methods: 'partner()' which uses 'belongsTo' to relate to the 'Partner' model; 'application()' which uses 'hasMany' to relate to the 'Application' model; and 'job()' which uses 'belongsToMany' to relate to the 'JobListing' model through a pivot table named 'applications', with 'cv_id' and 'job_id' as foreign keys and 'status' as a pivot column.

```

2025_09_12_102439_create_c_v_forms_table.php X CVForms.php X
app > Models > CVForms.php > CVForms > scopeFilter
8 class CVForms extends Model
9 {
10     public function partner()
11     {
12         return $this->belongsTo(Partner::class, 'partner_id');
13     }
14
15     public function application()
16     {
17         return $this->hasMany(Application::class, 'cv_id');
18     }
19
20     public function job()
21     {
22         return $this->belongsToMany(JobListing::class, 'applications', 'cv_id', 'job_id')
23             ->withPivot('status', 'remarks')
24             ->withTimestamps();
25     }
26 }

```

Figure: 5.3.5. CVForms Model for creating relationship for c_v_forms Table

The c_v_form model forms three relationships: a belongsTo with the Partner model, hasMany with Application model, and belongsToMany with Joblisting model via the applications pivot table.

Joblistings Table Structure and Model

```
2025_09_10_152516_create_joblistings_table.php X joblisting.php
database > migrations > 2025_09_10_152516_create_joblistings_table.php > class > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('joblistings', function (Blueprint $table) {
15             $table->id();
16             $table->string('job_code')->unique();
17             $table->string('job_type');
18             $table->integer('recruit_number');
19             $table->date('application_deadline');
20             $table->string('company_name');
21             $table->string('company_website');
22             $table->string('company_location');
23             $table->string('job_title');
24             $table->text('job_description');
25             $table->text('work_time');
26             $table->string('off_days');
27             $table->text('salary_benefits');
28             $table->text('deductables');
29             $table->text('after_deduction');
30             $table->string('airticket_exp');
31             $table->string('required_jp_level');
32             $table->string('age_range');
33             $table->string('job_gender');
34             $table->string('job_nationality');
35             $table->text('other_requirements') -> nullable();
36             $table->string('job_availability')->default('Available');
37             $table->timestamps();
38         });
39     }
40
41     /**
42     * Reverse the migrations.
43     */
44     public function down(): void
45     {
46         Schema::dropIfExists('joblistings');
47     }
48 };
49
```

Figure: 5.3.6. Laravel Migration for creating the *joblistings* Table

```
2025_09_10_152516_create_joblistings_table.php joblisting.php X
app > Models > joblisting.php > joblisting > booted > Closure
8  class joblisting extends Model
101 public function application()
102 {
103     return $this->hasMany(Application::class, 'job_id');
104 }
105
106 public function cvs()
107 {
108     return $this->belongsToMany(CVForms::class, 'applications', 'job_id', 'cv_id')
109         ->withPivot('status', 'remarks')
110         ->withTimestamps();
111 }
112
113
114 }
115
```

Figure: 5.3.7. Joblisting Model for creating relationship for *joblistings* Table

The *joblisting* model forms two relationships: *hasMany* with the *Application* model, and *belongsToMany* with the *CVForms* model via the *applications* pivot table.

Applications Table Structure and Model

```
2025_09_14_041729_create_applications_table.php × Application.php
database > migrations > 2025_09_14_041729_create_applications_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('applications', function (Blueprint $table) {
15             $table->id();
16
17             $table->foreignId('job_id')->constrained('joblistings')->onDelete('cascade');
18             $table->foreignId('cv_id')->constrained('c_v_forms')->onDelete('cascade');
19             $table->string('status')->default('Pending');
20             $table->text('remarks')->nullable();
21             $table->timestamps();
22
23             $table->unique(['cv_id', 'job_id']);
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('applications');
33     }
34 };
35
```

Figure: 5.3.8. Laravel Migration for creating the *applications* Table

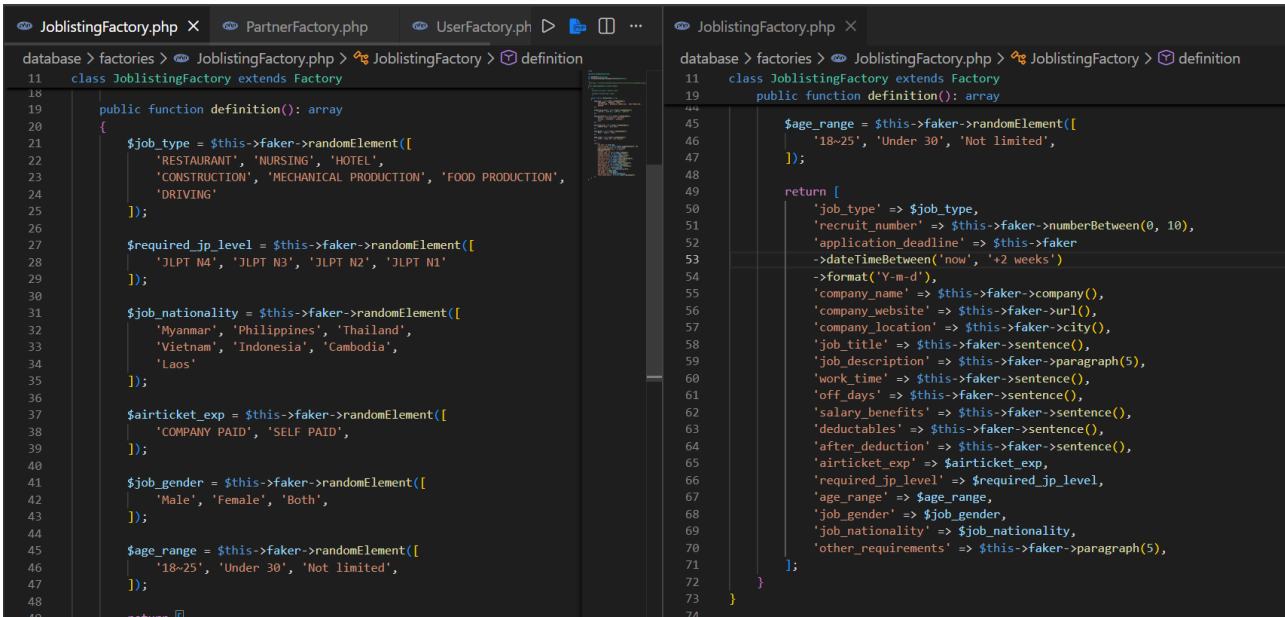
```
2025_09_14_041729_create_applications_table.php Application.php ×
app > Models > Application.php > application > scopeFilter
8 class application extends Model
19
20     public function cv()
21     {
22         return $this->belongsTo(CVForms::class, 'cv_id');
23     }
24
25     public function job()
26     {
27         return $this->belongsTo(Joblisting::class, 'job_id');
28     }
29
```

Figure: 5.3.9. Application Model for creating relationship for *applications* Table

The application model forms two relationships: belongsTo with the CVForms model and the Joblistings model.

Data Population with Factories and Database Seeder

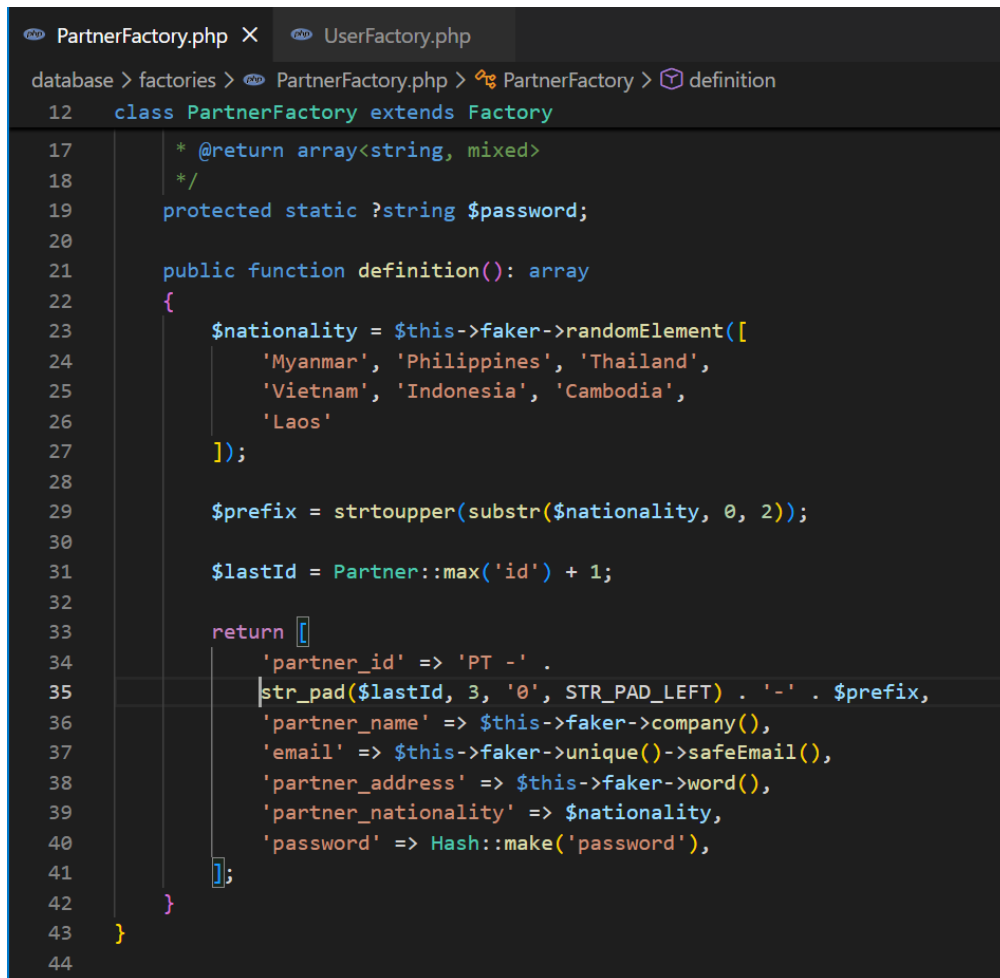
Joblisting Factory



```
11 class JoblistingFactory extends Factory
18
19 public function definition(): array
20 {
21     $job_type = $this->faker->randomElement([
22         'RESTAURANT', 'NURSING', 'HOTEL',
23         'CONSTRUCTION', 'MECHANICAL PRODUCTION', 'FOOD PRODUCTION',
24         'DRIVING'
25     ]);
26
27     $required_jp_level = $this->faker->randomElement([
28         'JLPT N4', 'JLPT N3', 'JLPT N2', 'JLPT N1'
29     ]);
30
31     $job_nationality = $this->faker->randomElement([
32         'Myanmar', 'Philippines', 'Thailand',
33         'Vietnam', 'Indonesia', 'Cambodia',
34         'Laos'
35     ]);
36
37     $airticket_exp = $this->faker->randomElement([
38         'COMPANY PAID', 'SELF PAID',
39     ]);
40
41     $job_gender = $this->faker->randomElement([
42         'Male', 'Female', 'Both',
43     ]);
44
45     $age_range = $this->faker->randomElement([
46         '18-25', 'Under 30', 'Not limited',
47     ]);
48
49 }
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
```

Figure: 5.3.10. Joblisting Factory for Job data population

Partner Factory



```
12 class PartnerFactory extends Factory
17
18 * @return array<string, mixed>
19 */
20 protected static ?string $password;
21
22 public function definition(): array
23 {
24     $nationality = $this->faker->randomElement([
25         'Myanmar', 'Philippines', 'Thailand',
26         'Vietnam', 'Indonesia', 'Cambodia',
27         'Laos'
28     ]);
29
30     $prefix = strtoupper(substr($nationality, 0, 2));
31
32     $lastId = Partner::max('id') + 1;
33
34     return [
35         'partner_id' => 'PT -' .
36         str_pad($lastId, 3, '0', STR_PAD_LEFT) . '-' . $prefix,
37         'partner_name' => $this->faker->company(),
38         'email' => $this->faker->unique()->safeEmail(),
39         'partner_address' => $this->faker->word(),
40         'partner_nationality' => $nationality,
41         'password' => Hash::make('password'),
42     ];
43 }
44
```

Figure: 5.3.11. Partner Factory for Partner data population

Admin User Factory

```
database > factories > UserFactory.php > ...
11  */
12  class UserFactory extends Factory
13  {
14      /**
15       * The current password being used by the factory.
16       */
17      protected static ?string $password;
18
19      /**
20       * Define the model's default state.
21       *
22       * @return array<string, mixed>
23       */
24      public function definition(): array
25      {
26          return [
27              'name' => fake()->name(),
28              'email' => fake()->unique()->safeEmail(),
29              'email_verified_at' => now(),
30              'password' => static::$password ??= Hash::make('password'),
31              'remember_token' => Str::random(10),
32          ];
33      }
34
35      /**
36       * Indicate that the model's email address should be unverified.
37       */
38      public function unverified(): static
39      {
40          return $this->state(fn (array $attributes) => [
41              'email_verified_at' => null,
42          ]);
43      }
44  }
45
```

Figure: 5.3.12. User Factory for Admin User data population

Database Seeder

```
DatabaseSeeder.php X
database > seeders > DatabaseSeeder.php > DatabaseSeeder > run
13
14  class DatabaseSeeder extends Seeder
15  {
16      /**
17       * Seed the application's database.
18       */
19      public function run(): void
20      {
21
22          User::factory()->create([
23              'name' => 'Phone Myint Maw',
24              'email' => 'admin@oshigoto.com',
25              'password' => 'pmm552002'
26          ]);
27
28          joblisting::factory(15)->create();
29
30          $partners = [
31              [
32                  'partner_name' => 'Toshin',
33                  'email' => 'mmtoshin@gmail.com',
34                  'partner_address' => 'Toshin Language School',
35                  'partner_nationality' => 'Myanmar',
36                  'password' => Hash::make('toshin2025'),
37              ],
38              [
39                  'partner_name' => 'Hikari',
40                  'email' => 'Hikari@gmail.com',
41                  'partner_address' => 'Hikari Language School',
42                  'partner_nationality' => 'Indonesia',
43                  'password' => Hash::make('hikari2025'),
44              ],
45              [
46                  'partner_name' => 'Nihongo',
47                  'email' => 'nihongo@gmail.com',
48                  'partner_address' => 'Nihongo Language School',
49                  'partner_nationality' => 'Laos',
50                  'password' => Hash::make('nihon2025'),
51              ],
52          ];
53
54          foreach($partners as $partnerData){
55              Partner::create($partnerData);
56          }
57      }
58  }
59
60
61
62
```

Figure: 5.3.13. Database for Data population from Factories

5.4. Coding and Implementation

For the implementation and coding part of the project, as explained in the technology review section of Chapter 2, HTML, Tailwind CSS, Traditional CSS, and Laravel were chosen as programming languages for developing the Oshigoto Japan System. Each major feature with the functional source code and corresponding system outputs will be shown in the sub-sections below. The full project source code is provided in the appendix.

Authentication and Authorization

Route Definition

The routes define the entry points for partner/admin login. GET loads the login page through the authenticate login functions in controllers, while POST sends inputs from the user interface to authenticate users. The screenshot below shows the defined routes for login and logout functions for partner and admin users.

```
//Log In Log Out

// Log IN CHOOSE Page
Route::get('/', function () {
    return view('login.login-choose');
});

// Log IN ADMIN Page
Route::get('/admin-login', [AdminController::class, 'login']);

// Log IN ADMIN Page ROUTE
Route::post('/admin/authenticate', [AdminController::class, 'adminLogin']);

// Log OUT ADMIN
Route::post('/admin/logout', [AdminController::class, 'logout']);

// Log IN PARTNER Page view
Route::get('/partner-login', [PartnerController::class, 'login']);

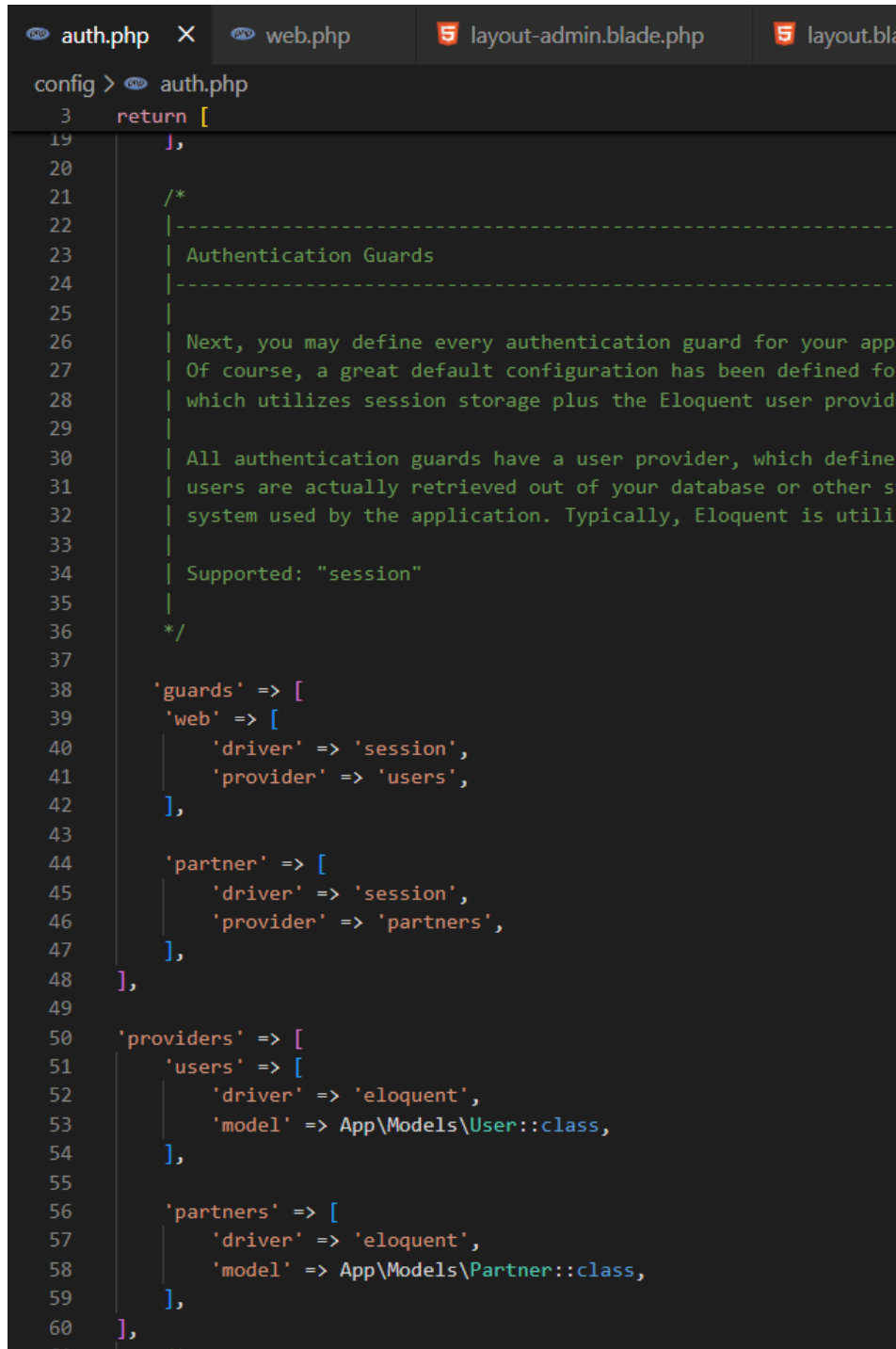
// Log IN PARTNER Page ROUTE
Route::post('/partner/authenticate', [PartnerController::class, 'partnerLogin']);

// Log OUT PARTNER
Route::post('/partner/logout', [PartnerController::class, 'logout']);
```

Figure: 5.4.1. Login, Logout and Authentication Route Definitions

Guards and Providers Configuration

Laravel is configured with two authentication guards: web (Admins) and partner, and the two providers link guards with the respective database tables through corresponding Eloquent models. This enables the separation of user roles and role-based authentication.



```
config > auth.php
3   return [
19      ],
20
21      /*
22      |-----
23      | Authentication Guards
24      |-----
25      |
26      | Next, you may define every authentication guard for your application.
27      | Of course, a great default configuration has been defined for you
28      | which utilizes session storage plus the Eloquent user provider.
29      |
30      | All authentication guards have a user provider, which defines how
31      | users are actually retrieved out of your database or other storage
32      | system used by the application. Typically, Eloquent is utilized.
33      |
34      | Supported: "session"
35      |
36      */
37
38      'guards' => [
39          'web' => [
40              'driver' => 'session',
41              'provider' => 'users',
42          ],
43
44          'partner' => [
45              'driver' => 'session',
46              'provider' => 'partners',
47          ],
48      ],
49
50      'providers' => [
51          'users' => [
52              'driver' => 'eloquent',
53              'model' => App\Models\User::class,
54          ],
55
56          'partners' => [
57              'driver' => 'eloquent',
58              'model' => App\Models\Partner::class,
59          ],
60      ],
61      /*
```

Figure: 5.4.2. Authentication Guards and Providers Configuration

Controller Logic

Both partner and admin users use authentication through Email/Password validation. The system uses Laravel's `Auth::guard()->attempt()` method for verifying credentials submitted by the users through the login form. In addition, when the users attempt to log out, the system invalidates and regenerates the session token, ensuring the previous session becomes unusable.



```
app > Http > Controllers > AdminUserController.php > AdminUserController > adminLogin
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\User;
6  use Illuminate\Http\Request;
7  use Illuminate\Validation\Rule;
8  use Illuminate\Support\Facades\Auth;
9
10 class AdminUserController extends Controller
11 {
12     //View Admin Log In Page
13     public function login(){
14         return view('login.login-admin');
15     }
16
17     //Authenticate Admin
18     public function adminLogin(Request $request){
19
20         $formFields = $request->validate([
21             'email' => ['required', 'email'],
22             'password' => 'required',
23         ]);
24
25         if (Auth::guard('web')->attempt($formFields, $request->filled('remember'))){
26
27             $request->session()->regenerate();
28             return redirect('/admin-home') -> with('message', 'You have logged in!');
29         }
30
31         return back()->withErrors(['email' => 'Invalid Credentials']) -> onlyInput('email');
32     }
33
34
35     //Admin Log Out
36     public function logout(Request $request){
37
38         Auth::guard('web')->logout();
39
40         $request->session()->invalidate();
41
42         $request->session()->regenerateToken();
43
44         return redirect('/')->with('message', 'You Have Been Logged Out');
45     }
46 }
47
48 }
49
```

Figure: 5.4.3. Authentication Controller Logic for Admin users

```
layout-admin.blade.php layout.blade.php PartnerController.php X AdminUserController.php
app > Http > Controllers > PartnerController.php > PartnerController > update
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\CVForms;
6 use App\Models\partner;
7 use Illuminate\Http\Request;
8 use Illuminate\Validation\Rule;
9 use Illuminate\Support\Facades\Auth;
10 use Illuminate\Support\Facades\Hash;
11 use Illuminate\Support\Facades\Crypt;
12
13 class PartnerController extends Controller
14 {
15     //View Partner Log In Page
16     public function login(){
17
18         return view('login.login-partner');
19     }
20
21
22     //Authenticate Partner
23     public function partnerLogin(Request $request){
24
25         $formFields = $request->validate([
26             'email' => ['required', 'email'],
27             'password' => 'required',
28         ]);
29
30         if (Auth::guard('partner')->attempt($formFields, $request->filled('remember'))) {
31
32             $request->session()->regenerate();
33             return redirect('/home') -> with('message', 'You have logged in!');
34
35         }
36
37         return back()->withErrors(['email' => 'Invalid Credentials']) -> onlyInput('email');
38     }
39
40     //Partner Log Out
41     public function logout(Request $request){
42
43         Auth::guard('partner')->logout();
44
45         $request->session()->invalidate();
46
47         $request->session()->regenerateToken();
48
49         return redirect('/')->with('message', 'You have been logged out.');
```

Figure: 5.4.3. Authentication Controller Logic for Partner users

Authorization in Layouts

By authenticating in layouts using `@auth()` and `@endauth`, the contents between the statements will be hidden for the users who are not logged in, and only the login button will appear.

```
@auth('partner')

<li>
  <span class="font-old-uppercase">
    Welcome, {{ auth('partner')->user()->partner_name }} !
  </span>
</li>

<li>
  <p>/</p>
</li>

<li>
  <form method="POST" action="/partner/logout" class="inline">
    @csrf
    <button type="submit" class="transition duration-300 hover:underline">
      <i class="fa-solid fa-door-closed col-span-1"></i>
      <span class="col-span-4">Logout</span>
    </button>
  </form>
</li>

</ul>
</nav>

<main>
  {{{slot}}}
</main>

<footer class="fixed bottom-0 left-0 w-full flex items-center justify-start font-bold
bg-black text-white h-10 md:justify-center">
  <p class="m1-2">Copyright &copy; 2025, All Rights reserved</p>
</footer>

@else

<div class="flex justify-end p-3 pr-10 font-bold text-2xl">
  <p>
    You are not Logged In
  </p>
</div>

<div class="flex justify-end p-3 pr-10 text-2xl">
  <a href="/" class="m1-5 inline-block border-2 font-bold
border-black text-white bg-black py-1 px-4 rounded-xl uppercase
hover:text-black hover:bg-white">
    Log In
  </a>
</div>

@endauth
```

Figure: 5.4.4. Authorization in Partner Page Layout

```

    @auth('web')

    <div class="font-bold text-lg text-center mb-5 text-blue-700">
      |   {{ auth('web')->user()->name }}
    </div>
    <ul class="text-lg font-bold p-5">...
    </ul>

  </aside>

</div>

<main class="mt-3">
  <div>
    |
    |   {{ $slot }}
    |
  </div>
</main>

@else

<div class="flex justify-end p-3 pr-10 font-bold text-2xl">
  <p>
    |   You are not Logged In
  </p>
</div>

<div class="flex justify-end p-3 pr-10 text-2xl">
  <a href="/" class="ml-5 inline-block border-2 font-bold
    |   border-black text-white bg-black py-1 px-4 rounded-xl uppercase
    |   hover:text-black hover:bg-white">
    |   Log In
  </a>
</div>

@endauth

```

Figure: 5.4.5. Authorization in Admin Page Layout

Blade Templates and components

Partner Page Layout and Navigation Bar

```
layout.blade.php X
resources > views > components > layout.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"
9     integrity="sha512-KfkfwYDsLkIlwQp6LF182Mq8XgU9AA1QvwfYqN4PheIQQ5vqcyVLLD9a4hXg13uQjoxT6ENos0Wazqkgel0g=="
10    crossorigin="anonymous" referrerpolicy="no-referrer" />
11   <script src="//unpkg.com/alpinejs"></script>
12   <script src="https://cdn.tailwindcss.com"></script>
13
14   <title></title>
15
16 </head>
17
18 <body class="mb-48">
19
20   <nav class="inline-flex w-full font-bold bg-white text-black justify-between items-center p-5 shadow-lg">
21     <h1 class="text-2xl">
22       Oshigoto <span class="text-red-700">JAPAN</span>
23     </h1>
24
25     <ul class="flex justify-center space-x-6 mr-6 text-lg">
26
27       @auth('partner')
28
29         <li>
30           <span class="font-6d uppercase">
31             Welcome, {{ auth('partner')->user()->partner_name }} !
32           </span>
33         </li>
34
35         <li>
36           <p/></p>
37         </li>
38
39         <li>
40
41           <form method="POST" action="/partner/logout" class="inline">
42             @csrf
43             <button type="submit" class="transition duration-300 hover:underline">
44               <i class="fa-solid fa-door-closed col-span-1"></i>
45               <span class="col-span-4">Logout</span>
46             </button>
47           </form>
48
49         </li>
50
51       </ul>
52     </nav>
53
54     <main>
55
56       {{{slot}}}
57
58     </main>
59
60     <footer class="fixed bottom-0 left-0 w-full flex items-center justify-start font-bold
61       bg-black text-white h-10 md:justify-center">
62       <p class="m1-2">Copyright ©copy; 2025, All Rights reserved</p>
63
64     </footer>
65
66     @else
67
68     <div class="flex justify-end p-3 pr-10 font-bold text-2xl">
69       <p>
70         You are not Logged In
71       </p>
72     </div>
73
74     <div class="flex justify-end p-3 pr-10 text-2xl">
75
76       <a href="/" class="m1-5 inline-block border-2 font-bold
77         border-black text-white bg-black py-1 px-4 rounded-xl uppercase
78         hover:text-black hover:bg-white">
79         Log In
80       </a>
81
82     </div>
83
84     @endauth
85
86   </body>
87
88 </html>
89
90
```

Figure: 5.4.7. Partner Page Overall Layout

```

layout.blade.php  _partner-nav.blade.php X
items-center.p-5 > div > a.transition.duration-700.inline-block.border-2.font-bold.border-black.text-black.py-2.px-4.rounded-xl.uppercase.
1 <section class="flex font-bold bg-white text-black justify-between items-center p-5">
2
3 <div>
4 <ul class="grid grid-cols-1 lg:grid-cols-3 space-x-6 text-lg">
5
6 <li class="col-span-1">
7 <a href="/home" class="inline-block font-bold border-black text-black uppercase hover:underline">
8 <i class="fa-solid fa-list"></i> Registered CV List
9 </a>
10 </li>
11
12 <li class="col-span-1">
13 <a href="/partner-applied-list"
14 class="inline-block font-bold border-black text-black uppercase hover:underline">
15 <i class="fa-solid fa-check-to-slot"></i> Applied Job List
16 </a>
17 </li>
18
19 <li class="col-span-1">
20 <a href="/apply-jobs" class="inline-block font-bold border-black text-black uppercase hover:underline">
21 <i class="fa-solid fa-building"></i> Apply New Jobs
22 </a>
23 </li>
24 </ul>
25 </div>
26
27 <div>
28 <a href="/new-cv-register" class="transition duration-700 inline-block border-2 font-bold border-black
29 text-black py-2 px-4 rounded-xl
30 uppercase hover:bg-black hover:text-white hover:animate-pulse shadow-lg">
31 <i class="fa-solid fa-plus"></i> Make new CV Forms
32 </a>
33 </div>
34 </section>
35
36
37

```

Figure: 5.4.8. Partner Page Navigation Bar Layout

```

layout.blade.php  _partner-nav.blade.php  partner-home.blade.php X
resources > views > partner > partner-home.blade.php > x-layout > main.flex.justif
1 <x-layout>
2
3 @include('components._partner-nav')
4
5 <main class="flex justify-center mb-20">...
217 </main>
218
219 </x-layout>
220

```

Figure: 5.4.9. Partner Home Page with Overall Partner Layout and Navigation Bar Integration

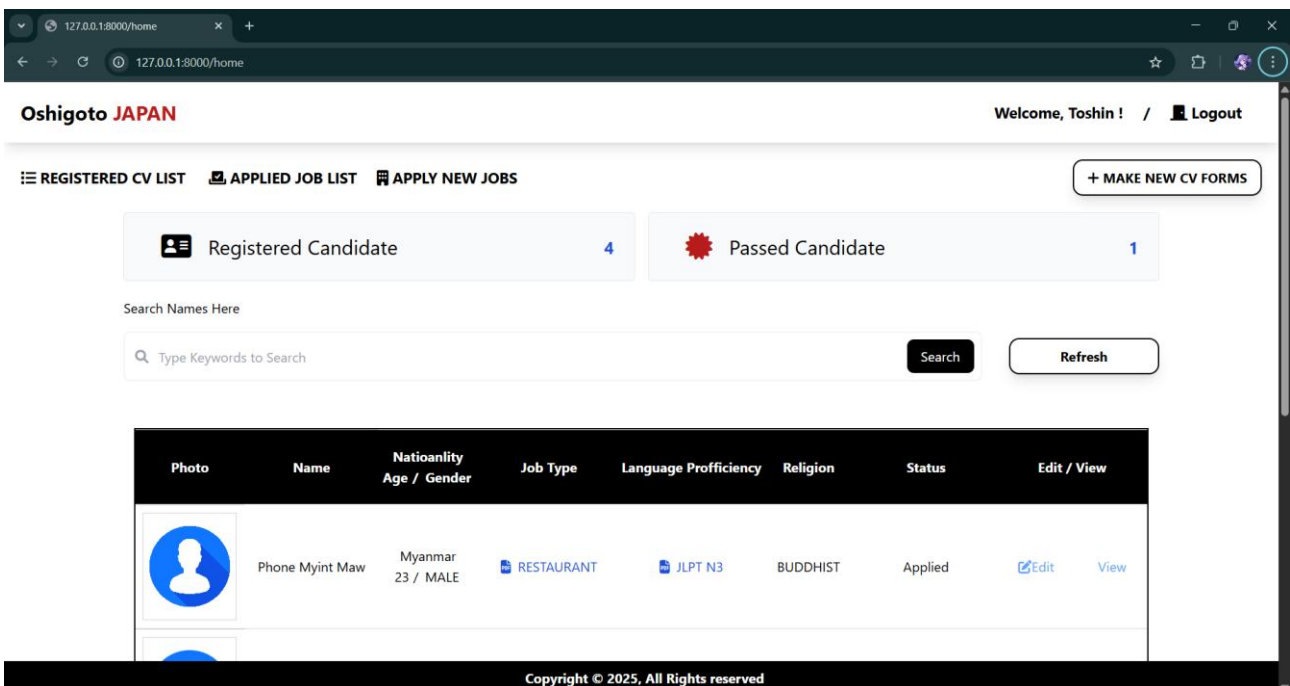


Figure: 5.4.10. Partner Home Page User Interface with Overall Partner Layout and Navigation

The <x-layout> component defines the base layout structure. All the contents in this tag is inserted into the '{{ \$slot }}' placeholder within the main layout file. The "@include('components._partner-nav')" statement loads the navigation bar component in the page, ensuring the partner navigation menu appears consistently across partner pages where this statement is used.

Admin Page and Navigation Layout

```

admin-home.blade.php
1 <doctype html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/css/all.min.css"
9 integrity="sha512-Kk6wGQVjVyT9hlcK+Kl9uB+WAX8q1WjK6FwSc78UwYEVz9UmJZZ9107606El2Waj87wF76oELG0emz0g10m="
10 crossorigin="anonymous" refererPolicy="no-referrer" />
11 <script src="https://cdn.jsdelivr.net/npm/axios" />
12 <script src="https://cdn.jsdelivr.net/npm/axios" />
13
14 <title></title>
15
16 </head>
17
18 <body>
19
20 <div x-data="{open: false}" class="flex bg-gray-100 relative">
21
22 <button @click="open = !open" class="fixed w-[50px] top-4 left-4 z-50 p-3 bg-black
23 text-white rounded-lg hover:bg-gray-700
24 transition">
25 <i class="fa-solid fa-bars">/i>
26 </button>
27
28 <aside :class="open ? 'translate-x-0 w-64' : '-translate-x-full w-64'" class="fixed top-0 left-0 h-screen
29 bg-gray-200 dark:bg-gray-300 p-4 transition-all
30 duration-300 ease-in-out shadow-lg z-40 overflow-y-auto">
31
32 <div class="font-bold text-2xl text-center mb-5 mt-20">
33 OSHIGOTO <span class="text-red-700">JAPAN</span>
34 </div>
35
36 @auth('web')
37
38 <div class="font-bold text-lg text-center mb-5 text-blue-700">
39 {{ auth('web')->user()->name }}
40 </div>
41
42 <ul class="text-lg font-bold p-5">
43
44 <li class="mb-5 justify-right items-center">
45 <a href="/admin-home" class="transition duration-300 grid grid-cols-5 uppercase
46 text-black bg-transparent
47 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
48 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
49 <i class="fa-solid fa-gear col-span-1">/i>
50 <span class="col-span-4">Dashboard</span>
51 </a>
52 </li>
53
54 <li class="mb-5 justify-right items-center">
55 <a href="/all-listed-jobs" class="transition duration-300 grid grid-cols-5 uppercase
56 text-black bg-transparent
57 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
58 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
59 <i class="fa-solid fa-thumbs-up col-span-1">/i>
60 <span class="col-span-4">Jobs</span>
61 </a>
62 </li>
63
64 <li class="mb-5 justify-right items-center">
65 <a href="/all-partners" class="transition duration-300 grid grid-cols-5 uppercase
66 text-black bg-transparent
67 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
68 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
69 <i class="fa-solid fa-handshake col-span-1">/i>
70 <span class="col-span-4">Partners</span>
71 </a>
72 </li>
73
74 <li class="mb-5 justify-right items-center">
75 <a href="/admin-applications" class="transition duration-300 grid grid-cols-5 uppercase
76 text-black bg-transparent
77 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
78 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
79 <i class="fa-solid fa-folder-open col-span-1">/i>
80 <span class="col-span-4">Applications</span>
81 </a>
82 </li>
83
84 <li class="mb-5 justify-right items-center">
85 <a href="/all-cv" class="transition duration-300 grid grid-cols-5 uppercase
86 text-black bg-transparent
87 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
88 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
89 <i class="fa-solid fa-scroll col-span-1">/i>
90 <span class="col-span-4">All CVs</span>
91 </a>
92 </li>
93
94 <li class="mb-5 justify-right items-center">
95 <a href="/admin-completed" class="transition duration-300 grid grid-cols-5 uppercase
96 text-black bg-transparent
97 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
98 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
99 <i class="fa-solid fa-crown col-span-1">/i>
100 <span class="col-span-4">Passed</span>
101 </a>
102 </li>
103
104 </ul>
105
106 </aside>
107
108 <main class="mt-3">
109 <div>
110
111 <div class="flex justify-end p-3 pr-10 font-bold text-2xl">
112 <p>
113 | You are not Logged In
114 </p>
115 </div>
116
117 <div class="flex justify-end p-3 pr-10 text-2xl">
118 <a href="/" class="ml-5 inline-block border-2 font-bold
119 border-black text-white bg-black py-1 px-4 rounded-xl uppercase
120 hover:text-black hover:bg-white">
121 Log In
122 </a>
123 </div>
124
125 @endauth
126
127 <div class="fixed z-50 bottom-0 left-0 w-full flex items-center justify-start font-bold
128 bg-black text-white h-10 justify-center">
129 <p class="ml-2">Copyright © Copy, 2025, All Rights reserved.</p>
130 </div>
131 </body>
132 </html>
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure: 5.4.11. Admin Overall Page and Navigation Bar Layout

```

admin-home.blade.php X layout-admin.blade.php
resources > views > company > admin-home.blade.php > ...
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 @include('components._message')
8
9 <div class="font-bold text-3xl mt-5 mb-10">
10 Admin Dashboard
11 </div>
12
13 <!-- STATISTICS SECTION 1 -->
14 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-5">...
15 </div>
16 <!-- STATISTICS SECTION 2 -->
17 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-5">...
18 </div>
19 <!-- SEARCH DATE SECTION -->
20 <div class="flex flex-col justify-center mb-5 bg-gray-50 border border-gray-200 rounded p-5 text-center">...
21 </div>
22 <!-- Jobs Bar Chart -->
23 <div class="flex-row justify-left bg-gray-50 border border-gray-200 rounded p-5 mb-10">...
24 </div>
25 <!-- Pie Chart -->
26 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-10">...
27 </div>
28 <!-- Calendar -->
29 <div class="flex justify-center mb-10">...
30 </div>
31 </div>
32 </div>
33 <!-- Chart JS Scripts -->
34 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
35 </script>
36 </x-layout-admin>
37

```

Figure: 5.4.12. Integration of Admin Overall Page and Navigation Bar Layout

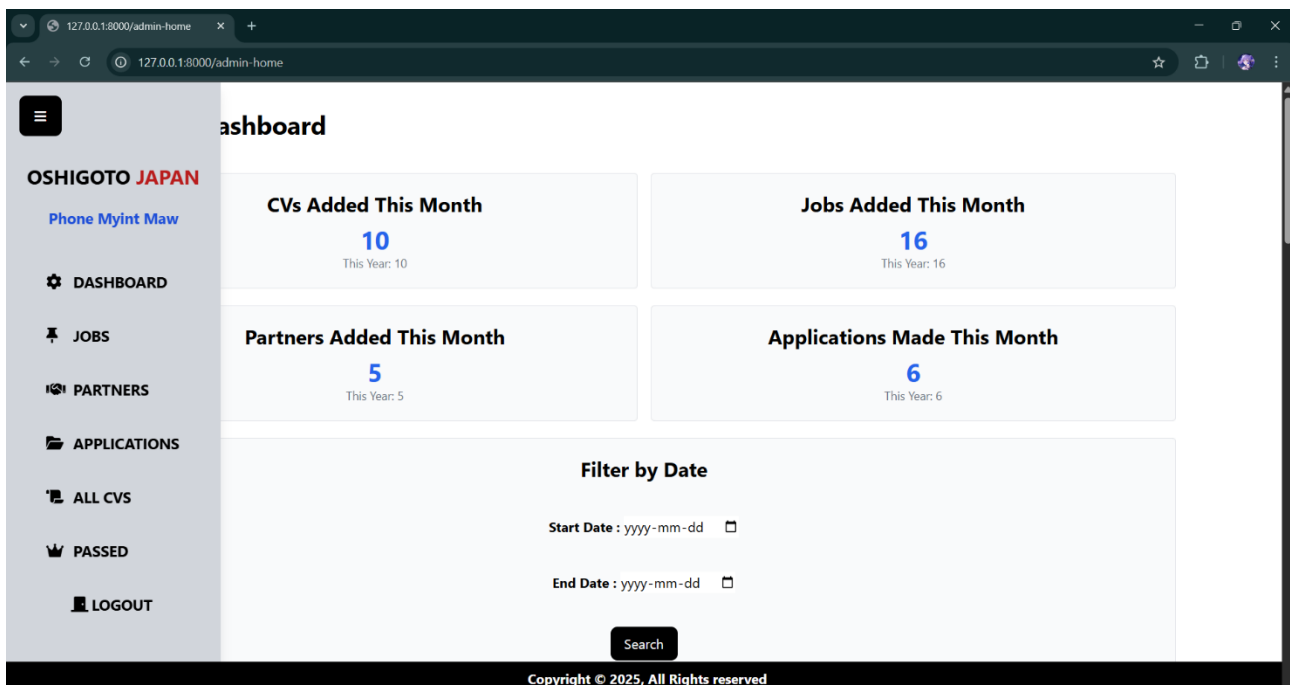
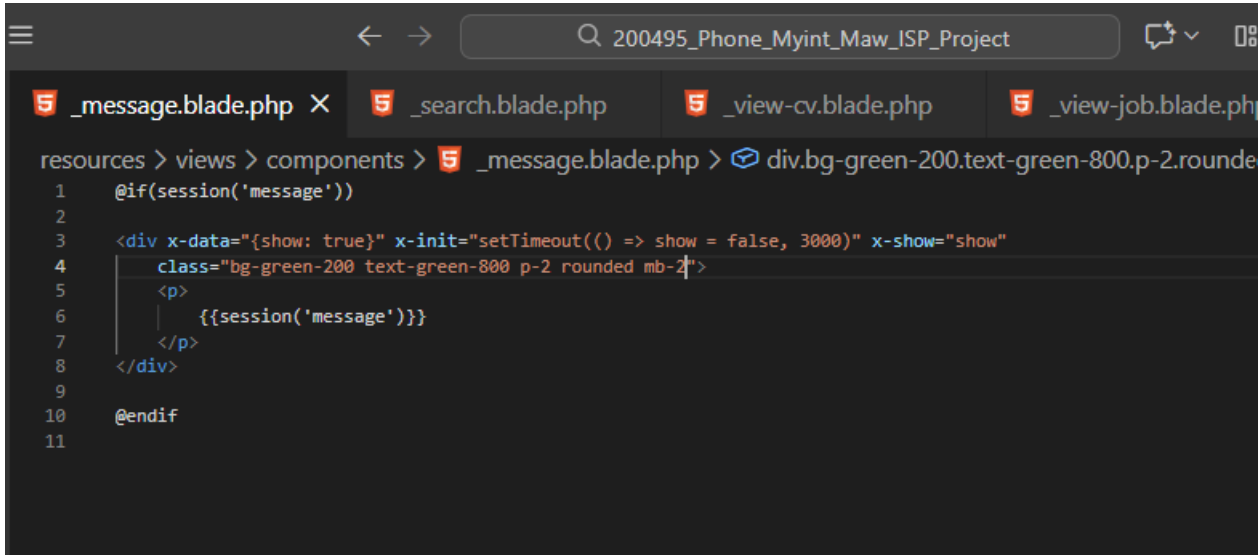


Figure: 5.4.13. Admin Home Page User Interface with Using Overall page and Navigation Bar Layout

The `<x-layout-admin>` component defines both the base layout structure and the side navigation bar.

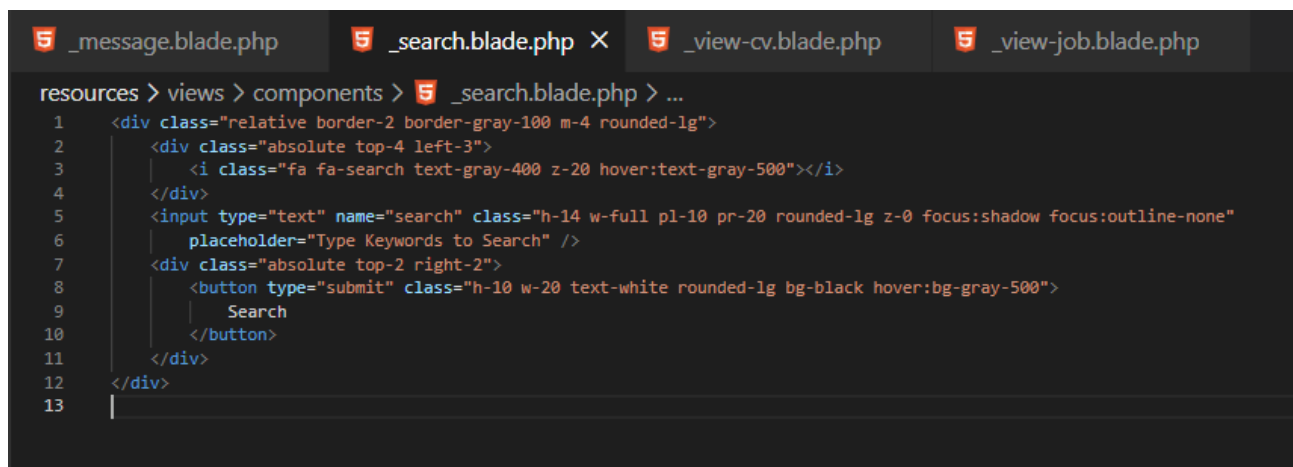
Reusable Components

With the blade layout structure, reusable components can be inserted into any pages using the '@include()' statement, which loads the specified component into the current view. The components include a message component for system messages, search components for keyword-filter function, a view CV for plain CV view function, and a view Job for Plain Job view function.

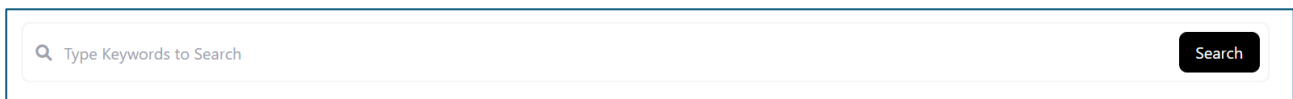


```
resources > views > components > _message.blade.php > div.bg-green-200.text-green-800.p-2.rounde
1 @if(session('message'))
2
3 <div x-data="{show: true}" x-init="setTimeout(() => show = false, 3000)" x-show="show"
4   class="bg-green-200 text-green-800 p-2 rounded mb-2">
5     <p>
6       {{session('message')}}
7     </p>
8   </div>
9
10 @endif
11
```

Figure: 5.4.14. Resuable Message Component



```
_message.blade.php _search.blade.php X _view-cv.blade.php _view-job.blade.php
resources > views > components > _search.blade.php > ...
1 <div class="relative border-2 border-gray-100 m-4 rounded-lg">
2   <div class="absolute top-4 left-3">
3     <i class="fa fa-search text-gray-400 z-20 hover:text-gray-500"></i>
4   </div>
5   <input type="text" name="search" class="h-14 w-full pl-10 pr-20 rounded-lg z-0 focus:shadow focus:outline-none"
6     placeholder="Type Keywords to Search" />
7   <div class="absolute top-2 right-2">
8     <button type="submit" class="h-10 w-20 text-white rounded-lg bg-black hover:bg-gray-500">
9       Search
10    </button>
11  </div>
12 </div>
13
```



A search input field with a magnifying glass icon on the left and a 'Search' button on the right. The input field contains the placeholder text 'Type Keywords to Search'.

Figure: 5.4.15. Resuable Search Component

```

message.blade.php search.blade.php view-cv.blade.php X view-job.blade.php
resources > views > components > view-cv.blade.php > div.max-w-4xl.mx-auto.bg-white.shadow-lg.rounded-xl.p-6
1 <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2 <!-- Title -->
3 <h2 class="text-2xl font-bold text-center mb-6">Curriculum Vitae</h2>
4
5 <!-- Basic Info -->
6 <table class="w-full border border-gray-300 text-sm mb-6">
7 <tr class="border border-gray-300">
8 <th class="bg-gray-200 px-3 py-2 text-left w-32">CV Code</th>
9 <td class="px-3 py-2 font-bold">{{ $cv->cv_code }}</td>
10 <th class="bg-gray-200 px-3 py-2 text-left">Name</th>
11 <td class="px-3 py-2">{{ $cv->cv_name }}</td>
12 <td rowspan="6" class="flex flex-col justify-center border-1 border-gray-300 text-center align-top w-36">
13 <div class="flex flex-col items-center justify-center">
14
15 @if($cv->cv_profile)
16
17 
19 <span class="text-xs text-gray-600">Photo</span>
20
21 @else
22
23 <p class="italic">Please Upload Profile Image</p>
24
25 @endif
26
27 </div>
28 </td>
29 </tr>
30 <tr class="border border-gray-300">
31 <th class="bg-gray-200 px-3 py-2 text-left w-32">Registered</th>
32 <td class="px-3 py-2">{{ $cv->created_at }}</td>
33 <th class="bg-gray-200 px-3 py-2 text-left">Gender</th>
34 <td class="px-3 py-2">{{ $cv->cv_gender }}</td>
35 </tr>
36 <tr class="border border-gray-300">
37 <th class="bg-gray-200 px-3 py-2 text-left">Nationality</th>
38 <td class="px-3 py-2">{{ $cv->cv_nationality }}</td>
39 <th class="bg-gray-200 px-3 py-2 text-left">Address</th>
40 <td class="px-3 py-2">{{ $cv->cv_address }}</td>
41 </tr>
42 <tr class="border border-gray-300">
43 <th class="bg-gray-200 px-3 py-2 text-left">Date of Birth</th>
44 <td class="px-3 py-2">{{ $cv->cv_dob }}</td>
45 <th class="bg-gray-200 px-3 py-2 text-left">Age</th>
46 <td class="px-3 py-2">{{ $cv->cv_age }}</td>
47 </tr>
48 <tr class="border border-gray-300">
49 <th class="bg-gray-200 px-3 py-2 text-left">Religion</th>
50 <td class="px-3 py-2">{{ $cv->cv_religion }}</td>
51 <th class="bg-gray-200 px-3 py-2 text-left">Marital Status</th>
52 <td class="px-3 py-2">{{ $cv->cv_marriage }}</td>
53 </tr>
54 <tr class="border border-gray-300">
55 <th class="bg-gray-200 px-3 py-2 text-left">Language Level</th>
56 <td class="px-3 py-2">{{ $cv->cv_lj_level }}</td>
57 <th class="bg-gray-200 px-3 py-2 text-left">Job Type</th>
58 <td class="px-3 py-2">{{ $cv->cv_job_type }}</td>
59 </tr>
60 </table>
61
62 <!-- Education Section -->
63 <h3 class="text-lg font-sembold mt-6 mb-3 text-center font-bold">School History</h3>
64 <table class="w-full border border-gray-300 text-sm mb-6">
65 <thead>
66 <tr class="border border-gray-300">
67 <th class="px-3 py-2 text-left">School Name</th>
68 <th class="px-3 py-2 text-left">Subject</th>
69 <th class="px-3 py-2 text-left">Start Date</th>
70 <th class="px-3 py-2 text-left">End Date</th>
71 <th class="px-3 py-2 text-left">Status</th>
72 </tr>
73 </thead>
74 <tbody>
75
76 @foreach ([1,2,3] as $i)
77
78 <tr class="border border-gray-300">
79 <td class="px-3 py-2">{{ $cv->{"cv_schoolhistory"}[$i]_name }}</td>
80 <td class="px-3 py-2">{{ $cv->{"cv_schoolhistory"}[$i]_subject }}</td>
81 <td class="px-3 py-2">{{ $cv->{"cv_schoolhistory"}[$i]_start }}</td>
82 <td class="px-3 py-2">{{ $cv->{"cv_schoolhistory"}[$i]_end }}</td>
83 <td class="px-3 py-2">{{ $cv->{"cv_schoolhistory"}[$i]_status }}</td>
84 </tr>
85
86 @endforeach
87
88 </tbody>
89 </table>
90

```

```

message.blade.php X search.blade.php view-cv.blade.php X view-job
resources > views > components > view-cv.blade.php > div.max-w-4xl.mx-auto.bg-white.sh
91 </table>
92 <!-- Work Experience -->
93 <h3 class="text-lg font-sembold mt-6 mb-3 text-center font-bold">Job History</h3>
94 <table class="w-full border border-gray-300 text-sm mb-6">
95 <thead>
96 <tr class="border border-gray-300">
97 <th class="px-3 py-2 text-left">Company Name</th>
98 <th class="px-3 py-2 text-left">Position</th>
99 <th class="px-3 py-2 text-left">Start Date</th>
100 <th class="px-3 py-2 text-left">End Date</th>
101 <th class="px-3 py-2 text-left">Status</th>
102 </tr>
103 </thead>
104 <tbody>
105
106 @foreach ([1,2,3] as $i)
107
108 <tr class="border border-gray-300">
109 <td class="px-3 py-2">{{ $cv->{"cv_jobhistory"}[$i]_name }}</td>
110 <td class="px-3 py-2">{{ $cv->{"cv_jobhistory"}[$i]_subject }}</td>
111 <td class="px-3 py-2">{{ $cv->{"cv_jobhistory"}[$i]_start }}</td>
112 <td class="px-3 py-2">{{ $cv->{"cv_jobhistory"}[$i]_end }}</td>
113 <td class="px-3 py-2">{{ $cv->{"cv_jobhistory"}[$i]_status }}</td>
114 </tr>
115
116 @endforeach
117
118 </tbody>
119 </table>
120
121 <!-- Other Notes -->
122 <h3 class="text-lg font-sembold mt-6 mb-3 text-center font-bold">Evaluation</h3>
123 <div class="border border-gray-300 h-28 p-2">{{ $cv->cv_evaluation }}</div>
124
125 <div class="flex justify-center m-10">
126
127 <a href="{{ url('/cv/' . $cv->cv_code . '/pdf') }}" target="_blank"
128 class="bg-green-600 text-white px-3 py-2 rounded">
129 <i class="fa fa-file-pdf"></i> View PDF
130 </a>
131
132 </div>
133

```

Figure: 5.4.16. Resuable View CV Component

```

resources > views > components > _view-job.blade.php > div.flex.justify-center.m-10 > a.bg-green-600.text-w
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213

```

Figure: 5.4.17. Resuable View Job Component

CRUD Operations (Partner and Admin Users)

The details of CRUD functions will be explained in the Appendix section. All the screenshots related to CRUD functions are shown in the sub-sections below.

Partner CV Creation

```
partner-new-cv-register.blade.php X CVForms.php CVController.php
resources > views > partner > partner-new-cv-register.blade.php > x-layout > div.flex.justify-c
1 <x-layout>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-3/4">
6
7 <a href="/home" class="mt-5 ml-5 inline-block border-2 font-bold
8 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
9 shadow-lg">
10 <i class="fa-solid fa-arrow-left"></i> Back to Main Page
11 </a>
12
13 <div class="mx-4">
14
15 @if(session('message'))
16 <div>{{ session('message') }}</div>
17 @endif
18
19 <div class="bg-gray-50 border border-gray-200 p-10 rounded mt-10">
20
21 <header class="text-center mb-10">
22
23 <h2 class="text-2xl font-bold uppercase mb-1">
24 Register New CV
25 </h2>
26 <p class="mb-4">Register for Job Application</p>
27
28 </header>
29
30 <form method="POST" action="/home" enctype="multipart/form-data">
31
32 @csrf
33
34 <!-- General Information -->
35 <div class="">...
36 </div>
37
38 <!-- School History -->
39 <div class="mb-6 mt-10">...
40 </div>
41
42 <!-- Work History -->
43 <div class="mb-6 mt-10">...
44 </div>
45
46 <!-- Others -->
47 <div class="mb-6 mt-10">...
48 </div>
49
50 <!-- Submit Button -->
51 <div class="m-10 text-center">
52 <button type="submit" class="transition duration-200 bg-black font-bold
53 text-white border-black border-2 px-4 py-3 rounded-xl hover:bg-white hover:text-black">
54 Register
55 </button>
56 </div>
57
58 </form>
59
60 </div>
61
62 </div>
63
64 </div>
65
66 </x-layout>
67
68
69
70
71
72
73
74
75
```

Figure: 5.4.18. CV Creation HTML Form

```

<div class="mb-6">
  <label for="cv_name" class="inline-block text-lg mb-2">
    Candidate Name
  </label>
  <input type="text" class="border border-gray-20 rounded p-2 w-full" name="cv_name"
    value="{{old('cv_name')}}" />

  @error('cv_name')
  <p class="text-red-500 text-xs mt-1">{{$message}}</p>
  @enderror
</div>

```

Figure: 5.4.19. Input Error Message for CV Creation HTML Form

```

partner-new-cv-register.blade.php  CVForms.php  CVController.php  web.php  X
routes > web.php > ...
63
64 // Partner CV Registration
65 Route::get('/new-cv-register', [CVController::class, 'create']);
66
67 // Partner New CV Storage
68 Route::post('/home', [CVController::class, 'store']);
69

```

Figure: 5.4.20. Route Definition for New CV Creation and Storage

```

partner-new-cv-register.blade.php X CVForms.php X CVController.php
app > Models > CVForms.php > CVForms > scopefilter
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class CVForms extends Model
9  {
10     /** @use HasFactory<Database\Factories\CVFormsFactory> */
11     use HasFactory;
12
13     protected $fillable = [
14
15         'cv_code',
16         'partner_id',
17         'cv_profile',
18         'cv_name',
19         'cv_gender',
20         'cv_age',
21         'cv_dob',
22         'cv_nationality',
23         'cv_address',
24         'cv_religion',
25         'cv_marriage',
26         'cv_job_type',
27         'cv_job_certificate',
28         'cv_jp_level',
29         'cv_jp_certificate',
30         'cv_jp_study_year',
31         'cv_jp_study_month',
32         'cv_desired_work_length',
33         'cv_schoolhistory1_name',
34         'cv_schoolhistory1_subject',
35         'cv_schoolhistory1_start',
36         'cv_schoolhistory1_end',
37         'cv_schoolhistory1_status',
38         'cv_schoolhistory2_name',
39         'cv_schoolhistory2_subject',
40         'cv_schoolhistory2_start',
41         'cv_schoolhistory2_end',
42         'cv_schoolhistory2_status',
43         'cv_schoolhistory3_name',
44         'cv_schoolhistory3_subject',
45         'cv_schoolhistory3_start',
46         'cv_schoolhistory3_end',
47         'cv_schoolhistory3_status',
48
49         //JOB HISTORY
50         'cv_jobhistory1_name',
51         'cv_jobhistory1_position',
52         'cv_jobhistory1_start',
53         'cv_jobhistory1_end',
54         'cv_jobhistory1_status',
55         'cv_jobhistory2_name',
56         'cv_jobhistory2_position',
57         'cv_jobhistory2_start',
58         'cv_jobhistory2_end',
59         'cv_jobhistory2_status',
60         'cv_jobhistory3_name',
61         'cv_jobhistory3_position',
62         'cv_jobhistory3_start',
63         'cv_jobhistory3_end',
64         'cv_jobhistory3_status',
65
66         //OTHERS
67         'cv_email',
68         'cv_phone',
69
70         'cv_passport_number',
71         'cv_passport_date',
72         'cv_passport',
73         'cv_jp_visit_time',
74         'cv_COE_Received',
75         'cv_COE_Rejected',
76         'cv_status',
77         'cv_evaluation',
78
79     ];
80

```

Figure: 5.4.21. Input definition for mass assignment in CVForm Model

```

partner-new-cv-register.blade.php CVForms.php CVController.php X
app > Http > Controllers > CVController.php > CVController > update
10 class CVController extends Controller
171 }
172
173 //Partner -> View Create New CV Page
174 public function create() {
175
176     return view('partner.partner-new-cv-register');
177 }
178
179
180 //Partner -> Store New CV
181 public function store(Request $request) {
182
183     $formFields = $request->validate([
184
185         'cv_profile' => 'nullable|file|mimes:jpg,jpeg,png|max:2048',
186         'cv_name' => 'required|string|max:255',
187         'cv_gender' => 'required|string|max:10',
188         'cv_age' => 'required|integer|min:15|max:100',
189         'cv_dob' => 'required|date',
190         'cv_nationality' => 'required|string|max:100',
191         'cv_address' => 'required|string|max:255',
192         'cv_religion' => 'required|string|max:100',
193         'cv_marriage' => 'required|string|max:50',
194         'cv_job_type' => 'required|string|max:255',
195         'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
196         'cv_jp_level' => 'required|string|max:50',
197         'cv_jp_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
198         'cv_jp_study_year' => 'required|integer|min:0|max:50',
199         'cv_jp_study_month' => 'required|integer|min:0|max:11',
200         'cv_desired_work_length' => 'required|string|max:100',
201
202         //SCHOOL HISTORY
203         'cv_schoolhistory1_name' => 'required|string|max:255',
204         'cv_schoolhistory1_subject' => 'required|string|max:255',
205         'cv_schoolhistory1_start' => 'required|date',
206         'cv_schoolhistory1_end' => 'required|date|after_or_equal:cv_schoolhistory1_start',
207         'cv_schoolhistory1_status' => 'required|string|max:100',
208
209         'cv_schoolhistory2_name' => 'nullable|string|max:255',
210         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
211         'cv_schoolhistory2_start' => 'nullable|date',
212         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal:cv_schoolhistory2_start',
213         'cv_schoolhistory2_status' => 'nullable|string|max:100',
214
215         'cv_schoolhistory3_name' => 'nullable|string|max:255',
216         'cv_schoolhistory3_subject' => 'nullable|string|max:255',
217         'cv_schoolhistory3_start' => 'nullable|date',
218         'cv_schoolhistory3_end' => 'nullable|date|after_or_equal:cv_schoolhistory3_start',
219         'cv_schoolhistory3_status' => 'nullable|string|max:100',
220
221         //JOB HISTORY
222         'cv_jobhistory1_name' => 'nullable|string|max:255',
223         'cv_jobhistory1_position' => 'nullable|string|max:255',
224         'cv_jobhistory1_start' => 'nullable|date',
225         'cv_jobhistory1_end' => 'nullable|date|after_or_equal:cv_jobhistory1_start',
226         'cv_jobhistory1_status' => 'nullable|string|max:100',
227
228         'cv_jobhistory2_name' => 'nullable|string|max:255',
229         'cv_jobhistory2_position' => 'nullable|string|max:255',
230         'cv_jobhistory2_start' => 'nullable|date',
231         'cv_jobhistory2_end' => 'nullable|date|after_or_equal:cv_jobhistory2_start',
232         'cv_jobhistory2_status' => 'nullable|string|max:100',
233
234         'cv_jobhistory3_name' => 'nullable|string|max:255',
235         'cv_jobhistory3_position' => 'nullable|string|max:255',
236         'cv_jobhistory3_start' => 'nullable|date',
237         'cv_jobhistory3_end' => 'nullable|date|after_or_equal:cv_jobhistory3_start',
238         'cv_jobhistory3_status' => 'nullable|string|max:100',
239
240         //OTHERS
241         'cv_email' => 'nullable|email|max:255',
242         'cv_phone' => 'nullable|string|max:20',
243
244         'cv_passport_number' => 'nullable|string|max:50',
245         'cv_passport_date' => 'nullable|date',
246         'cv_passport' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
247         'cv_jp_visit_time' => 'nullable|integer|min:0|max:50',
248         'cv_COE_Received' => 'nullable|integer|min:0|max:50',
249         'cv_COE_Rejected' => 'nullable|integer|min:0|max:50',
250
251     ]);
252
253     if ($request->hasFile('cv_profile')) {
254         $formFields['cv_profile'] = $request->file('cv_profile')
255         ->store('cv/cv_profile', 'public');
256     }
257
258     if ($request->hasFile('cv_job_certificate')) {
259         $formFields['cv_job_certificate'] = $request->file('cv_job_certificate')
260         ->store('cv/job_certificates', 'public');
261     }
262
263     if ($request->hasFile('cv_jp_certificate')) {
264         $formFields['cv_jp_certificate'] = $request->file('cv_jp_certificate')
265         ->store('cv/jp_certificates', 'public');
266     }
267
268     if ($request->hasFile('cv_passport')) {
269         $formFields['cv_passport'] = $request->file('cv_passport')
270         ->store('cv/passport', 'public');
271     }
272
273     $formFields['partner_id'] = auth('partner')->id();
274
275     $cv = CVForms::create($formFields);
276
277     $cv->cv_code = 'CV-' . str_pad($cv->id, 4, '0', STR_PAD_LEFT);
278     $cv->save();
279
280     return redirect('/home')->with('message', 'CV Created!');
281
282 }
283
284
285
286

```

Figure: 5.4.21. Input Validation and Database Storage in CV Controller

Partner CV Edit

```
web.php partner-edit-cv.blade.php CVController.php
div.flex.justify-center.mb-20 > div.w-3/4 > div.mx-4 > div.bg-gray-50.border.border-gray-200.p-5.rounded.mt-10
1 <x-layout>
3 <div class="flex justify-center mb-20">
4 <div class="w-3/4">
11
12 <div class="mx-4">
13
14 @include('components._message')
15
16 <div class="bg-gray-50 border border-gray-200 p-5 rounded mt-24">
17
18 <header class="text-center mb-10">
19
20 <h2 class="text-2xl font-bold uppercase mb-1">
21 | Edit NAME's CV
22 </h2>
23
24 </header>
25
26 <form method="post" action="/partner-view-cv/{{ $cv -> cv_code }}" enctype="multipart/form-data">
27 @csrf
28 @method('PUT')
29
30 <!-- General Infomration -->
31 > <div class="">...
329 </div>
330
331 <!-- School History -->
332 > <div class="mb-6 mt-10">...
559 </div>
560
561 <!-- Work History -->
562 > <div class="mb-6 mt-10">...
769 </div>
770
771 <!-- Others -->
772 > <div class="mb-6 mt-10">...
879 </div>
880
881 <!-- Submit Button -->
882 <div class="m-10 text-center">
883 | <button type="submit" class="transition duration-200 bg-black font-bold
884 | text-white border-black border-2
885 | px-4 py-3 rounded-xl hover:bg-white hover:text-black">
886 | Update
887 </button>
888 </div>
889
890 </form>
891
```

Figure: 5.4.22. CV Edit HTML Form

```
<div class="mb-6">
  <label for="cv_name" class="inline-block text-lg mb-2">
    Candidate Name
  </label>
  <input type="text" class="border border-gray-20 rounded p-2 w-full" name="cv_name"
    value="{{ $cv -> cv_name }}" />
  @error('cv_name')
  <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
  @enderror
</div>
```

Figure: 5.4.23. CV Edit HTML Form Input Error Display

```

web.php X
routes > web.php > ...
59 Route::get('/partner-view-cv/{cv_code}/edit', [CVController::class, 'edit']);
60
61 // Partner CV Edit Submit and Update
62 Route::put('/partner-view-cv/{cv_code}', [CVController::class, 'update']);
63

```

Figure: 5.4.24. Route Definition for CV Update

```

web.php partner-edit-cv.blade.php CVController.php X
C:\xampp\htdocs\200495_Phone_Myint_Maw_ISP_Project\routes\web.php index
10 class CVController extends Controller
49
50 //Partner -> Edit Single CV
51 public function edit($cv_code) {
52
53     $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
54
55     return view('partner.partner-edit-cv', [
56         'cv' => $cv
57     ]);
58 }
59
60
61 //Partner -> Update the Edited CV
62 public function update(Request $request, $cv_code) {
63
64     $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
65
66     $formFields = $request->validate([
67
68         'cv_profile' => 'nullable|file|mimes:jpg,jpeg,png|max:2048',
69         'cv_name' => 'required|string|max:255',
70         'cv_age' => 'required|integer|min:15|max:100',
71         'cv_dob' => 'required|date',
72         'cv_address' => 'required|string|max:255',
73         'cv_religion' => 'required|string|max:100',
74         'cv_marriage' => 'required|string|max:50',
75         'cv_job_type' => 'required|string|max:255',
76         'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
77         'cv_jp_level' => 'required|string|max:50',
78         'cv_jp_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
79         'cv_jp_study_year' => 'required|integer|min:0|max:50',
80         'cv_jp_study_month' => 'required|integer|min:0|max:11',
81         'cv_desired_work_length' => 'required|string|max:100',
82
83         //SCHOOL HISTORY
84         'cv_schoolhistory1_name' => 'required|string|max:255',
85         'cv_schoolhistory1_subject' => 'required|string|max:255',
86         'cv_schoolhistory1_start' => 'required|date',
87         'cv_schoolhistory1_end' => 'required|date|after_or_equal
88         :cv_schoolhistory1_start',
89
90         'cv_schoolhistory1_status' => 'required|string|max:100',
91
92         'cv_schoolhistory2_name' => 'nullable|string|max:255',
93         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
94         'cv_schoolhistory2_start' => 'nullable|date',
95         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal
96         :cv_schoolhistory2_start',
97
98         'cv_schoolhistory2_status' => 'nullable|string|max:100',
99
100        'cv_schoolhistory3_name' => 'nullable|string|max:255',
101        'cv_schoolhistory3_subject' => 'nullable|string|max:255',
102        'cv_schoolhistory3_start' => 'nullable|date',
103        'cv_schoolhistory3_end' => 'nullable|date|after_or_equal
104        :cv_schoolhistory3_start',
105
106        'cv_schoolhistory3_status' => 'nullable|string|max:100',
107
108        //JOB HISTORY
109        'cv_jobhistory1_name' => 'nullable|string|max:255',
110        'cv_jobhistory1_position' => 'nullable|string|max:255',
111        'cv_jobhistory1_start' => 'nullable|date',
112        'cv_jobhistory1_end' => 'nullable|date|after_or_equal
113        :cv_jobhistory1_start',
114
115        'cv_jobhistory1_status' => 'nullable|string|max:100',

```

```

web.php partner-edit-cv.blade.php CVController.php X
app > Http > Controllers > CVController.php > CVController > update
10 class CVController extends Controller
62 public function update(Request $request, $cv_code) {
113     'cv_jobhistory2_name' => 'nullable|string|max:255',
114     'cv_jobhistory2_position' => 'nullable|string|max:255',
115     'cv_jobhistory2_start' => 'nullable|date',
116     'cv_jobhistory2_end' => 'nullable|date|after_or_equal
117     :cv_jobhistory2_start',
118     'cv_jobhistory2_status' => 'nullable|string|max:100',
119
120     'cv_jobhistory3_name' => 'nullable|string|max:255',
121     'cv_jobhistory3_position' => 'nullable|string|max:255',
122     'cv_jobhistory3_start' => 'nullable|date',
123     'cv_jobhistory3_end' => 'nullable|date|after_or_equal
124     :cv_jobhistory3_start',
125     'cv_jobhistory3_status' => 'nullable|string|max:100',
126
127     //OTHERS
128     'cv_email' => 'nullable|email|max:255',
129     'cv_phone' => 'nullable|string|max:20',
130
131     'cv_passport_number' => 'nullable|string|max:50',
132     'cv_passport_date' => 'nullable|date',
133     'cv_passport' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
134     'cv_jp_visit_time' => 'nullable|integer|min:0|max:50',
135     'cv_COE_Received' => 'nullable|integer|min:0|max:50',
136     'cv_COE_Rejected' => 'nullable|integer|min:0|max:50',
137
138 });
139
140 if ($request->hasFile('cv_profile')) {
141     $formFields['cv_profile'] = $request->file('cv_profile')
142     ->store('cv/profile_images', 'public');
143 }
144
145 if ($request->hasFile('cv_job_certificate')) {
146     $formFields['cv_job_certificate'] = $request->file('cv_job_certificate')
147     ->store('cv/job_certificates', 'public');
148 }
149
150 if ($request->hasFile('cv_jp_certificate')) {
151     $formFields['cv_jp_certificate'] = $request->file('cv_jp_certificate')
152     ->store('cv/jp_certificates', 'public');
153 }
154
155 if ($request->hasFile('cv_passport')) {
156     $formFields['cv_passport'] = $request->file('cv_passport')
157     ->store('cv/passports', 'public');
158 }
159
160 $cv->update($formFields);
161
162 return back()
163 ->with('message', 'CV Updated!');
164
165 }
166

```

Figure: 5.4.25. Controller methods for CV Update

Partner CV Data Retrieve

```
web.php CVController.php partner-home.blade.php X
> views > partner > partner-home.blade.php > x-layout > main.flex.justify-center.mb-20 > div.w-5/6 > div.flex.justify-center
1 <x-layout>
5 <main class="flex justify-center mb-20">
7 <div class="w-5/6">
43 <div class="grid grid-cols-1">
59 </div>
60 <div class="flex justify-center mt-10">
61
62 @if($cvs -> isEmpty())
63
64 <div class="text-center text-gray-500 italic py-10">
65 No CVs have been submitted yet. Click <span class="text-black font-bold">Make New CV Forms</span> to
66 create CV
67 </div>
68
69 @else
70
71 <div class="h-screen overflow-y-auto scrollbar border-2 border-t-2 border-black">
72
73 <table class="w-full table-auto rounded-xl py-2 px-4">
74
75 <thead class="bg-black text-white">...
76 </thead>
77
78 @foreach($cvs as $cv)
79
80 <tbody>
81 <tr class="p-5">
82 <td class="border-t border-b border-gray-300 flex justify-center">
83 @if($cv->cv_profile)
84
85 
87
88 @endif
89 </td>
90 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
91 {{ $cv -> cv_name }}
92 </td>
93 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
94 <table class="w-full table-auto">
95 <tr class="text-center">
96 <td class="text-center">
97 {{ $cv -> cv_nationality }}
98 </td>
99 <tr>
100 <td>
101 {{ $cv -> cv_age }}
102 </td>
103 <td>
104 /
105 </td>
106 <td>
107 {{ $cv -> cv_gender }}
108 </td>
109 </tr>
110 </table>
111 </td>
112 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
113
114 @if($cv->cv_job_certificate)
115
116 <a href="{{ asset('storage/' . $cv->cv_job_certificate) }}" target="_blank"
117 class="text-blue-700 hover:underline p-3">
118 <i class="fa fa-file-pdf mr-1"></i> {{ $cv -> cv_job_type }}
119 </a>
120
121 @else
122
123 {{ $cv -> cv_job_type }}
124
125 @endif
126 </td>
127 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
128
129 @if($cv->cv_ip_certificate)
130
131 <a href="{{ asset('storage/' . $cv->cv_ip_certificate) }}" target="_blank"
132 class="text-blue-700 hover:underline p-3">
133 <i class="fa fa-file-pdf mr-1"></i> {{ $cv -> cv_ip_level }}
134 </a>
135
136 @else
137
138 {{ $cv -> cv_ip_level }}
139
140 @endif
141 </td>
142 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
143 {{ $cv -> cv_religion }}
144 </td>
145 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
146 {{ $cv -> cv_status }}
147 </td>
148 <td class="border-t border-b border-gray-300 text-center">
149 <a href="/partner-view-cv/{{ $cv -> cv_code }}/edit"
150 class="text-blue-400 px-6 py-2 rounded-xl hover:underline hover:text-blue-700"
151 <i class="fa-solid fa-pen-to-square"></i> Edit
152 </a>
153 <a href="/partner-view-cv/{{ $cv -> cv_code }}"
154 class="text-blue-400 px-6 py-2 rounded-xl hover:underline hover:text-blue-700"
155 View
156 </a>
157 </td>
158 </tr>
159 </tbody>
160 @endforeach
```

Figure: 5.4.26. Partner Home Page Retrieving CV Data through {{ }} syntax

```

web.php  _view-cv.blade.php  partner-view-cv.blade.php X  CVController.php  partner-home
resources > views > partner > partner-view-cv.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w-5/6
1  <x-layout>
2
3      <div class="flex justify-center mb-20">
4          <div class="w-5/6">
5
6              <a href="/home" class="mt-5 mb-10 ml-5 inline-block border-2 font-bold
7              border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
8              shadow-lg">
9                  <i class="fa-solid fa-arrow-left"></i> Return All CV
10             </a>
11
12             @include('components._view-cv')
13
14             <div class="flex justify-center mt-10">
15
16                 <a href="/partner-view-cv/{{ $cv -> cv_code }}/edit" class="bg-blue-600 text-white
17                 shadow hover:bg-blue-700 transition">
18                     <i class="fa-solid fa-pen-to-square"></i> Edit
19                 </a>
20
21             </div>
22
23         </div>
24     </x-layout>
25
26
27
28

```

Figure: 5.4.27. Partner View CV Page with view-cv component

```

web.php  _view-cv.blade.php X  CVController.php  partner-home.blade.php
resources > views > components > _view-cv.blade.php > div.max-w-4xl.mx-auto.bg-white.shadow-lg.rounded-xl.p-6
1  <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2
3      <!-- Basic Info -->
4      <table class="w-full border border-gray-300 text-sm mb-6">
5          <tr class="border border-gray-300">
6              <th class="bg-gray-200 px-3 py-2 text-left w-32">CV Code</th>
7              <td class="px-3 py-2 font-bold">{{ $cv -> cv_code }}</td>
8              <th class="bg-gray-200 px-3 py-2 text-left">Name</th>
9              <td class="px-3 py-2">{{ $cv -> cv_name }}</td>
10             <td rowspan="6" class="flex flex-col justify-center border-1 border-gray-300 text-center">
11                 <div class="flex flex-col items-center justify-center">
12                     @if($cv->cv_profile)
13
14                         
16                         <span class="text-xs text-gray-600">Photo</span>
17
18                     @else
19
20                         <p class="italic">Please Upload Profile Image</p>
21
22                     @endif
23
24                 </div>
25             </td>
26
27         </tr>
28
29         <tr class="border border-gray-300">
30             <th class="bg-gray-200 px-3 py-2 text-left w-32">Registered</th>
31             <td class="px-3 py-2">{{ $cv -> created_at }}</td>
32             <th class="bg-gray-200 px-3 py-2 text-left">Gender</th>
33             <td class="px-3 py-2">{{ $cv -> cv_gender }}</td>
34
35         </tr>
36         <tr class="border border-gray-300">
37             <th class="bg-gray-200 px-3 py-2 text-left">Nationality</th>
38             <td class="px-3 py-2">{{ $cv -> cv_nationality }}</td>
39             <th class="bg-gray-200 px-3 py-2 text-left">Address</th>
40             <td class="px-3 py-2">{{ $cv -> cv_address }}</td>
41
42         </tr>

```

Figure: 5.4.28. Data retrieval in view-cv component through implemented page using {{ }} Syntax

```
web.php X CVController.php partner-home.blade.php
routes > web.php > ...
47
48 //Partner Side
49
50 // PARTNER CV
51
52 // Partner Full CV List
53 Route::get('/home', [CVController::class, 'index']);
54 |
55 // Partner CV View
56 Route::get('/partner-view-cv/{cv_code}', [CVController::class, 'viewcv']);
57
```

Figure: 5.4.29. Route Definition for Data Retrieval in Home and view CV Pages

```
_view-cv.blade.php partner-view-cv.blade.php CVController.php X
> Http > Controllers > CVController.php > CVController > update
<?php
namespace App\Http\Controllers;

use App\Models\CVForms;
use Illuminate\Http\Request;
use Illuminate\Validation\Rule;
use Illuminate\Support\Facades\Auth;

class CVController extends Controller
{
    //Partner -> Default Home Page -> Shows CV Forms Created by Partner
    public function index(){
        $filters = request(
            [
                'cv_name',
                'search'
            ]
        );

        $partnerID = auth('partner')->id();

        $cvs = CVForms::where('partner_id', $partnerID)->latest()
            ->filter($filters)->get();

        $passed = CVForms::where('partner_id', $partnerID)->where('cv_status',
            'Final-Interview Pass')->count();

        $total = $cvs->count();

        return view('partner.partner-home', [
            'cvs' => $cvs,
            'total' => $total,
            'passed' => $passed,
        ]);
    }

    //Partner -> View Single CV
    public function viewcv($cv_code) {
        $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();

        return view('partner.partner-view-cv', [
            'cv' => $cv
        ]);
    }
}
```

Figure: 5.4.30. CV Controller Methods for Data Retrieval in Home and View CV Pages

Partner Job Data Retrieve

```
resources > views > partner > partner-all-jobs.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w-5/6 > div.grid.grid-cols-1.mb-10 > p.m-4
1
5 lass="flex justify-center mb-20">
7 iv class="w-5/6">
24 </div>
25
26 <main>
27
28 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
29
30 @foreach($jobs as $job)
31
32 <div class="bg-gray-50 border border-gray-200 rounded p-5">
33
34 <div class="m-3">
35 <div class="text-xl font-bold mb-3">
36 | {{{ $job -> job_code }}}
37 </div>
38
39 <h3 class="text-2xl">
40 | {{{ $job -> job_title }}}
41 </h3>
42
43 <div class="text-xl font-bold">
44 | {{{ $job -> company_name }}}
45 </div>
46 </div>
47
48 <div class="m-3">
49
50 <ul class="flex">
51
52 <li
53 | class="flex items-center justify-center bg-black text-white rounded-xl py-1 px-3 m-2 text-xs max-w-24">
54 <a href="/apply-jobs?search={{ $job -> job_type }}">{{{ $job -> job_type }}}</a>
55 </li>
56
57 <li
58 | class="flex items-center justify-center bg-black text-white rounded-xl py-1 px-3 m-2 text-xs max-w-24">
59 <a
60 | href="/apply-jobs?search={{ $job -> required_jp_level }}">{{{ $job -> required_jp_level }}}</a>
61 </li>
62 </ul>
63 </div>
64
65 <div class="m-3 grid lg:grid-cols-1 grid-cols-2">
66
67 <div class="col-span-1 text-lg">
68
69 <div class="flex items-center">
70 <i class="fa-solid fa-location-dot mr-2"></i>{{{ $job -> company_location }}}
71 </div>
72
73 <div class="flex items-center">
74 <i class="fas fa-flag mr-2"></i> {{{ $job -> job_nationality }}}
75 </div>
76
77 <div class="flex items-center">
78 <span class="font-bold mr-2 text-red-700">Application Deadline :
79 </span>{{{ $job -> application_deadline }}}
80 </div>
81 </div>
82
83 <div class="col-span-1 flex justify-center m-5">
84 <a href="/job-info/{{ $job -> job_code }}"
85 | class="border-black border-2 rounded-xl text-black bg-white py-1 px-2 hover:bg-black hover:text-white">
86 See More
87 </a>
88 </div>
89 </div>
90 </div>
91
92 @endforeach
93
94 </div>
95
96 <div class="m-5">
97 | {{{ $jobs->links() }}
98 </div>
99
100
101 </main>
```

Figure: 5.4.31. Partner All Job Page Retrieving Job Data through {{ }} syntax

```

web.php partner-job-application.blade.php partner-view-job.blade.php X
ems-center.justify-center. > a.transition.duration-300.m-5.border-2.font-bold.border-red-700.text-white.bg-red-700.py-2
1 <x-layout>
2
3 <div class="flex justify-center mb-20">
4 <div class="5/6">
5
6 <a href="/apply-jobs" class="mt-5 ml-5 mb-10 inline-block border-2 font-bold
7 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
8 shadow-lg">
9 <i class="fa-solid fa-arrow-left"></i> Back
10 </a>
11
12 @include('components._view-job')
13
14 <div class="flex items-center justify-center ">
15 <a href="/job-application/{{ $job -> job_code }}" class="transition duration-300 m-5 border-2
16 font-bold border-red-700
17 text-white bg-red-700 py-2 px-4 rounded-3xl uppercase shadow-lg hover:bg-white hover:text-red-700">
18 Apply this Job
19 </a>
20 </div>
21
22 </div>
23 </div>
24
25 </x-layout>
26

```

Figure: 5.4.32. Partner View Job Page with view-job component

```

web.php partner-job-application.blade.php partner-view-job.blade.php _view-job.blade.php X
resources > views > components > _view-job.blade.php > div.flex.justify-center.m-10 > a.bg-green-600.text-white.p
1 <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2
3 <div class="mt-10">
4
5 <header class="text-center mb-10">
6
7 <h2 class="text-2xl font-bold uppercase mb-1">
8 Job Information
9 </h2>
10 </header>
11
12 <div class="mb-10">
13 <table class="w-full">
14 <tbody class="border-2 border-black rounded-3xl">
15
16 <tr>
17
18 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
19 Information
20 </td>
21 <td class="border border-gray-700">
22 <ul class="p-5 font-bold">
23 <li>
24 Code : <span class="text-blue-700">{{ $job -> job_code }}</span>
25 </li>
26 <li>
27 Job Type : <span class="text-blue-700">{{ $job -> job_type }}</span>
28 </li>
29 <li>
30 Recruitment Number : <span class="text-blue-700">{{ $job -> recruit_number }}</span>
31 </li>
32 <li>
33 Application Deadline : <span
34 class="text-blue-700">{{ $job -> application_deadline }}</span>
35 </li>
36 <li>
37 Availability : <span class="text-blue-700">{{ $job -> job_availability }}</span>
38 </li>
39 </ul>
40 </td>
41 </tr>
42
43 <tr>
44 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
45 Company Name
46 </td>
47 <td class="p-5 text-1 border border-gray-700">
48 {{ $job -> company_name }}
49 </td>
50 </tr>
51
52 <tr>
53 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">

```

Figure: 5.4.33. View-job component Retrieving Job Data with {{ }} syntax through implemented page

```
web.php x partner-job-application.blade.php partner-view-job.blade.php _view-job.blade.php
routes > web.php > ...
69
70
71 //PARTNER JOB
72
73 // Partner Job Listing
74 Route::get('/apply-jobs',[JoblistingController::class, 'joblist']);
75
76 // Partner Show Job Info
77 Route::get('/job-info/{job_code}', [JoblistingController::class, 'viewjob']);
78
```

Figure: 5.4.34. Route Definition for Partner all job and view job info Pages

```
web.php JoblistingController.php x partner-job-application.blade.php partner-view-job
app > Http > Controllers > JoblistingController.php > JoblistingController > joblist
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\joblisting;
6 use Illuminate\Http\Request;
7 use Illuminate\Validation\Rule;
8 use Illuminate\Support\Facades\Auth;
9
10 class JoblistingController extends Controller
11 {
12
13 //Partner -> View All Jobs
14 public function joblist(){
15
16     $partner = auth('partner')->user();
17
18     $filters = request(
19         ['job_type',
20          'job_title',
21          'required_jp_level',
22          'search']);
23
24 //forPartner, filter -> Check joblisting Model
25 $jobs = joblisting::query()->forPartner($partner)
26 ->where('job_availability', 'Available')
27 ->latest() -> filter($filters) -> Paginate(6);
28
29 return view('partner.partner-all-jobs', [
30
31     'jobs' => $jobs
32
33     ]);
34
35 }
36
37 //Partner -> View Job
38 public function viewjob($job_code){
39
40     $job = joblisting::where('job_code', $job_code)->firstOrFail();
41
42     return view('partner.partner-view-job', [
43         'job' => $job
44     ]);
45
46 }
47
48 }
49
```

Figure: 5.4.35. Joblistings Controller Methods for Retrieving Job Data

Admin Job Create

```
web.php AdminJoblistingController.php joblisting.php admin-add-new-job.blade.php X
main.m-5 > div.mx-4 > div.max-w-9/10.mx-auto.mt-10 > div.flex.justify-center > div.lg:w-4/5.p-10.border-2.border-
1 admin>
2 lass="flex justify-center mb-20">
3
4 div class="w-4/5">
5
6 <main class="m-5">
7
8 <div class="mx-4">
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 <!-- CV ADD FORM -->
27 <div class="flex justify-center">
28
29 <div class="lg:w-4/5 p-10 border-2 border-black">
30
31 <form method="post" action="/all-listed-jobs" enctype="multipart/form-data">
32
33 @csrf
34
35 <div class="mb-6">...
36 </div>
37
38 <div class="mb-6">...
39 </div>
40
41 <div class="mb-6">
42
43 <label for="application_deadline" class="inline-block text-lg mb-2">
44 Application Deadline
45 </label>
46
47 <input type="date" class="border border-gray-200 rounded p-2 w-full"
48 name="application_deadline" value="{{old('application_deadline')}}"/>
49
50 @error('application_deadline')
51 <p class="text-red-500 text-xs mt-1">{{$message}}</p>
52 @enderror
53
54 </div>
55
56 <div class="mb-6">...
57 </div>
58
59 <div class="mb-6">...
60 </div>
61
62 <div class="mb-6">...
63 </div>
64
65 <div class="mb-6">...
66 </div>
67
68 <div class="mb-6">...
69 </div>
70
71 <div class="mb-6">...
72 </div>
73
74 <div class="mb-6">...
75 </div>
76
77 <div class="mb-6">...
78 </div>
79
80 <div class="mb-6">...
81 </div>
82
83 <div class="mb-6">...
84 </div>
85
86 <div class="mb-6">...
87 </div>
88
89 <div class="mb-6">...
90 </div>
91
92 <div class="mb-6">...
93 </div>
94
95 <div class="mb-6">...
96 </div>
97
98 <div class="mb-6">...
99 </div>
100
101 <div class="mb-6">...
102 </div>
103
104 <div class="mb-6">...
105 </div>
106
107 <div class="mb-6">...
108 </div>
109
110 <div class="mb-6">...
111 </div>
112
113 <div class="mb-6">...
114 </div>
115
116 <div class="mb-6">...
117 </div>
118
119 <div class="mb-6">...
120 </div>
121
122 <div class="mb-6">...
123 </div>
124
125 <div class="mb-6">...
126 </div>
127
128 <div class="mb-6">...
129 </div>
130
131 <div class="mb-6">...
132 </div>
133
134 <div class="mb-6">...
135 </div>
136
137 <div class="mb-6">...
138 </div>
139
140 <div class="mb-6">...
141 </div>
142
143 <div class="mb-6">...
144 </div>
145
146 <div class="mb-6">...
147 </div>
148
149 <div class="mb-6">...
150 </div>
151
152 <div class="mb-6">...
153 </div>
154
155 <div class="mb-6">...
156 </div>
157
158 <div class="mb-6">...
159 </div>
160
161 <div class="mb-6">...
162 </div>
163
164 <div class="mb-6">...
165 </div>
166
167 <div class="mb-6">...
168 </div>
169
170 <div class="mb-6">...
171 </div>
172
173 <div class="mb-6">...
174 </div>
175
176 <div class="mb-6">...
177 </div>
178
179 <div class="mb-6">...
180 </div>
181
182 <div class="mb-6">...
183 </div>
184
185 <div class="mb-6">...
186 </div>
187
188 <div class="mb-6">...
189 </div>
190
191 <div class="mb-6">...
192 </div>
193
194 <div class="mb-6">...
195 </div>
196
197 <div class="mb-6">...
198 </div>
199
200 <div class="mb-6">...
201 </div>
202
203 <div class="mb-6">...
204 </div>
205
206 <div class="mb-6">...
207 </div>
208
209 <div class="mb-6">...
210 </div>
211
212 <div class="mb-6">...
213 </div>
214
215 <div class="mb-6">...
216 </div>
217
218 <div class="mb-6">...
219 </div>
220
221 <div class="mb-6">...
222 </div>
223
224 <div class="mb-6">...
225 </div>
226
227 <div class="mb-6">...
228 </div>
229
230 <div class="mb-6">...
231 </div>
232
233 <div class="mb-6">...
234 </div>
235
236 <div class="mb-6">...
237 </div>
238
239 <div class="mb-6">...
240 </div>
241
242 <div class="mb-6">...
243 </div>
244
245 <div class="mb-6">...
246 </div>
247
248 <div class="mb-6">...
249 </div>
250
251 <div class="mb-6">...
252 </div>
253
254 <div class="mb-6">...
255 </div>
256
257 <div class="mb-6">...
258 </div>
259
260 <div class="mb-6">...
261 </div>
262
263 <div class="mb-6">...
264 </div>
265
266 <div class="mb-6">...
267 </div>
268
269 <div class="mb-6">...
270 </div>
271
272 <div class="mb-6">...
273 </div>
274
275 <div class="mb-6">...
276 </div>
277
278 <div class="mb-6">...
279 </div>
280
281 <div class="mb-6">...
282 </div>
283
284 <div class="mb-6">...
285 </div>
286
287 <div class="mb-6">...
288 </div>
289
290 <div class="mb-6">...
291 </div>
292
293 <div class="mb-6">...
294 </div>
295
296 <div class="mb-6">...
297 </div>
298
299 <div class="mb-6">...
300 </div>
301
302 <div class="mb-6">...
303 </div>
304
305 <div class="mb-6">...
306 </div>
307
308 <div class="mb-6">...
309 </div>
310
311 <div class="mb-6">...
312 </div>
313
314 <div class="mb-6">...
315 </div>
316
317 <div class="mb-6">...
318 </div>
319
320 <div class="mb-6">...
321 </div>
322
323 <div class="mb-6">...
324 </div>
325
326 <div class="mb-6">...
327 </div>
328
329 <div class="mb-6">...
330 </div>
331
332 <div class="mb-6">...
333 </div>
334
335 <div class="mb-6">...
336 </div>
337
338 <div class="mb-6">...
339 </div>
340
341 <div class="mb-6">...
342 </div>
343
344 <div class="mb-6">...
345 </div>
346
347 <div class="mb-6">...
348 </div>
349
350 <div class="mb-6">...
351 </div>
352
353 <div class="mb-6">...
354 </div>
355
356 <div class="mb-6">...
357 </div>
358
359 <div class="mb-6">...
360 </div>
361
362 <div class="mb-6">...
363 </div>
364
365 <div class="mb-6">...
366 </div>
367
368 <div class="mb-6">...
369 </div>
370
371 <div class="mb-6">...
372 </div>
373
374 <div class="mb-6">...
375 </div>
376
377 <div class="mb-6">...
378 </div>
379
380 <div class="mb-6">...
381 </div>
382
383 <div class="mb-6">...
384 </div>
385
386 <div class="mb-6">...
387 </div>
388
389 <div class="mb-6">...
390 </div>
391
392 <div class="mb-6 flex justify-center">
393
394 <button type="submit" class="transition duration-300 bg-green-700 text-white
395 rounded py-2 px-4 hover:bg-green-500">
396 <i class="fa fa-pencil mr-2"></i>Add
397 </button>
398
399 </div>
400
401 </form>
402
403 </div>
```

Figure: 5.4.36. HTML Form for Job Creation Input

```

web.php X AdminJoblistingController.php joblisting.php admin-add-new-job.blade.php
routes > web.php > ...
130
131 // Admin Job Add New
132 Route::get('/add-new-job', [AdminJoblistingController::class, 'add']);
133
134 // Admin New Job Store
135 Route::post('/all-listed-jobs', [AdminJoblistingController::class, 'store']);
136

```

Figure: 5.4.37. Route Definition for Admin Job Creation and Storage

```

web.php AdminJoblistingController.php X joblisting.php admin-add-new-job.blade.php
app > Http > Controllers > AdminJoblistingController.php > AdminJoblistingController > update
10 class AdminJoblistingController extends Controller
93
94 //Admin -> View Add New Job Page
95 public function add(){
96
97     return view('company.admin-add-new-job');
98
99 }
100
101 //Admin -> Store Added New Job Data
102 public function store(Request $request){
103
104     $formFields = $request->validate([
105         'job_type' => 'required',
106         'recruit_number' => 'required',
107         'application_deadline' => 'required',
108         'company_name' => 'required',
109         'company_website' => 'required',
110         'company_location' => 'required',
111         'job_title' => 'required',
112         'job_description' => 'required',
113         'work_time' => 'required',
114         'off_days' => 'required',
115         'salary_benefits' => 'required',
116         'deductables' => 'required',
117         'after_deduction' => 'required',
118         'airticket_exp' => 'required',
119         'required_jp_level' => 'required',
120         'age_range' => 'required',
121         'job_gender' => 'required',
122         'job_nationality' => 'required',
123         'other_requirements' => 'required',
124     ]);
125
126     $jobs = joblisting::create($formFields);
127     $prefix = strtoupper(substr($jobs -> job_type, 0, 3));
128
129     $jobs->job_code = 'JOB - ' .
130     str_pad($jobs->id, 4, '0', STR_PAD_LEFT) . ' - ' . $prefix;
131
132     $jobs->save();
133
134     return redirect('/all-listed-jobs')
135     ->with('message', 'Job Created!');
136
137 }
138
139 }
140
141

```

Figure: 5.4.38. AdminJoblistings Controller Methods for Admin Job Creation and Storage

```

web.php AdminJoblistingController.php joblisting.php x admin-add-new-job.blade.php
app > Models > joblisting.php > joblisting > booted > Closure
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class joblisting extends Model
9  {
10     /** @use HasFactory<Database\Factories\JoblistingFactory> */
11     use HasFactory;
12
13     protected $fillable = [
14         'job_type',
15         'recruit_number',
16         'application_deadline',
17         'company_name',
18         'company_website',
19         'company_location',
20         'job_title',
21         'job_description',
22         'work_time',
23         'off_days',
24         'salary_benefits',
25         'deductables',
26         'after_deduction',
27         'airticket_exp',
28         'required_jp_level',
29         'age_range',
30         'job_gender',
31         'job_nationality',
32         'other_requirements',
33         'job_availability',
34     ];
35

```

Figure: 5.4.39. Input Definition for Mass Assignment in Joblisting Model

```

web.php AdminJoblistingController.php joblisting.php admin-add-new-job.blade.php x
main.m-5 > div.mx-4 > div.max-w-9/10.mx-auto.mt-10 > div.flex.justify-center > div.lg:w-4/5.p-10.border-2.bor
1
3  <div class="mb-20">
5
13
14  >
96  </div>
97
98  <div class="mb-6">
99
100     <label for="company_name" class="inline-block text-lg mb-2">
101         Company Name
102     </label>
103
104     <input type="text" class="border border-gray-200 rounded p-2 w-full"
105         name="company_name" value="{{old('company_name')}}" />
106
107     @error('company_name')
108     <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
109     @enderror
110
111 </div>
112

```

Figure: 5.4.40. HTML Form Input Error Message Display

Admin Job Update

```
web.php AdminJoblistingController.php admin-edit-job.blade.php
justify-center.mb-20 > div.w-5/6 > main.m-5 > div.mx-4 > div.flex.justify-center > div.p-20.border-2.border-black.v
1
3   <div justify-center mb-20"&>
5   <div w-5/6"&>
13  <div class="m-5"&>
14  <div class="mx-4"&>
26  <div class="p-20 border-2 border-black w-3/4"&>
27  <form method="post" action="/all-listed-jobs/{{ $job -> job_code }}"
28  <enctype="multipart/form-data">
29
30  @csrf
31  @method('PUT')
32
33  <div class="mb-6"> ...
45  </div>
46
47  <div class="mb-6"> ...
57  </div>
58
59  <div class="mb-6"> ...
72  </div>
73
74  <div class="mb-6"> ...
87  </div>
88
89  <div class="mb-6"> ...
102 </div>
103
104 <div class="mb-6"> ...
116 </div>
117
118 <div class="mb-6"> ...
131 </div>
132
133 <div class="mb-6"> ...
146 </div>
147
148 <div class="mb-6"> ...
161 </div>
162
163 <div class="mb-6"> ...
176 </div>
177
178 <div class="mb-6"> ...
191 </div>
192
193 <div class="mb-6"> ...
206 </div>
207
208 <div class="mb-6"> ...
221 </div>
222
223 <div class="mb-6"> ...
237 </div>
238
239 <div class="mb-6"> ...
258 </div>
259
260 <div class="mb-6"> ...
282 </div>
283
284 <div class="mb-6"> ...
302 </div>
303
304 <div class="mb-6"> ...
320 </div>
321
322 <div class="mb-6"> ...
349 </div>
350
351 <div class="mb-6"> ...
365 </div>
366
367 <div class="mb-6"> ...
390 </div>
391
392 <div class="mb-6 flex justify-center">
393
394 <button type="submit" class="transition duration-300 bg-green-700 text-white r
395   hover:bg-green-500">
396   <i class="fa fa-pencil mr-2"></i>Edit
397 </button>
398
399 </div>
400
401 </form>
```

Figure: 5.4.41. HTML Form for Job Update Input

```

web.php X AdminJoblistingController.php admin-edit-job.blade.php
routes > web.php > ...
125 // Admin Job Edit
126 Route::get('/all-listed-jobs/{job_code}/edit', [AdminJoblistingController::class, 'edit']);
127
128 // Admin Job Edit and SUBMIT => UPDATE
129 Route::put('/all-listed-jobs/{job_code}', [AdminJoblistingController::class, 'update']);
130

```

Figure: 5.4.42. Route Definition for Job edit and Information Update

```

web.php X AdminJoblistingController.php X admin-edit-job.blade.php
app > Http > Controllers > AdminJoblistingController.php > AdminJoblistingController > update
10 class AdminJoblistingController extends Controller
47
48 //Admin -> Edit Selected Job
49 public function edit($job_code){
50
51     $job = joblisting::where('job_code', $job_code)->firstOrFail();
52
53
54     return view('company.admin-edit-job', [
55         'job' => $job
56     ]);
57 }
58
59
60 //Admin -> Update Edited Job
61 public function update(Request $request, $job_code){
62
63     $job = joblisting::where('job_code', $job_code)->firstOrFail();
64
65     $formFields = $request->validate([
66         'recruit_number' => 'required',
67         'application_deadline' => 'required',
68         'company_name' => 'required',
69         'company_website' => 'required',
70         'company_location' => 'required',
71         'job_title' => 'required',
72         'job_description' => 'required',
73         'work_time' => 'required',
74         'off_days' => 'required',
75         'salary_benefits' => 'required',
76         'deductables' => 'required',
77         'after_deduction' => 'required',
78         'airticket_exp' => 'required',
79         'required_jp_level' => 'required',
80         'age_range' => 'required',
81         'job_gender' => 'required',
82         'job_nationality' => 'required',
83         'other_requirements' => 'required',
84         'job_availability' => 'required'
85     ]);
86
87     $job->update($formFields);
88
89     return back()
90     ->with('message', 'Job Updated!');
91
92 }
93

```

Figure: 5.4.43. Admin Joblisting Controller Method for Editing and Updating Job Information

```

<div class="mb-6">
    <label for="application_deadline" class="inline-block text-lg mb-2">
        Application Deadline
    </label>
    <input type="date" class="border border-gray-200 rounded p-2 w-full"
        name="application_deadline" value="{{ $job -> application_deadline }}" />
    @error('application_deadline')
    <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
    @enderror
</div>

```

Figure: 5.4.44. HTML Edit Form Input Error Display

Admin Job Retrieve

```

admin-job.blade.php x AdminJoblistingController.php
docs\200495_Phone_Myint_Maw_ISP_Project\routes\web.php nin > div.flex.justify-center.mb-20 > div.w-5/6
ex justify-center mb-20"
="w-5/6">
on>
iv class="m-5 flex flex-col justify-center">
<thead>...
</thead>
@foreach($jobs as $job)
<tbody>
<tr>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<div class="text-sm font-bold lg:text-lg">
{{ $job -> job_code }}
</div>
</td>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<div class="text-sm font-bold lg:text-lg">
{{ $job -> company_name }}
</div>
</td>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<a class="text-sm font-bold lg:text-lg"
href="/all-listed-jobs?search={{ $job -> job_type }}">{{ $job -> job_type }}</a>
</td>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<a class="text-sm font-bold lg:text-lg"
href="/all-listed-jobs?search={{ $job -> required_jp_level }}">{{ $job -> required_jp_level }}</a>
</td>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<a class="text-sm font-bold lg:text-lga"
href="/all-listed-jobs?search={{ $job -> job_nationality }}">{{ $job -> job_nationality }}</a>
</td>
<td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
<ul>
<li class="flex justify-center mb-1">
<a href="/all-listed-jobs/{{ $job -> job_code }}"
class="items-center text-blue-700 hover:underline">
More Info
</a>
</li>
<li class="flex justify-center mb-1">
<a href="/all-listed-jobs/{{ $job -> job_code }}/edit"
class="items-center text-blue-700 hover:underline">
Edit
</a>
</li>
</ul>
</td>
</tr>
</tbody>
@endforeach
</table>

```

Figure: 5.4.45. Admin All Job Page Data Retrieval through {{ }} Syntax

```

resources > views > company > admin-show-job.blade.php > x-layout-admin > div.flex.justify-center.mb-20
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <div class="flex justify-between mb-20">
8
9 <a href="/all-listed-jobs" class="mt-5 ml-5 inline-block border-2 font-bold
10 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
11 shadow-lg">
12 <i class="fa-solid fa-arrow-left"></i> All Job List
13 </a>
14
15 <a href="/all-listed-jobs/{{ $job -> job_code }}/edit" class="transition duration-200 mt-5 ml-5
16 inline-block font-bold text-white bg-blue-700 py-1 px-2
17 rounded-xl uppercase hover:bg-blue-400">
18 <i class="fa fa-pencil mr-2"></i>Edit
19 </a>
20
21 </div>
22
23 <!-- JOB SECTION -->
24 @include('components._view-job')
25
26 </div>
27
28 </div>
29
30 </x-layout-admin>
31

```

Figure: 5.4.46. Admin Show Job Page Integrated with view-job component

```

-job.blade.php admin-show-job.blade.php _view-job.blade.php AdminJoblistingController.php
resources > views > components > _view-job.blade.php > div.flex.justify-center.m-10 > a.bg-green-600.text-white.p-2
1 <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2
3 <div class="mt-10">
4 <div class="mb-10">
5 <table class="w-full">
6 <tbody class="border-2 border-black rounded-3xl">
7
8 <tr>
9 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
10 Information
11 </td>
12 <td class="border border-gray-700">
13 <ul class="p-5 font-bold">
14 <li>
15 Code : <span class="text-blue-700">{{ $job -> job_code }}</span>
16 </li>
17 <li>
18 Job Type : <span class="text-blue-700">{{ $job -> job_type }}</span>
19 </li>
20 <li>
21 Recruitment Number : <span class="text-blue-700">{{ $job -> recruit_number }}</span>
22 </li>
23 <li>
24 Application Deadline : <span
25 class="text-blue-700">{{ $job -> application_deadline }}</span>
26 </li>
27 <li>
28 Availability : <span class="text-blue-700">{{ $job -> job_availability }}</span>
29 </li>
30 </ul>
31 </td>
32 </tr>
33
34 <tr>...
35 </tr>
36
37 </tbody>
38 </table>
39
40 </div>
41
42 </div>
43 </div>
44
45 </div>
46
47 </div>
48
49 </div>
50
51

```

Figure: 5.4.47. Joblistings Data Retrieval in view-job component through Integrated Pages

```
web.php × admin-job.blade.php admin-show-job.blade.php _view-job.blade.php Ac [
routes > web.php > ...
115
116
117 // ADMIN JOBS
118
119 // Admin All Jobs
120 Route::get('/all-listed-jobs', [AdminJoblistingController::class, 'joblistAdmin']);
121
122 // Admin Show Job
123 Route::get('/all-listed-jobs/{job_code}', [AdminJoblistingController::class, 'viewjobAdmin']);
124
```

Figure: 5.4.48. Route Definition for Job Data Retrieval

```
job.blade.php AdminJoblistingController.php × admin-show-job.blade.php _vie
app > Http > Controllers > AdminJoblistingController.php > AdminJoblistingController > upd
8 use Illuminate\Support\Facades\Auth;
9
10 class AdminJoblistingController extends Controller
11 {
12     //Admin -> Show All Jobs
13     public function joblistAdmin(){
14
15         $filters = request(
16             [
17                 'company_name',
18                 'job_type',
19                 'job_title',
20                 'required_jp_level',
21                 'job_nationality',
22                 'search'
23             ]
24         );
25
26         //filters -> Check joblisting Model
27         $jobs = joblisting::latest()->filter($filters)->get();
28
29         return view('company.admin-job', [
30             'jobs' => $jobs,
31         ]
32     );
33 }
34
35
36 //Admin -> Show Selected Job
37 public function viewjobAdmin($job_code){
38
39     $job = joblisting::where('job_code', $job_code)->firstOrFail();
40
41
42     return view('company.admin-show-job', [
43         'job' => $job
44     ]
45 );
46 }
47
```

Figure: 5.4.49. AdminJoblisting Controller Methods for Job View Pages

Admin Partner Account Create

```
web.php PartnerController.php admin-add-partner.blade.php Partner.php
resources > views > company > admin-add-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.
1
3 justify-center mb-20">
5 /-5/6">
14 is="m-5">
24 class="flex justify-center">
33 </div>
34
35 <form method="POST" action="/all-partners">
36
37 @csrf
38 <div>
39
40 <ul class="border border-gray-200 mb-5 p-5">
41
42 <li class="mb-5 p-3">
43 <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
44 Partner Name
45 </label>
46
47 <input type="text" class="border border-gray-200 rounded p-2 w-full"
48 name="partner_name" value="{{old('partner_name')}}" />
49
50 @error('partner_name')
51 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
52 @enderror
53 </li>
54
55 > <li class="mb-5 p-3">...
66 </li>
67
68 > <li class="mb-5 p-3">...
79 </li>
80
81 > <li class="mb-5 p-3">...
113 </li>
114
115 <li class="mb-5 p-3">
116 <label for="password" class="inline-block text-lg mb-2 font-bold">
117 Partner Password
118 </label>
119
120 <input type="text" class="border border-gray-200 rounded p-2 w-full"
121 name="password" />
122
123 @error('password')
124 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
125 @enderror
126
127 </li>
128
129 </ul>
130
131 </div>
132
133 <div class="flex justify-center">
134 <button class="transition duration-300 bg-blue-700 text-white rounded py-2 px-4
135 hover:bg-blue-500">
136 Register
137 </button>
138 </div>
139
140
141 </form>
142
143 </div>
144
```

Figure: 5.4.50. HTML Form Input for Partner Account Creation

```
web.php x PartnerController.php admin-add-partner.blade.php Partner.php
routes > web.php > ...
151
152 // Admin Add Partner
153 Route::get('/register-new-partner', [PartnerController::class, 'addNew']);
154
155 // Admin Store Partner
156 Route::post('/all-partners', [PartnerController::class, 'store']);
157
```

Figure: 5.4.51. Route Definition for Partner Account Creation and Storage

```
PartnerController.php x admin-add-partner.blade.php Partner.php
Controller > PartnerController.php > PartnerController
class PartnerController extends Controller

//Admin -> Partner Add Page
public function addNew(){

    return view('company.admin-add-partner');

}

//Admin -> New Partner Store
public function store(Request $request){

    $formFields = $request->validate([
        'partner_name' => 'required',
        'email' => ['required', 'email', 'unique:partners,email'],
        'partner_address' => 'required',
        'partner_nationality' => 'required',
        'password' => 'required|min:6',
    ]);

    $formFields['password'] = Hash::make($formFields['password']);

    $partner = partner::create($formFields);

    $prefix = strtoupper(substr($partner -> partner_nationality, 0, 2));
    $partner -> partner_id = 'PT' . $prefix . '-' .
    str_pad($partner -> id, 3, '0', STR_PAD_LEFT);
    $partner -> save();

    return redirect('/all-partners')->with('message', 'Partner Created!');

}
```

Figure: 5.4.52. Partner Controller Methods for Partner Account Storage

```
hp Partner.php X PartnerController.php admin-add-partner.blad
models > Partner.php > Partner
class Partner extends Authenticatable
/** @use HasFactory<\Database\Factories\PartnerFactory> */
use HasFactory, Notifiable;

protected $table = 'partners';

protected $fillable = [
    'partner_name',
    'email',
    'partner_address',
    'partner_nationality',
    'password',
];
```

Figure: 5.4.53. Input Definition in Partner Model for Mass Assignment

```
hp Partner.php PartnerController.php admin-add-partner.blade.php X
s > views > company > admin-add-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 >
center mb-20">
flex justify-center">
<ul class="border border-gray-200 mb-5 p-5">
  <li class="mb-5 p-3">
    <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
      Partner Name
    </label>
    <input type="text" class="border border-gray-200 rounded p-2 w-full"
      name="partner_name" value="{{old('partner_name')}}" />
    @error('partner_name')
    <p class="text-red-500 text-xs mt-1">{{$message}}</p>
    @enderror
  </li>
```

Figure: 5.4.54. Input Error Display for Account Creation Input

Admin Partner Account Update

```
web.php x Partner.php PartnerController.php admin-edit-partner.blade.php x
1/4 > form > div.mb-6.flex.justify-center > button.transition.duration-300.bg-green-700.text-white.rounded.py-2.px-4.hov
1
3   fy-center mb-20">
5   >
13  -5">
15  ="mx-4">
27  @include( components._message )
30
31  <form method="post" action="/all-partners/{{ $partner -> partner_id }}"
32  enctype="multipart/form-data">
33
34  @csrf
35  @method('PUT')
36
37  <div>
38
39  <ul class="border border-gray-200 mb-5 p-5">
40
41
42
43  <li class="mb-5 p-3">
44  <label for="PartnerID" class="inline-block text-lg mb-2 font-bold">
45  Partner ID
46  </label>
47
48  <h2 class="font-bold text-blue-700">
49  {{ $partner -> partner_id }}
50  </h2>
51
52  </li>
53
54  <li class="mb-5 p-3">
55  <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
56  Partner Name
57  </label>
58
59  <input type="text" class="border border-gray-200 rounded p-2 w-full"
60  name="partner_name" value="{{ $partner -> partner_name }}" />
61
62  @error('partner_name')
63  <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
64  @enderror
65  </li>
66
67  > <li class="mb-5 p-3"> ...
68  </li>
69
70  > <li class="mb-5 p-3"> ...
71  </li>
72
73  <li class="mb-5 p-3">
74  <label for="password" class="inline-block text-lg mb-2 font-bold">
75  Partner Password (Leave if No Update is Needed)
76  </label>
77
78  <input type="text" class="border border-gray-200 rounded p-2 w-full"
79  name="password" />
80
81  </li>
82
83  </ul>
84
85  </div>
86
87  <div class="mb-6 flex justify-center">
88
89  <button type="submit" class="transition duration-300 bg-green-700 text-white rounded
90  py-2 px-4 hover:bg-green-500">
91  <i class="fa fa-pencil mr-2"></i>Edit
92  </button>
93
94  </div>
95
96  </form>
97
98  div>
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
```

Figure: 5.4.55. HTML Form Input for Editing Partner Information

```

web.php X PartnerController.php admin-edit-partner.blade.php
routes > web.php > ...
145
146 // Admin View Edit Partner
147 Route::get('/all-partners/{partner_id}/edit', [PartnerController::class, 'edit']);
148
149 // Admin Edit and Update Partner
150 Route::put('/all-partners/{partner_id}', [PartnerController::class, 'update']);
151

```

Figure: 5.4.56. Route Definition for Partner Information Update and Storage

```

web.php X PartnerController.php X admin-edit-partner.blade.php
> Http > Controllers > PartnerController.php > PartnerController
class PartnerController extends Controller

//Admin -> Partner Edit Page View
public function edit($partner_id){

    $partner = partner::where('partner_id', $partner_id)->firstOrFail();

    return view('company.admin-edit-partner', [
        'partner' => $partner
    ]);
}

//Admin -> Partner Edit and Update
public function update(Request $request, $partner_id){

    $partner = partner::where('partner_id', $partner_id)->firstOrFail();

    $formFields = $request->validate([

        'partner_name' => 'required',
        'email' => ['required', 'email'],
        'partner_address' => 'required',
        'password' => 'nullable|string|min:6',

    ]);

    //Dont Overwrite with Null Password
    if (!empty($formFields['password'])) {
        $formFields['password'] = Hash::make($formFields['password']);
    }
    else {
        unset($formFields['password']);
    }

    $partner->update($formFields);

    return back()
    ->with('message', 'Partner Information Updated!');
}

```

Figure: 5.4.57. Partner Controller Methods for Updating Partner Information

```

<li class="mb-5 p-3">
    <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
        Partner Name
    </label>

    <input type="text" class="border border-gray-200 rounded p-2 w-full"
        name="partner_name" value="{{ $partner -> partner_name }}" />

    @error('partner_name')
    <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
    @enderror
</li>

```

Figure: 5.4.58. Input Error Display in Partner Edit Page

Admin Partner Account Retrieve

```
PartnerController.php | admin-show-partner.blade.php | admin-partner.blade.php
x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > section.m-5.flex.flex-col.justify-center
tify-center mb-20">
6">
ss="m-5 flex flex-col justify-center">
ss="border-black border-4 rounded-2xl text-black pb-5">
class="h-full overflow-y-auto scrollbar">
<table class="w-full table-auto bg-white text-black">
  <thead>...
  </thead>
  @foreach($partners as $partner)
  <tbody>
    <tr>
      <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
        <a class="text-lg lg:text-2xl font-bold"
          href="/all-partners?search={{$partner -> partner_name}}">
          {{$partner -> partner_name}}
        </a>
      </td>
      <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
        <h3 class="text-lg lg:text-2xl font-bold">
          {{$partner -> email}}
        </h3>
      </td>
      <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
        <a class="text-lg m-2 lg:text-xl w-24 hover:text-blue-500"
          href="/all-partners?search={{$partner -> partner_nationality}}">
          {{$partner -> partner_nationality}}
        </a>
      </td>
      <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
        <a class="hover:text-blue-700" href="/all-partners/{{$partner -> partner_id}}">
          <i class="fa-solid fa-circle-info mr-1"></i> More Info
        </a>
      </td>
    </tr>
  </tbody>
  @endforeach
</table>
```

Figure: 5.4.59. Partner Data Retrieval in All Partners Page using {{ }} Syntax

```

web.php PartnerController.php admin-show-partner.blade.php X admin-partner.blade.php
show-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > section.m-5.flex.flex-col.justify
1
2
3 .fy-center mb-20"
4
5 >
25 <div class="flex flex-col justify-center">
28 <div class="text-center mb:5">
34
35
36 <div class="flex justify-center">
37
38 <div class="w-3/4">
39
40 -- PARTNER SECTION -->
41 <div class="border-black border-4 rounded-2xl text-black overflow-hidden flex flex-col mb-10">
42
43 <div class="text-2xl font-bold text-center bg-gray-300 max-h border-b-2 p-3 text-center">
44
45 General Information
46
47 </div>
48
49 <div class="p-3">
50
51 <ul>
52 <li>
53 Partner ID: <span class="font-bold">{{ $partner -> partner_id }}</span>
54 </li>
55 <li>
56 Partner Name: <span class="font-bold">{{ $partner -> partner_name }}</span>
57 </li>
58 <li>
59 Partner Email: <span class="font-bold">{{ $partner -> email }}</span>
60 </li>
61 <li>
62 Partner Address: <span class="font-bold">{{ $partner -> partner_address }}</span>
63 </li>
64 <li>
65 Partner Nation: <span
66 class="font-bold">{{ $partner -> partner_nationality }}</span>
67 </li>
68 </ul>
69
70 </div>
71
72
73 </div>
74
75 </div>

```

```

85
86 <table class="w-full table-auto bg-white text-black">
87 <thead>...
88 </thead>
89
90 <tbody>
91
92 @foreach($cvs as $cv)
93
94 <tr>
95 <td class="border-t border-b border-gray-300 flex justify-center">
96 @if($cv->cv_profile)
97
98 
100
101 @endif
102 </td>
103 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
104 {{ $cv -> cv_name }}
105 </td>
106 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
107
108 @if($cv->cv_jp_certificate)
109
110 <a href="{{ asset('storage/' . $cv->cv_jp_certificate) }}"
111 target="_blank" class="text-blue-700 hover:underline p-3">
112 <i class="fa fa-file-pdf mr-1"></i> {{ $cv -> cv_jp_level }}
113 </a>
114
115 @else
116
117 {{ $cv -> cv_jp_level }}
118
119 @endif
120 </td>
121 <td
122 class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
123
124 @if($cv->cv_job_certificate)
125
126 <a href="{{ asset('storage/' . $cv->cv_job_certificate) }}"
127 target="_blank" class="text-blue-700 hover:underline p-3">
128 <i class="fa fa-file-pdf mr-1"></i> {{ $cv -> cv_job_type }}
129 </a>
130
131 @else
132
133 {{ $cv -> cv_job_type }}
134
135 @endif
136 </td>
137 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
138 <a href="/all-cv/{{ $cv -> cv_code }}"
139 class="text-blue-700 hover:underline">
140 <i class="fa fa-info-circle mr-2"></i> CV Info
141 </a>
142 </td>
143 </tr>
144 @endforeach
145 </tbody>
146 </table>

```

Figure: 5.4.60. Partner Detailed Information Retrieval with Partner Created CVs using {{ }} Syntax

```

web.php X PartnerController.php admin-show-partner.blade.php admin-partner.blade.p
routes > web.php > ...
138 // ADMIN PARTNERS
139
140 // Admin All Partners Show
141 Route::get('/all-partners', [PartnerController::class, 'partner']);
142
143 // Admin Show Partner
144 Route::get('/all-partners/{partner_id}', [PartnerController::class, 'viewPartner']);
145

```

Figure: 5.4.61. Route Definition for Showing Partner Data by Retrieval

```
web.php PartnerController.php X admin-show-partner.blade.php admin-partne
app > Http > Controllers > PartnerController.php > PartnerController
13 class PartnerController extends Controller
95 //Admin -> Partner Edit and Update
96 > public function update(Request $request, $partner_id){ ...
122 }
123
124 //Admin -> View Partner Info
125 ✓ public function viewPartner($partner_id){
126
127     $partner = partner::where('partner_id', $partner_id)->firstOrFail();
128     $cvs = $partner->cvs;
129
130 ✓     return view('company.admin-show-partner', [
131         'partner' => $partner,
132         'cvs' => $cvs
133     ]);
134 }
135
136 //Admin -> Show All Partners
137
138 ✓ public function partner(){
139
140 ✓     $filters = request([
141         'partner_name',
142         'partner_nationality',
143         'search'
144     ]);
145
146     //filter -> Check Partner Model
147     $partners = partner::latest()->filter($filters)->get();
148
149 ✓     return view('company.admin-partner', [
150         'partners' => $partners,
151     ]);
152 }
153
154 }
155
```

Figure: 5.4.62. Partner Controller Methods for Showing Partner Data

As the Partner and C_V_Forms tables are related, the CVs can be retrieved through the partner_id attribute, which is defined in the C_V_Forms table as a foreign key.

Admin CV Retrieve

```
web.php AdminCVController.php admin-all-cv.blade.php x admin-show-cv.blade.php
resources > views > company > admin-all-cv.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > div.mb
1
3 .fy-center mb-20">
5 '>
31 i="m-5 flex flex-col justify-center">
54 i="border-black border-4 rounded-2xl text-black pb-5">
60 iass= n-screen overflow-y-auto scrollbar >
61
62 :able class="w-full table-auto bg-white text-black">
63
64 <thead>...
92 </thead>
93
94 <tbody>
95 @foreach($cvs as $cv)
96
97 <tr>
98 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
99     {{$cv -> partner -> partner_name}}
100 </td>
101 <td class="border-t border-b border-gray-300 flex justify-center">
102     @if($cv->cv_profile)
103
104         
106
107     @endif
108 </td>
109 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
110     {{$cv -> cv_name}}
111 </td>
112 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
113     <a href="/all-cv?search={{$cv -> cv_jp_level}}">
114         {{$cv -> cv_jp_level}}
115     </a>
116     @if($cv -> cv_jp_certificate)
117
118     <a href="{{ asset('storage/' . $cv -> cv_jp_certificate) }}" target="_blank"
119         class="text-blue-700 hover:underline p-3">
120         <i class="fa fa-file-pdf mr-1"></i>
121     </a>
122
123     @endif
124 </td>
125 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
126     <a href="/all-cv?search={{$cv -> cv_job_type}}">
127         {{$cv -> cv_job_type}}
128     </a>
129     @if($cv -> cv_job_certificate)
130
131     <a href="{{ asset('storage/' . $cv -> cv_job_certificate) }}" target="_blank"
132         class="text-blue-700 hover:underline p-3">
133         <i class="fa fa-file-pdf mr-1"></i>
134     </a>
135
136     @endif
137 </td>
138 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
139     <a href="/all-cv?search={{$cv -> cv_nationality}}">
140         {{$cv -> cv_nationality}}
141     </a>
142 </td>
143 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
144     <a href="/all-cv?search={{$cv -> cv_status}}">
145         {{$cv -> cv_status}}
146     </a>
147 </td>
148 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
149     <a href="/all-cv/{{$cv -> cv_code}}" class="text-blue-700 hover:underline"><i
150         class="fa fa-info-circle mr-2"></i>More Info</a>
151 </td>
152 </tr>
153 </tbody>
154 @endforeach
155 </tbody>
156 </table>
157
158 </table>
```

Figure: 5.4.63. Admin All CV Page CV Data Retrieval using {{ }} Syntax

```

AdminCVController.php X admin-all-cv.blade.php X admin-show-cv.blade.php X
mp\htdocs\200495_Phone_Myint_Maw_ISP_Project\routes\web.php layout-admin > div.flex.justify-center.mb-20 > div.w-5/6
<x-layout-admin>
  <div class="flex justify-center mb-20">
    <div class="w-5/6">
      <div class="flex justify-between mb-10">
        <a href="/all-cv" class="mt-5 ml-5 inline-block border-2 font-bold border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white shadow-lg">
          <i class="fa-solid fa-arrow-left"></i> Go to All CVs
        </a>
      </div>
      <!-- CV SECTION -->
      @include('components._view-cv')
    </div>
  </div>
</x-layout-admin>

```

Figure: 5.4.64. Admin View CV Page with view-cv component

```

AdminCVController.php X _view-cv.blade.php X admin-all-cv.blade.php X admin-show-cv.blade.php
4xl.mx-auto.bg-white.shadow-lg.rounded-xl.p-6 > table.w-full.border.border-gray-300.text-sm.mb-6 > tr.b
<div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
  <table class="w-full border border-gray-300 text-sm mb-6">
    <tr class="border border-gray-300">
      <td class="px-3 py-2 <!-- cv --> cv_name">{{ $cv->cv_name }}</td>
      <td rowspan="6" class="flex flex-col justify-center border-1 border-gray-300 text-center">
        <div class="flex flex-col items-center justify-center">
          @if($cv->cv_profile)
            
            <span class="text-xs text-gray-600">Photo</span>
          @else
            <p class="italic">Please Upload Profile Image</p>
          @endif
        </div>
      </td>
    </tr>
    <tr class="border border-gray-300">
      <th class="bg-gray-200 px-3 py-2 text-left w-32">Registered</th>
      <td class="px-3 py-2">{{ $cv -> created_at }}</td>
      <th class="bg-gray-200 px-3 py-2 text-left">Gender</th>
      <td class="px-3 py-2">{{ $cv -> cv_gender }}</td>
    </tr>
    <tr class="border border-gray-300">
      <th class="bg-gray-200 px-3 py-2 text-left">Nationality</th>
      <td class="px-3 py-2">{{ $cv -> cv_nationality }}</td>
      <th class="bg-gray-200 px-3 py-2 text-left">Address</th>
      <td class="px-3 py-2">{{ $cv -> cv_address }}</td>
    </tr>
    <tr class="border border-gray-300">
      <th class="bg-gray-200 px-3 py-2 text-left">Date of Birth</th>
      <td class="px-3 py-2">{{ $cv -> cv_dob }}</td>
      <th class="bg-gray-200 px-3 py-2 text-left">Age</th>
      <td class="px-3 py-2">{{ $cv -> cv_age }}</td>
    </tr>
    <tr class="border border-gray-300">
      <th class="bg-gray-200 px-3 py-2 text-left">Religion</th>
      <td class="px-3 py-2">{{ $cv -> cv_religion }}</td>
      <th class="bg-gray-200 px-3 py-2 text-left">Marital Status</th>
      <td class="px-3 py-2">{{ $cv -> cv_marriage }}</td>
    </tr>
    <tr class="border border-gray-300">
      <th class="bg-gray-200 px-3 py-2 text-left">Language Level</th>
      <td class="px-3 py-2">{{ $cv -> cv_jp_level }}</td>
      <th class="bg-gray-200 px-3 py-2 text-left">Job Type</th>
      <td class="px-3 py-2">{{ $cv -> cv_job_type }}</td>
    </tr>
  </table>
</div>

```

Figure: 5.4.65. View-cv component Data Retrieval with {{ }} Syntax through the Implemented Page

```

Controller.php  admin-all-cv.blade.php  admin-show-cv.blade.php  admin-completed.blade.php X
resources > views > company > admin-completed.blade.php > x-layout-admin > div.flex.justify-center.mb-20 >
1
3   r-center mb-20">
5
24  'm-5 flex flex-col justify-center">
26  'border-black border-4 rounded-2xl text-black pb-5">
63
64  <tbody>
65      @foreach($jobs as $job)
66
67      @foreach($job->application as $app)
69
70      <tr>
71
72          <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
73              {{$app -> job -> job_code}}
74          </td>
75          <td class="border-t border-b border-gray-300 flex justify-center">
76              @if($app->cv->cv_profile)
77
78                  
80
81              @endif
82          </td>
83          <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
84              {{$app -> cv -> cv_name}}
85          </td>
86          <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
87              @if($app->cv->cv_job_certificate)
88
89                  <a href="{{ asset('storage/' . $app->cv->cv_job_certificate) }}" target="_blank"
90                      class="text-blue-700 hover:underline p-3">
91                      <i class="fa fa-file-pdf mr-1"></i> {{$app->cv -> cv_job_type}}
92                  </a>
93
94              @else
95
96                  {{$app-> cv -> cv_job_type}}
97
98              @endif
99          </td>
100         <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
101             {{$app -> cv -> partner -> partner_name}}
102         </td>
103         <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
104             {{$app -> cv -> cv_nationality}}
105         </td>
106         <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
107             <a href="/all-cv/{{$app -> cv -> cv_code}}"
108                 class="text-blue-700 hover:underline">
109                 <i class="fa fa-info-circle mr-2"></i>CV Info</a>
110         </td>
111     </tr>
112 </tbody>
113
114 @endforeach
115
116 @endforeach
117
118 </tbody>
119
120 </tbody>

```

Figure: 5.4.66. Data Retrieval of CVs with 'Final-Interview Pass' Status Filtered in Passed CVs Page

```

web.php x AdminCVController.php _view-cv.blade.php admin-all-cv.blade.php
routes > web.php > ...
157
158
159 // ADMIN CV
160
161 // Admin all CV List
162 Route::get('/all-cv', [AdminCVController::class, 'allCV']);
163
164 // Admin Show CV
165 Route::get('/all-cv/{cv_code}', [AdminCVController::class, 'showCV']);
166
167 // Admin CV Completed
168 Route::get('/admin-completed', [AdminCVController::class, 'passedCV']);
169

```

Figure: 5.4.67. Route Definition for CV Data Retrieval

```

AdminCVController.php x admin-all-cv.blade.php admin-show-cv.blade.php
> Controllers > AdminCVController.php > AdminCVController > passedCV
class AdminCVController extends Controller
{
    //Admin -> Show All CVs
    public function allCV(){
        $filters = request([
            'cv_name',
            'cv_jp_level',
            'cv_job_type',
            'cv_nationality',
            'job_nationality',
            'cv_status',
            'search'
        ]);

        $cvs = CVForms::with(['Partner'])
            ->has('partner')->filter($filters)
            ->get();

        return view('company.admin-all-cv', [
            'cvs' => $cvs
        ]);
    }

    //Admin -> Show Selected CV
    public function showCV($cv_code) {
        $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();

        return view('company.admin-show-cv', [
            'cv' => $cv
        ]);
    }

    //Admin -> Show all 'Final Interview Passed' CVs
    public function passedCV(){
        $jobs = JobListing::with(['application' => function ($query){
            $query->whereHas('cv', function ($cvQuery) {
                $cvQuery->where('cv_status', 'Final-Interview Pass');
            });
        }], 'application.cv.partner')->get();

        $totalPassed = CVForms::where('cv_status', 'Final-Interview Pass')
            ->count();

        return view('company.admin-completed', [
            'jobs' => $jobs,
            'totalPassed' => $totalPassed
        ]);
    }
}

```

Figure: 5.4.68. AdminCV Controller Methods for CV Data Retrieval

In the All Passed CV page, the system displays all passed CVs that have been marked as 'Final-Interview Pass', along with the corresponding jobs each CV applied to. As both c_v_Forms and joblistings tables are linked through the applications table, the applications table is used as a central source for the system to fetch both the CV information and the associated job details.

Application Process

The details of how Application process works will be described in the Appendix. The related code screenshots will be shown in the sub-sections below.

Partner Application Process

```
partner-job-application.blade.php X ApplicationController.php AdminCVController.php 2025_1
resources > views > partner > partner-job-application.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w
1
3   = "flex justify-center mb-20">
5   lass="w-5/6">
35  iv class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-4 md:space-y-0 mx-4">
36
37  <!-- Select Section -->
38  <div class="border-black border-4 rounded-2xl text-black pb-5">
39
40  <h1 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
41    Applicable Candidate List
42  </h1>
43
44  <div class="h-96 overflow-y-auto scrollbar">
45
46    <table class="w-full table-auto bg-white text-black">
47
48      @foreach($availableCvs as $cv)
49
50        <tbody>
51
52          <tr>
53            <td class="border-t border-b border-gray-300 items-center">
54              <img src="" alt="">
55            </td>
56            <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
57              {{ $cv->cv_name }}
58            </td>
59            <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
60              <table class="w-full table-auto">
61
62                <tr>
63                  <td>
64                    {{ $cv->cv_age }}
65                  </td>
66                  <td>
67                    /
68                  </td>
69                  <td>
70                    {{ $cv->cv_gender }}
71                  </td>
72                </tr>
73              </table>
74            </td>
75            <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
76              <form>
77                <input type="hidden" name="cv_id" value="{{ $cv->id }}">
78                <button type="submit" class="transition duration-300 border
79                  rounded-2xl border-black py-1 px-2
80                  hover:bg-green-700 hover:border-green-700 hover:text-white">
81                  <i class="fa fa-plus"></i> Add
82                </button>
83              </form>
84            </td>
85          </tr>
86        </tbody>
87      @endforeach
88    </table>
89  </div>
90
91  </div>
92
93  </div>
94
95  </div>
96
97  </div>
98
99  </div>
100
```

Figure: 5.4.69. Applicable Candidate CV List Section with Add button

```

partner-job-application.blade.php X ApplicationController.php AdminCVController.php 2025_1
div.w-5/6 > div.grid.grid-cols-1.lg:grid-cols-2.gap-4.space-y-4.md:space-y-0.mx-4 > div.border-black.border-4.rou

"flex justify-center mb-20">
ass="w-5/6">
v class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-4 md:space-y-0 mx-4">

<!-- Selected Section -->
<div class="border-black border-4 rounded-2xl text-black">
<h1 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
    Applied Candidates
</h1>
@include('components._message')
<div class="h-96 overflow-y-auto scrollbar">

    <table class="w-full table-auto bg-white text-black">

        @foreach($appliedCvs as $cv)

            <tbody>

                <tr>

                    <td class="border-t border-b border-gray-300 items-center">
                        <img src="" alt="">
                    </td>

                    <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
                        {{ $cv -> cv_name }}
                    </td>

                    <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
                        <table class="w-full table-auto">

                            <tr>

                                <td>
                                    {{ $cv -> cv_age }}
                                </td>

                                <td>
                                    /
                                </td>

                                <td>
                                    {{ $cv -> cv_gender }}
                                </td>

                            </tr>

                        </table>
                    </td>

                    <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
                        <form
                            action="{{ route('job.removeCV', ['job_id' => $job->id, 'cv_id' => $cv->id]) }}"
                            method="POST">
                            @csrf
                            @method('DELETE')
                            <input type="hidden" name="cv_id" value="{{ $cv->id }}">
                            <button type="submit" class="transition duration-300 border rounded-lg
                                border-black
                                py-1 px-2 hover:border-red-700 hover:bg-red-700 hover:text-white">
                                <i class="fa fa-trash"></i> Remove
                            </button>
                        </form>
                    </td>

                </tr>

            </tbody>

        @endforeach

    </table>

</div>
</div>

```

Figure: 5.4.70. Applied CV List Section with Remove Button

```

hp partner-applied-job.blade.php X partner-job-application.blade.php ApplicationC
s > views > partner > partner-applied-job.blade.php > x-layout > main.flex.justify-center.mb-20 > di

="flex_justify-center mb-20">
ass="w-5/6">
y class="m-5 flex flex-col items-center justify-center mt-10">
@foreach($jobs as $job)

<div class="h-[600px] w-5/6 overflow-x-scroll scrollbar border-2 border-t-2 border-black mb-10">

<table class="w-full table-auto rounded-xl py-2 px-4 mb-5">

<thead>
<tr class="bg-gray-50 text-center">
<th class="px-2 py-4">Job Code:</th>
<th class="px-2 py-4 text-blue-700">{{ $job->job_code }}</th>
<th class="px-2 py-4">|</th>
<th class="px-2 py-4">Company Name:</th>
<th class="px-2 py-4 text-blue-700">{{ $job->company_name }}</th>
<th class="px-2 py-4 text-blue-700">{{ $job->job_availability }}</th>
</tr>
</thead>

<thead class="bg-black text-white">
<tr>
<th class="px-4 py-3 text-center">Photo</th>
<th class="px-4 py-3 text-center">Name</th>
<th class="px-4 py-3 text-center">Nationality / Age / Gender</th>
<th class="px-4 py-3 text-center">Job Type</th>
<th class="px-4 py-3 text-center">Language</th>
<th class="px-4 py-3 text-center">Status</th>
</tr>
</thead>

<tbody>

@foreach($job->application as $app)
<tr class="border-t border-b border-gray-300 hover:bg-gray-50 text-center">
<td class="py-3 flex justify-center">
@if($app->cv->cv_profile)



@endif
</td>
<td class="py-3">{{ $app->cv->cv_name }}</td>
<td class="py-3">
{{ $app->cv->cv_nationality }} / {{ $app->cv->cv_age }} / {{ $app->cv->cv_gender }}
</td>
<td class="py-3">

@if($app->cv->cv_job_certificate)

<a href="{{ asset('storage/' . $app->cv->cv_job_certificate) }}" target="_blank"
class="text-blue-700 hover:underline p-3">
<i class="fa fa-file-pdf mr-1"></i> {{ $app->cv -> cv_job_type}}
</a>

@else

{{ $app->cv -> cv_job_type}}

@endif

</td>
<td class="py-3">

@if($app->cv->cv_jp_certificate)

<a href="{{ asset('storage/' . $app->cv->cv_jp_certificate) }}" target="_blank"
class="text-blue-700 hover:underline p-3">
<i class="fa fa-file-pdf mr-1"></i> {{ $app->cv -> cv_jp_level}}
</a>

@else

{{ $app->cv -> cv_jp_level}}

@endif

</td>
<td class="py-3">{{ $app->cv->cv_status }}</td>
</tr>
@endforeach

</tbody>

```

Figure: 5.4.71. All Applied Job Page on Partner Side (The same as Admin Side)

```

web.php X partner-applied-job.blade.php partner-job-application.blade.php Applicat
routes > web.php > ...
79
80 // PARTNER JOB APPLICATION
81
82 // Partner Job Application Page View
83 Route::get('/job-application/{job_code}', [ApplicationController::class, 'viewapplyjob']);
84
85 // Partner Job Application Add
86 Route::post('/job-application/{job_id}/add/{cv_id}', [ApplicationController::class, 'addCV']);
87 ->name('job.addCV');
88
89 //Partner Job Application Remove
90 Route::delete('/job-application/{job_id}/remove/{cv_id}', [ApplicationController::class, 'removeCV']);
91 ->name('job.removeCV');
92
93 // Partner Applied List
94 Route::get('/partner-applied-list', [ApplicationController::class, 'appliedCV']);
95

```

Figure: 5.4.72. Route Definition for Viewing Applied Job list, Viewing Application Page, Adding CVs, and Removing CVs

```

http > Controllers > ApplicationController.php > ApplicationController > removeCV
class ApplicationController extends Controller
//Partner -> View Apply Job Page
public function viewapplyjob($job_code){

    $partnerId = auth('partner')->id();

    $job = joblisting::where('job_code', $job_code)->firstOrFail();

    $availableCvs = CVForms::where('partner_id', $partnerId)
->where('cv_status', 'Not Applied')
->where('cv_job_type', $job->job_type)
->get();

    $appliedCvs = $job->cvs()->where('partner_id', $partnerId)->get();

    return view('partner.partner-job-application', [
        'job' => $job,
        'availableCvs' => $availableCvs,
        'appliedCvs' => $appliedCvs
    ]);
}

//Partner -> Add CV in Apply Job Page
public function addCV($job_id, $cv_id){

    $cv = CVForms::where('id', $cv_id)->where('cv_status', 'Not Applied')
->first();

    Application::create([
        'job_id' => $job_id,
        'cv_id' => $cv_id,
    ]);

    $cv->update(['cv_status' => 'Applied']);

    return back()->with('message', $cv->cv_name . ' added to job successfully!');
}

//Partner -> Remove CV in Apply Job Page
public function removeCV($job_id, $cv_id){

    $application = Application::where('job_id', $job_id)
->where('cv_id', $cv_id)
->first();

    if (!$application) {
        return back()->with('error', 'Application not found. ');
    }

    $application->delete();

    $cv = CVForms::find($cv_id);
    $cv->update(['cv_status' => 'Not Applied']);

    return back()->with('message', $cv->cv_name . ' removed from job successfully!');
}

```

```

//Partner -> View Applied CVs and Job
public function appliedCV(){

    $partnerId = auth('partner')->id();

    $filters = request([
        'job_code',
        'company_name',
        'search'
    ]);

    $jobs = Joblisting::whereHas('application.cv', function ($query) use ($partnerId) {
        $query->where('partner_id', $partnerId);
    })
->with(['application' => function ($query) use ($partnerId) {
    $query->whereHas('cv', function ($q) use ($partnerId) {
        $q->where('partner_id', $partnerId);
    })
->with('cv');
    }])
->latest()->filter($filters)->paginate(5);

    return view('partner.partner-applied-job', [
        'jobs' => $jobs
    ]);
}

```

Figure: 5.4.73. Application Controller Methods for Viewing Application related Pages, and Adding and Removing Functions

```

Application.php X AdminCVContr
p > Models > Application.php > applicati
8 class application extends Model
11 use HasFactory;
12
13 protected $fillable = [
14     'cv_id',
15     'job_id',
16 ];
17

```

Figure: 5.4.74. Defining assignable attributes for mass assignment and creation

Admin Application Evaluation

```

.php X admin-evaluate.blade.php X admin-application.blade.php AdminCVController.php
orm > div.flex.justify-center > button.border-black.border-2.bg-black.text-white.rounded.py-2.px-4.hover.bg-white.f
-center mb-20">
d grid-cols-1">
col-span-1 ">
<form method="post" action="/admin-applications/{{cv->cv_code}}/evaluate"
enctype="multipart/form-data">
@csrf
@method('PUT')
<div class="font-bold text-xl text-blue-900 mb-10">
    {{cv -> cv_name}}
</div>
<div class="mb-10">
<label for="cv_status" class="inline-block text-lg mb-2 font-bold">
    Interview Result Status
</label>
<select name="cv_status" id="cv_status"
    class="border border-gray-200 rounded p-2 w-full">
    <option value="{{cv -> cv_status}}">{{cv -> cv_status}}</option>
    <option value="Not Applied" @selected(old('cv_status')=='Applied' )>Applied
    </option>
    <option value="Not Applied" @selected(old('cv_status')=='Not Applied' )>Not
    Applied
    </option>
    <option value="Pre-Interview Pass"
        @selected(old('cv_status')=='Pre-Interview Pass' )>
        Pre-Interview Pass</option>
    <option value="Final-Interview Pass"
        @selected(old('cv_status')=='Final-Interview Pass' )>
        Final-Interview Pass</option>
    <option value="Final Pass" @selected(old('cv_status')=='Left' )>Left</option>
    </select>
</div>
<div class="mb-10">
<label for="cv_evaluation" class="inline-block text-lg mb-2">
    Interview Evaluation
</label>
<textarea type="text" class="border border-gray-200 rounded p-2 w-full"
    name="cv_evaluation" rows="4" value="{{cv -> cv_evaluation}}">
</textarea>
</div>
<div class="flex justify-center">
<button type="submit" class="border-black border-2 bg-black text-white
    rounded py-2 px-4 hover:bg-white hover:text-black">
    Evaluate
</button>
</div>
</form>

```

Figure: 5.4.75. HTML Form Inputs for Applicant CV Evaluation

```
web.php × admin-evaluate.blade.php AdminCVController.php
routes > web.php > ...
176 // Admin Show CV Evaluate
177 Route::get('/admin-applications/{cv_code}/evaluate', [AdminCVController::class, 'seeEvaluate']);
178
179 // Admin EVALUATION UPDATE
180 Route::put('/admin-applications/{cv_code}/evaluate', [AdminCVController::class, 'evaluate']);
181
182
```

Figure: 5.4.76. Route Definition for CV Evaluation and Storing Updates

```
AdminCVController.php ×
Http > Controllers > AdminCVController.php > AdminCVController > pa
class AdminCVController extends Controller

//Admin -> See Evaluate CV Page
public function seeEvaluate($cv_code){

    $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();

    return view('company.admin-evaluate', [
        'cv' => $cv
    ]);
}

//Admin -> Store Evaluated Message
public function evaluate(Request $request, $cv_code){
    $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();

    $formFields = $request->validate([
        'cv_status' => 'required',
        'cv_evaluation' => 'nullable|string|max:255'
    ]);

    $cv->update($formFields);

    return back()
        ->with('message', 'CV Evaluated!');
}
```

Figure: 5.4.77. AdminCV Controller Methods for Viewing Evaluation Page and Storage of Evaluation data

File Upload Functionality

The file upload feature works the same as the CV Creation and Editing process, with the additional step for handling uploaded files. When the file is submitted and the controller validates the request, the file attribute and the storage directory are specified. The uploaded files are saved in the designated storage path, and the file path is stored in the database for later retrieval. For file storage specification, the system has to link with Laravel's storage directory using the terminal.

```
web.php partner-new-cv-register.blade.php X
resources > views > partner > partner-new-cv-register.blade.php > x-layout > div.flex.justify-center.mt-10
1
3   = "flex justify-center mb-20">
5   lass="w-3/4">
13  iv class="mx-4">
19  <div class="bg-gray-50 border border-gray-200 p-10 rounded mt-10">
41  <div class="mb-6">
42  <div class="border border-gray-200 rounded p-2 w-full">
43  <label for="cv_profile" class="font-bold text-lg">
44  | Candidate Photo (JPG/PNG)
45  </label>
46  <input type="file" name="cv_profile" accept=".jpg,.jpeg,.png"
47  | class="border border-gray-200 rounded p-2 w-full">
48  </div>
49  </div>
```

Figure: 5.4.78. File Upload section in CV Creation Form

```
if ($request->hasFile('cv_profile')) {
    $formFields['cv_profile'] = $request->file('cv_profile')
    ->store('cv/cv_profile', 'public');
}

if ($request->hasFile('cv_job_certificate')) {
    $formFields['cv_job_certificate'] = $request->file('cv_job_certificate')
    ->store('cv/job_certificates', 'public');
}

if ($request->hasFile('cv_jp_certificate')) {
    $formFields['cv_jp_certificate'] = $request->file('cv_jp_certificate')
    ->store('cv/jp_certificates', 'public');
}

if ($request->hasFile('cv_passport')) {
    $formFields['cv_passport'] = $request->file('cv_passport')
    ->store('cv/passport', 'public');
}
```

Figure: 5.4.79. File and Storage Directory Specification in the Controller

```
PS C:\xampp\htdocs\200495_Phone_Myint_Maw_ISP_Project> php artisan storage:link
```

Figure: 5.4.80. Terminal Command for Linking Storage Directory with the System

Dashboard (Chart.js and Google Calendar)

For Chart.js, the dashboard retrieves statistical data from the controller and passes it to the dashboard page, where the charts are rendered using JavaScript. For Google Calendar, it is implemented with the embed.

```
web.php x admin-home.blade.php x DashboardController.php
C:\xampp\htdocs\200495_Phone_Myint_Maw_ISP_Project\routes\web.php document.addEventListener("DOMContentLoaded") callback
1 <x-layout-admin>
3 <div class="flex justify-center mb-20">
5 <div class="w-5/6">
90
91 <!-- Jobs Bar Chart -->
92 <div class="flex-row justify-left bg-gray-50 border border-gray-200 rounded p-5 mb-10">
93 <div class="flex items-center mb-10">
94 <i class="fa-solid fa-thumbtack fa-xl mr-5"></i>
95
96 <h3 class="text-xl font-bold mr-5">
97 | Total Jobs Created
98 </h3>
99
100 <h3 class="text-xl font-bold text-blue-700">
101 | {{{totalJobs}}
102 </h3>
103 </div>
104
105 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
106 | <canvas class="max-h-[600px]" id="jobsByTypeChart"></canvas>
107 </div>
108 </div>
109
```

```
<!-- Pie Chart -->
<div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-10">
<!-- Partner Pie Chart -->
<div class="col-span-1 bg-gray-50 border border-gray-200 rounded p-5 mb-10">
<h2 class="text-2xl font-bold mb-10">
| Partner Chart
</h2>
<div class="mb-10 flex items-center justify-center text-xl font-bold text-blue-700">
| <canvas id="partnersByNationalityChart" class="max-h-[300px]"></canvas>
</div>
<div class="flex">
| <h3 class="text-xl font-bold mr-5">
| Total Partners :
| </h3>
| <h3 class="text-xl font-bold text-blue-700">
| {{{totalPartners}}
| </h3>
</div>
</div>
</div>
```

```
admin-home.blade.php x admin-home.blade.php x DashboardController.php
in-home.blade.php x layout-admin x script x document.addEventListener("DOMContentLoaded") callback
<x-layout-admin>
<div class="flex justify-center mb-20">
<div class="w-5/6">
<div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-10">
</div>
<!-- CV Pie Chart -->
<div class="col-span-1 bg-gray-50 border border-gray-200 rounded p-5 mb-10">
<h2 class="text-2xl font-bold mb-2">
| CV Chart
</h2>
<div class="m-10 flex items-center justify-center text-xl font-bold text-blue-700">
| <canvas id="cvsByNationalityChart" class="max-h-[300px]"></canvas>
</div>
<div class="flex">
| <h3 class="text-xl font-bold mr-5">
| Total CVs :
| </h3>
| <h3 class="text-xl font-bold text-blue-700">
| {{{totalCVs}}
| </h3>
</div>
<div class="flex">
| <h3 class="text-xl font-bold mr-5">
| Passed Applicants :
| </h3>
| <h3 class="text-xl font-bold text-blue-700">
| {{{totalPassed}}
| </h3>
</div>
</div>
</div>
```

Figure: 5.4.81. HTML sections for displaying Statistical data and Charts

```

web.php | admin-home.blade.php | DashboardController.php
C:\xampp\htdocs\200495_Phone_Myint_Maw_ISP_Project\routes\web.php | it.addEventListener('DOMContentLoaded')
1 <x-layout-admin>
191 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
192
193 <script>
194 <script>
195 document.addEventListener('DOMContentLoaded', () => {
196
197 //Job Type Barchart
198 fetch('/chart/jobs-by-type')
199 .then(response => response.json())
200 .then(data => {
201 const labels = data.map(item => item.job_type);
202 const counts = data.map(item => item.count);
203
204 new Chart(document.getElementById('jobsByTypeChart'), {
205 type: 'bar',
206 data: {
207 labels: labels,
208 datasets: [{
209 label: 'Jobs by Type',
210 data: counts,
211 borderWidth: 1
212 }]}],
213 },
214 options: {
215 responsive: true,
216 plugins: {
217 legend: {
218 display: false
219 }},
220 scales: {
221 y: {
222 beginAtZero: true
223 }},
224 }
225 });
226
227 });
228
229
web.php | admin-home.blade.php | DashboardController.php
1 <x-layout-admin>
2 <script>
3 document.addEventListener('DOMContentLoaded', () => {
4
5 //Partners by Nationality
6 fetch('/chart/partners-by-nationality')
7 .then(response => response.json())
8 .then(data => {
9 const labels = data.map(item => item.partner_nationality);
10 const counts = data.map(item => item.count);
11
12 new Chart(document.getElementById('partnersByNationalityChart'), {
13 type: 'pie',
14 data: {
15 labels: labels,
16 datasets: [{
17 label: 'Partners by Nationality',
18 data: counts,
19 backgroundColor: [
20 '#3882F6', '#10B981', '#F59E08', '#EF4444',
21 '#8B5CF6', '#8B5CF6',
22 '#EC4899', '#14B8A6'
23 ]},
24 }]}],
25 },
26 options: {
27 responsive: true,
28 plugins: {
29 legend: {
30 position: 'bottom'
31 }},
32 }
33 });
34
35 });
36
37 });
38
39

```

```

admin-home.blade.php | DashboardController.php
blade.php | <x-layout-admin> | <script> | document.addEventListener('DOMContentLoaded')
out-admin>
<script>
document.addEventListener('DOMContentLoaded', () => {
// CVs by Nationality
fetch('/chart/cvs-by-nationality')
.then(response => response.json())
.then(data => {
const labels = data.map(item => item.cv_nationality);
const counts = data.map(item => item.count);

new Chart(document.getElementById('cvsByNationalityChart'), {
type: 'pie',
data: {
labels: labels,
datasets: [{
label: 'CVs by Nationality',
data: counts,
}]}],
options: {
responsive: true,
plugins: {
legend: {
position: 'bottom'
}
}
}
});
});

```

Figure: 5.4.82. Chart.js JavaScript for Chart Rendering

```

web.php | admin-home.blade.php | DashboardController.php
routes > web.php > ...
107
108 Route::get('/chart/jobs-by-type', [DashboardController::class, 'jobsbytype']);
109
110 Route::get('/chart/cvs-by-nationality', [DashboardController::class, 'cvsbynationality']);
111
112 Route::get('/chart/partners-by-nationality', [DashboardController::class, 'partnersbynationality']);
113

```

Figure: 5.4.83. Route Definition for Requesting Statistical Data

```

php      admin-home.blade.php      DashboardController.php X
http > Controllers > DashboardController.php > DashboardController
class DashboardController extends Controller
{
    //Admin -> Default Home Page -> Dashboard
    public function home(){...
    }

    //Admin -> Default Home Page -> Barchart
    public function jobsbytype(){

        return Cache::remember('chart_jobs_by_type', 60, function() {
            return JobListing::selectRaw('job_type, COUNT(*) as count')
                ->groupBy('job_type')
                ->get();
        });
    }

    //Admin -> Default Home Page -> Partner Pie Chart
    public function partnersbynationality(){

        return Cache::remember('chart_partners_by_nationality', 60, function () {
            return Partner::selectRaw('partner_nationality, COUNT(*) as count')
                ->groupBy('partner_nationality')
                ->get();
        });
    }

    //Admin -> Default Home Page -> CV Pie Chart
    public function cvsbynationality(){

        return CVForms::selectRaw('cv_nationality, COUNT(*) as count')
            ->groupBy('cv_nationality')
            ->get();
    }
}

```

Figure: 5.4.84. Dashboard Controller Methods for returning statistical data to Dashboard Page

```

admin-home.blade.php X      DashboardController.php
s > company > admin-home.blade.php > x-layout-admin > script > document.addEventListener("DOMContentLoaded") callback >
ut-admin>
v class="flex justify-center mb-20">
<div class="w-5/6">
    <!-- Calendar-->
    <div class="flex justify-center mb-10">
        <div class="relative overflow-y-auto" style="height: 600px;">
            <h1 class="text-3xl font-bold m-5">Company Calendar</h1>
            <iframe
                src="https://calendar.google.com/calendar/embed?src=phonemyintmaw.ge%40gmail.com&ctz=Asia%2FTokyo"
                style="border: 0" width="1000" height="600" frameborder="0" scrolling="no">
            </iframe>
        </div>
    </div>
</div>
</div>

```

Figure: 5.4.85. Google Calendar Embed within Admin Dashboard Page

Filtering (Keyword/Search)

For the keyword-based search feature, filtering logic is defined in the models specify how the data should be filtered based on user inputs using filtering query scopes. The controllers uses these scopes through the filter() method to retrieve only the matching results. As all the keyword-based filtering mechanisms for CVs, jobs, partners, and applications are the same, only one example is presented.

```
joblisting.php x AdminJoblistingController.php
p > Models > joblisting.php > joblisting > application
8 class joblisting extends Model
9
10 }
11
12 //Filter Function
13 public function scopeFilter($query, array $filters){
14     if(!empty($filters['company_name'])){
15         $query->where('company_name', 'like', '%' . $filters('company_name') . '%');
16     }
17     if(!empty($filters['job_type'])){
18         $query->where('job_type', 'like', '%' . $filters('job_type') . '%');
19     }
20     if(!empty($filters['job_title'])){
21         $query->where('job_title', 'like', '%' . $filters('job_title') . '%');
22     }
23     if(!empty($filters['required_jp_level'])){
24         $query->where('required_jp_level', 'like', '%' . $filters('required_jp_level') . '%');
25     }
26     if(!empty($filters['job_nationality'])){
27         $query->where('job_nationality', 'like', '%' . $filters('job_nationality') . '%');
28     }
29     if (!empty($filters['search'])) {
30         $query->where(function ($q) use ($filters) {
31             $q->Where('company_name', 'like', '%' . $filters['search'] . '%')
32             -> orWhere('job_type', 'like', '%' . $filters['search'] . '%')
33             -> orWhere('job_title', 'like', '%' . $filters['search'] . '%')
34             -> orWhere('required_jp_level', 'like', '%' . $filters['search'] . '%')
35             -> orWhere('job_nationality', 'like', '%' . $filters['search'] . '%');
36         });
37     }
38 }
39 }
```

Figure: 5.4.86. Defining Job Filtering Logic in Joblisting Model

```

AdminJoblistingController.php X
> Controllers > AdminJoblistingController.php > AdminJoblistingController >
class AdminJoblistingController extends Controller
//Admin -> Show All Jobs
public function joblistAdmin(){
    $filters = request(
        [
            'company_name',
            'job_type',
            'job_title',
            'required_jp_level',
            'job_nationality',
            'search'
        ]
    );
    //filters -> Check joblisting Model
    $jobs = joblisting::latest()-> filter($filters)->get();
    return view('company.admin-job', [
        'jobs' => $jobs,
    ]);
}

```

Figure: 5.4.87. Retrieving the Keywords from the search form to filter Jobs

PDF View Rendering

The system uses the composer package `barryvdh/laravel-dompdf` for generating Viewable PDF file from HTML format. When the user requests PDF view, the template data is sent through the defined routes to the corresponding controller method. The DOMPDF service renders the HTML and CSS template into a downloadable PDF file in the methods. As the HTML templates for CV and Jobs are identical to the `view-cv` and `view-job` templates except for the traditional CSS usage, only one HTML template with the route definition and the controller will be shown.

```

PDFController.php X web.php X cv.blade.php joblisting.blade.php
routes > web.php > ...
183
184 // CV/JOB PDF Download
185
186 Route::get('/cv/{cv_code}/pdf', [PDFController::class, 'cvPdf']);
187 Route::get('/job/{job_code}/pdf', [PDFController::class, 'jobPdf']);
188
189

```

Figure: 5.4.88. Route Definition for PDF Rendering

```

PDFController.php X web.php cv.blade.php joblisting.blade.php
app > Http > Controllers > PDFController.php > PDFController > jobPdf
8 use Illuminate\Http\Request;
9
10 class PDFController extends Controller
11 {
12     //Generate CV PDF
13     public function cvPdf($cv_code){
14         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
15         $pdf = Pdf::setPaper('A4', 'portrait')
16             ->loadView('view-as-pdf.cv', compact('cv'));
17
18         return $pdf->stream("CV-{$cv->cv_code}.pdf");
19     }
20 }
21
22 //Generate JOB PDF
23 public function jobPdf($job_code){
24     $job = joblisting::where('job_code', $job_code)->firstOrFail();
25     $pdf = Pdf::setPaper('A4', 'portrait')
26         ->loadView('view-as-pdf.joblisting', compact('job'));
27
28     return $pdf->stream("Job-{$job->job_code}.pdf");
29 }
30
31 }
32

```

Figure: 5.4.89. PDF Controller Methods for Generating HTML Template into viewable PDF format

```

Controller.php  web.php  cv.blade.php X
ces > views > view-as-pdf > cv.blade.php > html > body >
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>{{ $cv->cv_name }} - Curriculum Vitae</title>
<style>
body {
font-family: DejaVu Sans, sans-serif;
background-color: #ffff;
display: flex;
justify-content: center;
font-size: 12px;
color: #000;
}
h2,
h3 {
text-align: center;
}
h2 {
font-size: 18px;
font-weight: bold;
margin-top: 40px;
margin-bottom: 50px;
}
h3 {
font-size: 14px;
font-weight: bold;
margin-top: 25px;
margin-bottom: 10px;
}
table {
width: 100%;
border-collapse: collapse;
margin-bottom: 10px;
}
th,
td {
border: 1px solid #d1d5db;
padding: 6px 8px;
text-align: left;
vertical-align: top;
}
th {
background-color: #e5e7eb;
font-weight: bold;
}
.font-bold {
font-weight: bold;
}
.text-center {
text-align: center;
}
.photo {
width: 100px;
height: 120px;
object-fit: cover;
border: 1px solid #d1d5db;
}
.photo-cell {
text-align: center;
vertical-align: top;
width: 120px;
}
.evaluation-box {
border: 1px solid #d1d5db;
height: 200px;
padding: 10px;
}
</style>
</head>

```

```

Views > view-as-pdf > cv.blade.php > html >
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.

```

Figure: 5.4.90. HTML Template for CV Form with Traditional CSS

5.5. Risk Management

Effective risk management was necessary to ensure that the development and testing were conducted without any disruptions. The potential risks were identified and monitored throughout the project development. The table below summarizes the risks and the strategies used to minimize impacts.

Risks	Impact	Likelihood	Mitigation Strategy
Database Connection Failure	High	Medium	Careful .env file configuration
Security Vulnerabilities	High	Low	Use Laravel's validation, authentication, and guarded routes
Loss of Project files	High	Low	Saving backup files
Functional Bugs	High	Low	Iterative testing before finalizing
Browser Compatibility Issues	High	Medium	Testing with various browsers
Responsiveness Issues	Medium	Medium	Apply Tailwind responsive classes and test on multiple browsers
Delay in Feature Implementations	High	Low	Prioritise Core features and follow Agile Iterations

Table: 5.5.1. Risk Impacts and Management Strategies

5.6. Problems Encountered

Throughout the development of the Oshigoto Japan System, there were several technical problems and issues, and the following issues were addressed during the implementation.

Database Configuration

MySQL is used as the system's database, and the default database of Laravel is SQLite. Due to this, the database connection attempts failed due to incorrect settings in the .env file. By configuring the database driver, host IP, host port, database name, username, and password, the issue was resolved.

Authentication Guard Conflicts

Using a separate authentication guard for admin and partner users initially caused incorrect redirections due to incomplete configuration in the auth.php file. The problem is solved by adjusting guard and provider settings in the auth.php file and ensuring each controller uses its respective guards correctly.

Migration and Seeding Errors

Some migration files failed, and seeding caused an error due to not declaring the relationships between the tables, while the foreign keys were already declared in the migration files. The issue is solved by correctly defining the model relationships.

Create and Edit Errors

While testing the creation and edition of CV and Job, they failed due to data mismatch and missing attributes in the model's \$fillable property. This failure is solved by reviewing all the fillable attributes, and the actual inputs are defined correctly.

File Upload Storage Issue

There was an issue where the file was uploaded but not saved due to not linking the system with Laravel's storage. It is resolved by running `php artisan storage:link` on the terminal and correcting file paths.

PDF Generation Error

At first, the PDF file view of CVs and Jobs from the view pages did not render correctly due to Tailwind CSS not being supported by DOMPDF, which caused a distorted layout and missing styles. So, the separate file for the PDF template was created using traditional CSS to ensure correct formatting.

5.7. Data Migration

As Oshigoto Japan did not previously have a digital job application system, and no database to migrate into the new platform, the data migration process is focused on preparing initial system data.

Creating Initial System Users

For testing purposes, the admin account and several partner accounts were generated through Laravel's seeders to allow testing of authentication and role-based access.

Generating Test and Demo Data

Laravel seeders and factories were used for generating sample joblistings data to populate the database, allowing the verification of the system features such as filtering, viewing, and application.

Manual Data Entry Through the System

PDF Files and CV Creation entries are manually inserted to test the user input feature and verify form handling and validation.

5.8. User Manual

Partner Side Manual

Log In

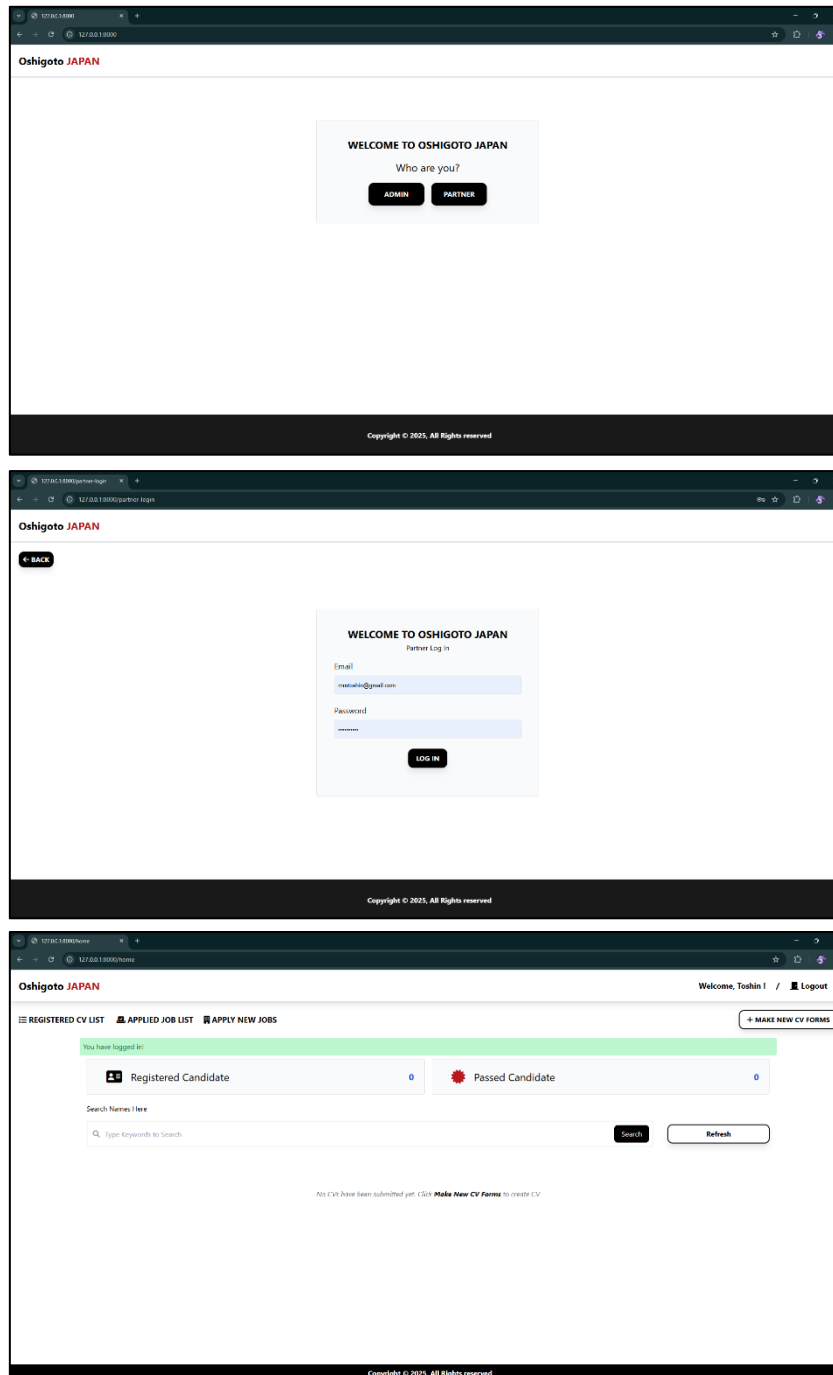


Figure: 5.8.1. Log In Process of Partner User Side

Figure:5.8.1. explains how the partner logs in to the job application web portal. Firstly, the user must choose the user type. If the user is the partner, the partner will have to fill email and password to log in. When logged in, the partner will then be redirected to the partner home page.

CV Creation

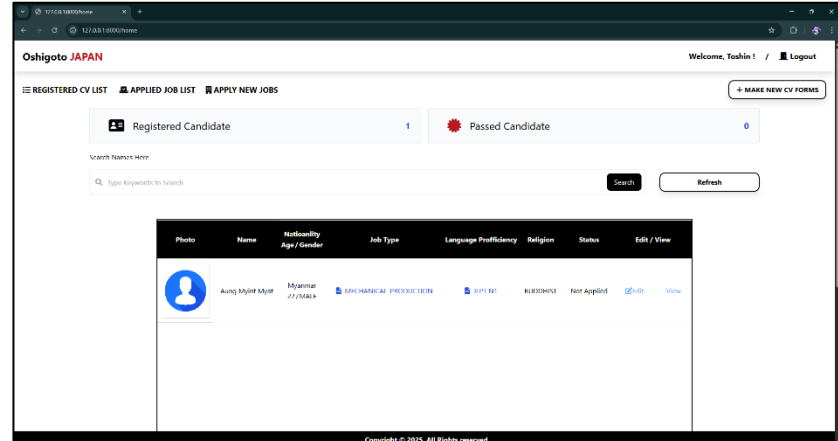
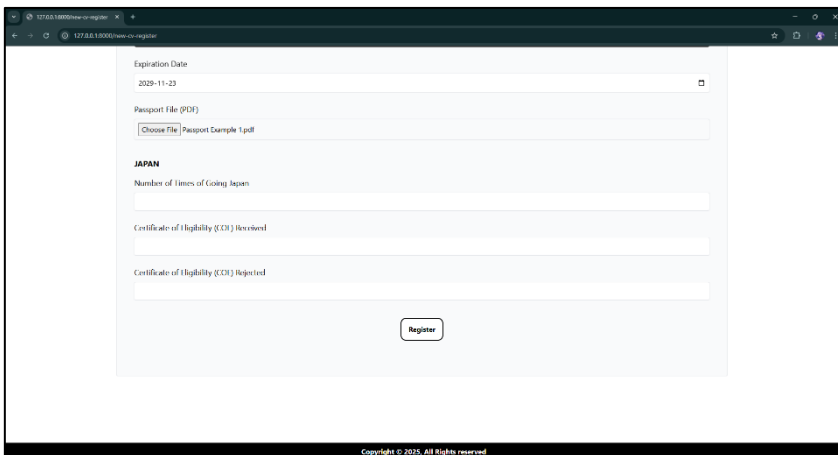
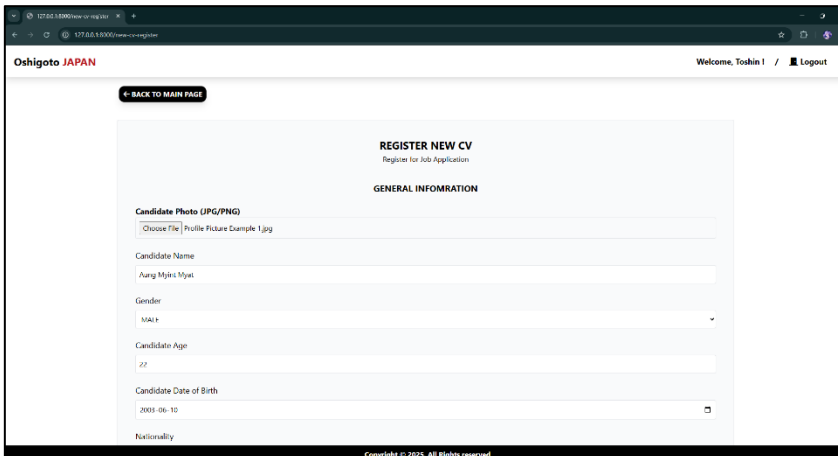
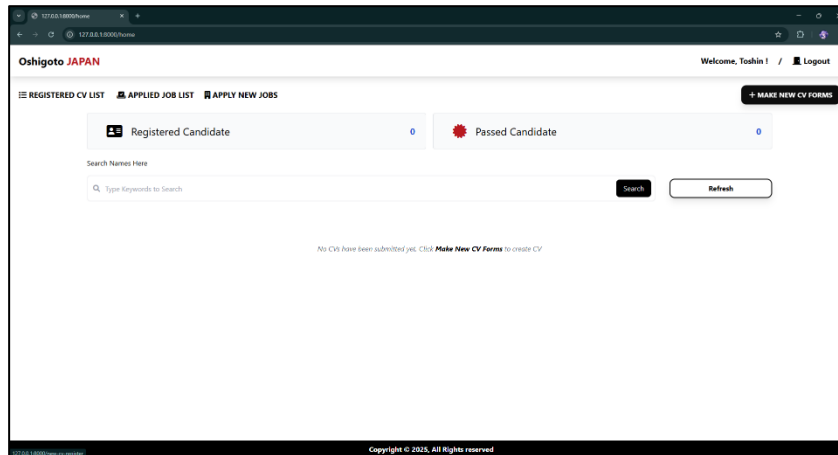


Figure: 5.8.2. CV Creation Process of Partner User

Figure: 5.8.2. shows the process of CV Creation. The Partner can create a new CV starting from clicking the Make New CV Forms button, which redirects the user to the CV creation page. Now, the partner can fill required information and click Register to store data. After clicking the Register Button, the CV is saved, and the user is redirected to the Home Page, where the added CV is displayed in a table.

Edit CV

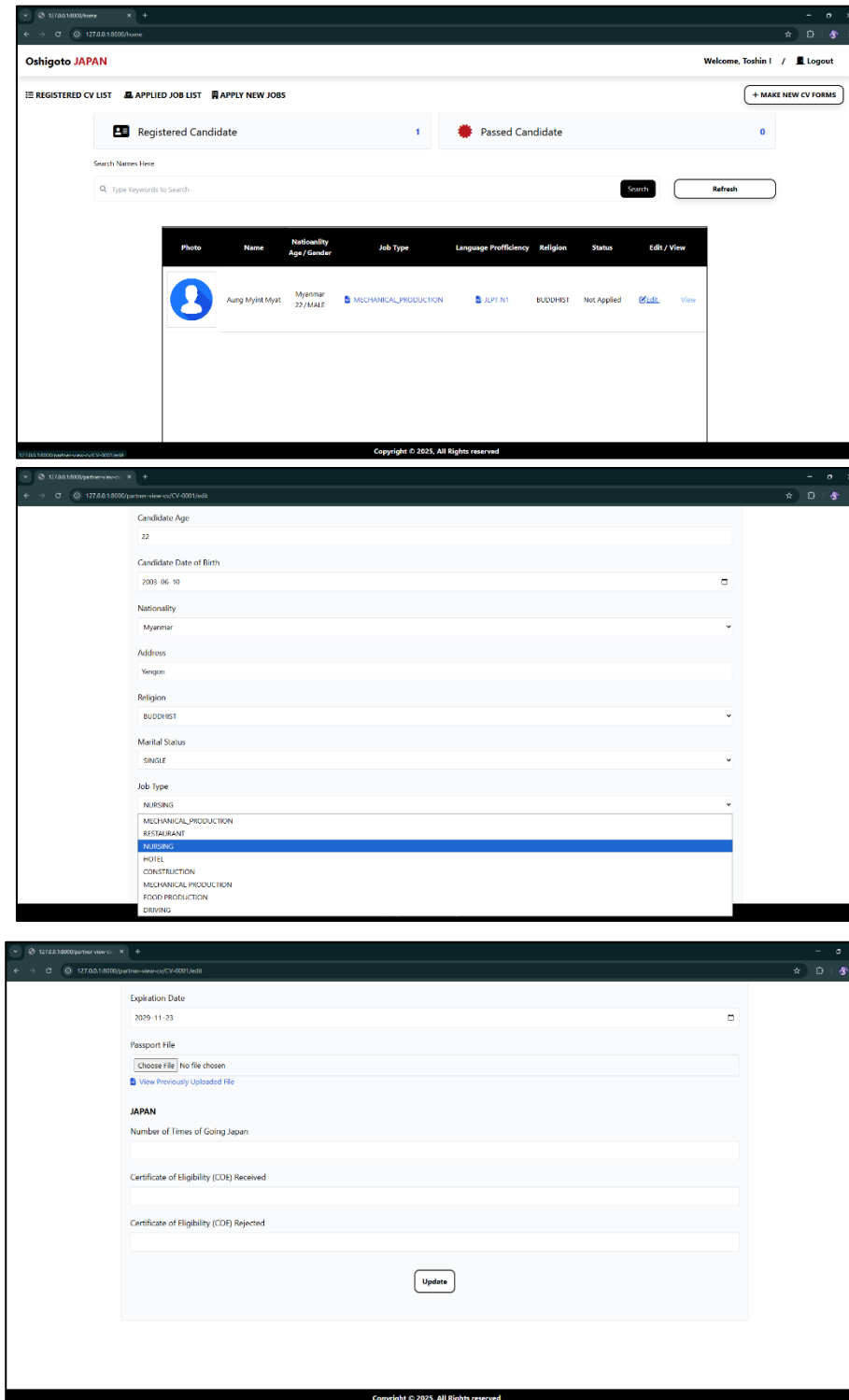


Figure: 5.8.3. CV Edit Process of Partner User 1

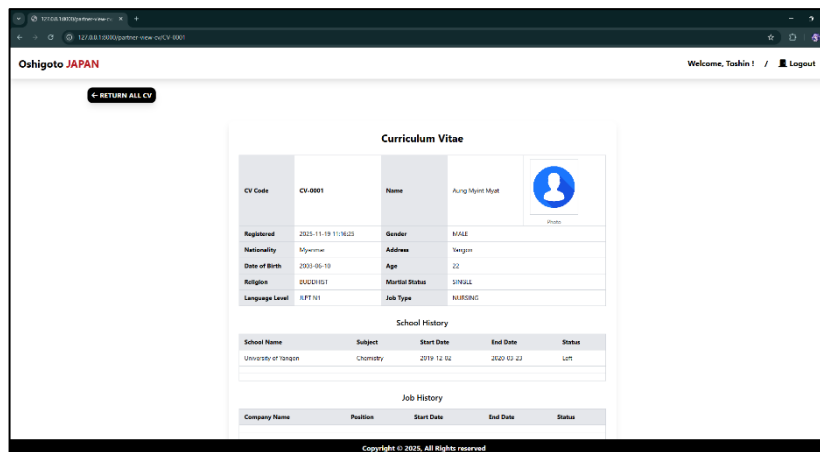
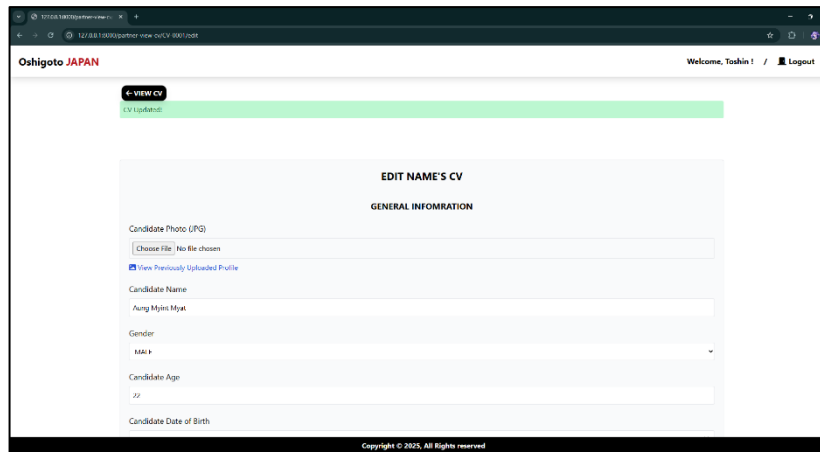


Photo	Name	Natioanlity Age / Gender	Job Type	Language Proficiency	Religion	Status	Edit / View
	Aung Myint Myat	Myanmar 22/MALE	NURSING	JLPT N1	BUDDHIST	Not Applied	Edit View

Figure: 5.8.4. CV Edit Process of Partner User 2

Figure: 5.8.3. shows the edit process, where the user clicks the edit link to access the Edit Page of the selected CV. After editing the desired information and clicking the Update button, the CV is updated, and the success message will appear at the top of the edit page as shown in Figure: 5.8.4. After updating, the user can return from the View CV button, through the CV view page, to the All CV Page (Home), where the updated information is displayed.

Job Application

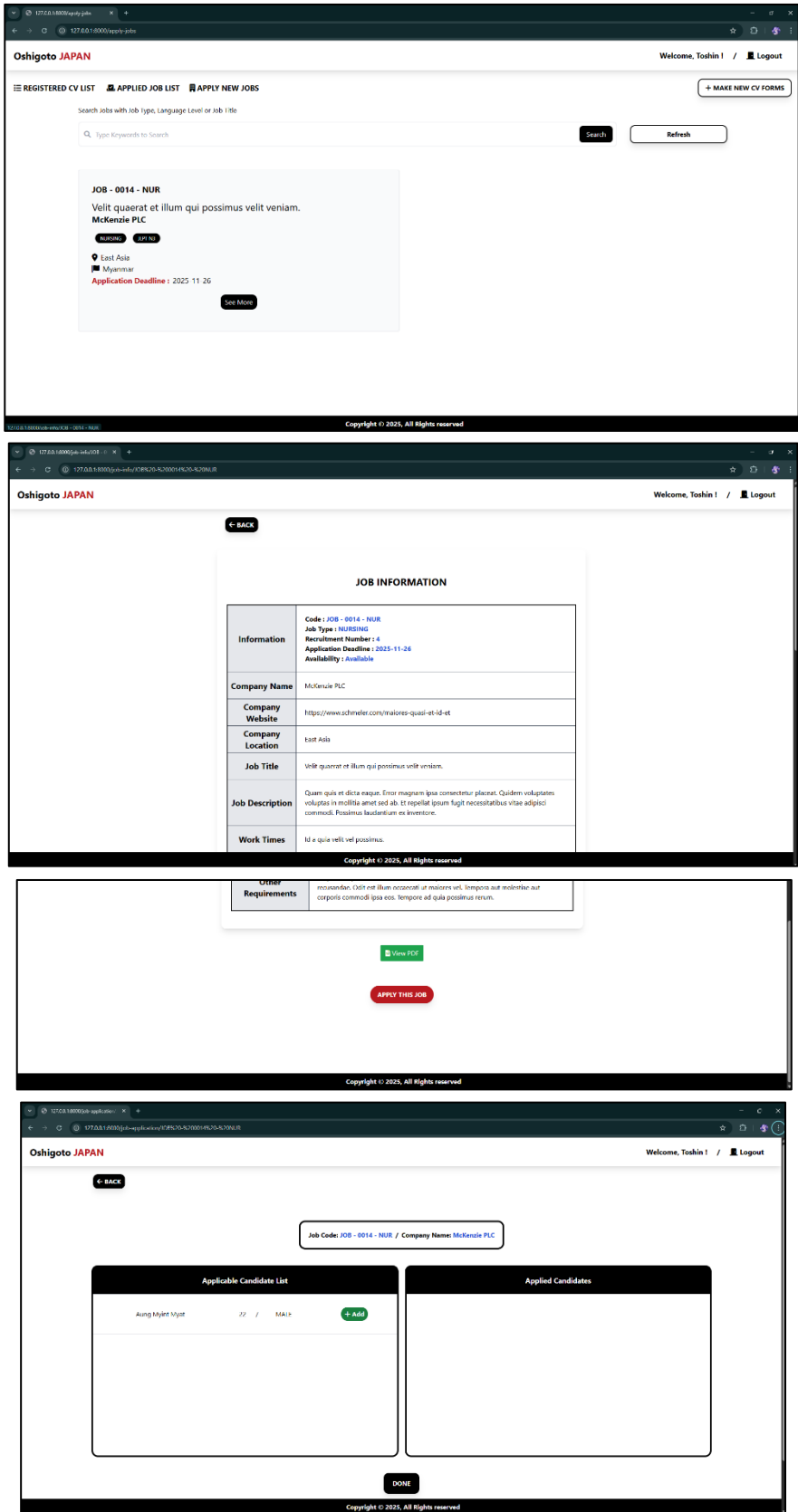


Figure: 5.8.5. Job Application Process of Partner User 1

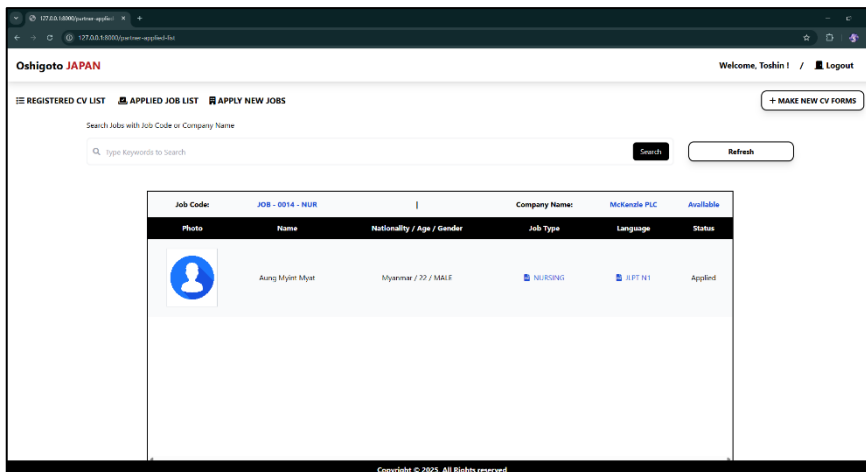
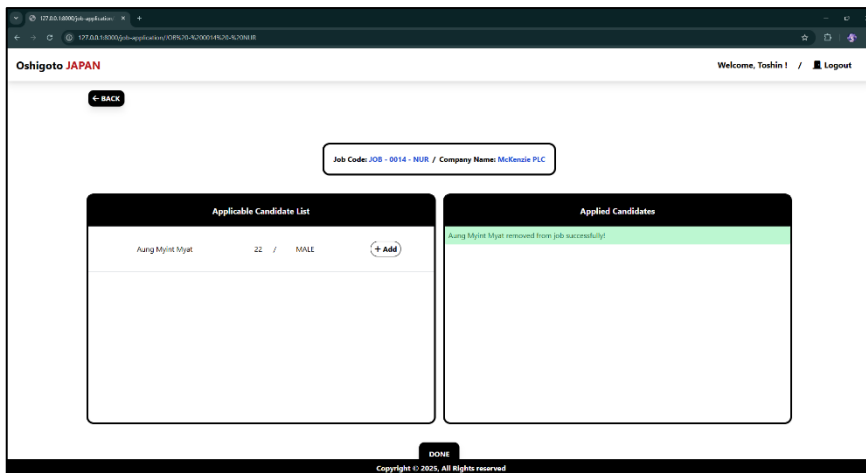
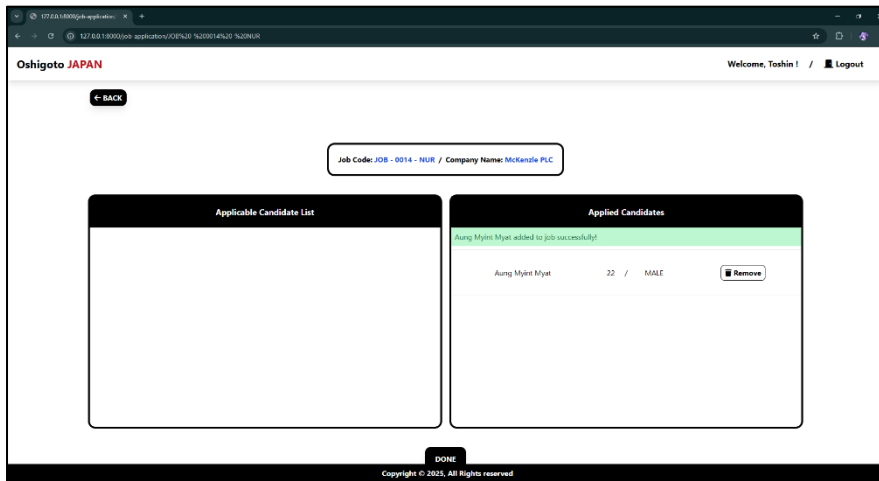


Figure: 5.8.6. Job Application Process of Partner User 2

Figure: 5.8.5 shows how the user navigates from the Job list page, through the job view page's Apply button, to the application page. After reaching the Application Page, the partner can add a job-type filtered CV from the Applicable Candidate List on the left side to the Applied Candidates on the right side. The Addition and Removal from both sides of the list are shown in the first two screenshots in Figure: 5.8.6. After adding, click the done button, which redirects to the Applied Job List Page, where the job with the applied candidate record is shown.

Log Out

When the partner clicks the Log Out Button from the Navigation Bar, the system ends the user's session and redirects to the initial login page shown below with the log out message.

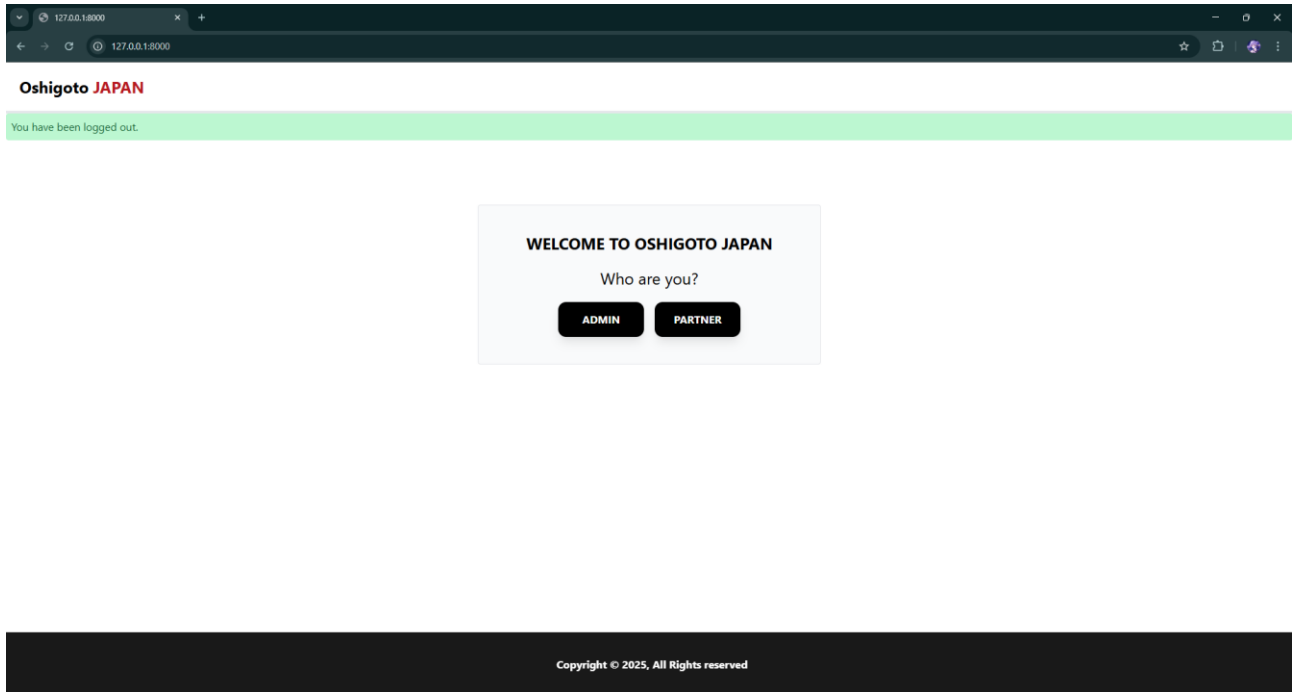


Figure: 5.8.7. Partner Log Out

Admin Side Manual

Log In

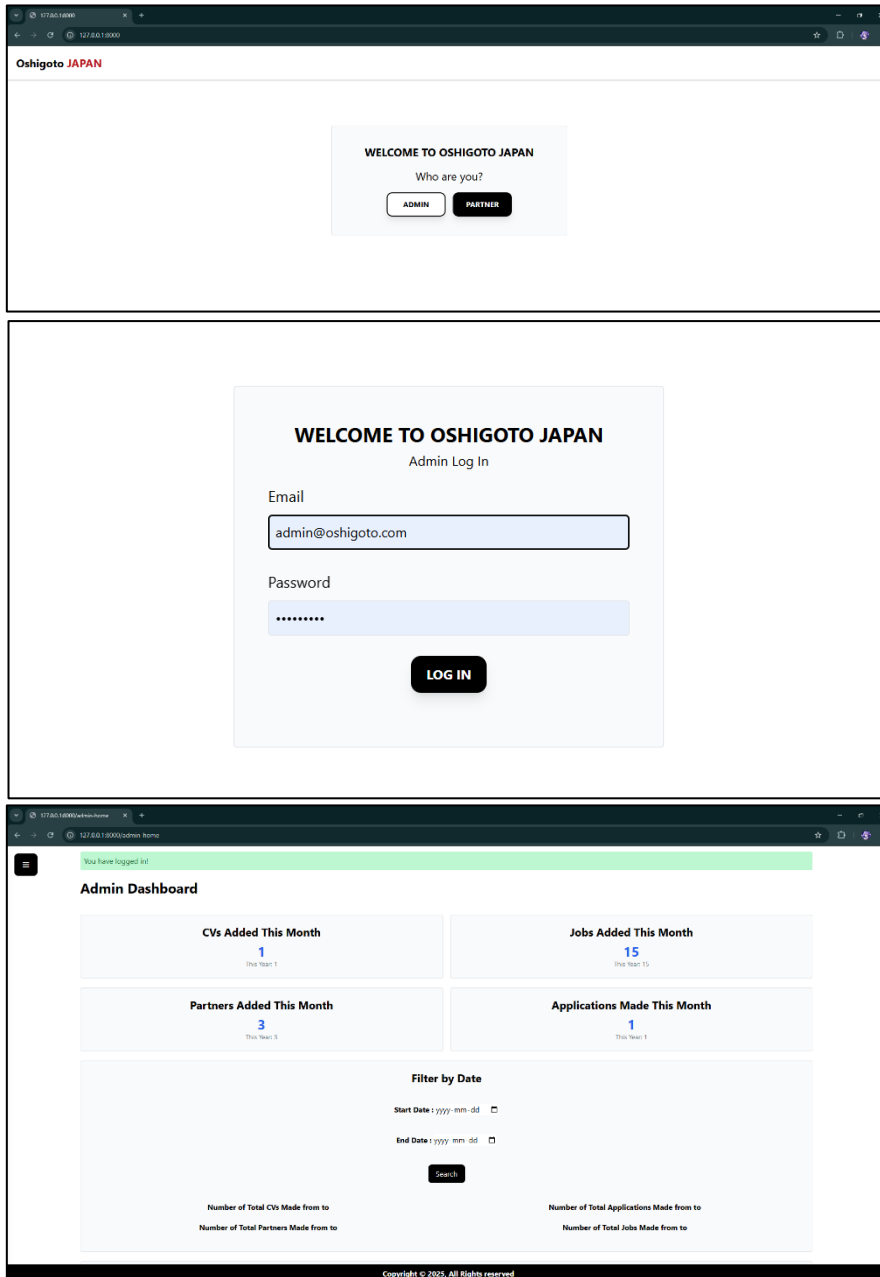


Figure: 5.8.8. Admin Log In

Figure: 5.8.8. shows the Log In process of the Admin User. The admin has to choose the Admin Side Log In Page and fill necessary email and password to log in. After clicking the Log In button, it redirects the user to the Admin Dashboard Page.

Date Range Search

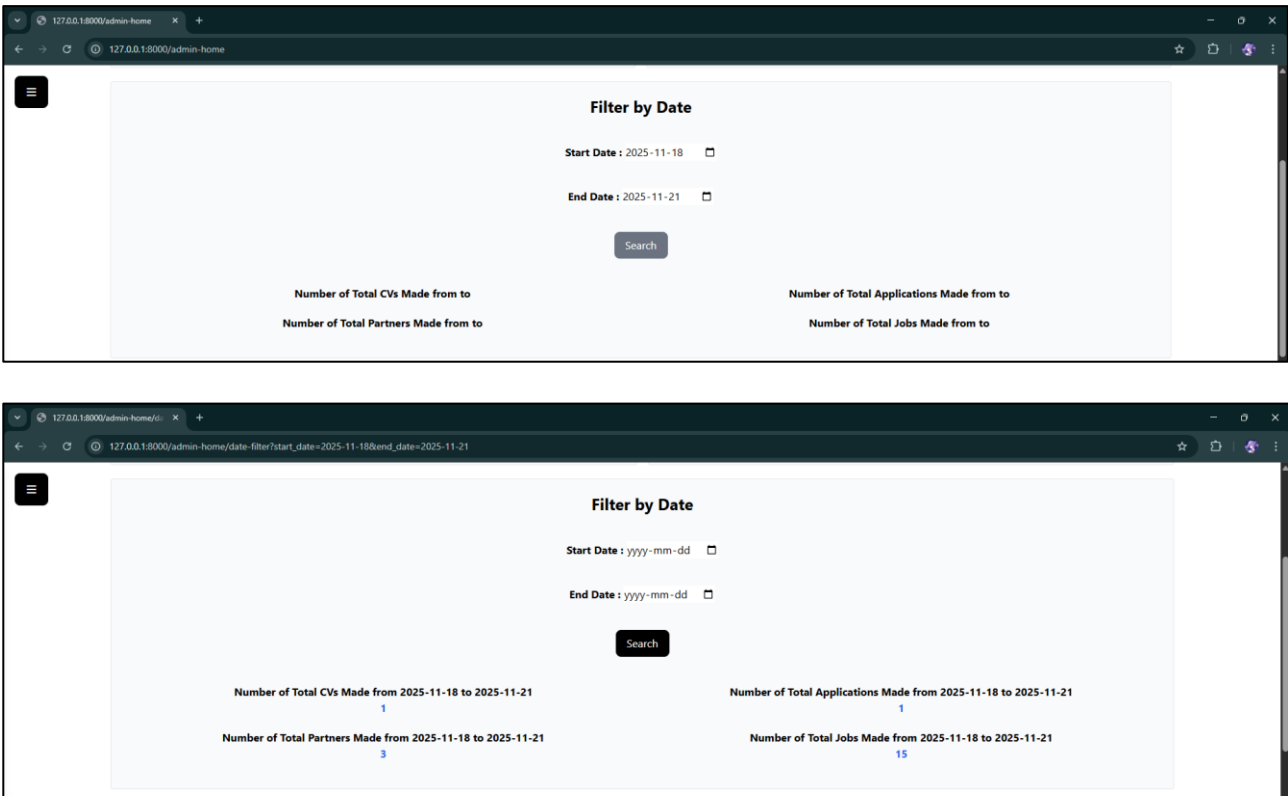


Figure: 5.8.9. Date Range Filtration

To filter the statistics using Date Range, the user has to enter the start date and end date to search. After clicking the Search Button, the statistics according to the entered date range as shown in Figure: 5.8.9.

Add Jobs

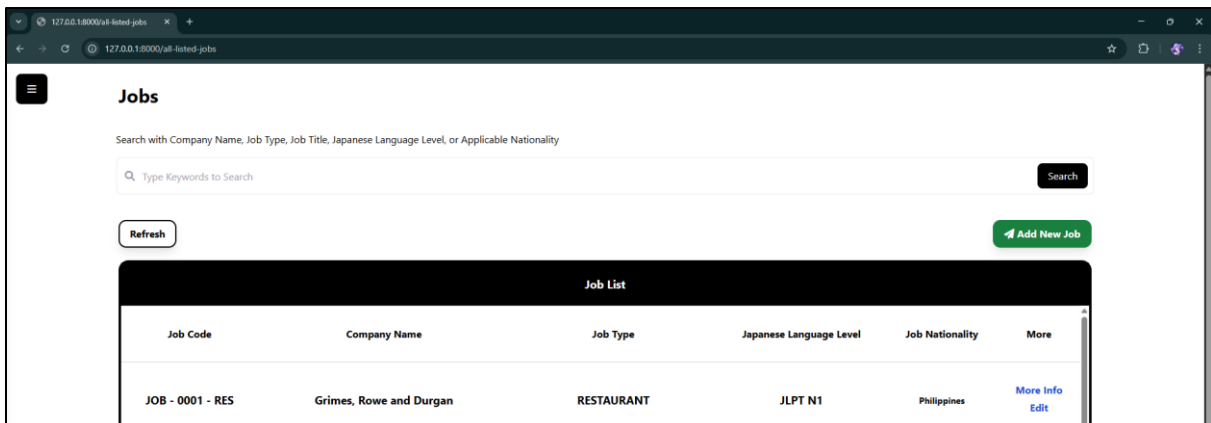


Figure: 5.8.10. Job Creation of Admin Side User 1

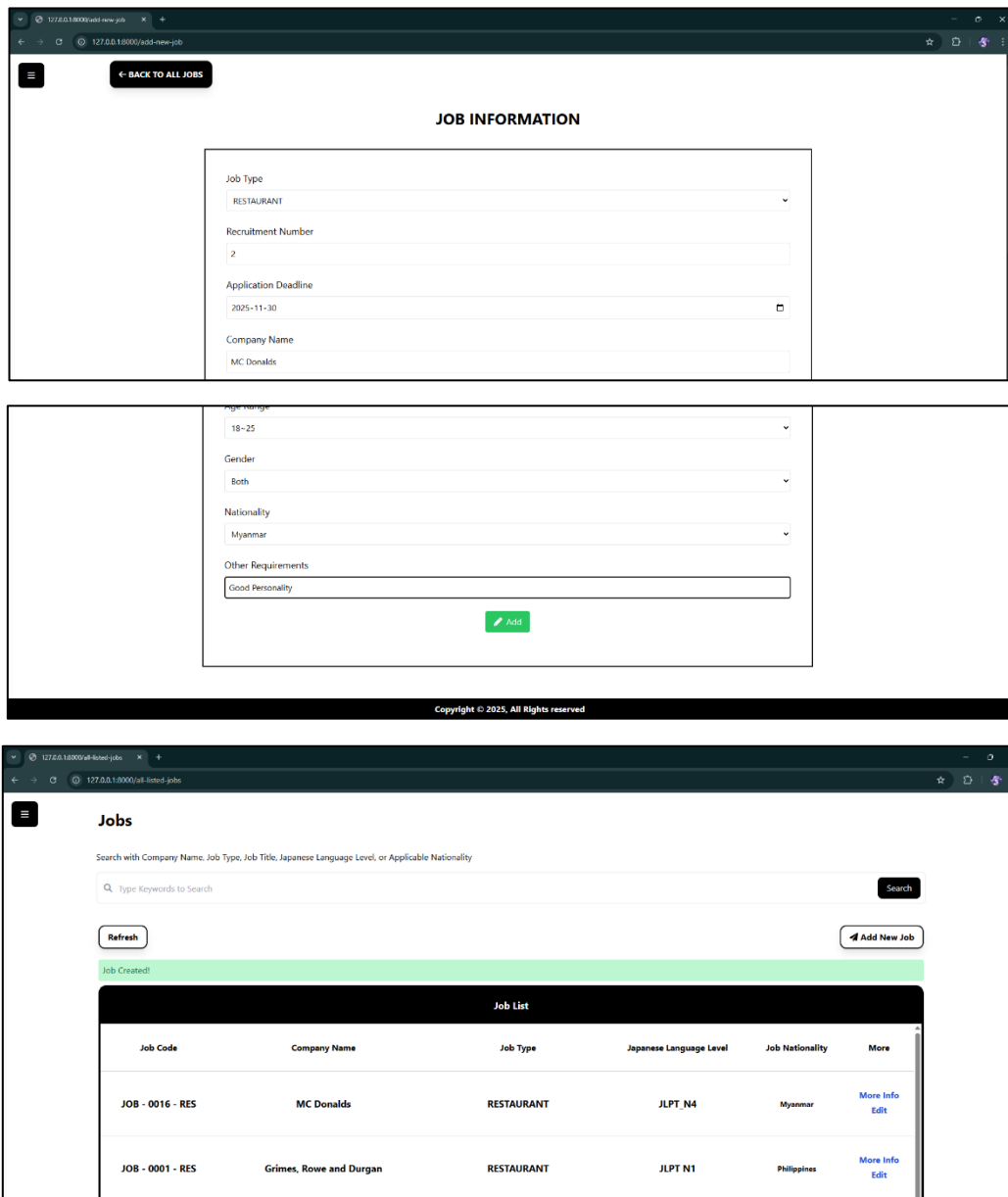


Figure: 5.8.11. Job Creation of Admin Side User 2

Figure: 5.8.10. shows the Job List Page, where the user can create a job by clicking the Add New Job button. Clicking the button redirects the user to the job creation page. After filling necessary information and clicking the add button, the system saves the data and redirects the user to the Job List Page, showing a successful creation message as shown in Figure: 5.8.11.

Edit Job

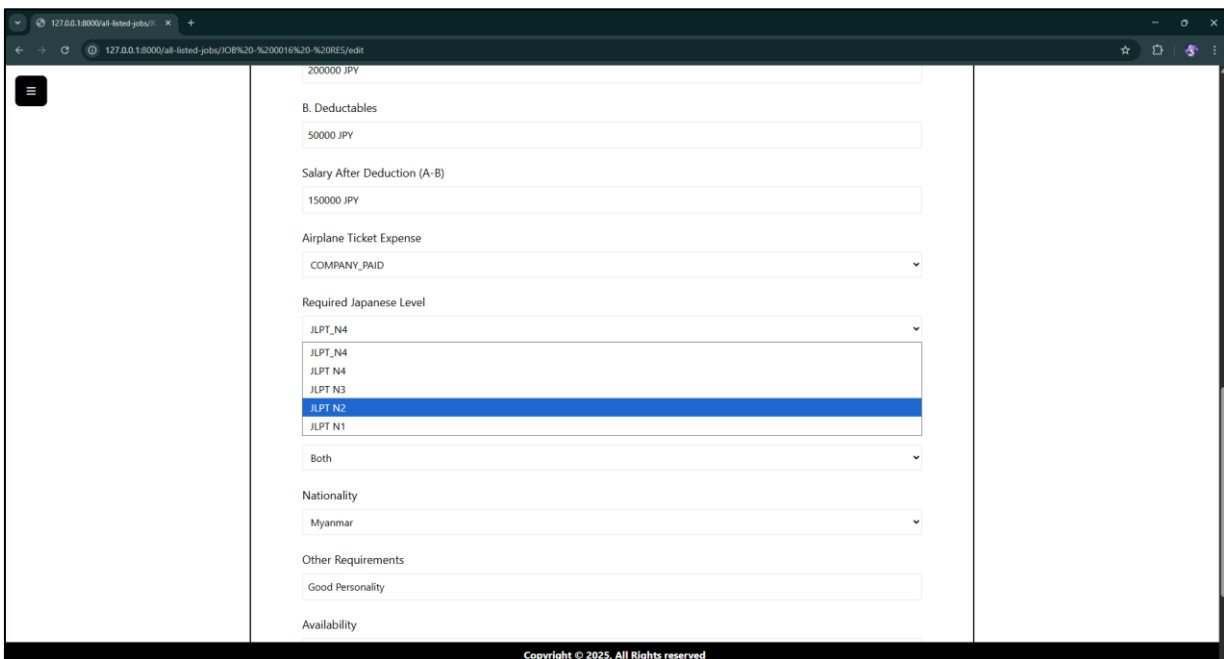
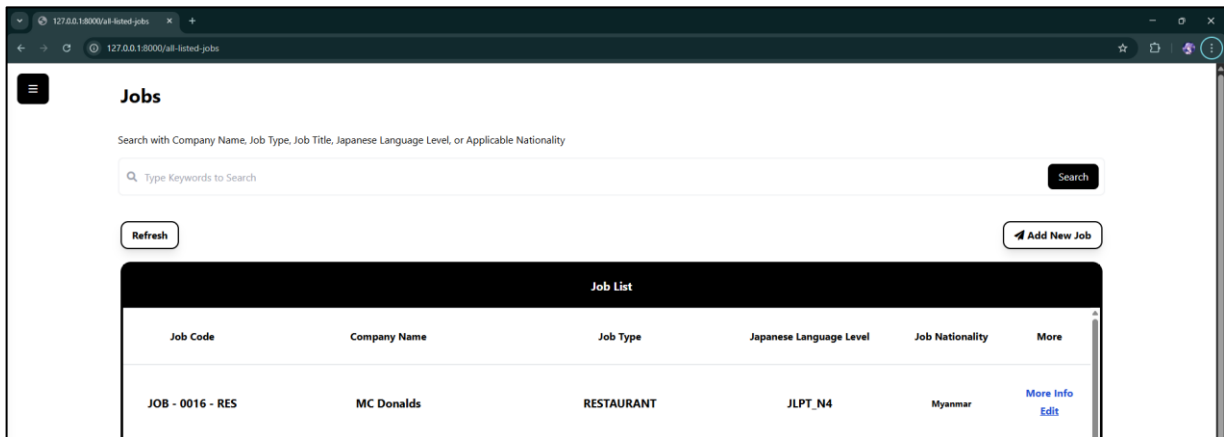


Figure: 5.8.12. Job Edition of Admin Side User 1

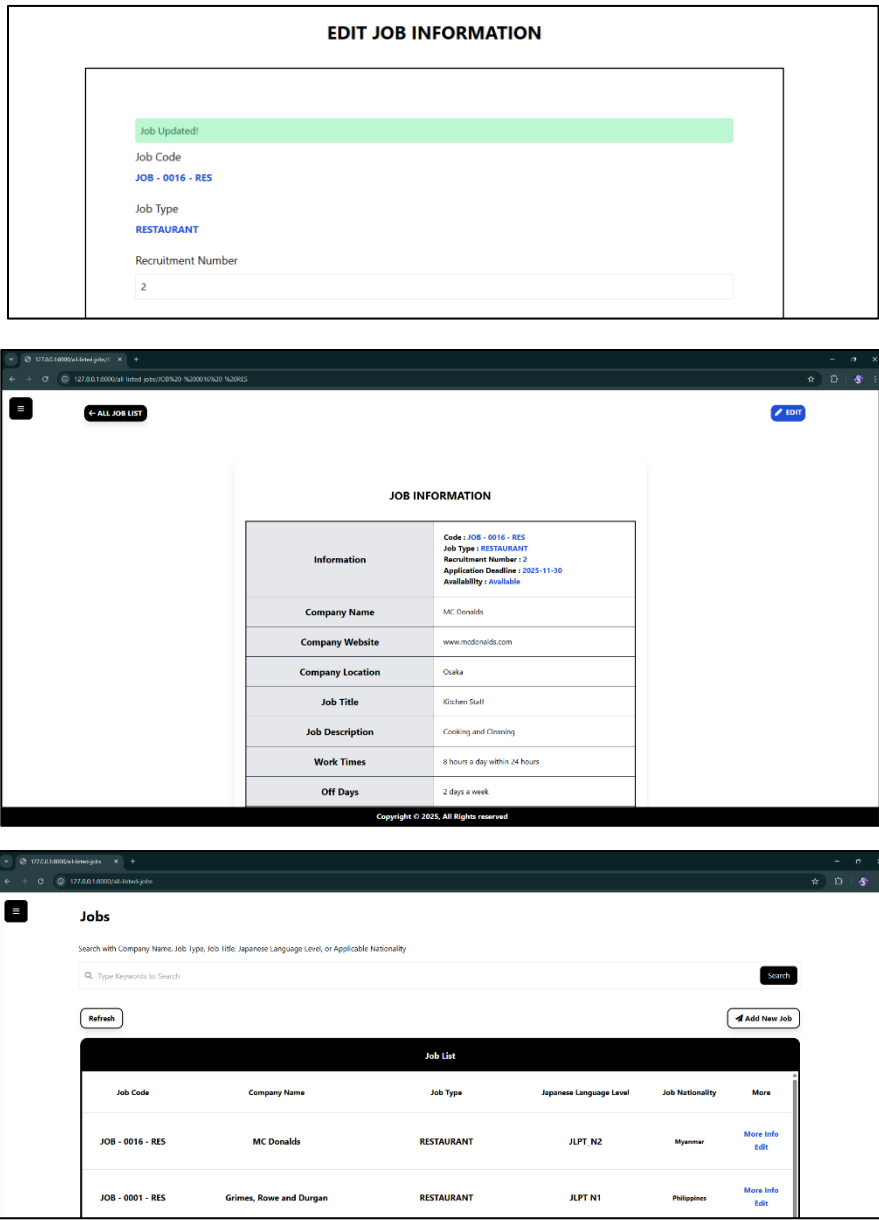


Figure: 5.8.13. Job Edition of Admin Side User 2

Figure: 5.8.12. shows the edit process where the user clicks the edit link to go to the Edit Page of the selected Job. After changing desired information and clicking the Edit button, the Job information is updated, and the success message will appear at the top of the edit page as shown in Figure: 5.8.13. After updating, the user can go from the View Job button, through the Job view page, to the Job List Page, where the updated information is displayed.

Add Partner

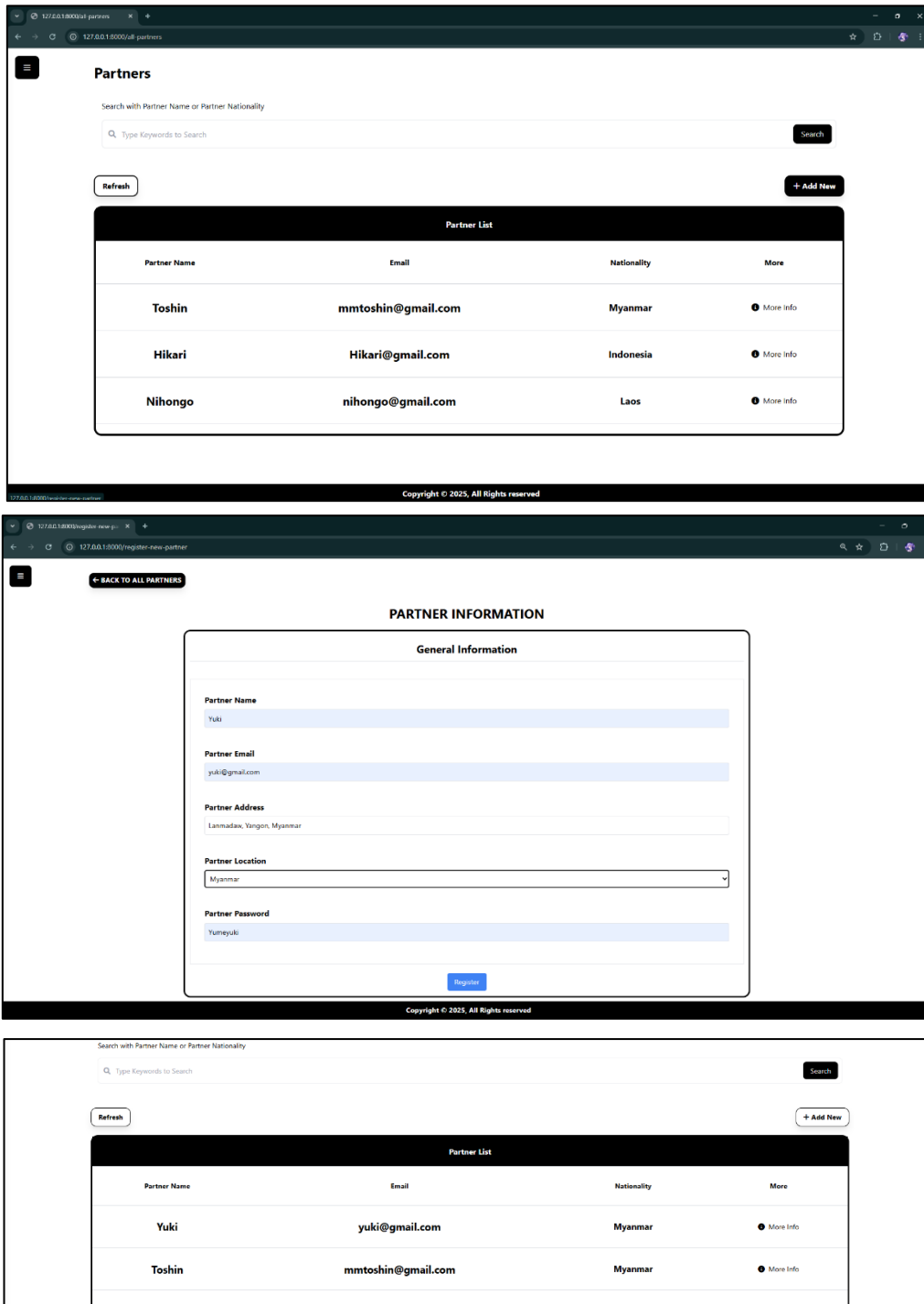


Figure: 5.8.14. Partner Creation of Admin Side User 1

Figure: 5.8.14. presents how the partner account is created by the Admin user. When the Add New button is clicked, the user is directed to the Partner account creation page, where the user fills in the necessary information. When the register button is clicked, the data is stored, and the user is redirected to the Partners Page, where the added partner is displayed.

Edit Partner

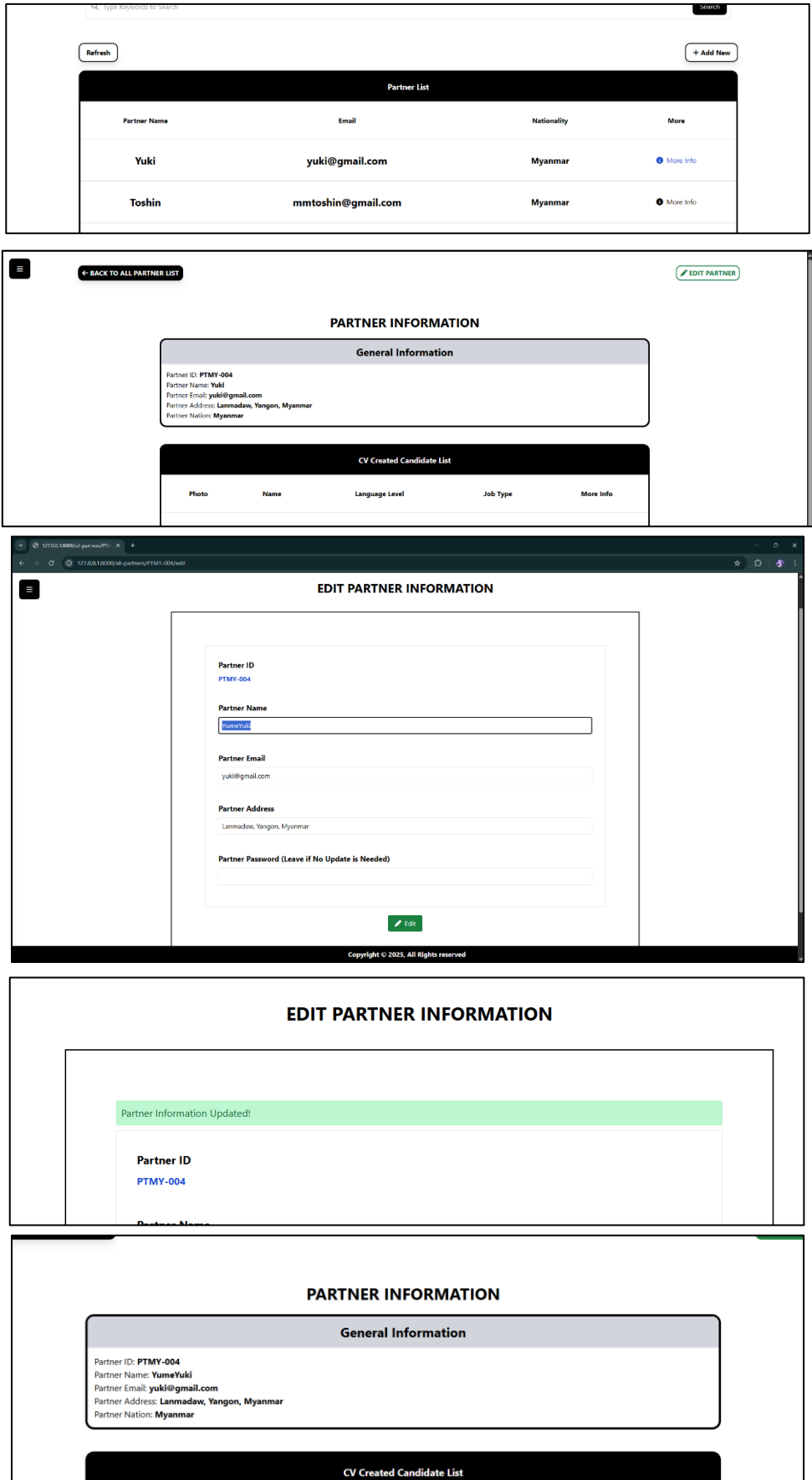


Figure: 5.8.15. Partner Edition of Admin Side User

Figure: 5.8.15 shows the partner account edition process of the admin side user. First, the user has to click the More Info link to go View Partner page, where general information and submitted CVs of the selected partner can be seen. When the edit button is clicked, the user is redirected to the edit page, where the user edits the desired information and saves the edit. The edit is successful when the green message appears as shown in the fourth picture. When the user goes back to the View Partner page, the information there will be updated.

Applicant Evaluation and Checking Passed CV

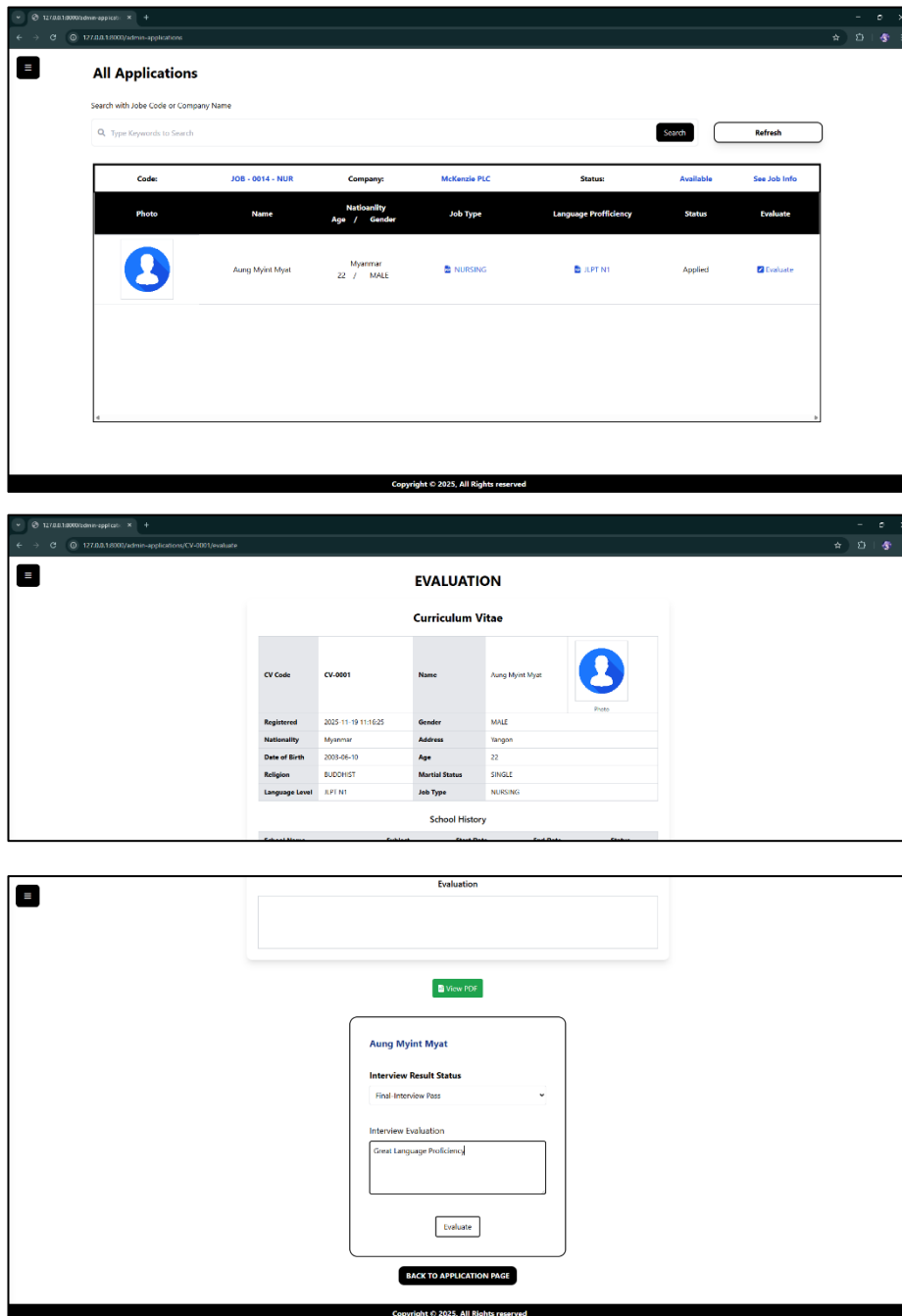


Figure: 5.8.16. Applicant Evaluation Process of Admin Side User

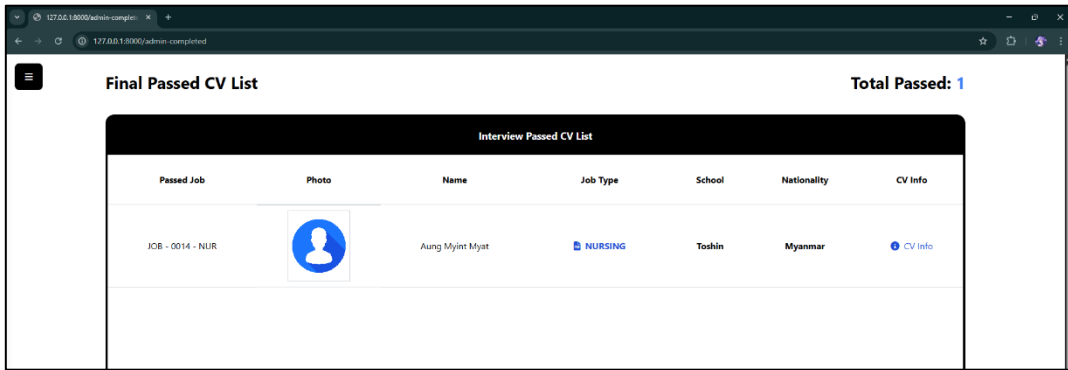
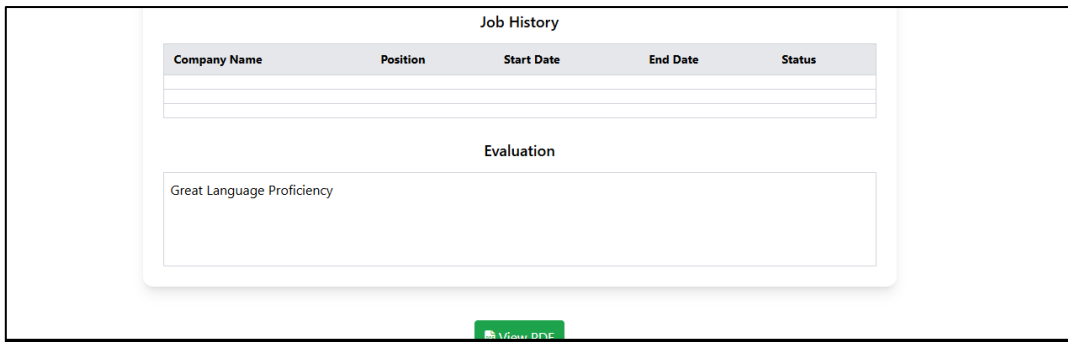


Figure: 5.8.17. Updated Information After Evaluation by Admin User

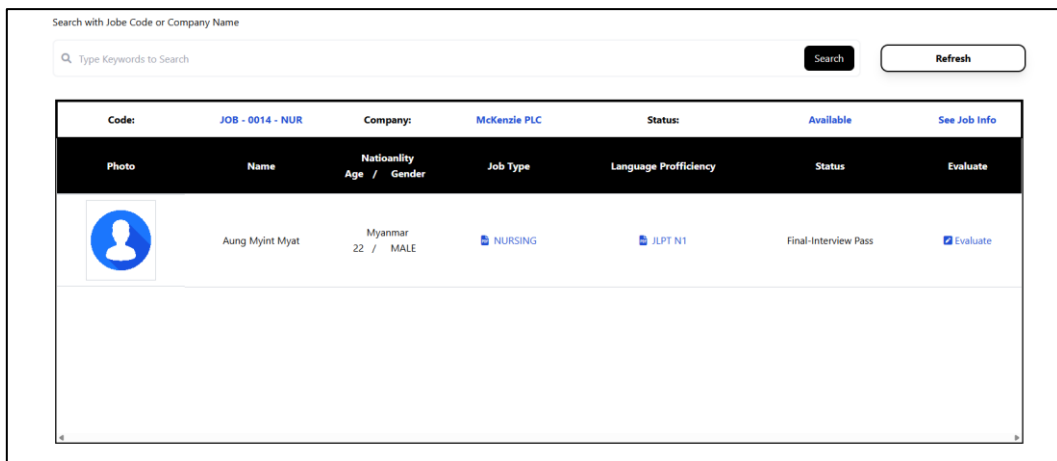


Figure: 5.8.18. Passed CV Page with Evaluated Applicant with Final Interview Pass Status

As shown in the Figure: 5.8.16, the admin user can see the applications made by the partners. When the evaluate link is clicked, the link redirects the admin to the Evaluation page, where the admin can evaluate the selected candidate’s CV just like in the third screenshot of Figure: 5.8.16. Once evaluated by filling in information and clicking the Evaluate button, the evaluation data on the CV is updated on the CV template and the application record, just like in Figure: 5.8.17.

When the candidate’s CV status is changed to Final Interview Pass, that CV will be displayed in the All Passed CV list as shown in Figure: 5.8.18.

Filteration Functions

The screenshot displays a web application interface for managing jobs. At the top, there is a search bar with the text "Restaurant" entered. Below the search bar are buttons for "Refresh" and "Add New Job". The main content area is a table titled "Job List" with the following columns: Job Code, Company Name, Job Type, Japanese Language Level, Job Nationality, and More. The table contains five rows of data, all of which are restaurant-related jobs. The footer of the application reads "Copyright © 2025, All Rights reserved".

Job Code	Company Name	Job Type	Japanese Language Level	Job Nationality	More
JOB - 0016 - RES	MC Donalds	RESTAURANT	JLPT_N2	Myanmar	More Info Edit
JOB - 0001 - RES	Grimes, Rowe and Durgan	RESTAURANT	JLPT N1	Philippines	More Info Edit
JOB - 0002 - CON	Stoltenberg, Marks and Rutherford	CONSTRUCTION	JLPT N3	Cambodia	More Info Edit
JOB - 0003 - CON	Douglas, Ratke and Rice	CONSTRUCTION	JLPT N1	Cambodia	More Info Edit

Job Code	Company Name	Job Type	Japanese Language Level	Job Nationality	More
JOB - 0016 - RES	MC Donalds	RESTAURANT	JLPT_N2	Myanmar	More Info Edit
JOB - 0001 - RES	Grimes, Rowe and Durgan	RESTAURANT	JLPT N1	Philippines	More Info Edit
JOB - 0010 - RES	Klein Inc	RESTAURANT	JLPT N2	Indonesia	More Info Edit
JOB - 0015 - RES	Glover, Schroeder and Bartoletti	RESTAURANT	JLPT N2	Vietnam	More Info Edit

Figure: 5.8.19. Filtering Function using Keyword-based Search Feature

For the filtering function, the keyword-based job search feature is presented as an example, as shown in Figure: 5.8.19. By typing keywords in the search bar and clicking the search button, the system will filter the data only the user wants.

5.9. Project Management

Work Breakdown Structure



Figure: 5.9.1. Work Breakdown Structure of Oshigoto Japan System Project

Gantt Chart

As the timeline until the final interim report remains the same as shown in the interim report, only the schedule from the starting point of learning backend language through finalizing the final report will be presented.

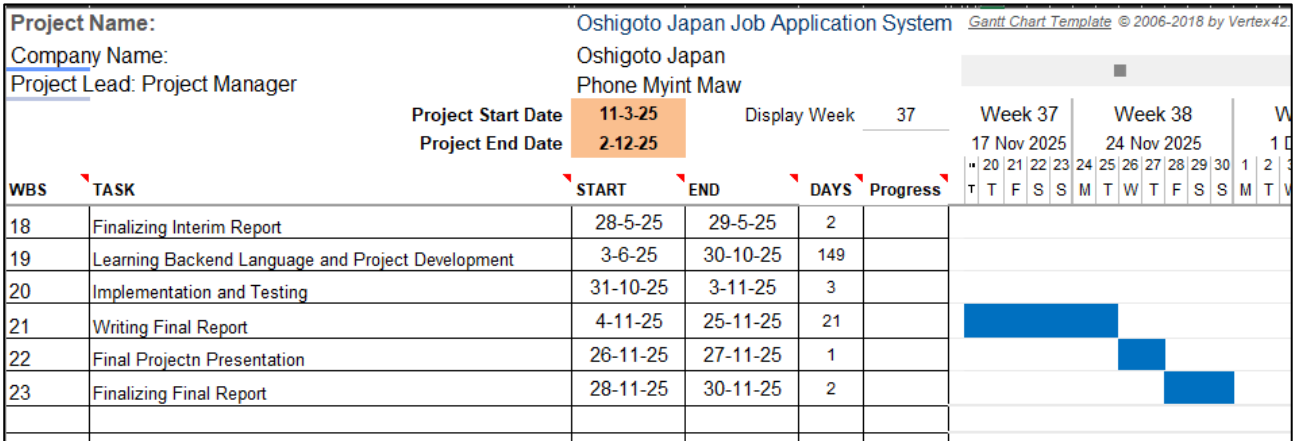
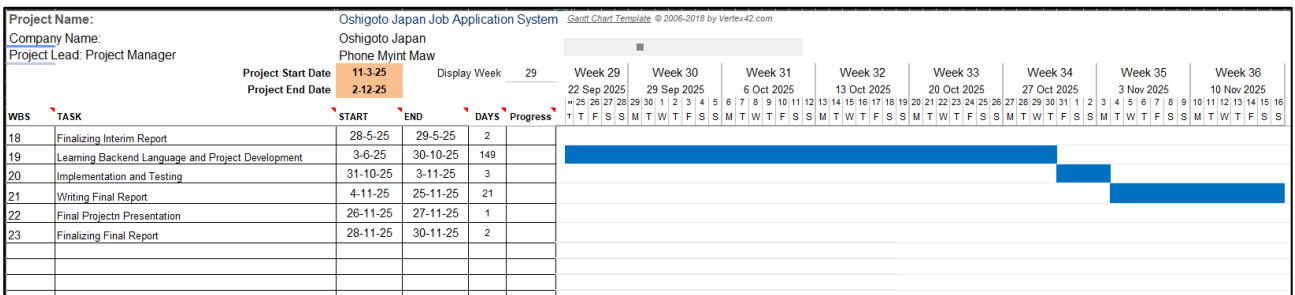
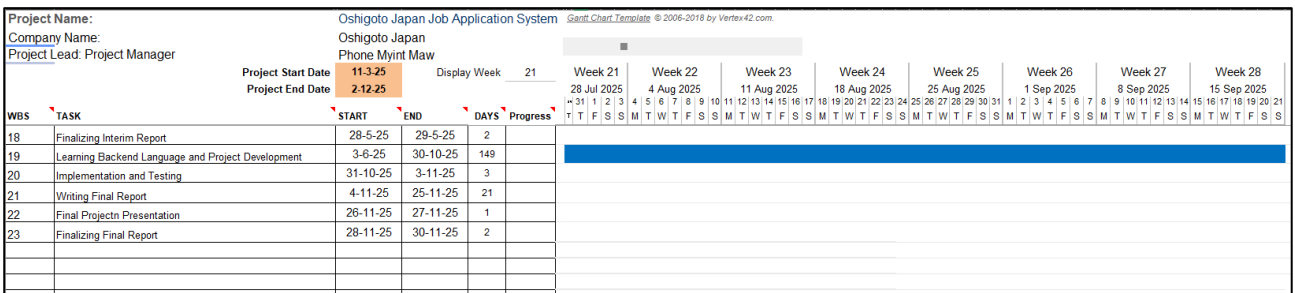
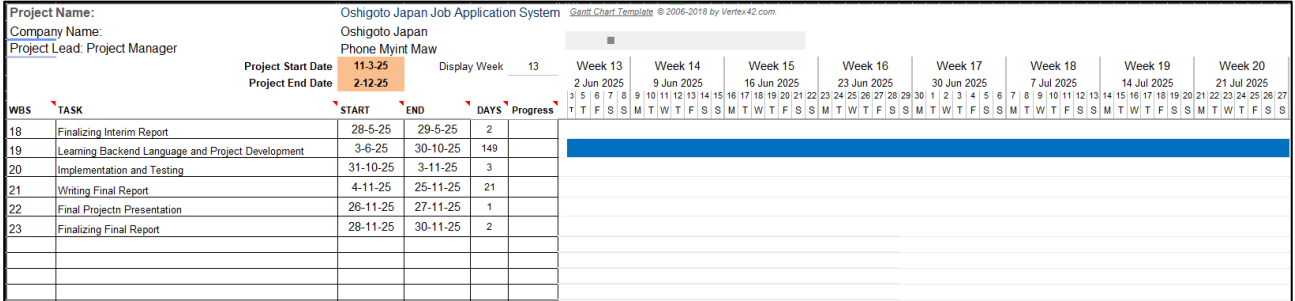


Figure: 5.9.2. Gantt Chart of Oshigoto Japan System Project

5.10. Training Schedule

Admin Side Schedule

No	Target User	Target Function	Estimated Training Length
1	Admin	Admin Log In	10 mins
2	Admin	Overall Explanation	15 mins
3	Admin	Dashboard Explanation	10 mins
4	Admin	Date Filteration	10 mins
5	Admin	Job Creation	10 mins
6	Admin	Job Information Edition	10 mins
7	Admin	Job View Info and PDF File Viewing	5 mins
8	Admin	Partner Account Creation	5 mins
9	Admin	Partner Account Edition	5 mins
10	Admin	Partner Account View and Related CV Check	10 mins
11	Admin	CV Information and File Views with PDF Views	15 mins
12	Admin	Application Record Explanation	10 mins
13	Admin	Applicant Evaluation	10 mins
14	Admin	All Registered CV File and Information View with PDF Views	10 mins
15	Admin	Final Passed CV List View	5 mins
16	Admin	Filtering Function	10 mins
Total			150 mins

Table: 5.10.1. Admin Side System Usage Training Schedule

Partner Side Schedule

No	Target User	Target Function	Estimated Training Length
1	Partner	Partner Log In	10 mins
2	Partner	Overall Explanation	15 mins
3	Partner	CV Creation	15 mins
4	Partner	CV Edition	15 mins
5	Partner	CV information and file View	10 mins
6	Partner	CV PDF File View	10 mins
7	Partner	Job Selection, Job View Information, and PDF View	10 mins
8	Partner	Job Application	15 mins
9	Partner	Applied Job View	10 mins
10	Partner	Filtering Function	10 mins
Total			120 mins

Table: 5.10.2. Partner Side System Usage Training Schedule

5.11. Conclusion

This chapter presented the full implementation process of the Oshigoto Japan System, covering database structure, coding, and third-party tool integration. The chapter also outlined key functionalities such as authentication, CRUD operations, CV and Job application workflow, file handling, PDF rendering, filtering, and dashboard visualisation. In addition, the technical issues encountered during the implementation and the solutions for addressing such issues were also covered. Finally, the user manual and the training schedule are also presented to support effective system usage.

Chapter 6 – Testing and Results

6.1. Unit Testing

Log In Error Test

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.1	User Log In – The user must use the correct email and password to log in.	In Figure: 6.1.1. Email: admin@oshigoto.com Password: pmm552002	System displays “Invalid Credentials” Message	System displays “Invalid Credentials” Message in Figure: 6.1.2.

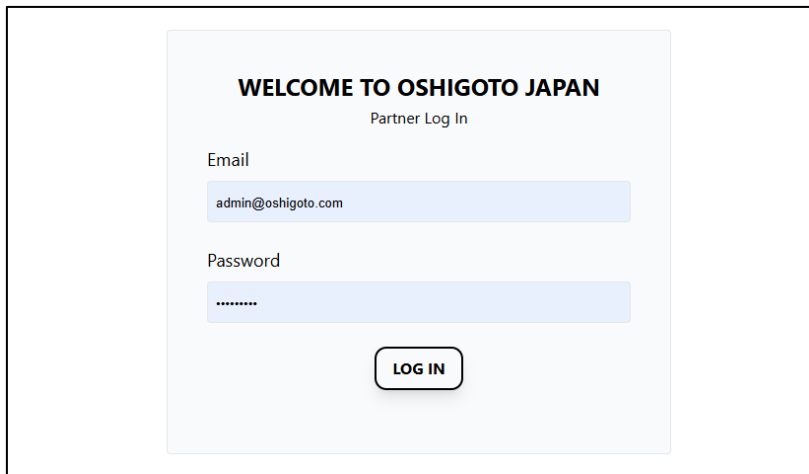


Figure: 6.1.1. Entering Admin Email and Password on Partner Side

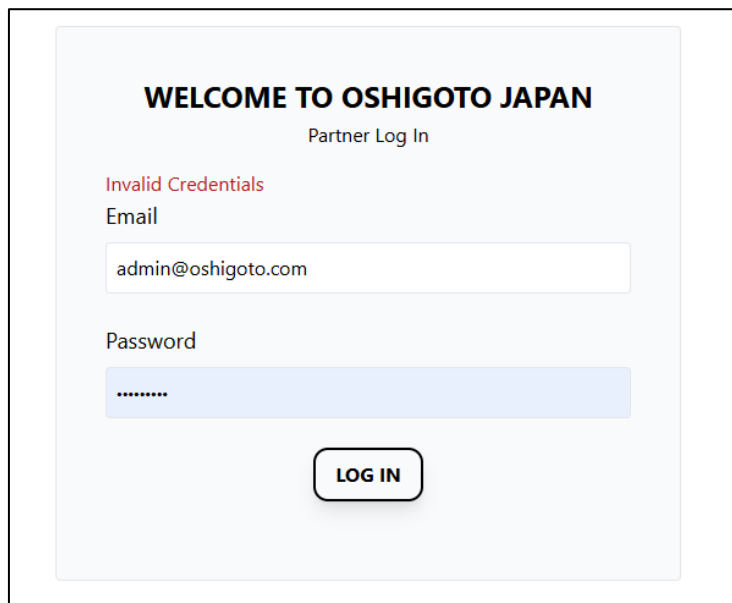


Figure: 6.1.2. Error Message Due to Admin account data not existing in Partner Accounts

Log In Test

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.2	User Log In – The user must use the correct email and password to log in.	In Figure: 6.1.3. Email: admin@oshigoto.com Password: pmm552002	Logs in to the Admin Page, and the system shows “You have logged in” Message	Logs in to Admin Page, and the system shows “You have logged in” Message in Figure: 6.1.4.

Figure: 6.1.3. Entering Admin Email and Password on Admin Side

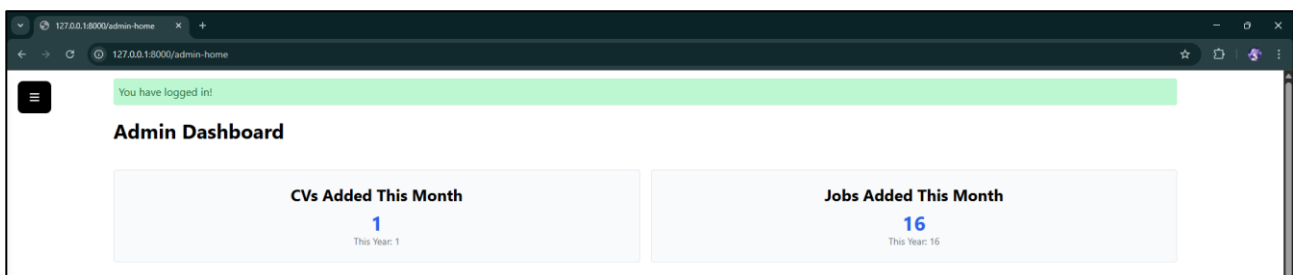


Figure: 6.1.4. Log In Success Message after Successful Log In

Partner Create CV Error

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.3	CV Creation Error – The user must fill all required fields for CV Creation.	In Figure: 6.1.5. All Inputs of Required Fields: Null	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs in Figure: 6.1.6.

REGISTER NEW CV
Register for Job Application

GENERAL INFORMATION

Candidate Photo (JPG/PNG)
Choose File | No file chosen

Candidate Name

Gender
-- Select One --

Candidate Age

Candidate Date of Birth
yyyy-mm-dd

Figure: 6.1.5. No Inputs were filled in CV Creation

Candidate Photo (JPG/PNG)
Choose File | No file chosen

Candidate Name

Gender
-- Select One --

Candidate Age

Candidate Date of Birth

Please select an item in the list.

Candidate Name

The cv name field is required.

Gender
MALE

Candidate Age

The cv age field is required.

Candidate Date of Birth

Figure: 6.1.6. Error for No Inputs on Required Fields in CV Creation

Partner Create CV

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.4	To Test Successful CV Creation	In Figure: 6.1.7. All Required Inputs are Filled and Submitted	System displays "CV Created!" Message	System displays "CV Created!" Message in Figure: 6.1.8.

REGISTER NEW CV
Register for Job Application

GENERAL INFORMATION

Candidate Photo (JPG/PNG)
Choose File | Profile Picture Example 2.jpg

Candidate Name
Zaw Gyi

Gender
MALE

Candidate Age
20

Candidate Date of Birth
2000-05-10

Figure: 6.1.7. Filled Inputs on All Required Fields on CV Creation Form

REGISTERED CV LIST | APPLIED JOB LIST | APPLY NEW JOBS | + MAKE NEW CV FORMS

CV Created!

Registered Candidate 2 | Passed Candidate 1

Search Names Here
Type Keywords to Search | Search | Refresh

Photo	Name	Nationality Age / Gender	Job Type	Language Proficiency	Religion	Status	Edit / View
	Zaw Gyi	Myanmar 20 / MALE	RESTAURANT	JLPT N4	CHRISTIAN	Not Applied	Edit View

Figure: 6.1.8. "CV Created" Message After Successful CV Creation

Partner Edit CV

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.5	To Test Successful Editing of CV and Errors	In Figure: 6.1.9. The Desired Field is Edited and Submitted	System Displays "CV Updated!" Message	System Displays "CV Updated!" Message in Figure: 6.1.10. Error on leaving blank on input field in Figure: 6.1.10.

A screenshot of a web form for editing a CV. The form includes a dropdown menu for 'CHRISTIAN' (likely Religion), a 'Marital Status' dropdown menu with options 'SINGLE', 'MARRIED' (highlighted in blue), and 'MARRIED WITH CHILD', and a 'Job Type Certificate (PDF)' field with a 'Choose File' button and 'No file chosen' text.

A screenshot of a web page titled 'EDIT NAME'S CV' under the heading 'GENERAL INFORMATION'. It features a green notification bar at the top that says 'CV Updated!'. Below this, there is a 'Candidate Photo (JPG)' field with a 'Choose File' button and 'No file chosen' text, and a link to 'View Previously Uploaded Profile'. The 'Candidate Name' field contains the text 'Zaw Gyi'.

Figure: 6.1.9. Field Edited and Submitted

A screenshot of a web form showing an 'Address' field. The field contains the text 'Insein, Yangon'. Below the field, a red error message reads: 'The cv address field is required.'

Figure: 6.1.10. Error Displays when Edited Field is left Blank

Admin Job Create Error

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.6	Job Creation Error – The user must fill all required fields for Job Creation.	In Figure: 6.1.11. All Inputs of Required Fields: Null	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs in Figure: 6.1.12.

The screenshot shows a web interface for 'JOB INFORMATION'. At the top left, there is a hamburger menu icon and a button labeled '← BACK TO ALL JOBS'. The form contains the following fields: 'Job Type' (a dropdown menu with the placeholder text '-- Select Job Type --'), 'Recruitment Number' (an empty text input field), 'Application Deadline' (a date picker with the placeholder 'yyyy-mm-dd'), and 'Company Name' (an empty text input field).

Figure: 6.1.11. No Inputs were filled in Job Creation

This screenshot is a close-up of the 'JOB INFORMATION' form. The 'Job Type' dropdown menu is highlighted with a red border and contains the message 'Please select an item in the list.' next to an exclamation mark icon. The 'Recruitment Number' field is also visible below it.

This screenshot shows the 'JOB INFORMATION' form with error messages for three fields. The 'Job Type' dropdown is set to 'RESTAURANT'. The 'Recruitment Number' field has a red error message: 'The recruit number field is required.' The 'Application Deadline' date picker has a red error message: 'The application deadline field is required.' The 'Company Name' field has a red error message: 'The company name field is required.'

Figure: 6.1.12. Errors for No Inputs on Required Fields in Job Creation

Admin Job Create

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.7	To Test Successful Job Creation	In Figure: 6.1.13. All Required Inputs are Filled and Submitted	System displays "Job Created!" Message	System displays "Job Created!" Message in Figure: 6.1.14.

JOB INFORMATION

Job Type

Recruitment Number

Application Deadline

Company Name

Company Website

Company Location

Figure: 6.1.13. Filled Inputs on All Required Fields on Job Creation

Job Created!

Job List

Job Code	Company Name	Job Type	Japanese Language Level	Job Nationality	More
JOB - 0017 - RES	ABC	RESTAURANT	JLPT N3	Myanmar	More Info Edit

Figure: 6.1.14. "Job Created" Message After Successful Job Creation

Admin Job Edit

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.8	To Test Successful Editing of CV and Errors	In Figure: 6.1.15. The Desired Field is Edited and Submitted	System Displays "Job Updated!" Message	System Displays "Job Updated!" Message in Figure: 6.1.15. Error on leaving blank in the input field in Figure 6.1.16.

2025-11-30

Company Name
ABCD

Company Website
abc.com

EDIT JOB INFORMATION

Job Updated!

Job Code
JOB - 0017 - RES

Job Type
RESTAURANT

Recruitment Number

Figure: 6.1.15. Field Edited and Submitted

Company Name

ABCD

The company name field is required.

Figure: 6.1.16. Error Displays when Edited Field is left Blank

Admin Partner Create Error

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.9	Partner Creation Error – The user must fill all required fields for Partner Creation.	In Figure: 6.17. All Inputs of Required Fields: Null	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs	System displays “Please Select as Item in the List” for Option Selection Input and field error for normal inputs in Figure: 6.1.18.

The screenshot shows a form titled "PARTNER INFORMATION" with a sub-section "General Information". It contains five input fields: "Partner Name", "Partner Email", "Partner Address", "Partner Location" (a dropdown menu showing "-- Select Location --"), and "Partner Password". A blue "register" button is at the bottom. A copyright notice "Copyright © 2025, All Rights reserved" is at the very bottom.

Figure: 6.1.17. No Inputs Filled in Partner Creation

This figure shows two parts of the error state. The top part is a close-up of the "Partner Location" dropdown menu, which displays "-- Select Location --" and a tooltip that says "Please select an item in the list." The bottom part shows the "Partner Name" and "Partner Email" text input fields. Below the "Partner Name" field, a red error message reads "The partner name field is required." Below the "Partner Email" field, a red error message reads "The email field is required."

Figure: 6.1.18. Errors for No Inputs on Required Fields in Partner Creation

Admin Partner Create

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.10	To Test Successful Partner Creation	In Figure: 6.1.19. All Required Inputs are Filled and Submitted	System displays "Partner Created!" Message	System displays "Partner Created!" Message in Figure: 6.1.20.

The screenshot shows a form titled "PARTNER INFORMATION" with a sub-section "General Information". The fields are filled with the following data:

- Partner Name: Jishin
- Partner Email: jishin@gmail.com
- Partner Address: Insein, Yangon
- Partner Location: Indonesia (selected from a dropdown menu)
- Partner Password: jishin2025

A "Register" button is visible at the bottom of the form. The footer of the page reads "Copyright © 2025, All Rights reserved".

Figure: 6.1.19. Filled Inputs on All Required Fields on Partner Creation

The screenshot shows the "Partners" page. At the top, there is a search bar with the text "Search with Partner Name or Partner Nationality" and a "Search" button. Below the search bar are "Refresh" and "+ Add New" buttons. A green banner displays the message "Partner Created!". Below this is a table titled "Partner List" with the following data:

Partner Name	Email	Nationality	More
Jishin	jishin@gmail.com	Indonesia	More Info

Figure: 6.1.20. "Partner Created" Message After Successful Partner Creation

Admin Partner Edit

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.11	To Test Successful Editing Partner and Errors	In Figure: 6.1.21. The Desired Field is Edited and Submitted	System Displays "Partner Information Updated!" Message	System Displays "Partner Information Updated!" Message in Figure: 6.1.15. Error on leaving blank on the input field in Figure: 6.1.22.

The screenshot shows a form with three sections:

- Partner ID:** PTIN-005
- Partner Name:** Jishin Japan
- Partner Email:** jishin@gmail.com

 The form is displayed in a light gray border, and the text is in a clean, sans-serif font.

The screenshot shows the same form as Figure 6.1.21, but with a green success message at the top: "Partner Information Updated!". The form fields are still visible below the message.

Figure: 6.1.21. Field Edited and Submitted

The screenshot shows the form with the "Partner Name" field. The text "Jishin Japan" is present in the input field. Below the field, a red error message reads: "The partner name field is required."

Figure: 6.1.22. Error Displays when Edited Field is left Blank

Admin Dashboard Date Filter

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.12	To Test Date Range Function Works Properly	In Figure: 6.1.23. Start Date: 2025-11-01 End Date: 2025-11-21	System Displaying Statistical Data According to Date Range Entered	System Displaying Statistical Data According to Date Range Entered in Figure: 6.1.24.

Filter by Date

Start Date : 2025-11-01

End Date : 2025-11-21

Number of Total CVs Made from to Number of Total Applications Made from to

Number of Total Partners Made from to Number of Total Jobs Made from to

Figure: 6.1.23. Date Range Input in Filter by Date Function

Filter by Date

Start Date : yyyy-mm-dd

End Date : yyyy-mm-dd

Number of Total CVs Made from 2025-11-01 to 2025-11-21 Number of Total Applications Made from 2025-11-01 to 2025-11-21

2 1

Number of Total Partners Made from 2025-11-01 to 2025-11-21 Number of Total Jobs Made from 2025-11-01 to 2025-11-21

5 17

Figure: 6.1.24. Filtered Statistical Data Between the Entered Date Range

Filteration Feature

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
UT.13	To Test the Filtration Function	In Figure: 6.1.25. Search Input: Restaurant	Jobs with Restaurant Job Type Filtered in Job List Table	Jobs with Restaurant Job Type Filtered in Job List Table in Figure: 6.1.26.

Search with Company Name, Job Type, Job Title, Japanese Language Level, or Applicable Nationality

Search

Figure: 6.1.25. Keyword Input in Search box for Filtration

Job List					
Job Code	Company Name	Job Type	Japanese Language Level	Job Nationality	More
JOB - 0017 - RES	ABCD	RESTAURANT	JLPT N3	Myanmar	More Info Edit
JOB - 0016 - RES	MC Donalds	RESTAURANT	JLPT_N2	Myanmar	More Info Edit
JOB - 0001 - RES	Grimes, Rowe and Durgan	RESTAURANT	JLPT N1	Philippines	More Info Edit
JOB - 0010 - RES	Klein Inc	RESTAURANT	JLPT N2	Indonesia	More Info Edit
JOB - 0015 - RES	Glover, Schroeder and Bartoletti	RESTAURANT	JLPT N2	Vietnam	More Info Edit

Figure: 6.1.26. Filtered Data According to Keyword Input

6.2. Integrated Testing

Logging Out and Going Back

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
INT.1	The User Logs Out	In Figure: 6.2.1. Log Out Button	The System Ends Session and Logs Out the User	The System Ends Session and Logs Out User in Figure: 6.2.2.
	Clicking the Back Button after Logging Out	Browser Back Button and Refresh Page	No Content in Page after Refresh	No Content in Page after Refresh in Figure: 6.2.3.

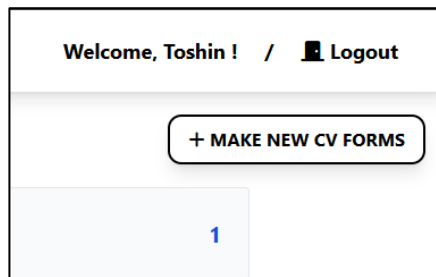


Figure: 6.2.1. Log Out Button on Partner Side Page

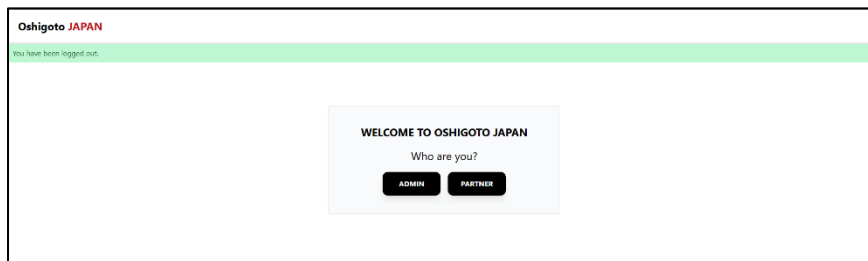


Figure: 6.2.2. System Log Out Message after clicking Log Out Button

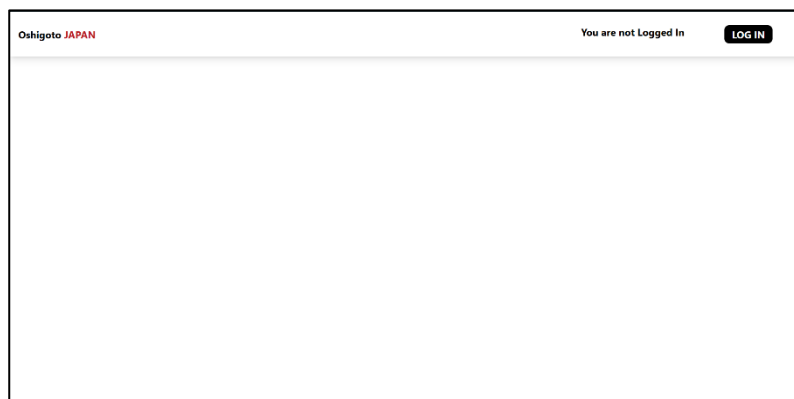


Figure: 6.2.3. Hidden Contents when not Logged In

Partner Job Application and View Applied Job

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
INT.2	The User Selects a Job to go Application Page	In Figure: 6.2.4. See More Button on Apply New Jobs Page Apply Button on See More Info Page	The User is Guided to the Application Page	The User is Guided to the Application Page in Figure: 6.2.5.
	The User Adds Applicants to Application Record	In Figure: 6.2.5. Add Button	The Selected Applicant is Put into the Applied List	The Selected Applicant is added to the applied list in Figure 6.2.6.
	The User Redirected to the Applied Job Page	In Figure: 6.2.6. Done Button	The User is redirected to the Applied Job List Page	The User is redirected to the Applied Job List Page in Figure: 6.2.7

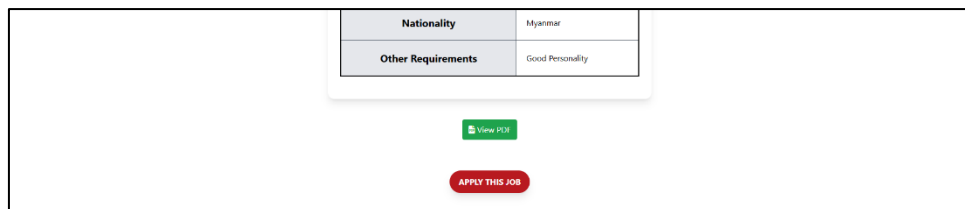
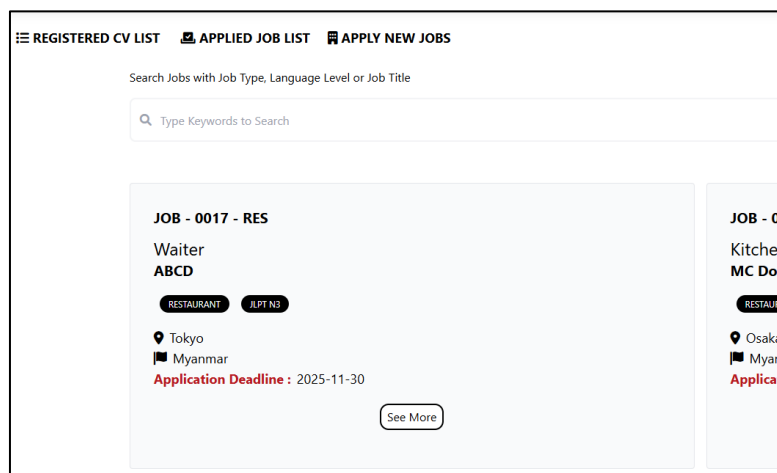


Figure: 6.2.4. Buttons for Guiding User to Application Page



Figure: 6.2.5. Job Application Page

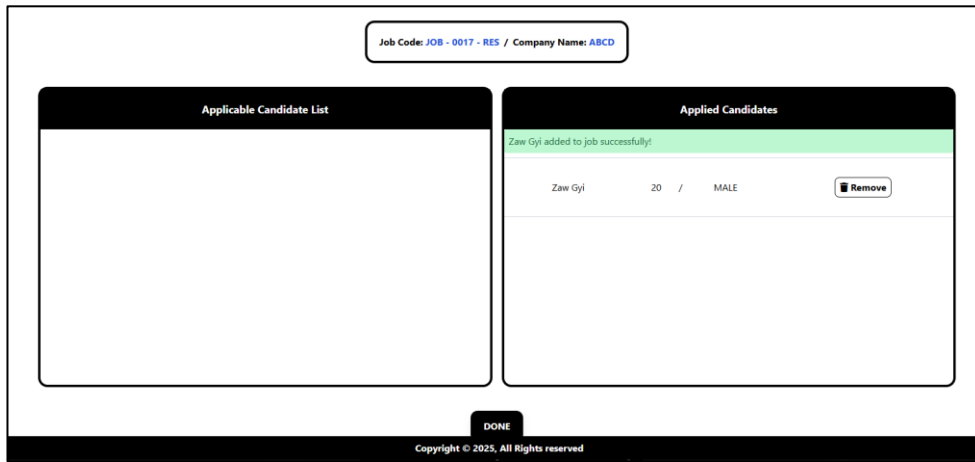


Figure: 6.2.6. Applicant Put into Applied Candidate List by Clicking Add Button

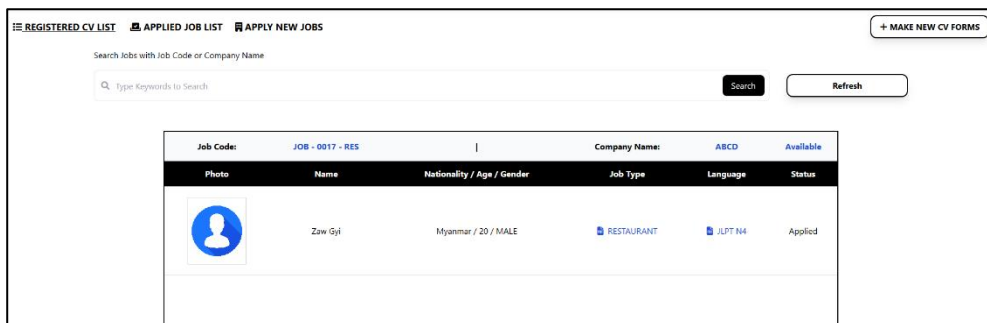


Figure: 6.2.7. The User Redirected to Applied Job List Page

Admin Applicant Evaluation

Scenario	Purpose	Inputs	Anticipated Outcome	Actual Outcome
INT.3	The User Selects the Applied Applicant in the All Applications Page	In Figure: 6.2.8. Evaluate link	The User is Redirected to the Evaluation Page	The User is Redirected to the Evaluation Page in Figure: 6.2.9.
	The User Evaluates the Applicant	In Figure: 6.2.10 Status: Final-Interview Pass Evaluation: Not Bad Evaluate Button	The System Displays "CV Evaluated!" After clicking Evaluation	The System Displays "CV Evaluated!" After clicking Evaluation in Figure: 6.2.11.
	The User goes back to the Applications Page	In Figure: 6.2.11. Back to Applications Page Button	The user is redirected to the All Applications Page	The user is redirected to the All Applications Page in Figure: 6.2.12.




Code:	JOB - 0017 - RES	Company:	ABCD	Status:	Available	See Job Info
Photo	Name	Nationality Age / Gender	Job Type	Language Proficiency	Status	Evaluate
	Zaw Gyi	Myanmar 20 / MALE	 RESTAURANT	 JLPT N4	Applied	<input checked="" type="checkbox"/> Evaluate

Figure: 6.2.8. Applicant's Evaluation Link in Application Record


EVALUATION					
Curriculum Vitae					
CV Code	CV-0002	Name	Zaw Gyi		Photo
Registered	2025-11-20 09:43:32	Gender	MALE		
Nationality	Myanmar	Address	Insein, Yangon		
Date of Birth	2000-05-10	Age	20		
Religion	CHRISTIAN	Marital Status	MARRIED		
Language Level	JLPT N4	Job Type	RESTAURANT		
School History					
School Name	Subject	Start Date	End Date	Status	
University of Yangon	Physics	2022-12-01	2025-11-20	Studying	

Figure: 6.2.9. The User Redirected to Selected Applicant's Evaluation Page

Figure: 6.2.10. Evaluation Inputs for Applicant Evaluation

Figure: 6.2.11. “CV Evaluated” Message after Successful Evaluation

All Applications

Search with Job Code or Company Name

🔍 Type Keywords to Search Search Refresh

Code:	JOB - 0017 - RES	Company:	ABCD	Status:	Available	See Job Info
Photo	Name	Nationality Age / Gender	Job Type	Language Proficiency	Status	Evaluate
	Zaw Gyi	Myanmar 20 / MALE	RESTAURANT	JLPT N4	Final-Interview Pass	Evaluate

Figure: 6.2.12. The User Redirected to All Applications Page

Chapter 7 – Conclusions and Future Improvements

7.1. Conclusion

The project successfully developed a functional web-based job application system for Oshigoto Japan, addressing the organization's challenges and needs. Throughout the development process, the system was designed, implemented, and refined iteratively, using a combination of Laravel Framework and Tailwind CSS.

The completed system provides functionalities for the company's Language-School-Partner-to-Admin workflow, including CV Submission, Job Application, Job Creation, Partner Account Creation, and Applicant Evaluation. In addition to such functions, keyword-based filtering, Calendar integration, statistical reporting, and chart visualization were integrated into the system to enhance user experience. The system's security is ensured with authentication and role-based authorization.

7.2. Future improvements

Although the current system is tailored to Oshigoto Japan's organizational scope and requirements, several enhancements could be made in the future to strengthen its functionality as the organization grows. Such enhancements are as follows.

Automated Email Notifications

Automating email alerts to notify both partners and admin about job applications, newly posted jobs, interview schedules, and applicant evaluations and results.

Fully Integrated Scheduling

The existing embedded Google Calendar only provides a basic view of the schedules. By integrating a fully functional scheduling API with Google Calendar, it would allow the admin to create, edit, and manage interview events from the system.

Partner-Based Application Record

As the Application Page from the Admin side shows all the applied applicants from several partners, dividing into job-and partner-based application records will help differentiate the application records.

Multi-language support

The existing system is entirely in English, and the users can be from different countries, who might not be proficient in English. Implementing a multi-language interface would make the users more confident in using the system in their own language.

Cloud Storage

As the file storage is tied to Laravel's storage path, it would be better for the system to be integrated with the company's own cloud storage.

Admin Account Creation

As the seeding function creates the admin account and does not have an admin account creation feature due to the organization's scope, as the company grows and the person handling the admin account increases, it would be best to add an admin account management module for the following features.

- Creating new admin accounts
- Assigning Roles or Permission Levels
- Editing or Disabling Existing Admin Users
- Resetting Passwords securely

References

1. Gordon, C. (2025), What is the Recruitment Management System (RMS)?, *RECRUITERS LINEUP*. Available at: <https://www.recruiterslineup.com/what-is-the-recruitment-management-system/> (Accessed: 4th November 2025)
2. Noborder (n.d), Noborder Main Page. Available at: <https://noborder.cloud/> (Accessed: 4th November 2025)
3. Manatal (n.d), Manatal Main Page. Available at: <https://www.manatal.com/staffing-agencies> (Accessed: 5th November 2025)
4. GeeksforGeeks (2025), What is Agile Methodology?. Available at: <https://www.geeksforgeeks.org/software-testing/what-is-agile-methodology/> (Accessed: 5th November 2025)
5. Visual Paradigm (n.d), What is Use Case Diagram?. Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> (Accessed: 6th November 2025)
6. Visual Paradigm (n.d), What is Sequence Diagram?. Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> (Accessed: 6th November 2025)
7. GeeksForGeeks (2025), Introduction to Flowcharts. Available at: <https://www.geeksforgeeks.org/competitive-programming/an-introduction-to-flowcharts/> (Accessed: 7th November 2025)

Appendix

1. Development Flow using Agile Methodology

Planning

The planning phase involves identifying the company's background, the project's aims and objectives, defining project scopes, and outlining functional and non-functional requirements, challenges, and expected deliverables. The initial planning was carried out in the proposal stage, where the main idea of the system, its purpose, and requirements were documented.

Requirement Analysis

During the requirement analysis, the initial requirements were expanded and refined in the form of an interim report. In this stage, the detailed functional and non-functional requirements, detailed workflow descriptions, and user role definitions were developed. As the project follows an iterative Agile approach, these requirements continued to evolve, and changes were made throughout the project.

Designing

In this stage, after doing the requirement analysis, the system's structural and visual components were planned and designed. These included creating a database schema, use case diagrams, sequence diagrams, and user interface layouts. These designs provided a clear understanding of how the system works, how different components of the system interact with each other, how the user would navigate the system, and how the system would look after development. These designs were initially presented in the interim report.

Implementation

The implementation phase involves the development of the system based on the designs and requirements created and gathered in the earlier stages. In this stage, following Agile's iterative approach, each page and feature were developed, tested, and refined to ensure that all necessary features were developed and integrated smoothly within the system.

The implementation Oshigoto Japan System stage was carried out through 4 iterative phases. The first phase began with preparing the project environment and learning the Laravel framework first. After familiarizing with Laravel, the initial user interface components and basic feature developments were made, which included role-based navigation bars, CV form, job creation and partner account creation, and management in this phase.

In the second phase, role-based authentication for Partner accounts and Admin accounts was implemented, making login and logout functions usable, and ensuring that only the intended features are accessible to intended user types. Following this, in the third phase, the application function for partners, which allowed them to submit CVs under jobs and view the application status,

and the admin side, which allowed the admin to see the applications made by the partners and evaluate the candidates.

In the final stage, the admin dashboard, in which the admin can view the statistical summaries of each key category, including total jobs, CVs, applications, and partners, displayed in both monthly and yearly metrics, was implemented. The dashboard also has a search function by date range, where the admin can filter the total number of jobs, CVs, partners, and applications based on the created dates. Visualizations such as the bar charts and pie charts were implemented using the Chart.js library to present data clearly. At the end of the dashboard, the Google Calendar component was embedded to help the admin track the schedules of incoming interviews or events.

As an additional enhancement for better user experience, keyword-based search functions and filtering functions were implemented to help users quickly locate specific jobs, CVs, and application records. For the CVs and jobs, a PDF file viewing feature was added for more versatile and accessible document handling. Finally, responsive design was applied throughout the system to ensure that all interfaces adapt smoothly in different screen sizes across different devices, making the system more user-friendly. The code screenshots and other details of implementations are provided in Chapter 5.

Testing

The testing of the core features of the Oshigoto Japan Job Application System was conducted iteratively with the development to verify each component functioned as intended. Both functional and usability testing were performed to ensure that Partner and Admin interfaces were reliable, fully integrated, worked together, and user-friendly. This phase also highlighted some initial design flaws and was refined to improve overall system structure and performance. Testing and the results are discussed in Chapter 6.

Deployment

After the testing phase, the Oshigoto Japan System is set up in a local development environment for testing and demonstration in the deployment phase. The system was hosted in a local Apache server using XAMPP configuration with MySQL serving as the database management system. Laravel's migration and seeding features were used to create and populate the database structure. After the deployment, all the core features were tested to ensure each functioned correctly within the server environment.

As the project is intended for demonstration and evaluation purposes, the deployment on a live production server was not conducted. However, the system will be deployable in a web hosting environment, which supports PHP, Laravel, and MySQL.

2. CATWOE Analysis

Elements	Description	Wants
Customers	Language School Partners from Different Countries	Easy Usage, Improve Quality of Workflow, Simple Design
Actors	Agency Side Staff	Easy Usage, Secure System, Simple Design
Transformation	<p>Partner Side: CV Submission, File Upload, Job Application, Application Record Checking</p> <p>Agency Side: Job Management, Partner Account Management Evaluation, Applicant Record Tracking</p>	Smooth data exchange and file uploads, Easy-to-use System UI.
Worldview	Provide Jobs, Easy to Use/Understand	Clear User Manual Explanation, Clear information of posted jobs
Owners	Owner/Founder	<p>A system that fits the requirements of the school-to-Admin workflow.</p> <p>A System in which the implementation is welcomed by the users</p>
Environment	Local Authorities	Legal and Privacy Law Compliance.

3. Use Case Diagram Descriptions and Explanations

Overall Use Case Diagram

The overall use case diagram illustrates the complete functional scopes of the Oshigoto Japan System by showing how the two main actors, the partner and the admin interact with the system. The Partner side will be responsible for creating and editing CVs, applying CVs to jobs that are available, and tracking application records while the admin side will manage partner accounts, create jobs, edit jobs, and view and evaluate CVs.

Detailed Use Case Diagram

Partner CV Creation

ID	Detailed Use Case Diagram 1	
Figure	3.3.2.	
Name	Partner CV Creation	
Actor	Language School Partner	
Triggers	Update and Open CV List Page	
Preconditions	Required to Log in to Partner Account	
Post Conditions	Updated CV List	
Description	Step	Action
	1	Click the Make New CV Form Button
	2	Fill the Necessary information
	3	Upload Profile Picture and Files (Optional)
	4	Click the Register Button
	5	Redirected to CV List Page
	6	Displays Creation Successful Message and CV list Updated
Alternative	Step	Action
	1	Click the Make New CV Form Button
	2	Fill the Necessary information
	3	Upload Profile Picture and Files (Optional)
	4	Click the Register Button
	5	Error Message due to failure of filling required information

Partner CV Application

ID	Detailed Use Case Diagram 2	
Figure	3.3.3.	
Name	Partner CV Application	
Actor	Language School Partner	
Triggers	Update and Open Applied Jobs Page	
Preconditions	Have Created CV, Matching with Available Job	
Post Conditions	Updated Application List	
Description	Step	Action
	1	Open Apply Job Page
	2	Select Desired Job with the See More line
	3	Go to the application page though Apply Button
	4	Select Candidate from Available Applicants Side
	5	Click the Done button
	6	Redirected to Applied Job page with updated Application Record
Alternative	Step	Action
	1	Open Apply Job Page
	2	Select Desired Job with the See More line
	3	Go to the application page though Apply Button
	4	Remove Candidate from Applied Candidate Side
	5	Click the Done button
	6	Redirected to Applied Job page with updated Application Record

Admin Job Creation

ID	Detailed Use Case Diagram 3	
Figure	3.3.4.	
Name	Admin Job Creation	
Actor	Admin	
Triggers	Update and Open All Jobs Page	
Preconditions	Open All Jobs Page	
Post Conditions	Updated Job List	
Description	Step	Action
	1	Open Jobs Page
	2	Click Add New Job Button
	3	Fill Necessary Job Information
	4	Click the Add Button
	5	Redirected to Jobs Page
	6	Displays Creation Successful Message and Job list Updated
Alternative	Step	Action
	1	Open Jobs Page
	2	Click Add New Job Button
	3	Fill Necessary Job Information
	4	Click the Add Button
	5	Error due to failure to fill all required fields

Admin Partner Creation

ID	Detailed Use Case Diagram 4	
Figure	3.3.5.	
Name	Admin Partner Creation	
Actor	Admin	
Triggers	Update and Open All Partners Page	
Preconditions	Open All Partners Page	
Post Conditions	Updated Partner List	
Description	Step	Action
	1	Open Partners Page
	2	Click Add New Button
	3	Fill Necessary Partner Information
	4	Click the Register Button
	5	Redirected to Partners Page
	6	Displays Creation Successful Message and Partner list Updated
Alternative	Step	Action
	1	Open Partners Page
	2	Click Add New Button
	3	Fill Necessary Partner Information
	4	Click the Register Button
	5	Error due to failure to fill all required fields

Admin Applicant CV Evaluation

ID	Detailed Use Case Diagram 5	
Figure	3.3.6.	
Name	Admin Applicant CV Evaluation	
Actor	Admin	
Triggers	Updates Applicant Status and Evaluation field	
Preconditions	Open Applications Page	
Post Conditions	Updated Applicant Status and Evaluation Fields	
Description	Step	Action
	1	Open Applications Page
	2	Click Evaluate link of Desired Applicant
	3	Change Status and Fill Evaluation comment
	4	Click the Evaluate Button
	5	Displays Evaluation Successful Message, and Applicant Status and Evaluation fields Updated
Alternative	Step	Action
	1	Open Applications Page
	2	Click Evaluate link of desired Applicant
	3	Change Status and Fill Evaluation comment
	4	Click the Evaluate Button
	5	Displays Evaluation Successful Message, and Applicant Status and Evaluation fields Updated
	6	Click the Back to All Applications Page
	7	Select Another Evaluate link of Desired Applicant

4. Sequence Diagram Explanations

Partner Sequence Diagram

The Partner sequence diagram illustrates the overall interaction flow for the language school partner. It begins with the login process, where the user can log in successfully or get a login error due to wrong email or password. When logged in successfully, the user can now create a CV before applying to available job. To create one, the user has to go to CV creation page and fill necessary information.

After CV creation, the user can now apply available jobs, which can be seen at the Job Listings Page, where the user can select a job and go to Application Page to apply. After application, the user is directed to the Applied Job page to view the applied CV list, the applied job, and its status.

Admin Sequence Diagram

The Admin sequence diagram models the overall workflow of the agency side admin user. The process begins with the login action, which may result either successful login or error message. Once authenticated, the admin now can create jobs at Job Listing Creation Page through create button in Job Listings Page. After all the necessary job information is filled and the job is submitted, it is stored in the database. The user is then directed to the Job Listings Page, where the newly created job is added.

For the evaluation, once the partner side user submits CV to a job, the record will show in All Application Page, where admin can view all applicants for each applied job. The admin can select the applied candidates and evaluate them in the Evaluation Page. After submitting the evaluate button, the evaluation is stored in the database and the admin now can return to All Application Page through the direct link button.

5. Entity Relationship Diagram Explanation

The representation of the system's data model and the relationships between the main entities is illustrated below as an Entity Relationship Diagram. A partner can create multiple CVs, and each CV can be applied to different jobs through the application table. Likewise, a job can receive multiple applications, while one application can only be made to one job. This makes the Applications table a link entity between the c_v_forms and joblistings tables. For the users table, which represents the admin, as its main purpose is for the administrative tasks and not tied to individual admin accounts, the users table can act independently with no direct foreign key connections.

6. Class Diagram

The class diagram shows the structure of the Oshigoto Japan System. It defines main entities, attributes, and the relationships between them. The entities include Partner, CV_Form, Job_Listings, Admin, and Application. The relationships between these entities show the user

interaction with the system and how the data is transmitted, stored, and maintained throughout different processes. Together, the classes form the foundation of the system's database and guide overall implementation of the proposed system.

7. System Architecture Explanation

The Presentation Layer represents the user interface of both Partner and Admin users. It is developed using HTML, Tailwind CSS, and traditional CSS for all the visual structure, layout, and styling for all pages, including form inputs, page displays, and PDF file views, which were mentioned in the technological review of Chapter 2.

For the Application Layer, the system is implemented using the Laravel framework. It contains all the features that manage the system's operations. The features include routing, request handling, authentication, authorization, CRUD functionalities, and dashboard analytics generation. Additionally, third-party integrations, such as Chart.js and Google Calendar, are also included in this layer to generate interactive charts and schedules.

The Data Layer is made up of a MySQL relational database, which stores all system data. Additionally, Laravel's eloquent ORM manages database communication, ensuring efficient data retrieval and maintaining integrity.

8. CRUD Operations of Oshigoto Japan System Users

Create

For creation operations, the users fill in the required information through an HTML <form> component. When the form is submitted, the data is sent to the controller methods through the defined POST routes. The inputs are then validated in the controller and are stored in the database. To allow mass assignment, the fillable attributes are defined in the model's \$fillable property. When the required input is not filled or filled with the wrong data type, there will be an error message. When the data is stored successfully, there will be a success message displayed using the message component.

Update

For update operations, the users first select an existing record, such as a CV, job listing, or partner account, which is retrieved and displayed in an editable form in edit pages. When the users submit the edited information, it is sent to the controller's update method through dedicated PUT routes. The updated inputs are validated in the controllers to ensure data integrity before applying changes to the database. The model's \$fillable property, which enables mass assignment, allows the system to update only the specified fields. If the validation fails, there will be error messages displayed under the input fields. The success message will appear through using the message component after the edit date is stored successfully.

Retrieve

For the retrieve function, it is used for viewing CVs, Jobs, or Application Records. The user pages retrieve data by executing predefined queries and controller methods through the routes, and are displayed using blade syntax “{{ }}”.

Delete

For deletion, it is only implemented in the job application process and is explained in a next part.

9. Application Operations of Oshigoto System Users

Partner Side

The CV application process is implemented using insert and delete operations on the application pivot table on the partner side. When the partner navigates to the application page for the selected job, the interface displays two sections: one containing unapplied CVs filtered with the selected job's job type, with the applicant's desired job type, and the applicant's CV status, while the other contains the selected applicants for the job. The structure allows partners to easily view available CVs to apply and the applied CVs for the selected job.

When the add button of the available CV is submitted with an HTML form, the application record is created in the Application Controller. The record will have the record id, and the job_id and cv_id, which are the foreign keys and represent the IDs of CVs and jobs. When the remove button is clicked, the created application record for the selected CV will be deleted. After adding the applicants to the application record, the record can be seen in the Applied Job Page. The records will show the records based on the applied jobs instead of the overall application timeline on both sides.

Admin Side

On the admin side, the user can view the application records in the All Applications page, where the admin can select the candidate and proceed to the evaluation page. In the evaluation page, the interface displays two sections: the selected applicant's CV and the evaluation part. The evaluation functions the same as the edit operation, which allows the admin to update the CV status and add comments. Once submitted, the updated information is stored in the related tables, and changes are made across the relevant pages.

10. All Folders and Files

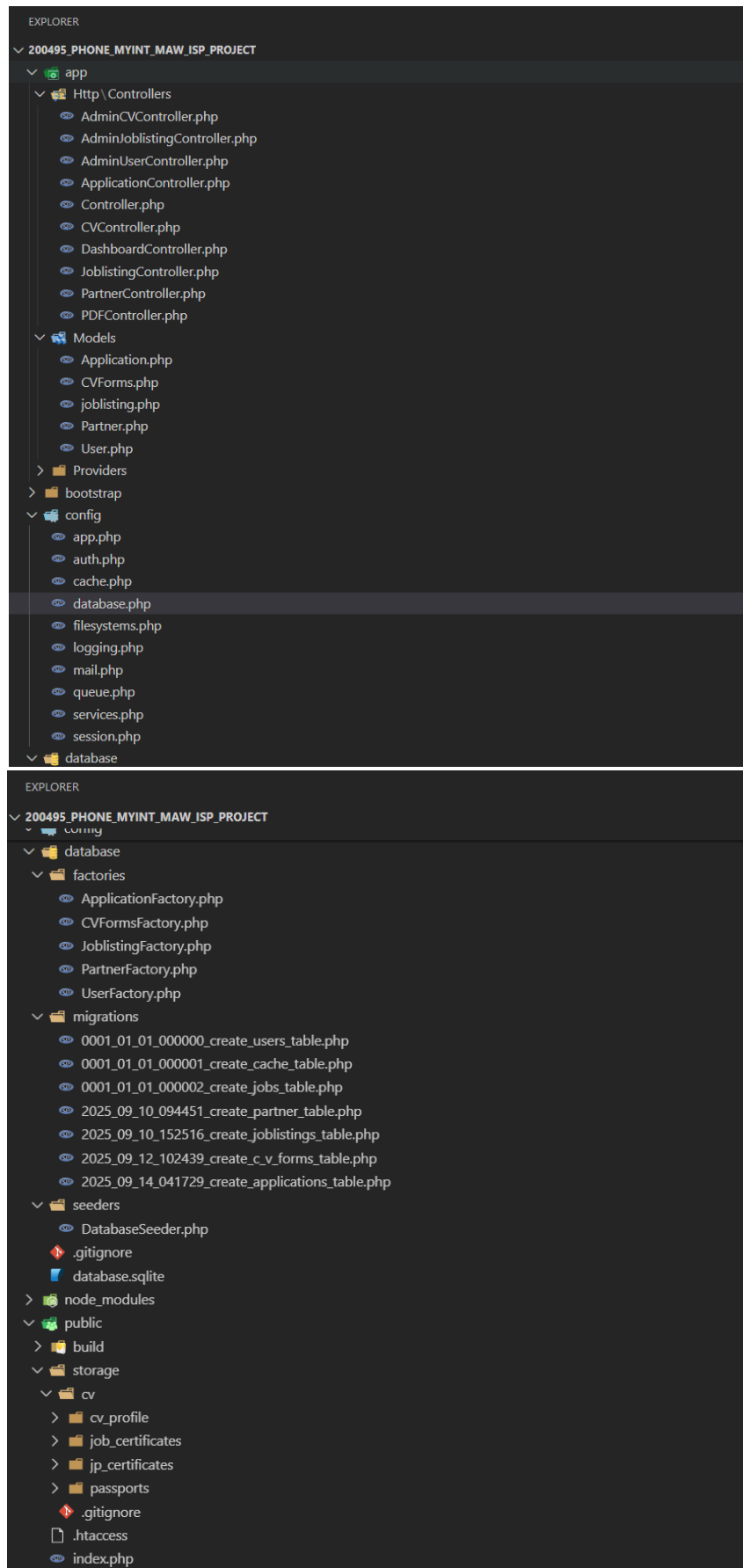


Figure: 10.1. All Folders and Files of Oshigoto Japan ISP Project 1

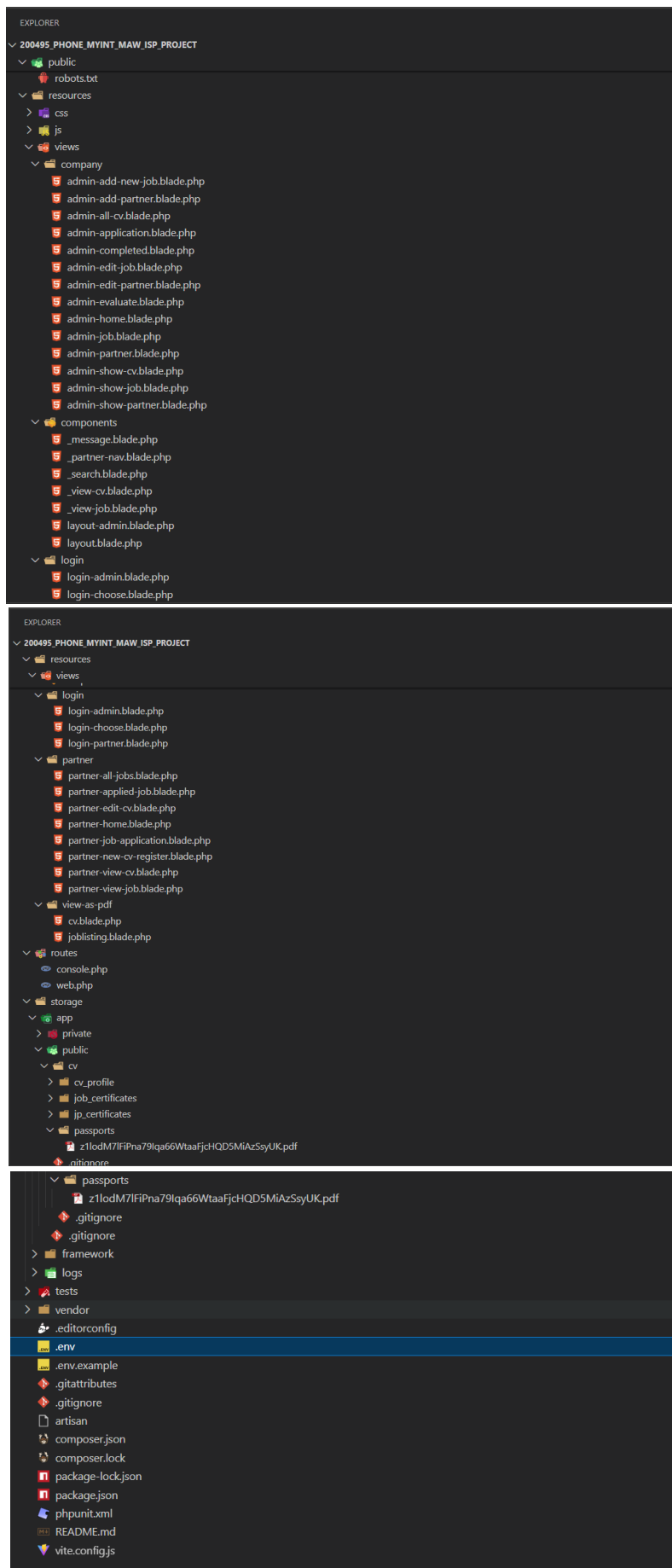


Figure: 10.2. All Folders and Files of Oshigoto Japan ISP Project 2

11. Source Codes

11.1. Migration Files

The migration folder, containing migration files can be found in database folder.

Admin Users Table Structure

```
0001_01_01_000000_create_users_table.php X 2025_09_10_094451_create_partner_table
database>\migrations> 0001_01_01_000000_create_users_table.php>0001_01_01_00
0001_01_01_000000_create_users_table.php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->rememberToken();
21             $table->timestamps();
22         });
23
24         Schema::create('password_reset_tokens', function (Blueprint $table) {
25             $table->string('email')->primary();
26             $table->string('token');
27             $table->timestamp('created_at')->nullable();
28         });
29
30         Schema::create('sessions', function (Blueprint $table) {
31             $table->string('id')->primary();
32             $table->foreignId('user_id')->nullable()->index();
33             $table->string('ip_address', 45)->nullable();
34             $table->text('user_agent')->nullable();
35             $table->longText('payload');
36             $table->integer('last_activity')->index();
37         });
38     }
39
40     /**
41      * Reverse the migrations.
42      */
43     public function down(): void
44     {
45         Schema::dropIfExists('users');
46         Schema::dropIfExists('password_reset_tokens');
47         Schema::dropIfExists('sessions');
48     }
49 };
50
```

Figure: 11.1.1. Migration File for Admin Users Table

Partner Users Table Structure

```
2025_09_10_094451_create_partner_table.php X 2025_09_10_152516_create_joblistings_ta
database > migrations > 2025_09_10_094451_create_partner_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('partners', function (Blueprint $table) {
15             $table->id();
16             $table->string('partner_id')->unique();
17             $table->string('partner_name');
18             $table->string('email')->unique();
19             $table->string('partner_address');
20             $table->string('partner_nationality');
21             $table->string('password');
22             $table->rememberToken();
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('partners');
33     }
34 };
35
```

Figure: 11.1.2. Migration File for Partner Users Table

Joblistings Table Structure

```
2025_09_10_152516_create_joblistings_table.php X 2025_09_12_102439_create_c_v_forms_table
database > migrations > 2025_09_10_152516_create_joblistings_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('joblistings', function (Blueprint $table) {
15             $table->id();
16             $table->string('job_code')->unique();
17             $table->string('job_type');
18             $table->integer('recruit_number');
19             $table->date('application_deadline');
20             $table->string('company_name');
21             $table->string('company_website');
22             $table->string('company_location');
23             $table->string('job_title');
24             $table->text('job_description');
25             $table->text('work_time');
26             $table->string('off_days');
27             $table->text('salary_benefits');
28             $table->text('deductables');
29             $table->text('after_deduction');
30             $table->string('airticket_exp');
31             $table->string('required_jp_level');
32             $table->string('age_range');
33             $table->string('job_gender');
34             $table->string('job_nationality');
35             $table->text('other_requirements') -> nullable();
36             $table->string('job_availability')->default('Available');
37             $table->timestamps();
38         });
39     }
40
41     /**
42      * Reverse the migrations.
43      */
44     public function down(): void
45     {
46         Schema::dropIfExists('joblistings');
47     }
48 };
49
```

Figure: 11.1.3. Migration File for Joblistings Table

CV Forms Table Structure

```
2025_09_12_102439_create_cv_forms_table.php X 2025_09_14_041729_create_applications_table.php ... 2025_09_12_102439_create_cv_forms_table.php X
database > migrations > 2025_09_12_102439_create_cv_forms_table.php > ...
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('cv_forms', function (Blueprint $table) {
15             $table->id();
16             $table->string('cv_code')->unique();
17             $table->string('cv_profile')->nullable(); //File
18             $table->foreignId('partner_id')->nullable()->constrained('partners')
19                 ->onDelete('cascade');
20
21             $table->string('cv_name');
22             $table->string('cv_gender');
23             $table->integer('cv_age'); //
24             $table->date('cv_dob');
25             $table->string('cv_education');
26             $table->string('cv_address');
27             $table->string('cv_religion');
28             $table->string('cv_marriage');
29             $table->string('cv_job_type');
30
31             $table->string('cv_job_certificate')->nullable(); //File
32             $table->string('cv_ip_level');
33
34             $table->string('cv_ip_certificate')->nullable(); //File
35
36             $table->integer('cv_ip_study_year');
37             $table->integer('cv_ip_study_month');
38             $table->string('cv_desired_work_length');
39
40             //SCHOOL HISTORY
41             $table->string('cv_schoolhistory_name');
42             $table->string('cv_schoolhistory_subject');
43             $table->date('cv_schoolhistory_start');
44             $table->date('cv_schoolhistory_end');
45             $table->string('cv_schoolhistory_status');
46
47             $table->string('cv_schoolhistory2_name')->nullable();
48             $table->string('cv_schoolhistory2_subject')->nullable();
49             $table->date('cv_schoolhistory2_start')->nullable();
50             $table->date('cv_schoolhistory2_end')->nullable();
51             $table->string('cv_schoolhistory2_status')->nullable();
52
53             $table->string('cv_schoolhistory3_name')->nullable();
54             $table->string('cv_schoolhistory3_subject')->nullable();
55             $table->date('cv_schoolhistory3_start')->nullable();
56             $table->date('cv_schoolhistory3_end')->nullable();
57             $table->string('cv_schoolhistory3_status')->nullable();
58
59             //JOB HISTORY
60
61         });
62     }
63
64     /**
65      * Reverse the migrations.
66      */
67     public function down(): void
68     {
69         Schema::dropIfExists('cv_forms');
70     }
71 };
72
73 return new class extends Migration
74 {
75     public function up(): void
76     {
77         Schema::create('cv_forms', function (Blueprint $table) {
78             //JOB HISTORY
79             $table->string('cv_jobhistory_name')->nullable();
80             $table->string('cv_jobhistory_position')->nullable();
81             $table->date('cv_jobhistory_start')->nullable();
82             $table->date('cv_jobhistory_end')->nullable();
83             $table->string('cv_jobhistory_status')->nullable();
84
85             $table->string('cv_jobhistory2_name')->nullable();
86             $table->string('cv_jobhistory2_position')->nullable();
87             $table->date('cv_jobhistory2_start')->nullable();
88             $table->date('cv_jobhistory2_end')->nullable();
89             $table->string('cv_jobhistory2_status')->nullable();
90
91             $table->string('cv_jobhistory3_name')->nullable();
92             $table->string('cv_jobhistory3_position')->nullable();
93             $table->date('cv_jobhistory3_start')->nullable();
94             $table->date('cv_jobhistory3_end')->nullable();
95             $table->string('cv_jobhistory3_status')->nullable();
96
97             //OTHERS
98             $table->string('cv_email')->nullable();
99             $table->string('cv_phone')->nullable();
100
101             $table->string('cv_passport_number')->nullable();
102             $table->date('cv_passport_date')->nullable();
103
104             $table->string('cv_passport')->nullable(); //File
105
106             $table->integer('cv_ip_visit_time')->nullable();
107             $table->integer('cv_OCR_Received')->nullable();
108             $table->integer('cv_OCR_Rejected')->nullable();
109
110             $table->string('cv_status')->default('Not Applied'); //For Admins Only
111             $table->text('cv_evaluation')->nullable(); //For Admins Only
112
113             $table->timestamps();
114         });
115     }
116
117     /**
118      * Reverse the migrations.
119      */
120     public function down(): void
121     {
122         Schema::dropIfExists('cv_forms');
123     }
124 };
125
```

Figure: 11.1.4. Migration File for CV Forms Table

Applications Table Structure

```
2025_09_14_041729_create_applications_table.php X
C:\xampp\htdocs\200495_Phone_Myint_Maw_ISP_Project\database\migrations\2025_09_14_04
1729_create_applications_table.php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('applications', function (Blueprint $table) {
15             $table->id();
16
17             $table->foreignId('job_id')->constrained('joblistings')->onDelete('cascade');
18             $table->foreignId('cv_id')->constrained('c_v_forms')->onDelete('cascade');
19             $table->timestamps();
20
21             $table->unique(['cv_id', 'job_id']);
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('applications');
31     }
32 };
33
```

Figure: 11.1.5. Migration File for Applications Table

11.2. Factory Files for Defining Test Data Structure for Seeding

Factory folder, containing factory files can be found in database folder.

Admin Users Data Factory File Code

```
ApplicationFactory.php  CVFormsFactory.php  PartnerFactory.php  UserFactory.php  JoblistingFactory.php
database > factories > UserFactory.php > ...
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6  use Illuminate\Support\Facades\Hash;
7  use Illuminate\Support\Str;
8
9  /**
10 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
11 */
12 class UserFactory extends Factory
13 {
14     /**
15      * The current password being used by the factory.
16      */
17     protected static ?string $password;
18
19     /**
20      * Define the model's default state.
21      *
22      * @return array<string, mixed>
23      */
24     public function definition(): array
25     {
26         return [
27             'name' => fake()->name(),
28             'email' => fake()->unique()->safeEmail(),
29             'email_verified_at' => now(),
30             'password' => static::$password ??= Hash::make('password'),
31             'remember_token' => Str::random(10),
32         ];
33     }
34
35     /**
36      * Indicate that the model's email address should be unverified.
37      */
38     public function unverified(): static
39     {
40         return $this->state(fn (array $attributes) => [
41             'email_verified_at' => null,
42         ]);
43     }
44 }
```

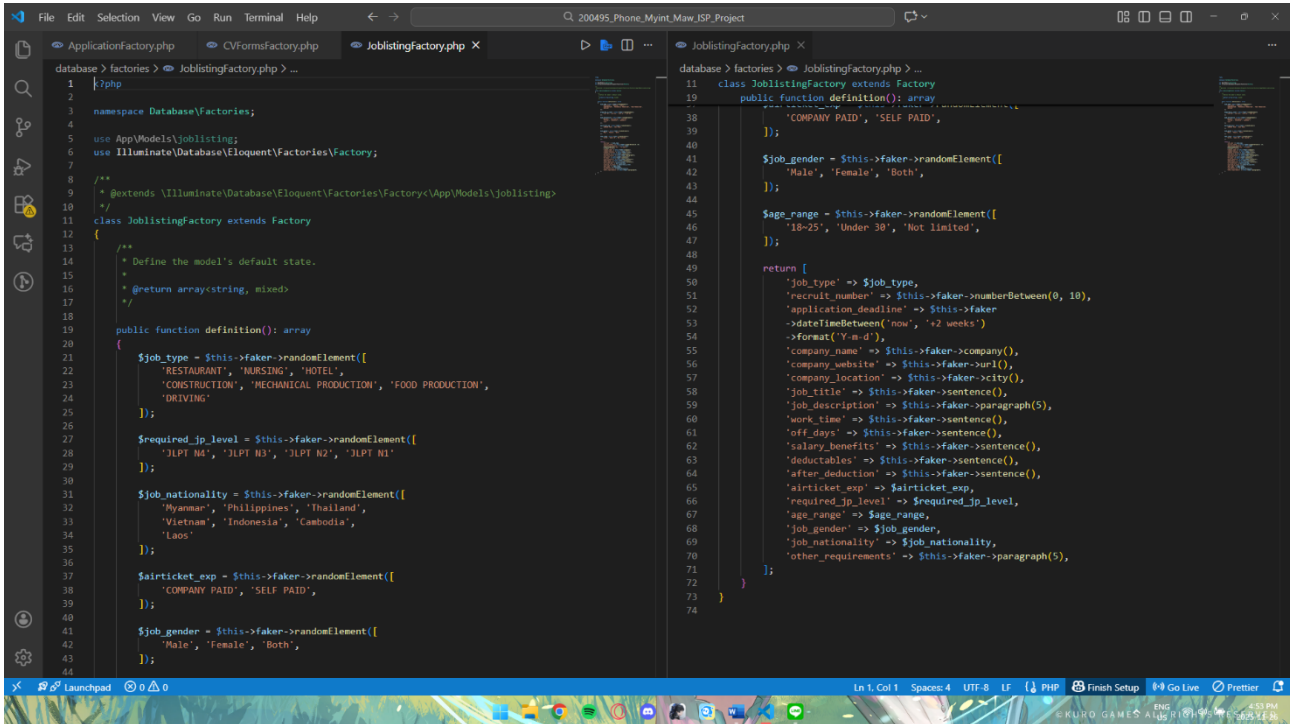
Figure: 11.2.1. Factory File for Admin Users Data

Partner User Data Factory File Code

```
ApplicationFactory.php  CVFormsFactory.php  PartnerFactory.php  JoblistingFactory.php
database > factories > PartnerFactory.php > ...
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6  use App\Models\Partner;
7  use Illuminate\Support\Facades\Hash;
8
9  /**
10 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\partner>
11 */
12 class PartnerFactory extends Factory
13 {
14     /**
15      * Define the model's default state.
16      *
17      * @return array<string, mixed>
18      */
19     protected static ?string $password;
20
21     public function definition(): array
22     {
23         $nationality = $this->faker->randomElement([
24             'Myanmar', 'Philippines', 'Thailand',
25             'Vietnam', 'Indonesia', 'Cambodia',
26             'Laos'
27         ]);
28
29         $prefix = strtoupper(substr($nationality, 0, 2));
30
31         $lastId = Partner::max('id') + 1;
32
33         return [
34             'partner_id' => 'PT - ' .
35             str_pad($lastId, 3, '0', STR_PAD_LEFT) . '-' . $prefix,
36             'partner_name' => $this->faker->company(),
37             'email' => $this->faker->unique()->safeEmail(),
38             'partner_address' => $this->faker->word(),
39             'partner_nationality' => $nationality,
40             'password' => Hash::make('password'),
41         ];
42     }
43
44 }
```

Figure: 11.2.2. Factory File for Partner Users Data

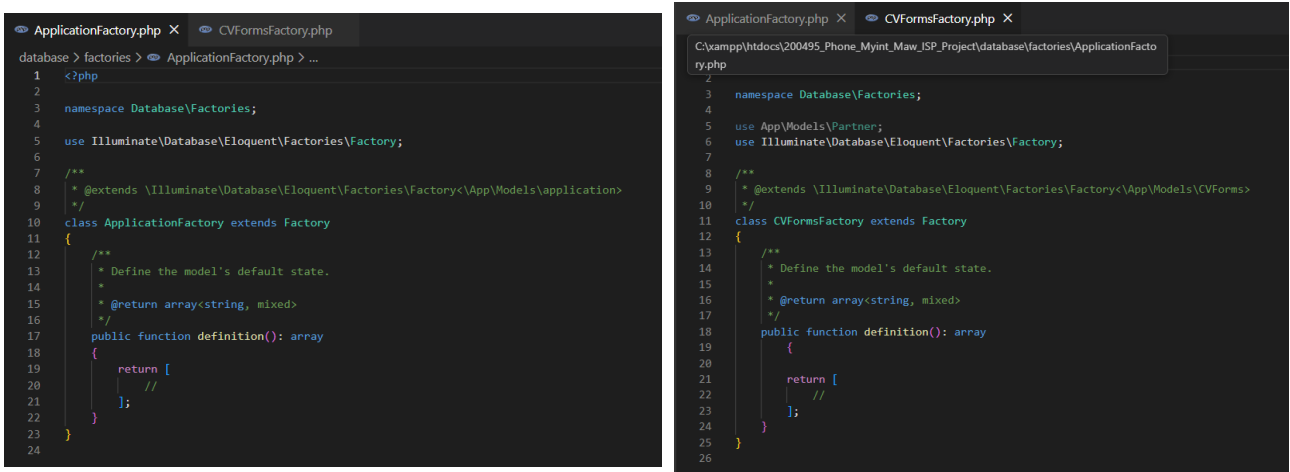
Joblistings Data Factory File Code



```
1 <?php
2
3 namespace Database\Factories;
4
5 use App\Models\Joblisting;
6 use Illuminate\Database\Eloquent\Factories\Factory;
7
8 /**
9  * @extends \Illuminate\Database\Eloquent\Factories\Factory<App\Models\Joblisting>
10 */
11 class JoblistingFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array<string, mixed>
17      */
18     public function definition(): array
19     {
20         $job_type = $this->faker->randomElement([
21             'RESTAURANT', 'NURSING', 'HOTEL',
22             'CONSTRUCTION', 'MECHANICAL PRODUCTION', 'FOOD PRODUCTION',
23             'DRIVING'
24         ]);
25
26         $required_jp_level = $this->faker->randomElement([
27             'JLPT N4', 'JLPT N3', 'JLPT N2', 'JLPT N1'
28         ]);
29
30         $job_nationality = $this->faker->randomElement([
31             'Myanmar', 'Philippines', 'Thailand',
32             'Vietnam', 'Indonesia', 'Cambodia',
33             'Laos'
34         ]);
35
36         $airticket_exp = $this->faker->randomElement([
37             'COMPANY PAID', 'SELF PAID',
38         ]);
39
40         $job_gender = $this->faker->randomElement([
41             'Male', 'Female', 'Both',
42         ]);
43
44     }
45 }
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
```

Figure: 11.2.3. Factory File for Joblistings Data

Applications and CV Form Data Factory File Code



```
ApplicationFactory.php
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6
7 /**
8  * @extends \Illuminate\Database\Eloquent\Factories\Factory<App\Models\application>
9 */
10 class ApplicationFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition(): array
18     {
19         return [
20             //
21         ];
22     }
23 }
24
```

```
CVFormsFactory.php
1
2
3 namespace Database\Factories;
4
5 use App\Models\Partner;
6 use Illuminate\Database\Eloquent\Factories\Factory;
7
8 /**
9  * @extends \Illuminate\Database\Eloquent\Factories\Factory<App\Models\CVForms>
10 */
11 class CVFormsFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array<string, mixed>
17      */
18     public function definition(): array
19     {
20         return [
21             //
22         ];
23     }
24 }
25
26
```

Figure: 11.2.4. Factory File for Applications and CV Form Data

11.3. Database Seeder file for Seeding Test Data

The Seeders folder, containing DatabaseSeeder.php file, can be found under the database folder.

```
DatabaseSeeder.php X
database > seeders > DatabaseSeeder.php > DatabaseSeeder > run
1 <?php
2
3 namespace Database\Seeders;
4
5 use App\Models\User;
6 use App\Models\joblisting;
7 use App\Models\CVForms;
8 use Illuminate\Support\Facades\Hash;
9 use App\Models\partner;
10 use Database\Factories\CVFormsFactory;
11 // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
12 use Illuminate\Database\Seeder;
13
14 class DatabaseSeeder extends Seeder
15 {
16     /**
17      * Seed the application's database.
18      */
19     public function run(): void
20     {
21
22         User::factory()->create([
23             'name' => 'Phone Myint Maw',
24             'email' => 'admin@oshigoto.com',
25             'password' => 'pmm552002'
26         ]);
27
28
29         joblisting::factory(15)->create();
30
31         $partners = [
32             [
33                 'partner_name' => 'Toshin',
34                 'email' => 'mmtoshin@gmail.com',
35                 'partner_address' => 'Toshin Language School',
36                 'partner_nationality' => 'Myanmar',
37                 'password' => Hash::make('toshin2025'),
38             ],
39             [
40                 'partner_name' => 'Hikari',
41                 'email' => 'Hikari@gmail.com',
42                 'partner_address' => 'Hikari Language School',
43                 'partner_nationality' => 'Indonesia',
44                 'password' => Hash::make('hikari2025'),
45             ],
46             [
47                 'partner_name' => 'Nihongo',
48                 'email' => 'nihongo@gmail.com',
49                 'partner_address' => 'Nihongo Language School',
50                 'partner_nationality' => 'Laos',
51                 'password' => Hash::make('nihon2025'),
52             ],
53         ];
54
55         foreach($partners as $partnerData){
56
57             Partner::create($partnerData);
58
59         }
60     }
61 }
62
```

Figure: 11.3.1. Database seeder File for Seeding Test Data

11.4. Model files

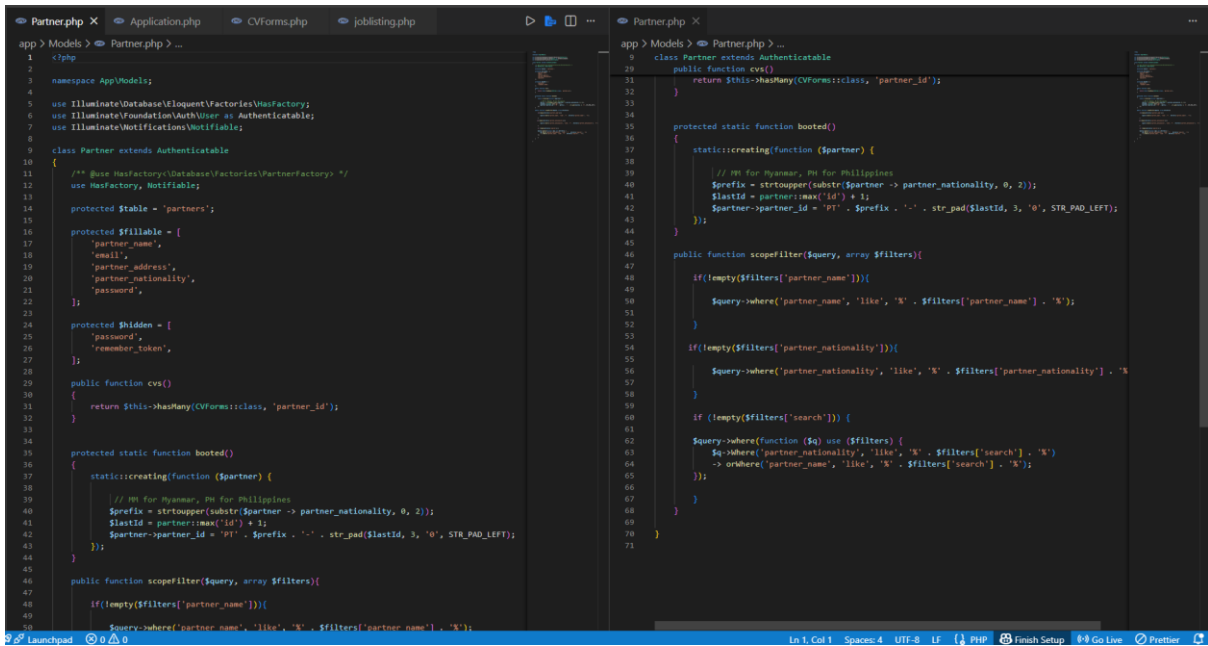
The Model folder, containing model files can be found under the App folder.

Admin User Model File

```
User.php X Application.php X CVForms.php joblisting.p
app > Models > User.php > ...
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<\Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var list<string>
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24     ];
25
26     /**
27      * The attributes that should be hidden for serialization.
28      *
29      * @var list<string>
30      */
31     protected $hidden = [
32         'password',
33         'remember_token',
34     ];
35
36     /**
37      * Get the attributes that should be cast.
38      *
39      * @return array<string, string>
40      */
41     protected function casts(): array
42     {
43         return [
44             'email_verified_at' => 'datetime',
45             'password' => 'hashed',
46         ];
47     }
48 }
49
```

Figure: 11.4.1. Admin User Model File Codes

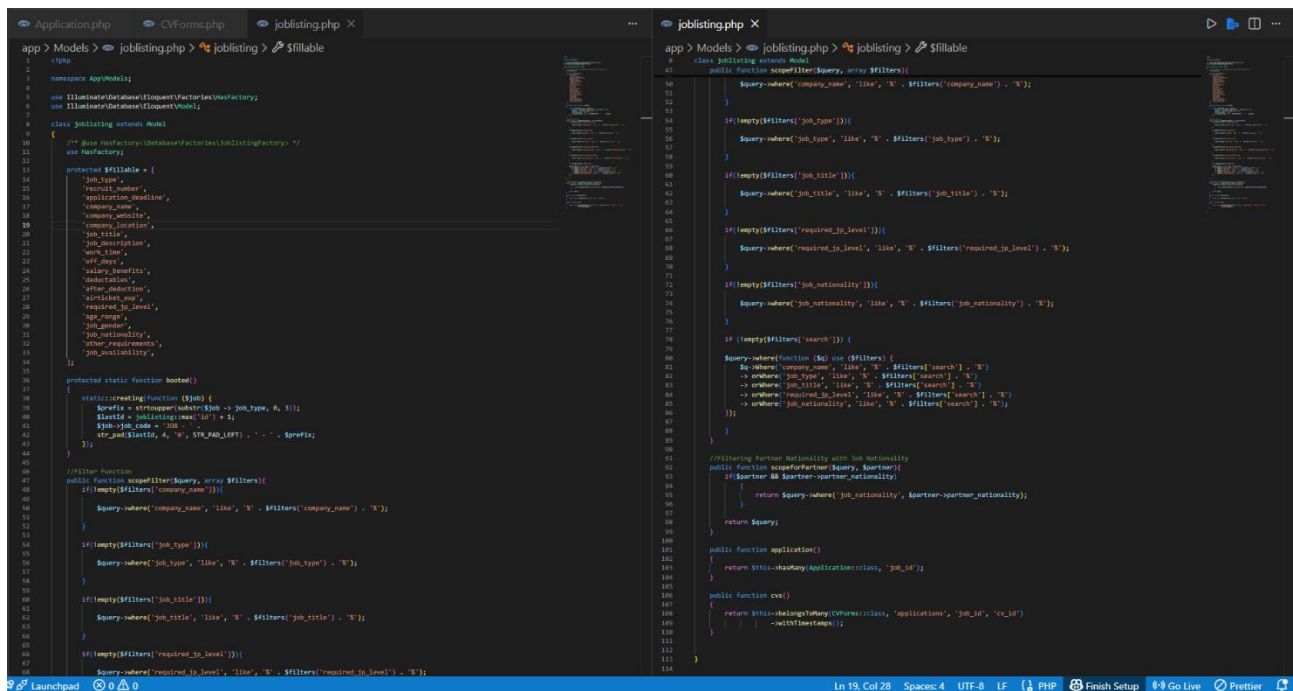
Partner User Model File



```
app > Models > Partner.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8
9 class Partner extends Authenticatable
10 {
11     /** @use HasFactory<Database\Factories\PartnerFactory> */
12     use HasFactory, Notifiable;
13
14     protected $table = 'partners';
15
16     protected $fillable = [
17         'partner_name',
18         'email',
19         'partner_address',
20         'partner_nationality',
21         'password',
22     ];
23
24     protected $hidden = [
25         'password',
26         'remember_token',
27     ];
28
29     public function cvs()
30     {
31         return $this->hasMany(CVForms::class, 'partner_id');
32     }
33
34     protected static function booted()
35     {
36         static::creating(function ($partner) {
37             // PH For Myanmar, PH for Philippines
38             $prefix = strtoupper(substr($partner->partner_nationality, 0, 2));
39             $lastId = partner::max('id') + 1;
40             $partner->partner_id = 'PT' . $prefix . '-' . str_pad($lastId, 3, '0', STR_PAD_LEFT);
41         });
42     }
43
44     public function scopeFilter($query, array $filters)
45     {
46         if (empty($filters['partner_name'])) {
47             $query->where('partner_name', 'like', '%' . $filters['partner_name'] . '%');
48         }
49     }
50
51     class Partner extends Authenticatable
52     {
53         public function cvs()
54         {
55             return $this->hasMany(CVForms::class, 'partner_id');
56         }
57
58         protected static function booted()
59         {
60             static::creating(function ($partner) {
61                 // PH For Myanmar, PH for Philippines
62                 $prefix = strtoupper(substr($partner->partner_nationality, 0, 2));
63                 $lastId = partner::max('id') + 1;
64                 $partner->partner_id = 'PT' . $prefix . '-' . str_pad($lastId, 3, '0', STR_PAD_LEFT);
65             });
66         }
67
68         public function scopeFilter($query, array $filters)
69         {
70             if (empty($filters['partner_name'])) {
71                 $query->where('partner_name', 'like', '%' . $filters['partner_name'] . '%');
72             }
73
74             if (empty($filters['partner_nationality'])) {
75                 $query->where('partner_nationality', 'like', '%' . $filters['partner_nationality'] . '%');
76             }
77
78             if (empty($filters['search'])) {
79                 $query->where(function ($q) use ($filters) {
80                     $q->where('partner_nationality', 'like', '%' . $filters['search'] . '%')
81                     ->orWhere('partner_name', 'like', '%' . $filters['search'] . '%');
82                 });
83             }
84         }
85     }
86 }
```

Figure: 11.4.2. Partner User Model File Codes

Joblistings Model File



```
app > Models > joblistings.php > joblistings > $fillable
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7
8 class Joblistings extends Model
9 {
10     /** @use HasFactory<Database\Factories\JoblistingsFactory> */
11     use HasFactory;
12
13     protected $fillable = [
14         'job_type',
15         'recruit_number',
16         'application_deadline',
17         'company_name',
18         'company_website',
19         'company_location',
20         'job_title',
21         'job_description',
22         'work_time',
23         'salary_range',
24         'education',
25         'other_requirements',
26         'highest_exp',
27         'required_to_level',
28         'app_range',
29         'job_gender',
30         'job_nationality',
31         'other_requirements',
32         'job_availability',
33     ];
34
35     protected static function booted()
36     {
37         static::creating(function ($job) {
38             $prefix = strtoupper(substr($job->job_type, 0, 3));
39             $lastId = joblistings::max('id') + 1;
40             $job->job_code = $job . $prefix . '-' . str_pad($lastId, 4, '0', STR_PAD_LEFT);
41         });
42     }
43
44     //filter function
45     public function scopeFilter($query, array $filters)
46     {
47         if (empty($filters['company_name'])) {
48             $query->where('company_name', 'like', '%' . $filters['company_name'] . '%');
49         }
50
51         if (empty($filters['job_type'])) {
52             $query->where('job_type', 'like', '%' . $filters['job_type'] . '%');
53         }
54
55         if (empty($filters['job_title'])) {
56             $query->where('job_title', 'like', '%' . $filters['job_title'] . '%');
57         }
58
59         if (empty($filters['required_to_level'])) {
60             $query->where('required_to_level', 'like', '%' . $filters['required_to_level'] . '%');
61         }
62
63         if (empty($filters['job_nationality'])) {
64             $query->where('job_nationality', 'like', '%' . $filters['job_nationality'] . '%');
65         }
66
67         if (empty($filters['search'])) {
68             $query->where(function ($q) use ($filters) {
69                 $q->where('company_name', 'like', '%' . $filters['search'] . '%')
70                 ->orWhere('job_type', 'like', '%' . $filters['search'] . '%')
71                 ->orWhere('job_title', 'like', '%' . $filters['search'] . '%')
72                 ->orWhere('required_to_level', 'like', '%' . $filters['search'] . '%')
73                 ->orWhere('job_nationality', 'like', '%' . $filters['search'] . '%');
74             });
75         }
76     }
77
78     //filtering Partner Nationality with Job Nationality
79     public function scopeForPartner($query, $partner)
80     {
81         return $query->where('job_nationality', $partner->partner_nationality);
82     }
83
84     return $query;
85 }
86
87 public function application()
88 {
89     return $this->hasMany(Application::class, 'job_id');
90 }
91
92 public function cvs()
93 {
94     return $this->hasMany(CVForms::class, 'application', 'job_id', 'cv_id')
95     ->withTimestamps();
96 }
97 }
```

Figure: 11.4.3. Joblistings Model File Codes

CV Forms Model File

```

1 <?php
2 namespace App\Models;
3
4 use Illuminate\Database\Eloquent\Factories\HasFactory;
5 use Illuminate\Database\Eloquent\Model;
6
7 class CVForms extends Model
8 {
9     /** @use HasFactory<Database\Factories\CVFormsFactory> */
10    use HasFactory;
11
12    protected $fillable = [
13
14        'cv_code',
15        'partner_id',
16        'cv_profile',
17        'cv_name',
18        'cv_gender',
19        'cv_age',
20        'cv_dob',
21        'cv_nationality',
22        'cv_address',
23        'cv_religion',
24        'cv_marriage',
25        'cv_job_type',
26        'cv_job_certificate',
27        'cv_jp_level',
28        'cv_jp_certificate',
29        'cv_jp_study_year',
30        'cv_jp_study_month',
31        'cv_desired_work_length',
32        'cv_schoolhistory1_name',
33        'cv_schoolhistory1_subject',
34        'cv_schoolhistory1_start',
35        'cv_schoolhistory1_end',
36        'cv_schoolhistory1_status',
37        'cv_schoolhistory2_name',
38        'cv_schoolhistory2_subject',
39        'cv_schoolhistory2_start',
40        'cv_schoolhistory2_end',
41        'cv_schoolhistory2_status',
42        'cv_schoolhistory3_name',
43        'cv_schoolhistory3_subject',
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2
```

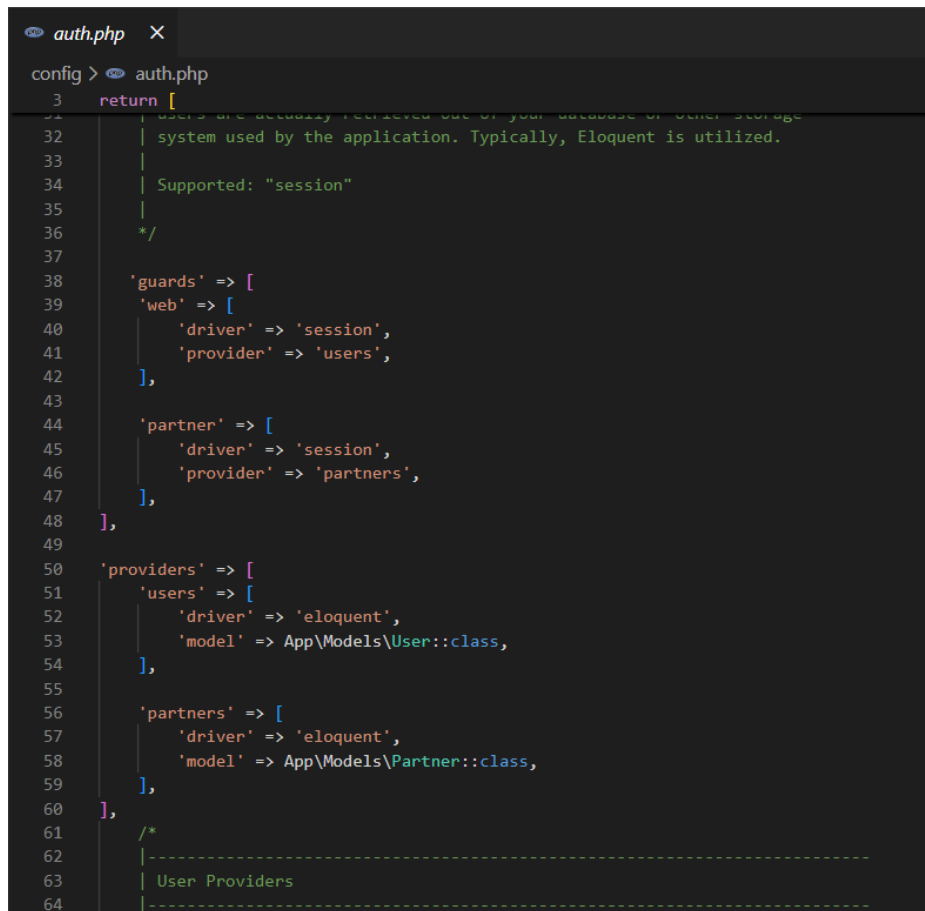
Applications Model File

```
Application.php X
app > Models > Application.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class application extends Model
9  {
10     /** @use HasFactory<\Database\Factories\ApplicationFactory> */
11     use HasFactory;
12
13     protected $fillable = [
14         'cv_id',
15         'job_id',
16     ];
17
18     public function cv()
19     {
20         return $this->belongsTo(CVForms::class, 'cv_id');
21     }
22
23     public function job()
24     {
25         return $this->belongsTo(Joblisting::class, 'job_id');
26     }
27
28     public function scopeFilter($query, array $filters){ //For Filter Function
29
30         if(!empty($filters['job_code'])){
31             $query->where('job_code', 'like', '%' . $filters['job_code'] . '%');
32         }
33
34         if(!empty($filters['company_name'])){
35             $query->where('company_name', 'like', '%' . $filters['company_name'] . '%');
36         }
37
38         if (!empty($filters['search'])) {
39
40             $query->where(function ($q) use ($filters) {
41                 $q->where('company_name', 'like', '%' . $filters['search'] . '%')
42                 -> orWhere('job_code', 'like', '%' . $filters['search'] . '%');
43             });
44         }
45     }
46 }
47
48
49
50
51
52
```

Figure: 11.4.5. Applications Model File Codes

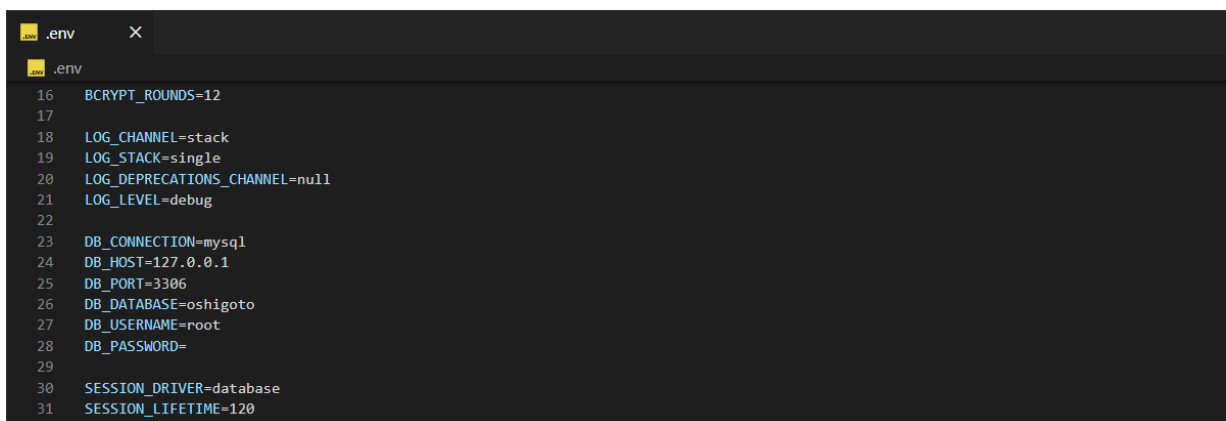
11.5. Configurations

The auth.php file can be found in config folder.



```
31  | users are actually retrieved out of your database or other storage
32  | system used by the application. Typically, Eloquent is utilized.
33  |
34  | Supported: "session"
35  |
36  | */
37  |
38  | 'guards' => [
39  |     'web' => [
40  |         'driver' => 'session',
41  |         'provider' => 'users',
42  |     ],
43  |     'partner' => [
44  |         'driver' => 'session',
45  |         'provider' => 'partners',
46  |     ],
47  | ],
48  | ],
49  |
50  | 'providers' => [
51  |     'users' => [
52  |         'driver' => 'eloquent',
53  |         'model' => App\Models\User::class,
54  |     ],
55  |     'partners' => [
56  |         'driver' => 'eloquent',
57  |         'model' => App\Models\Partner::class,
58  |     ],
59  | ],
60  | ],
61  | /*
62  | |-----
63  | | User Providers
64  | |-----
65  | */
```

Figure: 11.5.1. Configuration in auth file for Authentication



```
16 BCRYPT_ROUNDS=12
17
18 LOG_CHANNEL=stack
19 LOG_STACK=single
20 LOG_DEPRECATIONS_CHANNEL=null
21 LOG_LEVEL=debug
22
23 DB_CONNECTION=mysql
24 DB_HOST=127.0.0.1
25 DB_PORT=3306
26 DB_DATABASE=oshigoto
27 DB_USERNAME=root
28 DB_PASSWORD=
29
30 SESSION_DRIVER=database
31 SESSION_LIFETIME=120
32 SESSION_ENCRYPT=...
```

Figure: 11.5.2. Configuration in env file for Database connection

11.6. Components

The component file can be found in components folder in views folder, which is in the resources file.

Partner Page Layout File

```
resources > views > components > layout.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"
9 integrity="sha512-Kfkfw0DslogbKJ9GpAjZoe0ifLqoi1OPpKe0MgOK9O/XpAi49QJi4T8eM7Z+Qqoige"
10 crossorigin="anonymous" referrerpolicy="no-referrer" />
11 <script src="//unpkg.com/alpinejs" /></script>
12 <script src="https://cdn.tailwindcss.com" /></script>
13
14 <title></title>
15
16 </head>
17
18 <body class="mb-48">
19
20 <nav class="inline-flex w-full font-bold bg-white text-black justify-between items-center p-5 shadow-lg">
21 <div class="text-2xl">
22 <span class="text-red-700">JAPAN</span>
23 </div>
24
25 <ul class="flex justify-center space-x-6 mr-6 text-lg">
26
27 <li>@auth('partner')
28
29 <li>
30 <span class="font-old-uppercase">
31 Welcome, {{ auth('partner')->user()->partner_name }} !
32 </span>
33 </li>
34
35 <li>
36 <p />
37 </li>
38
39 <li>
40
41 <form method="POST" action="/partner/logout" class="inline">
42 @csrf
43 <button type="submit" class="transition duration:300 hover:underline">
44 <i class="fa-solid fa-door-closed col-span-1"></i>
45 <span class="col-span-4">Logout</span>
46 </button>
47 </form>
48
49 </li>
50
51 </ul>
52 </nav>
53
54 <main>
55
56 <div class="flex justify-center space-x-6 mr-6 text-lg">
57
58 </div>
59
60 <div class="flex justify-end p-3 pr-10 font-bold text-2xl">
61 <div class="flex justify-end p-3 pr-10 text-2xl">
62 <div class="flex justify-end p-3 pr-10 text-2xl">
63 <div class="flex justify-end p-3 pr-10 text-2xl">
64 </div>
65
66 @else
67
68 <div class="flex justify-end p-3 pr-10 font-bold text-2xl">
69 <div class="flex justify-end p-3 pr-10 text-2xl">
70 <div class="flex justify-end p-3 pr-10 text-2xl">
71 <div class="flex justify-end p-3 pr-10 text-2xl">
72 </div>
73
74 <div class="flex justify-end p-3 pr-10 text-2xl">
75 <div class="flex justify-end p-3 pr-10 text-2xl">
76 <div class="flex justify-end p-3 pr-10 text-2xl">
77 <div class="flex justify-end p-3 pr-10 text-2xl">
78 <div class="flex justify-end p-3 pr-10 text-2xl">
79 <div class="flex justify-end p-3 pr-10 text-2xl">
80 <div class="flex justify-end p-3 pr-10 text-2xl">
81 <div class="flex justify-end p-3 pr-10 text-2xl">
82 </div>
83
84 @endauth
85
86 </body>
87
88 </html>
89
```

Figure: 11.6.1. layout.blade.php file

Partner Navigation Bar Layout File

```
resources > views > components > _partner-nav.blade.php > ...
1 <section class="flex font-bold bg-white text-black justify-between items-center p-5">
2
3 <div>
4 <ul class="grid grid-cols-1 lg:grid-cols-3 space-x-6 text-lg">
5
6 <li class="col-span-1">
7 <a href="/home" class="inline-block font-bold border-black text-black uppercase hover:underline">
8 <i class="fa-solid fa-list"></i> Registered CV List
9 </a>
10 </li>
11
12 <li class="col-span-1">
13 <a href="/partner-applied-list"
14 class="inline-block font-bold border-black text-black uppercase hover:underline">
15 <i class="fa-solid fa-check-to-slot"></i> Applied Job List
16 </a>
17 </li>
18
19 <li class="col-span-1">
20 <a href="/apply-jobs" class="inline-block font-bold border-black text-black uppercase hover:underline">
21 <i class="fa-solid fa-building"></i> Apply New Jobs
22 </a>
23 </li>
24
25 </ul>
26 </div>
27
28 <div>
29 <a href="/new-cv-register" class="transition duration:700 inline-block border-2 font-bold border-black
30 text-black py-2 px-4 rounded-xl
31 uppercase hover:bg-black hover:text-white hover:animate-pulse shadow-lg">
32 <i class="fa-solid fa-plus"></i> Make new CV Forms
33 </a>
34 </div>
35
36 </section>
37
```

Figure: 11.6.2. _partner-nav.blade.php file

Admin Page Layout File

```

layout-admin.blade.php X
resources > views > components > layout-admin.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"
9 integrity="sha512-Kfkfw0Ds3kIwQp8161VCABqXVL09mk+hI149t6m1490Cx14yDRlzmP1t9daB5Th7284e500OdzMzqd1VaPg4K399K"
10 crossorigin="anonymous" referrerpolicy="no-referrer" />
11 <script src="https://cdn.tailwindcss.com/"></script>
12
13 <title></title>
14
15
16 </head>
17
18 <body>
19
20 <div x-data="{open: false}" class="flex bg-gray-100 relative">
21
22 <button @click="open = !open" class="fixed w-[50px] top-4 left-4 z-50 p-3 bg-black
23 text-white rounded-lg hover:bg-gray-700
24 transition">
25 <i class="fa-solid fa-bars"/></i>
26 </button>
27
28 <aside :class="open ? 'translate-x-0 w-64' : 'translate-x-full w-64'" class="fixed top-0 left-0 h-screen
29 bg-gray-200 dark:bg-gray-200 p-4 transition-all
30 duration-300 ease-in-out shadow-lg z-40 overflow-y-auto">
31
32 <div class="font-bold text-2xl text-center mb-5 @t-20">
33 <span class="text-red-700">JAPAN</span>
34 </div>
35
36 <h2>Auth</h2>
37
38 <div class="font-bold text-lg text-center mb-5 text-blue-700">
39 {{ auth('web')->user()->name }}
40 </div>
41 <ul class="text-lg font-bold p-5">
42
43 <li class="mb-5 justify-right items-center">
44 <a href="/admin-home" class="transition duration-300 grid grid-cols-5 uppercase
45 text-black bg-transparent
46 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
47 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
48 <i class="fa-solid fa-gear col-span-1"/></i>
49 <span class="col-span-4">Dashboard</span>
50 </a>
51
52 </li>
53
54 <li class="mb-5 justify-right items-center">
55 <a href="/all-listed-jobs" class="transition duration-300 grid grid-cols-5 uppercase
56 text-black bg-transparent
57 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
58 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
59 <i class="fa-solid fa-thumbtack col-span-1"/></i>
60 <span class="col-span-4">Jobs</span>
61 </a>
62 </li>
63
64 <li class="mb-5 justify-right items-center">
65 <a href="/all-partners" class="transition duration-300 grid grid-cols-5 uppercase
66 text-black bg-transparent
67 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
68 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
69 <i class="fa-solid fa-handshake col-span-1"/></i>
70 <span class="col-span-4">Partners</span>
71 </a>
72 </li>
73
74 <li class="mb-5 justify-right items-center">
75 <a href="/admin-applications" class="transition duration-300 grid grid-cols-5 uppercase
76 text-black bg-transparent
77 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
78 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
79 <i class="fa-solid fa-folder-open col-span-1"/></i>
80 <span class="col-span-4">Applications</span>
81 </a>
82 </li>
83
84 <li class="mb-5 justify-right items-center">
85 <a href="/all-cvs" class="transition duration-300 grid grid-cols-5 uppercase
86 text-black bg-transparent
87 hover:bg-gray-200 hover:text-gray-900 rounded-lg inline-flex items-center
88 dark:hover:bg-gray-600 dark:hover:text-white py-2 px-3">
89 <i class="fa-solid fa-scroll col-span-1"/></i>
90 <span class="col-span-4">All CVs</span>
91 </a>
92 </li>
93
94 </ul>
95
96 </div>
97
98 <div class="flex justify-between items-center">
99 <div class="font-bold text-2xl">{{ $title }}</div>
100 <div class="font-bold text-2xl">You are not logged in</div>
101 </div>
102
103 <div class="flex justify-end p-3 pr-18 font-bold text-2xl">
104 <div class="flex justify-center items-center">
105 <a href="#" class="mb-5 inline-block border-2 font-bold
106 border-black text-white bg-black py-4 rounded-xl uppercase
107 hover:text-black hover:bg-white">
108 Log In
109 </a>
110 </div>
111 </div>
112
113 </body>
114
115 </html>
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure: 11.6.3. layout-admin.blade.php file

Search Function File

```
_search.blade.php X
resources > views > components > _search.blade.php > div.relative.border-2.border-gray-100.m-4.rounded-lg
1 <div class="relative border-2 border-gray-100 m-4 rounded-lg">
2   <div class="absolute top-4 left-3">
3     <i class="fa fa-search text-gray-400 z-20 hover:text-gray-500"></i>
4   </div>
5   <input type="text" name="search" class="h-14 w-full pl-10 pr-20 rounded-lg z-0 focus:shadow focus:outline-none"
6     placeholder="Type Keywords to Search" />
7   <div class="absolute top-2 right-2">
8     <button type="submit" class="h-10 w-20 text-white rounded-lg bg-black hover:bg-gray-500">
9       Search
10    </button>
11  </div>
12 </div>
13
```

Figure: 11.6.4. _search.blade.php file

Message Function File

```
_message.blade.php X
resources > views > components > _message.blade.php > ...
1 @if(session('message'))
2
3   <div x-data="{show: true}" x-init="setTimeout(() => show = false, 3000)" x-show="show"
4     class="bg-green-200 text-green-800 p-2 rounded mb-2">
5     <p>
6       {{session('message')}}
7     </p>
8   </div>
9
10 @endif
11
```

Figure: 11.6.5. _message.blade.php file

CV Form View Layout File

```
resources > views > components > _view-cv.blade.php > div.max-w-4xl.mx-auto.bg-white.shadow-lg.rounded-xl.p-6
1 <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2 <!-- title -->
3 <h2 class="text-2xl font-bold text-center mb-6">Curriculum Vitae</h2>
4
5 <!-- Basic Info -->
6 <table class="w-full border border-gray-300 text-sm mb-6">
7 <tr class="border border-gray-300">
8 <th class="bg-gray-200 px-3 py-2 text-left w-32">CV Code</th>
9 <td class="px-3 py-2 font-bold">{{ $cv->cv_code }}</td>
10 <th class="bg-gray-200 px-3 py-2 text-left">Name</th>
11 <td class="px-3 py-2">{{ $cv->cv_name }}</td>
12 <td colspan="6" class="flex flex-col justify-center border-1 border-gray-300 text-center align-top w-36">
13 <div class="flex flex-col items-center justify-center">
14 <!-- if($cv->cv_profile) -->
15
16
17 
19 <span class="text-xs text-gray-600">Photo</span>
20
21 <!-- else -->
22 <p class="italic">Please Upload Profile Image</p>
23 </div>
24 </td>
25 </tr>
26 </table>
27
28 <tr class="border border-gray-300">
29 <th class="bg-gray-200 px-3 py-2 text-left w-32">Registered</th>
30 <td class="px-3 py-2">{{ $cv->created_at }}</td>
31 <th class="bg-gray-200 px-3 py-2 text-left">Gender</th>
32 <td class="px-3 py-2">{{ $cv->cv_gender }}</td>
33 </tr>
34 <tr class="border border-gray-300">
35 <th class="bg-gray-200 px-3 py-2 text-left">Nationality</th>
36 <td class="px-3 py-2">{{ $cv->cv_nationality }}</td>
37 <th class="bg-gray-200 px-3 py-2 text-left">Address</th>
38 <td class="px-3 py-2">{{ $cv->cv_address }}</td>
39 </tr>
40 <tr class="border border-gray-300">
41 <th class="bg-gray-200 px-3 py-2 text-left">Date of Birth</th>
42 <td class="px-3 py-2">{{ $cv->cv_dob }}</td>
43 <th class="bg-gray-200 px-3 py-2 text-left">Age</th>
44 <td class="px-3 py-2">{{ $cv->cv_age }}</td>
45 </tr>
46 <tr class="border border-gray-300">
47 <th class="bg-gray-200 px-3 py-2 text-left">Religion</th>
48 <td class="px-3 py-2">{{ $cv->cv_religion }}</td>
49 </tr>
50 </table>
51
52 <!-- Education Section -->
53 <h3 class="text-lg font-semibold mt-6 mb-3 text-center font-bold">School History</h3>
54 <table class="w-full border border-gray-300 text-sm mb-6">
55 <thead>
56 <tr class="bg-gray-200 border border-gray-300">
57 <th class="px-3 py-2 text-left">School Name</th>
58 <th class="px-3 py-2 text-left">Subject</th>
59 <th class="px-3 py-2 text-left">Start Date</th>
60 <th class="px-3 py-2 text-left">End Date</th>
61 <th class="px-3 py-2 text-left">Status</th>
62 </thead>
63 <tbody>
64 <!-- foreach -->
65 <tr class="border border-gray-300">
66 <td class="px-3 py-2">{{ $cv->cv_schoolhistory($i)_name }}</td>
67 <td class="px-3 py-2">{{ $cv->cv_schoolhistory($i)_subject }}</td>
68 <td class="px-3 py-2">{{ $cv->cv_schoolhistory($i)_start }}</td>
69 <td class="px-3 py-2">{{ $cv->cv_schoolhistory($i)_end }}</td>
70 <td class="px-3 py-2">{{ $cv->cv_schoolhistory($i)_status }}</td>
71 </tr>
72 </tbody>
73 </table>
74
75 <!-- Work Experience -->
76 <h3 class="text-lg font-semibold mt-6 mb-3 text-center font-bold">Job History</h3>
77 <table class="w-full border border-gray-300 text-sm mb-6">
78 <thead>
79 <tr class="bg-gray-200 border border-gray-300">
80 <th class="px-3 py-2 text-left">Company Name</th>
81 <th class="px-3 py-2 text-left">Position</th>
82 <th class="px-3 py-2 text-left">Start Date</th>
83 <th class="px-3 py-2 text-left">End Date</th>
84 <th class="px-3 py-2 text-left">Status</th>
85 </thead>
86 <tbody>
87 <!-- foreach -->
88 <tr class="border border-gray-300">
89 <td class="px-3 py-2">{{ $cv->cv_jobhistory($i)_name }}</td>
90 <td class="px-3 py-2">{{ $cv->cv_jobhistory($i)_subject }}</td>
91 <td class="px-3 py-2">{{ $cv->cv_jobhistory($i)_start }}</td>
92 <td class="px-3 py-2">{{ $cv->cv_jobhistory($i)_end }}</td>
93 <td class="px-3 py-2">{{ $cv->cv_jobhistory($i)_status }}</td>
94 </tr>
95 </tbody>
96 </table>
97
98 <!-- Other Notes -->
99 <h3 class="text-lg font-semibold mt-6 mb-3 text-center font-bold">Evaluation</h3>
100 <div class="border border-gray-300 h-28 p-2">{{ $cv->cv_evaluation }}</div>
101 </div>
102
103 <div class="flex justify-center m-10">
104 <a href="{{ url('/cv/' . $cv->cv_code . '/pdf') }}" target="_blank"
105 class="bg-green-600 text-white px-3 py-2 rounded">
106 <i class="fa fa-file-pdf"></i> View PDF
107 </a>
108 </div>
109 </div>
110
111
112
113
```

Figure: 11.6.6. _view-cv.blade.php file

Job View Layout File

```
resources > views > components > _view-job.blade.php > div.max-w-4xl.mx-auto.bg-white.shadow-lg.rounded-xl.p-6
1 <div class="max-w-4xl mx-auto bg-white shadow-lg rounded-xl p-6">
2
3 <div class="w-10">
4
5 <header class="text-center mb-10">
6
7 <h2 class="text-2xl font-bold uppercase mb-1">
8 Job Information
9 </h2>
10 </header>
11
12 <div class="mb-10">
13 <table class="w-full">
14 <tbody class="border-2 border-black rounded-3xl">
15 <tr>
16 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
17 Information
18 </td>
19 <td class="border border-gray-700">
20 <ul class="p-5 font-bold">
21 <li>Code : <span class="text-blue-700">{{ $job -> job_code }}</span>
22 </li>
23 <li>Job Type : <span class="text-blue-700">{{ $job -> job_type }}</span>
24 </li>
25 <li>Recruitment Number : <span class="text-blue-700">{{ $job -> recruit_number }}</span>
26 </li>
27 <li>Application Deadline : <span
28 class="text-blue-700">{{ $job -> application_deadline }}</span>
29 </li>
30 <li>Availability : <span class="text-blue-700">{{ $job -> job_availability }}</span>
31 </li>
32 </ul>
33 </td>
34 </tr>
35 <tr>
36 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
37 Company Name
38 </td>
39 <td class="p-5 text-1 border border-gray-700">
40 {{ $job -> company_name }}
41 </td>
42 </tr>
43 <tr>
44 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
45 Company Website
46 </td>
47 <td class="p-5 text-1 border border-gray-700">
48 {{ $job -> company_website }}
49 </td>
50 </tr>
51 </tbody>
52 </table>
53
54 <div class="text-2xl font-bold mb-5">
55 Requirements
56 </div>
57 <table class="w-full">
58 <tbody class="border-2 border-black rounded-3xl">
59 <tr>
60 <td class="max-w-1/3 text-center text-xl font-bold bg-gray-200 border border-gray-700">
61 Japanese Level
62 </td>
63 <td class="p-5 text-1 border border-gray-700">
64 {{ $job -> required_jp_level }}
65 </td>
66 </tr>
67 </tbody>
68 </table>
69
70 <div class="flex justify-center w-10">
71 <a href="{{ url('/job/' . $job -> job_code . '/pdf') }}" target="_blank"
72 class="bg-gray-500 text-white px-2 py-2 rounded"
73 >View PDF
74 </a>
75 </div>
76
77 </div>
78
79 </div>
80
81 </div>
82
83 </div>
84
85 </div>
86
87 </div>
88
89 </div>
90
91 </div>
92
93 </div>
94
95 </div>
96
97 </div>
98
99 </div>
100
101 </div>
102
103 </div>
104
105 </div>
106
107 </div>
108 </div>
109
110 </div>
111
112 </div>
113
114 </div>
115
116 </div>
117
118 </div>
119
120 </div>
121
122 </div>
123
124 </div>
125
126 </div>
127
128 </div>
129
130 </div>
131
132 </div>
133
134 </div>
135
136 </div>
137
138 </div>
139
140 </div>
141
142 </div>
143
144 </div>
145
146 </div>
147
148 </div>
149
150 </div>
151
152 </div>
153
154 </div>
155
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
810 </div>
811
812 </div>
813
814 </div>
815
816 </div>
817
818 </div>
819
820 </div>
821
822 </div>
823
824 </div>
825
826 </div>
827
828 </div>
829
830 </div>
831
832 </div>
833
834 </div>
835
836 </div>
837
838 </div>
839
840 </div>
841
842 </div>
843
844 </div>
845
846 </div>
847
848 </div>
849
850 </div>
851
852 </div>
853
854 </div>
855
856 </div>
857
858 </div>
859
860 </div>
861
862 </div>
863
864 </div>
865
866 </div>
867
868 </div>
869
870 </div>
871
872 </div>
873
874 </div>
875
876 </div>
877
878 </div>
879
880 </div>
881
882 </div>
883
884 </div>
885
886 </div>
887
888 </div>
889
890 </div>
891
892 </div>
893
894 </div>
895
896 </div>
897
898 </div>
899
900 </div>
901
902 </div>
903
904 </div>
905
906 </div>
907
908 </div>
909
910 </div>
911
912 </div>
913
914 </div>
915
916 </div>
917
918 </div>
919
920 </div>
921
922 </div>
923
924 </div>
925
926 </div>
927
928 </div>
929
930 </div>
931
932 </div>
933
934 </div>
935
936 </div>
937
938 </div>
939
940 </div>
941
942 </div>
943
944 </div>
945
946 </div>
947
948 </div>
949
950 </div>
951
952 </div>
953
954 </div>
955
956 </div>
957
958 </div>
959
960 </div>
961
962 </div>
963
964 </div>
965
966 </div>
967
968 </div>
969
970 </div>
971
972 </div>
973
974 </div>
975
976 </div>
977
978 </div>
979
980 </div>
981
982 </div>
983
984 </div>
985
986 </div>
987
988 </div>
989
990 </div>
991
992 </div>
993
994 </div>
995
996 </div>
997
998 </div>
999
1000 </div>
```

Figure: 11.6.7. _view-job.blade.php file

11.7. Login

The files can be found in the login folder of views folder.

Login Choice Page File

```
login-choose.blade.php X login-partner.blade.php login-admin.blade.php
resources > views > login > login-choose.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="If=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"
9 | integrity="sha512-KfkfwYDsLkIlwQp6LFn18zNdLgXu9YAA1QvwINks4PhcElQSwvcyVLD9aMhXd13uQjoXtEKKNosOwaZqXge10g=="
10 | crossorigin="anonymous" referrerpolicy="no-referrer" />
11 <script src="//unpkg.com/alpinejs"></script>
12 <script src="https://cdn.tailwindcss.com"></script>
13
14 <title></title>
15
16 </head>
17
18 <body class="mb-48">
19
20 <nav class="flex font-bold bg-white text-black justify-between items-center p-5 border-b-4">
21 | <h1 class="text-2xl">
22 | | Oshigoto <span class="text-red-700">JAPAN</span>
23 | </h1>
24 </nav>
25
26 @include('components._message')
27
28 <div class="bg-gray-50 border border-gray-200 p-10 rounded max-w-lg mx-auto mt-24">
29 | <header class="text-center">
30 | | <h2 class="text-2xl font-bold uppercase mb-1 mb-5">
31 | | | Welcome to Oshigoto Japan
32 | | </h2>
33 | | <p class="text-2xl">
34 | | | Who are you?
35 | | </p>
36 | </header>
37
38 | <div class="flex justify-center">
39 | | <div class="text-center">
40 | | | <a href="/admin-login" class="w-32 mt-5 mr-2 inline-block border-2 font-bold
41 | | | border-black text-white bg-black p-3 rounded-xl uppercase hover:text-black hover:bg-white
42 | | | shadow-lg transition duration-200">
43 | | | | Admin
44 | | | </a>
45 | | </div>
46 | | <div class="text-center">
47 | | | <a href="/partner-login" class="w-32 mt-5 ml-2 inline-block border-2 font-bold
48 | | | border-black text-white bg-black p-3 rounded-xl uppercase hover:text-black hover:bg-white
49 | | | shadow-lg transition duration-200">
50 | | | | Partner
51 | | | </a>
52 | | </div>
53 | </div>
54 </div>
55
56 <footer class="fixed bottom-0 left-0 w-full flex items-center justify-start font-bold bg-black
57 | text-white h-24 mt-24 opacity-90 md:justify-center">
58 | <p class="ml-2">Copyright &copy; 2025, All Rights reserved</p>
59
60 </footer>
61 </body>
62
63 </html>
64
```

Figure: 11.7.1. login-choose.blade.php file

Partner Login Page

```
login-partner.blade.php X login-admin.blade.php X
resources > views > login > login-partner.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/css/all.min.css"
9 integrity="sha512-Ek44vW79oRn5KeOqwnYivvLvWw1D9a9wX13wJxtE8N0s0w2qgeI8g=
10 crossorigin="anonymous" referrerpolicy="no-referrer" />
11 <script src="//unpkg.com/alpinejs" /></script>
12 <script src="https://cdn.tailwindcss.com" /></script>
13
14 <title></title>
15
16 </head>
17
18 <body class="mb-48">
19
20 <nav class="flex font-bold bg-white text-black justify-between items-center p-5 border-b-4">
21 <h1 class="text-2xl">
22 Oshigoto <span class="text-red-700">JAPAN</span>
23 </h1>
24 </nav>
25
26 <a href="#" class="mt-5 ml-5 inline-block border-2 font-bold
27 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
28 shadow-lg">
29 <i class="fa-solid fa-arrow-left"></i> Back
30 </a>
31
32 <div class="bg-gray-50 border border-gray-200 p-10 rounded max-w-lg mx-auto mt-24">
33 <header class="text-center">
34 <h2 class="text-2xl font-bold uppercase mb-1">
35 Welcome to Oshigoto Japan
36 </h2>
37 <p class="mb-4">Partner Log In</p>
38 </header>
39
40 @if ($errors->any())
41 <div class="text-red-700">
42 <ul>
43 @foreach ($errors->all() as $error)
44 <li{{ $error }}</li>
45 @endforeach
46 </ul>
47 </div>
48 @endif
49
50 <form method="POST" action="/partner/authenticate">
51
52 <div class="mb-6">
53 <input type="text" class="border border-gray-200 rounded p-2 w-full" value=""
54 <label for="partner_email" class="inline-block text-lg mb-2">
55 Email
56 </label>
57
58 <input type="email" class="border border-gray-200 rounded p-2 w-full" value="{{ old('email') }}"
59 <label for="email" class="border border-gray-200 rounded p-2 w-full" value="{{ old('email') }}"
60 <input type="password" class="border border-gray-200 rounded p-2 w-full" name="password" />
61 </div>
62
63 <div class="mb-6">
64 <input type="password" class="border border-gray-200 rounded p-2 w-full" name="password" />
65 <label for="password" class="inline-block text-lg mb-2">
66 Password
67 </label>
68 </div>
69
70 <div class="mb-6 text-center">
71 <button type="submit" class="transition duration-200 inline-block border-2 font-bold border-black
72 text-black py-2 px-4 rounded-xl uppercase hover:bg-black hover:text-white shadow-lg">
73 Log In
74 </button>
75 </div>
76 </form>
77
78 </div>
79
80 <div class="fixed bottom-0 left-0 w-full flex items-center justify-start font-bold bg-black
81 text-white h-24 mt-24 opacity-90 md:justify-center">
82 <p class="ml-2">Copyright ©copy, 2025, All Rights reserved.</p>
83 </div>
84
85 </body>
86
87 </html>
88
89
```

Figure: 11.7.2. login-partner.blade.php file

Admin Login Page

```
login-admin.blade.php X login-admin.blade.php X
resources > views > login > login-admin.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/css/all.min.css"
9 integrity="sha512-Ek44vW79oRn5KeOqwnYivvLvWw1D9a9wX13wJxtE8N0s0w2qgeI8g=
10 crossorigin="anonymous" referrerpolicy="no-referrer" />
11 <script src="//unpkg.com/alpinejs" /></script>
12 <script src="https://cdn.tailwindcss.com" /></script>
13
14 <title></title>
15
16 </head>
17
18 <body class="mb-48">
19
20 <nav class="flex font-bold bg-white text-black justify-between items-center p-5 border-b-4">
21 <h1 class="text-2xl">
22 Oshigoto <span class="text-red-700">JAPAN</span>
23 </h1>
24 </nav>
25
26 <a href="#" class="mt-5 ml-5 inline-block border-2 font-bold
27 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
28 shadow-lg">
29 <i class="fa-solid fa-arrow-left"></i> Back
30 </a>
31
32 <div class="bg-gray-50 border border-gray-200 p-10 rounded max-w-lg mx-auto mt-24">
33 <header class="text-center">
34 <h2 class="text-2xl font-bold uppercase mb-1">
35 Welcome to Oshigoto Japan
36 </h2>
37 <p class="mb-4">Admin Log In</p>
38 </header>
39
40 @if ($errors->any())
41 <div class="text-red-700">
42 <ul>
43 @foreach ($errors->all() as $error)
44 <li{{ $error }}</li>
45 @endforeach
46 </ul>
47 </div>
48 @endif
49
50 <form method="POST" action="/admin/authenticate">
51
52 <div class="mb-6">
53 <input type="text" class="border border-gray-200 rounded p-2 w-full" value=""
54 <label for="email" class="inline-block text-lg mb-2">
55 Email
56 </label>
57
58 <input type="email" class="border border-gray-200 rounded p-2 w-full" name="email"
59 <input type="password" class="border border-gray-200 rounded p-2 w-full" name="password" />
60 <label for="password" class="inline-block text-lg mb-2">
61 Password
62 </label>
63 </div>
64
65 <div class="mb-6 text-center">
66 <button type="submit" class="transition duration-200 inline-block border-2 font-bold border-black
67 text-black py-2 px-4 rounded-xl uppercase hover:bg-black hover:text-white shadow-lg">
68 Log In
69 </button>
70 </div>
71 </form>
72
73 </div>
74
75 <div class="fixed bottom-0 left-0 w-full flex items-center justify-start font-bold bg-black
76 text-white h-24 mt-24 opacity-90 md:justify-center">
77 <p class="ml-2">Copyright ©copy, 2025, All Rights reserved.</p>
78 </div>
79
80 </body>
81
82 </html>
83
84
```

Figure: 11.7.3. login-admin.blade.php file

11.8. Partner

The Partner files can be found in the Partner Folder in Views Folder.

Partner Home Page

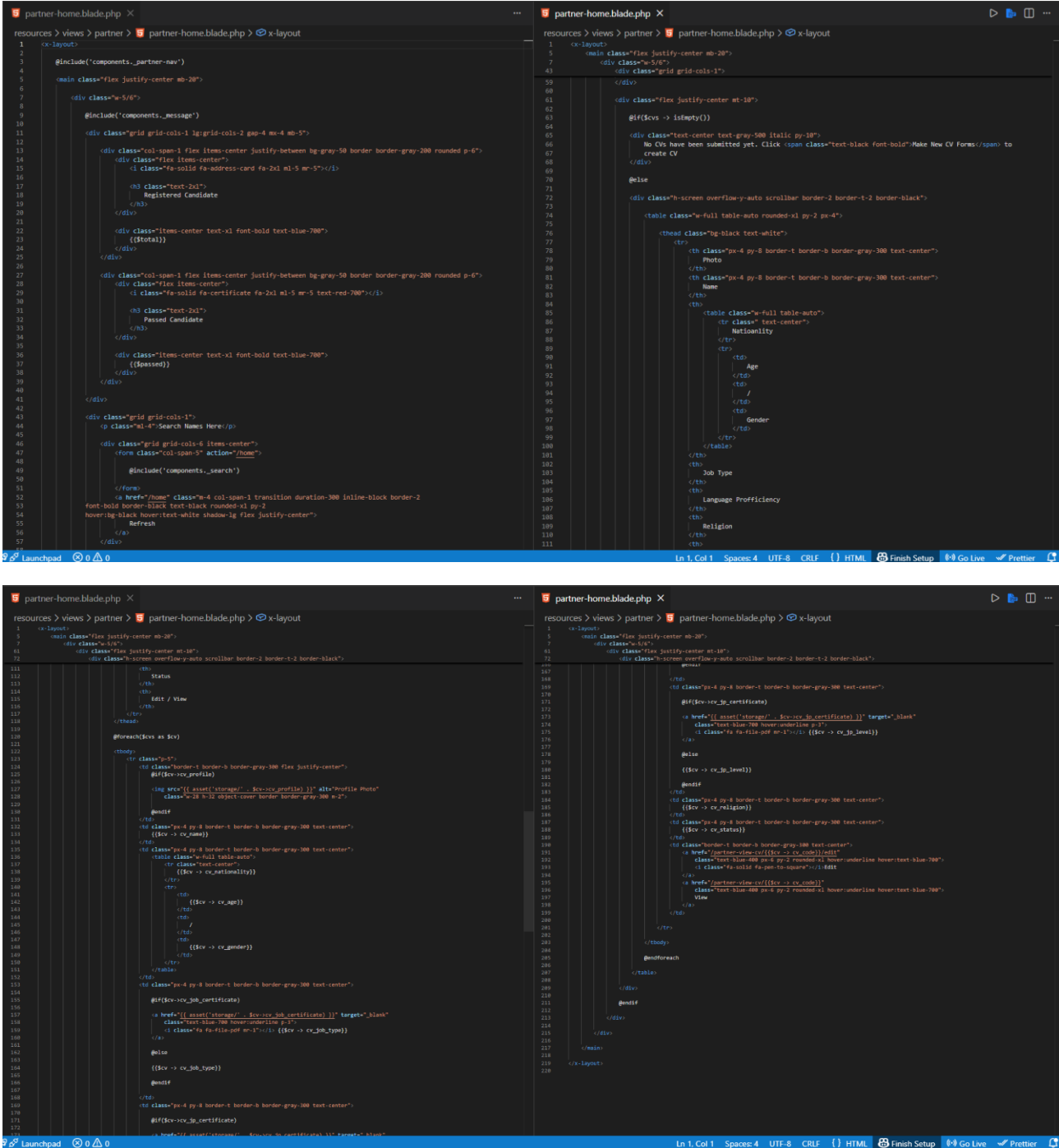


Figure: 11.8.1. partner-home.blade.php

Partner side New CV Registration Page

```

resources > views > partner > partner-new-cv-register.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w-3/4
1 <x-layout>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-3/4">
6
7 <a href="/home" class="m-5 inline-block border-2 font-bold
8 border-black text-white bg-black py-1 px-2 rounded-01 uppercase hover:text-white
9 shadow-lg">
10 <i class="fa-solid fa-arrow-left"/> Back to Main Page
11 </a>
12
13 <div class="m-4">
14
15 @if(session('message'))
16 <div class="text-red-500 text-xs mt-1">{{session('message')}}</div>
17 @endif
18
19 <div class="bg-gray-50 border border-gray-200 p-10 rounded mt-10">
20
21 <header class="text-center mb-10">
22
23 <h2 class="text-2xl font-bold uppercase mb-1">
24 Register New CV
25 </h2>
26 <p class="m-4">Register for Job Application</p>
27
28 </header>
29
30 <form method="POST" action="/home" enctype="multipart/form-data">
31
32 @csrf
33
34 <!-- General Information -->
35 <div class="mb-6">
36 <div class="text-xl font-bold mb-6 uppercase text-center">
37 General Information
38 </div>
39 </div>
40
41 <div class="mb-6">
42
43 <label for="cv_profile" class="font-bold text-lg">
44 Candidate Photo (JPG/PNG)
45 </label>
46 <input type="file" name="cv_profile" accept=".jpg,.png"
47 class="border border-gray-200 rounded p-2 w-full">
48 </div>
49
50 <div class="mb-6">
51
52 <label for="cv_name" class="inline-block text-lg mb-2">
53 Candidate Name
54 </label>
55 <input type="text" class="border border-gray-20 rounded p-2 w-full" name="cv_name"
56 value="{{old('cv_name')}}"/>
57
58
59
60
61
62
63
64
65
66 <div class="mb-6">
67 <label for="cv_gender" class="inline-block text-lg mb-2">
68 Gender
69 </label>
70
71 <select name="cv_gender" id="cv_gender"
72 class="border border-gray-200 rounded p-2 w-full" required>
73
74 <option value=""--> Select One --</option>
75 <option value="MALE" @selected(old('cv_gender'))=="MALE">MALE</option>
76 <option value="FEMALE" @selected(old('cv_gender'))=="FEMALE">FEMALE</option>
77 </select>
78
79 </div>
80
81
82 <div class="mb-6">
83
84 <label for="cv_age" class="inline-block text-lg mb-2">
85 Candidate Age
86 </label>
87 <input type="number" class="border border-gray-200 rounded p-2 w-full" name="cv_age"
88 value="{{old('cv_age')}}"/>
89
90 <error('cv_age')
91 @ class="text-red-500 text-xs mt-1">{{session('message')}}</error>
92 </div>
93
94
95 <div class="mb-6">
96
97 <label for="cv_dob" class="inline-block text-lg mb-2">
98 Candidate Date of Birth
99 </label>
100 <input type="date" class="border border-gray-200 rounded p-2 w-full" name="cv_dob"
101 value="{{old('cv_dob')}}"/>
102
103 <error('cv_dob')
104 @ class="text-red-500 text-xs mt-1">{{session('message')}}</error>
105 </div>
106
107 </div>
108
109
110
111 <div class="mb-6">
112 <label for="cv_nationality" class="inline-block text-lg mb-2">
113 Nationality
114 </label>
115 <select name="cv_nationality" id="cv_nationality"
116 class="border border-gray-200 rounded p-2 w-full" required>
117
118 <option value=""--> Select Nationality --</option>
119 <option value="Myanmar" @selected(old('cv_nationality'))=="Myanmar">Myanmar</option>
120 <option value="Philippines" @selected(old('cv_nationality'))=="Philippines">
121 Philippines</option>
122 <option value="Thailand" @selected(old('cv_nationality'))=="Thailand">Thailand
123 </option>
124 <option value="Vietnam" @selected(old('cv_nationality'))=="Vietnam">Vietnam
125 </option>
126 <option value="Indonesia" @selected(old('cv_nationality'))=="Indonesia">Indonesia
127 </option>
128 <option value="Cambodia" @selected(old('cv_nationality'))=="Cambodia">Cambodia
129 </option>
130 <option value="Laos" @selected(old('cv_nationality'))=="Laos">Laos</option>
131 </select>
132
133 </div>
134
135 <div class="mb-6">
136
137 <label for="cv_address" class="inline-block text-lg mb-2">
138 Address
139 </label>
140 <input type="text" class="border border-gray-200 rounded p-2 w-full" name="cv_address"
141 value="{{old('cv_address')}}"/>
142
143 <error('cv_address')
144 @ class="text-red-500 text-xs mt-1">{{session('message')}}</error>
145 </div>
146
147
148
149
150 <div class="mb-6">
151
152 <label for="cv_religion" class="inline-block text-lg mb-2">
153 Religion
154 </label>
155 <select name="cv_religion" id="cv_religion"
156 class="border border-gray-200 rounded p-2 w-full" required>
157
158 <option value=""--> Select Religion --</option>
159 <option value="BUDDHIST" @selected(old('cv_religion'))=="BUDDHIST">BUDDHIST
160 </option>
161 <option value="CHRISTIAN" @selected(old('cv_religion'))=="CHRISTIAN">CHRISTIAN
162 </option>
163 <option value="ISLAM" @selected(old('cv_religion'))=="ISLAM">ISLAM</option>
164 <option value="HINDU" @selected(old('cv_religion'))=="HINDU">HINDU</option>
165 <option value="Other" @selected(old('cv_religion'))=="Other">Other</option>
166 </select>
167
168 </div>
169
170 <div class="mb-6">
171
172 <label for="cv_marriage" class="inline-block text-lg mb-2">
173 Marital Status
174 </label>
175 <select name="cv_marriage" id="cv_marriage"
176 class="border border-gray-200 rounded p-2 w-full" required>
177
178 <option value=""--> Select Marital Status --</option>
179 <option value="SINGLE" @selected(old('cv_marriage'))=="SINGLE">SINGLE</option>
180 <option value="MARRIED" @selected(old('cv_marriage'))=="MARRIED">MARRIED</option>
181 <option value="MARRIED_WITH_CHILD"
182 @selected(old('cv_marriage'))=="MARRIED_WITH_CHILD">
183 MARRIED WITH CHILD</option>
184 </select>
185
186 </div>
187
188 <div class="mb-6">
189
190 <label for="cv_job_type" class="inline-block text-lg mb-2">
191 Job Type
192 </label>
193 <select name="cv_job_type" id="cv_job_type"
194 class="border border-gray-200 rounded p-2 w-full" required>
195
196 <option value=""--> Select Job Type --</option>
197 <option value="RESTAURANT" @selected(old('cv_job_type'))=="RESTAURANT">RESTAURANT
198 </option>
199 <option value="NURSING" @selected(old('cv_job_type'))=="NURSING">NURSING</option>
200 <option value="HOTEL" @selected(old('cv_job_type'))=="HOTEL">HOTEL</option>
201 <option value="CONSTRUCTION" @selected(old('cv_job_type'))=="CONSTRUCTION">
202 CONSTRUCTION</option>
203 <option value="MECHANICAL_PRODUCTION"
204 @selected(old('cv_job_type'))=="MECHANICAL_PRODUCTION">
205 MECHANICAL PRODUCTION</option>
206 </select>
207
208 </div>
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure: 11.8.2. partner-new-cv-register.blade.php 1


```
resources > views > partner > partner-new-cv-register.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w-3/4
1 <!-- layout -->
2 <div class="flex justify-center mb-20">
3   <div class="w-3/4">
4     <div class="mb-4">
5       <div class="bg-gray-50 border border-gray-200 p-10 rounded mt-10">
6         </div>
7       </div>
8     </div>
9     <div class="font-bold text-lg uppercase mb-4">Work History 2 (/2)</div>
10    <div class="mb-6">
11      <label for="cv_jobhistory2_name" class="inline-block text-lg mb-2">
12        Company Name
13      </label>
14      <input type="text" class="border border-gray-200 rounded p-2 w-full"
15        name="cv_jobhistory2_name" value="{{old('cv_jobhistory2_name')}}"/>
16    </div>
17    <div class="mb-6">
18      <label for="cv_jobhistory2_position" class="inline-block text-lg mb-2">
19        Position
20      </label>
21      <input type="text" class="border border-gray-200 rounded p-2 w-full"
22        name="cv_jobhistory2_position" value="{{old('cv_jobhistory2_position')}}"/>
23    </div>
24    <div class="mb-6">
25      <label for="cv_jobhistory2_start" class="inline-block text-lg mb-2">
26        Start Date
27      </label>
28      <input type="date" class="border border-gray-200 rounded p-2 w-full"
29        name="cv_jobhistory2_start" value="{{old('cv_jobhistory2_start')}}"/>
30    </div>
31    <div class="mb-6">
32      <label for="cv_jobhistory2_end" class="inline-block text-lg mb-2">
33        End Date
34      </label>
35      <input type="date" class="border border-gray-200 rounded p-2 w-full"
36        name="cv_jobhistory2_end" value="{{old('cv_jobhistory2_end')}}"/>
37    </div>
38    <div class="mb-6">
39      <label for="cv_jobhistory2_status" class="inline-block text-lg mb-2">
40        Status
41      </label>
42      <select name="cv_jobhistory2_status" id="cv_jobhistory2_status"
43        class="border border-gray-200 rounded p-2 w-full">
44        <option value=""--> Select Status </option>
45        <option value="Working" @selected(old('cv_jobhistory2_status')== 'Working')>
46          Working
47        </option>
48        <option value="Left" @selected(old('cv_jobhistory2_status')== 'Left')>Left
49        </option>
50      </select>
51    </div>
52    <div class="font-bold text-lg uppercase mb-4">Work History 3 (/2)</div>
53    <div class="mb-6">
54      <label for="cv_jobhistory3_name" class="inline-block text-lg mb-2">
55        Company Name
56      </label>
57      <input type="text" class="border border-gray-200 rounded p-2 w-full"
58        name="cv_jobhistory3_name" value="{{old('cv_jobhistory3_name')}}"/>
59    </div>
60    <div class="mb-6">
61      <label for="cv_jobhistory3_position" class="inline-block text-lg mb-2">
62        Position
63      </label>
64      <input type="text" class="border border-gray-200 rounded p-2 w-full"
65        name="cv_jobhistory3_position" value="{{old('cv_jobhistory3_position')}}"/>
66    </div>
67    <div class="mb-6">
68      <label for="cv_jobhistory3_start" class="inline-block text-lg mb-2">
69        Start Date
70      </label>
71      <input type="date" class="border border-gray-200 rounded p-2 w-full"
72        name="cv_jobhistory3_start" value="{{old('cv_jobhistory3_start')}}"/>
73    </div>
74    <div class="mb-6">
75      <label for="cv_jobhistory3_end" class="inline-block text-lg mb-2">
76        End Date
77      </label>
78      <input type="date" class="border border-gray-200 rounded p-2 w-full"
79        name="cv_jobhistory3_end" value="{{old('cv_jobhistory3_end')}}"/>
80    </div>
81  </div>
82 </div>
83 </div>
```

```
resources > views > partner > partner-new-cv-register.blade.php > x-layout > div.flex.justify-center.mb-20 > div.w-3/4
727 <div class="font-bold text-lg uppercase mb-4">Work History 3 (/2)</div>
728 <div class="mb-6">
729   <label for="cv_jobhistory3_status" class="inline-block text-lg mb-2">
730     Status
731   </label>
732   <select name="cv_jobhistory3_status" id="cv_jobhistory3_status"
733     class="border border-gray-200 rounded p-2 w-full">
734     <option value=""--> Select Status </option>
735     <option value="Working" @selected(old('cv_jobhistory3_status')== 'Working')>
736       Working
737     </option>
738     <option value="Left" @selected(old('cv_jobhistory3_status')== 'Left')>Left
739     </option>
740   </select>
741 </div>
742 </div>
743 <div class="font-bold text-xl uppercase mb-6 text-center">
744   <h1>Others</h1>
745 </div>
746 <div class="mb-6">
747   <label for="cv_email" class="inline-block text-lg mb-2">
748     Email Address
749   </label>
750   <input type="email" class="border border-gray-200 rounded p-2 w-full"
751     name="cv_email" value="{{old('cv_email')}}"/>
752 </div>
753 <div class="mb-6">
754   <label for="cv_phone" class="inline-block text-lg mb-2">
755     Phone Number
756   </label>
757   <input type="tel" class="border border-gray-200 rounded p-2 w-full"
758     name="cv_phone" value="{{old('cv_phone')}}"/>
759 </div>
760 <div class="font-bold text-lg uppercase mt-10 mb-4">
761   <h2>Passport (/2)</h2>
762 </div>
763 <div class="mb-6">
764   <label for="cv_passport_number" class="inline-block text-lg mb-2">
765     Passport Number
766   </label>
767   <input type="text" class="border border-gray-200 rounded p-2 w-full"
768     name="cv_passport_number" value="{{old('cv_passport_number')}}"/>
769 </div>
770 <div class="mb-6">
771   <label for="cv_passport_date" class="inline-block text-lg mb-2">
772     Expiration Date
773   </label>
774   <input type="date" class="border border-gray-200 rounded p-2 w-full"
775     name="cv_passport_date" value="{{old('cv_passport_date')}}"/>
776 </div>
777 <div class="mb-6">
778   <label for="cv_passport_file" class="inline-block text-lg mb-2">
779     Passport File (PDF)
780   </label>
781   <input type="file" name="cv_passport_accept" class="border border-gray-200 rounded p-2 w-full"
782   </input>
783 </div>
784 <div class="font-bold text-lg uppercase mt-10 mb-4">
785   <h2>Japan (/2)</h2>
786 </div>
787 <div class="mb-6">
788   <label for="cv_visit_time" class="inline-block text-lg mb-2">
789     Number of Times of Going Japan
790   </label>
791 </div>
```

Figure: 11.8.5. partner-new-cv-register.blade.php 4


```

partner-edit-cv.blade.php X
resources > views > partner > partner-edit-cv.blade.php > x-layout
1 <x-layout>
2 <div class="flex justify-center mb-20">
3 <div class="w-3/4">
4 <div class="m-4">
5 <div class="bg-gray-50 border border-gray-200 p-5 rounded-at-24">
6
7 <div class="mb-6">
8 <label for="cv_schoolhistory3_end" class="inline-block text-lg mb-2">
9 End Date
10 </label>
11 <input type="date" class="border border-gray-200 rounded p-2 w-full"
12 name="cv_schoolhistory3_end" value="{{ $cv->cv_schoolhistory3_end }}" />
13 </div>
14
15 <div class="mb-6">
16 <label for="cv_schoolhistory3_status" class="inline-block text-lg mb-2">
17 Status
18 </label>
19
20 <select name="cv_schoolhistory3_status" id="cv_schoolhistory3_status"
21 class="border border-gray-200 rounded p-2 w-full">
22 <option value="{{ $cv->cv_schoolhistory3_status }}">
23 {{ $cv->cv_schoolhistory3_status }} </option>
24 <option value="Studying" @selected(old($cv_schoolhistory3_status)=='Studying'
25 )>
26 Studying </option>
27 <option value="Graduated" @selected(old($cv_schoolhistory3_status)=='Graduated'
28 )>
29 Graduated </option>
30 <option value="left" @selected(old($cv_schoolhistory3_status)=='left' )>left
31 </option>
32 </select>
33 </div>
34
35 </div>
36 </div>
37
38 <!-- Work History -->
39 <div class="m-6 mt-10">
40
41 <div class="font-bold text-xl uppercase mb-6 text-center">
42 <h2>Work History</h2>
43 </div>
44
45 <div class="mt-10">
46 <h2 class="font-bold text-lg uppercase mb-4">Work History 1</h2>
47
48 <div class="mb-6">
49 <label for="cv_jobhistory1_name" class="inline-block text-lg mb-2">
50

```

```

partner-edit-cv.blade.php X
resources > views > partner > partner-edit-cv.blade.php > x-layout
1 <x-layout>
2 <div class="flex justify-center mb-20">
3 <div class="w-3/4">
4 <div class="m-4">
5 <div class="bg-gray-50 border border-gray-200 p-5 rounded-at-24">
6
7 <div class="mb-6">
8 <input type="text" class="border border-gray-200 rounded p-2 w-full"
9 name="cv_jobhistory1_name" value="{{ $cv->cv_jobhistory1_name }}" />
10 </div>
11
12 <div class="mb-6">
13 <label for="cv_jobhistory1_position" class="inline-block text-lg mb-2">
14 Position
15 </label>
16 <input type="text" class="border border-gray-200 rounded p-2 w-full"
17 name="cv_jobhistory1_position" value="{{ $cv->cv_jobhistory1_position }}" />
18 </div>
19
20 <div class="mb-6">
21 <label for="cv_jobhistory1_start" class="inline-block text-lg mb-2">
22 Start Date
23 </label>
24 <input type="date" class="border border-gray-200 rounded p-2 w-full"
25 name="cv_jobhistory1_start" value="{{ $cv->cv_jobhistory1_start }}" />
26 </div>
27
28 <div class="mb-6">
29 <label for="cv_jobhistory1_end" class="inline-block text-lg mb-2">
30 End Date
31 </label>
32 <input type="date" class="border border-gray-200 rounded p-2 w-full"
33 name="cv_jobhistory1_end" value="{{ $cv->cv_jobhistory1_end }}" />
34 </div>
35
36 <div class="mb-6">
37 <label for="cv_jobhistory1_status" class="inline-block text-lg mb-2">
38 Status
39 </label>
40
41 <select name="cv_jobhistory1_status" id="cv_jobhistory1_status"
42 class="border border-gray-200 rounded p-2 w-full">
43 <option value="{{ $cv->cv_jobhistory1_status }}">
44 {{ $cv->cv_jobhistory1_status }} </option>
45 <option value="Working" @selected(old($cv_jobhistory1_status)=='Working' )>
46 Working </option>
47 <option value="left" @selected(old($cv_jobhistory1_status)=='left' )>left
48 </option>
49 </select>
50 </div>
51
52 </div>
53
54 <div class="mt-10">
55 <h2 class="font-bold text-lg uppercase mb-4">Work History 3</h2>
56
57 <div class="mb-6">
58 <label for="cv_jobhistory3_name" class="inline-block text-lg mb-2">
59 Company Name
60 </label>
61 <input type="text" class="border border-gray-200 rounded p-2 w-full"
62 name="cv_jobhistory3_name" value="{{ $cv->cv_jobhistory3_name }}" />
63 </div>
64
65 <div class="mb-6">
66 <label for="cv_jobhistory3_position" class="inline-block text-lg mb-2">
67 Position
68 </label>
69 <input type="text" class="border border-gray-200 rounded p-2 w-full"
70 name="cv_jobhistory3_position" value="{{ $cv->cv_jobhistory3_position }}" />
71 </div>
72
73 <div class="mb-6">
74 <label for="cv_jobhistory3_start" class="inline-block text-lg mb-2">
75 Start Date
76 </label>
77 <input type="date" class="border border-gray-200 rounded p-2 w-full"
78 name="cv_jobhistory3_start" value="{{ $cv->cv_jobhistory3_start }}" />
79 </div>
80
81 <div class="mb-6">
82 <label for="cv_jobhistory3_end" class="inline-block text-lg mb-2">
83 End Date
84 </label>
85 <input type="date" class="border border-gray-200 rounded p-2 w-full"
86 name="cv_jobhistory3_end" value="{{ $cv->cv_jobhistory3_end }}" />
87 </div>
88
89 <div class="mb-6">
90 <label for="cv_jobhistory3_status" class="inline-block text-lg mb-2">
91 Status
92 </label>
93
94 <select name="cv_jobhistory3_status" id="cv_jobhistory3_status"
95 class="border border-gray-200 rounded p-2 w-full">
96 <option value="{{ $cv->cv_jobhistory3_status }}">
97 {{ $cv->cv_jobhistory3_status }} </option>
98 <option value="Working" @selected(old($cv_jobhistory3_status)=='Working' )>
99 Working </option>
100 <option value="left" @selected(old($cv_jobhistory3_status)=='left' )>left
101 </option>
102 </select>
103 </div>
104
105 </div>
106
107 <div class="mt-10">
108 <h2 class="font-bold text-lg uppercase mb-4">Work History 3</h2>
109
110 <div class="mb-6">
111 <label for="cv_jobhistory3_name" class="inline-block text-lg mb-2">
112 Company Name
113 </label>
114 <input type="text" class="border border-gray-200 rounded p-2 w-full"
115 name="cv_jobhistory3_name" value="{{ $cv->cv_jobhistory3_name }}" />
116 </div>
117
118 <div class="mb-6">
119 <label for="cv_jobhistory3_position" class="inline-block text-lg mb-2">
120 Position
121 </label>
122 <input type="text" class="border border-gray-200 rounded p-2 w-full"
123 name="cv_jobhistory3_position" value="{{ $cv->cv_jobhistory3_position }}" />
124 </div>
125
126 <div class="mb-6">
127 <label for="cv_jobhistory3_start" class="inline-block text-lg mb-2">
128 Start Date
129 </label>
130 <input type="date" class="border border-gray-200 rounded p-2 w-full"
131 name="cv_jobhistory3_start" value="{{ $cv->cv_jobhistory3_start }}" />
132 </div>
133
134 <div class="mb-6">
135 <label for="cv_jobhistory3_end" class="inline-block text-lg mb-2">
136 End Date
137 </label>
138 <input type="date" class="border border-gray-200 rounded p-2 w-full"
139 name="cv_jobhistory3_end" value="{{ $cv->cv_jobhistory3_end }}" />
140 </div>
141
142 <div class="mb-6">
143 <label for="cv_jobhistory3_status" class="inline-block text-lg mb-2">
144 Status
145 </label>
146
147 <select name="cv_jobhistory3_status" id="cv_jobhistory3_status"
148 class="border border-gray-200 rounded p-2 w-full">
149 <option value="{{ $cv->cv_jobhistory3_status }}">
150 {{ $cv->cv_jobhistory3_status }} </option>
151 <option value="Working" @selected(old($cv_jobhistory3_status)=='Working' )>
152 Working </option>
153 <option value="left" @selected(old($cv_jobhistory3_status)=='left' )>left
154 </option>
155 </select>
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163

```

Figure: 11.8.10. partner-edit-cv.blade.php 4

```

partner-edit-cv.blade.php X
resources > views > partner > partner-edit-cv.blade.php > x-layout
1 <x-layout>
2 <div class="flex justify-center mb-20">
3 <div class="w-3/4">
4 <div class="mb-4">
5 <div class="bg-gray-50 border border-gray-200 p-5 rounded mt-24">
6
778 <label>
779 Start Date
780 <input type="date" class="border border-gray-200 rounded p-2 w-full"
781 name="cv_jobhistory3_start" value="{{ $cv->cv_jobhistory3_start }}" />
782 </div>
783 </div>
784 <div class="mb-6">
785 <label form="cv_jobhistory3_end" class="inline-block text-lg mb-2">
786 End Date
787 <input type="date" class="border border-gray-200 rounded p-2 w-full"
788 name="cv_jobhistory3_end" value="{{ $cv->cv_jobhistory3_end }}" />
789 </div>
790 <div class="mb-6">
791 <label form="cv_jobhistory3_status" class="inline-block text-lg mb-2">
792 Status
793 <select name="cv_jobhistory3_status" id="cv_jobhistory3_status"
794 class="border border-gray-200 rounded p-2 w-full">
795 <option value="{{ $cv->cv_jobhistory3_status }}">
796 {{ $cv->cv_jobhistory3_status }}
797 </option>
798 <option value="Working" @selected(old("cv_jobhistory3_status")== "Working")>
799 Working
800 </option>
801 <option value="Left" @selected(old("cv_jobhistory3_status")== "Left")>
802 Left
803 </option>
804 </select>
805 </div>
806 </div>
807 <div class="mb-6">
808 <div class="font-bold text-xl uppercase mb-6 text-center">
809 Others
810 </div>
811 <div class="mb-6">
812 <div class="font-bold text-xl uppercase mb-6 text-center">
813 Others
814 </div>
815 </div>
816 </div>
817 </div>
818 </div>
819 </div>
820 </div>
821 </div>
822 </div>
823 </div>
824 </div>
825 </div>
826 </div>
827 </div>
828 </div>
829 </div>
830 </div>
831 </div>
832 </div>
833 </div>
834 </div>
835 </div>
836 </div>
837 </div>
838 </div>
839 </div>
840 </div>
841 </div>
842 </div>
843 </div>
844 </div>
845 </div>
846 </div>
847 </div>
848 </div>
849 </div>
850 </div>
851 </div>
852 </div>
853 </div>
854 </div>
855 </div>
856 </div>
857 </div>
858 </div>
859 </div>
860 </div>
861 </div>
862 </div>
863 </div>
864 </div>
865 </div>
866 </div>
867 </div>
868 </div>
869 </div>
870 </div>
871 </div>
872 </div>
873 </div>
874 </div>
875 </div>
876 </div>
877 </div>
878 </div>
879 </div>
880 </div>
881 </div>
882 </div>
883 </div>
884 </div>
885 </div>
886 </div>
887 </div>
888 </div>
889 </div>
890 </div>
891 </div>
892 </div>
893 </div>
894 </div>
895 </div>
896 </div>
897 </div>
898 </div>
899 </div>
900 </div>
901 </div>
902 </div>
903 </div>
904 </div>
905 </div>
906 </div>
907 </div>
908 </div>
909 </div>
910 </div>
911 </div>
912 </div>
913 </div>
914 </div>
915 </div>
916 </div>
917 </div>
918 </div>
919 </div>
920 </div>
921 </div>
922 </div>
923 </div>
924 </div>
925 </div>
926 </div>
927 </div>
928 </div>
929 </div>
930 </div>
931 </div>
932 </div>
933 </div>
934 </div>
935 </div>
936 </div>
937 </div>
938 </div>
939 </div>
940 </div>
941 </div>
942 </div>
943 </div>
944 </div>
945 </div>
946 </div>
947 </div>
948 </div>
949 </div>
950 </div>
951 </div>
952 </div>
953 </div>
954 </div>
955 </div>
956 </div>
957 </div>
958 </div>
959 </div>
960 </div>
961 </div>
962 </div>
963 </div>
964 </div>
965 </div>
966 </div>
967 </div>
968 </div>
969 </div>
970 </div>
971 </div>
972 </div>
973 </div>
974 </div>
975 </div>
976 </div>
977 </div>
978 </div>
979 </div>
980 </div>
981 </div>
982 </div>
983 </div>
984 </div>
985 </div>
986 </div>
987 </div>
988 </div>
989 </div>
990 </div>
991 </div>
992 </div>
993 </div>
994 </div>
995 </div>
996 </div>
997 </div>
998 </div>
999 </div>
1000 </div>

```

```

partner-edit-cv.blade.php X
resources > views > partner > partner-edit-cv.blade.php > x-layout
1 <x-layout>
2 <div class="flex justify-center mb-20">
3 <div class="w-3/4">
4 <div class="mb-4">
5 <div class="bg-gray-50 border border-gray-200 p-5 rounded mt-24">
6
831 <div class="font-bold text-xl uppercase mb-6 text-center">
832 Others
833 </div>
834 <div class="font-bold text-xl uppercase mb-6 text-center">
835 Others
836 </div>
837 </div>
838 </div>
839 </div>
840 </div>
841 </div>
842 </div>
843 </div>
844 </div>
845 </div>
846 </div>
847 </div>
848 </div>
849 </div>
850 </div>
851 </div>
852 </div>
853 </div>
854 </div>
855 </div>
856 </div>
857 </div>
858 </div>
859 </div>
860 </div>
861 </div>
862 </div>
863 </div>
864 </div>
865 </div>
866 </div>
867 </div>
868 </div>
869 </div>
870 </div>
871 </div>
872 </div>
873 </div>
874 </div>
875 </div>
876 </div>
877 </div>
878 </div>
879 </div>
880 </div>
881 </div>
882 </div>
883 </div>
884 </div>
885 </div>
886 </div>
887 </div>
888 </div>
889 </div>
890 </div>
891 </div>
892 </div>
893 </div>
894 </div>
895 </div>
896 </div>
897 </div>
898 </div>
899 </div>
900 </div>
901 </div>
902 </div>
903 </div>
904 </div>
905 </div>
906 </div>
907 </div>
908 </div>
909 </div>
910 </div>
911 </div>
912 </div>
913 </div>
914 </div>
915 </div>
916 </div>
917 </div>
918 </div>
919 </div>
920 </div>
921 </div>
922 </div>
923 </div>
924 </div>
925 </div>
926 </div>
927 </div>
928 </div>
929 </div>
930 </div>
931 </div>
932 </div>
933 </div>
934 </div>
935 </div>
936 </div>
937 </div>
938 </div>
939 </div>
940 </div>
941 </div>
942 </div>
943 </div>
944 </div>
945 </div>
946 </div>
947 </div>
948 </div>
949 </div>
950 </div>
951 </div>
952 </div>
953 </div>
954 </div>
955 </div>
956 </div>
957 </div>
958 </div>
959 </div>
960 </div>
961 </div>
962 </div>
963 </div>
964 </div>
965 </div>
966 </div>
967 </div>
968 </div>
969 </div>
970 </div>
971 </div>
972 </div>
973 </div>
974 </div>
975 </div>
976 </div>
977 </div>
978 </div>
979 </div>
980 </div>
981 </div>
982 </div>
983 </div>
984 </div>
985 </div>
986 </div>
987 </div>
988 </div>
989 </div>
990 </div>
991 </div>
992 </div>
993 </div>
994 </div>
995 </div>
996 </div>
997 </div>
998 </div>
999 </div>
1000 </div>

```

Figure: 11.8.11. partner-edit-cv.blade.php 5

Partner side Job Selection Page

```

1  partner-all-jobs.blade.php
2  out:
3
4  <div class="flex justify-center mb-20">
5
6
7  <div class="w-5/6">
8
9  <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
10  <div class="w-1/4"> Search Jobs with Job Type, Language Level or Job Title </div>
11
12  <div class="grid grid-cols-0 items-center">
13  <form class="col-span-5" action="/apply-jobs">
14  @include('components._search')
15  </form>
16
17  <a href="/apply-jobs" class="m-4 col-span-1 transition duration-300 inline-block border-2
18  font-bold border-blue text-black rounded-xl py-2
19  hover:bg-black hover:text-white shadow-lg flex justify-center">
20  Refresh
21  </a>
22  </div>
23
24  </div>
25
26  <main>
27
28  <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
29  @foreach($jobs as $job)
30
31  <div class="bg-gray-50 border border-gray-200 rounded-p-5">
32
33  <div class="m-3">
34  <div class="text-xl font-bold m-3">
35  {{ $job -> job_code }}
36  </div>
37
38  <div class="text-2xl">
39  {{ $job -> job_title }}
40  </div>
41
42  <div class="text-xl font-bold">
43  {{ $job -> company_name }}
44  </div>
45
46  </div>
47
48  <div class="m-3">
49
50  <div class="flex">
51
52  <div class="flex items-center justify-center bg-black text-white rounded-xl py-3 px-2 text-xs max-w-24">
53  <a href="/apply-jobs/search/{{ $job -> job_type }}" {{ $job -> job_type }}</a>
54  </div>
55
56  <div class="flex items-center justify-center bg-black text-white rounded-xl py-3 px-2 text-xs max-w-24">
57  <a href="/apply-jobs/search/{{ $job -> job_type }}" {{ $job -> job_type }}</a>
58  </div>
59  </div>
60
61  </div>
62
63  </div>
64
65  </div>
66
67  <div class="m-3 grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
68  <div class="flex justify-center mb-20">
69  <div class="w-5/6">
70
71  <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
72  <div class="w-1/4"> Search Jobs with Job Type, Language Level or Job Title </div>
73
74  <div class="grid grid-cols-0 items-center">
75  <form class="col-span-5" action="/apply-jobs">
76  @include('components._search')
77  </form>
78
79  <a href="/apply-jobs" class="m-4 col-span-1 transition duration-300 inline-block border-2
80  font-bold border-blue text-black rounded-xl py-2
81  hover:bg-black hover:text-white shadow-lg flex justify-center">
82  Refresh
83  </a>
84  </div>
85
86  </div>
87
88  <main>
89
90  <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
91  @foreach($jobs as $job)
92
93  <div class="bg-gray-50 border border-gray-200 rounded-p-5">
94
95  <div class="m-3">
96  <div class="text-xl font-bold m-3">
97  {{ $job -> job_code }}
98  </div>
99
100  <div class="text-2xl">
101  {{ $job -> job_title }}
102  </div>
103  <div class="text-xl font-bold">
104  {{ $job -> company_name }}
105  </div>
106
107  </div>
108
109  <div class="m-3">
110
111  <div class="flex">
112
113  <div class="flex items-center justify-center bg-black text-white rounded-xl py-3 px-2 text-xs max-w-24">
114  <a href="/apply-jobs/search/{{ $job -> job_type }}" {{ $job -> job_type }}</a>
115  </div>
116
117  <div class="flex items-center justify-center bg-black text-white rounded-xl py-3 px-2 text-xs max-w-24">
118  <a href="/apply-jobs/search/{{ $job -> job_type }}" {{ $job -> job_type }}</a>
119  </div>
120  </div>
121
122  </div>
123
124  </div>
125
126  </div>
127
128  </div>
129
130  </div>
131

```

Figure: 11.8.12. partner-all-jobs.blade.php

Partner Side Applied Job List Page

```

1  partner-applied-job.blade.php
2  resources > views > partner > partner-applied-job.blade.php > x-layout
3
4  <div class="flex justify-center mb-20">
5  @include('components._partner.nav')
6
7  <main class="flex justify-center mb-20">
8
9  <div class="w-5/6">
10
11  <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 space-y-0 mx-4">
12  <div class="w-1/4"> Search Jobs with Job Code or Company Name </div>
13
14  <div class="grid grid-cols-0 items-center">
15  <form class="col-span-5" action="/partner-applied-list">
16  @include('components._search')
17  </form>
18
19  <a href="/partner-applied-list" class="m-4 col-span-1 transition duration-300 inline-block border-2
20  font-bold border-blue text-black rounded-xl py-2
21  hover:bg-black hover:text-white shadow-lg flex justify-center">
22  Refresh
23  </a>
24  </div>
25
26  </div>
27
28  @include('components._message')
29
30  <div class="m-5 flex flex-col items-center justify-center mt-10">
31
32  @if($jobs -> isEmpty())
33  <div class="text-center text-gray-500 italic py-10">
34  No job applications have been submitted yet. Go to <a href="/apply-jobs" class="text-black font-bold">Apply New
35  Jobs</a> to apply jobs
36  </div>
37
38  @else
39  @foreach($jobs as $job)
40
41  <div class="m-5 [<div class="w-5/6 overflow-x-scroll scrollbar border-2 border-t-2 border-black mb-10">
42  <table class="w-full table-auto rounded-xl py-2 px-4 mb-5">
43
44  <thead>
45  <tr class="bg-gray-50 text-center">
46  <th class="py-2 px-4" Job Code </th>
47  <th class="py-2 px-4 text-blue-700">{{ $job->job_code }} </th>
48  <th class="py-2 px-4" </th>
49  <th class="py-2 px-4" Company Name </th>
50  <th class="py-2 px-4 text-blue-700">{{ $job->company_name }} </th>
51  <th class="py-2 px-4 text-blue-700">{{ $job->job_availability }} </th>
52  </tr>
53  </thead>
54
55  <tbody>
56  <tr class="bg-black text-white">
57  <td class="py-4 px-3 text-center">Photo </td>
58  <td class="py-4 px-3 text-center">Name </td>
59  <td class="py-4 px-3 text-center">Nationality / Age / Gender </td>
60  <td class="py-4 px-3 text-center">Job Type </td>
61  <td class="py-4 px-3 text-center">Language </td>
62  <td class="py-4 px-3 text-center">Status </td>
63  </tr>
64  </tbody>
65  </table>
66
67  </div>
68
69  </div>
70
71  </div>
72
73  </div>
74
75  </div>
76
77  </div>
78
79  </div>
80
81  </div>
82
83  </div>
84
85  </div>
86
87  </div>
88
89  </div>
90
91  </div>
92
93  </div>
94
95  </div>
96
97  </div>
98
99  </div>
100
101  </div>
102
103  </div>
104
105  </div>
106
107  </div>
108
109  </div>
110
111

```

Figure: 11.8.13. partner-applied-job.blade.php

Partner Side Job Application Page

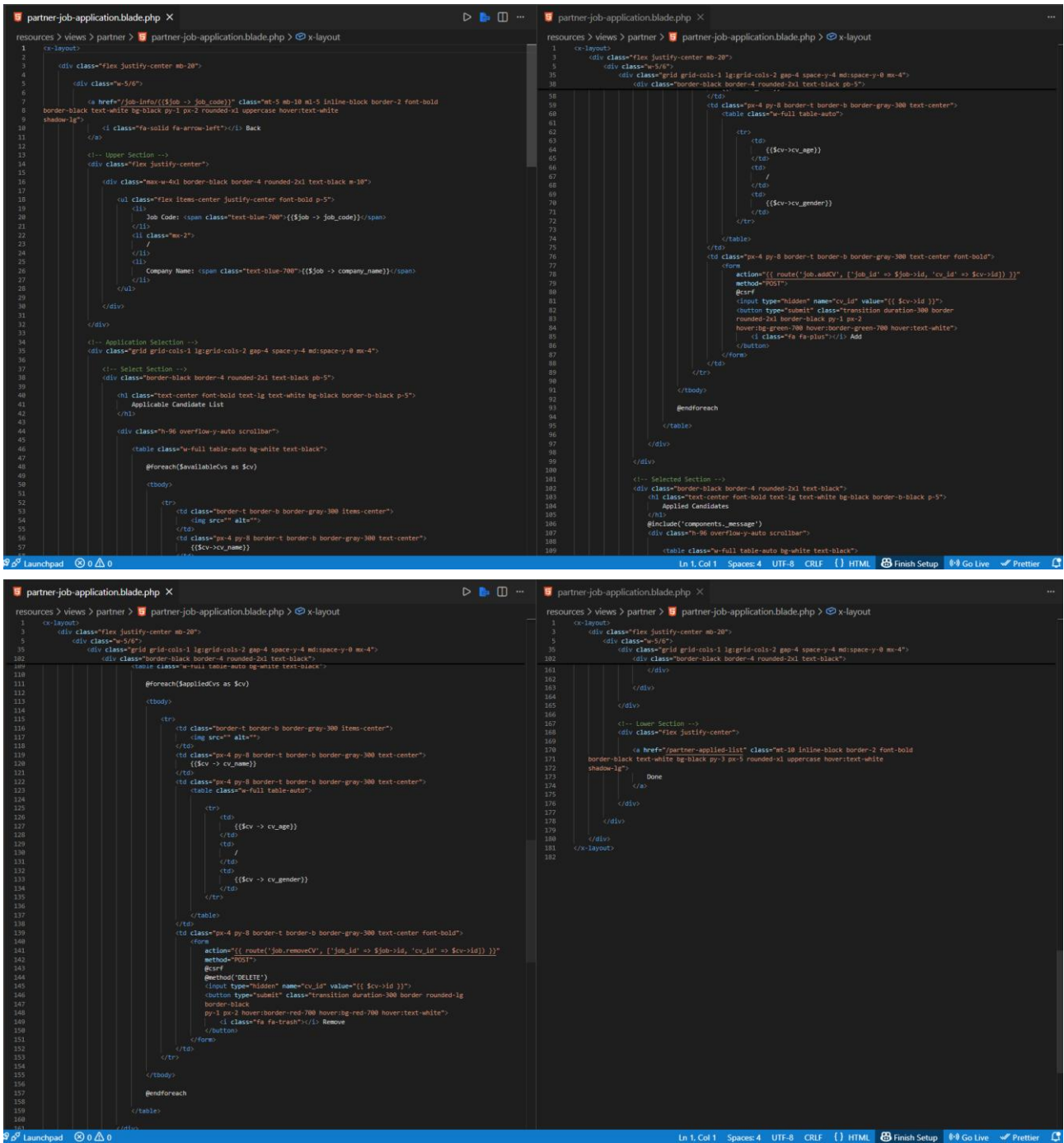
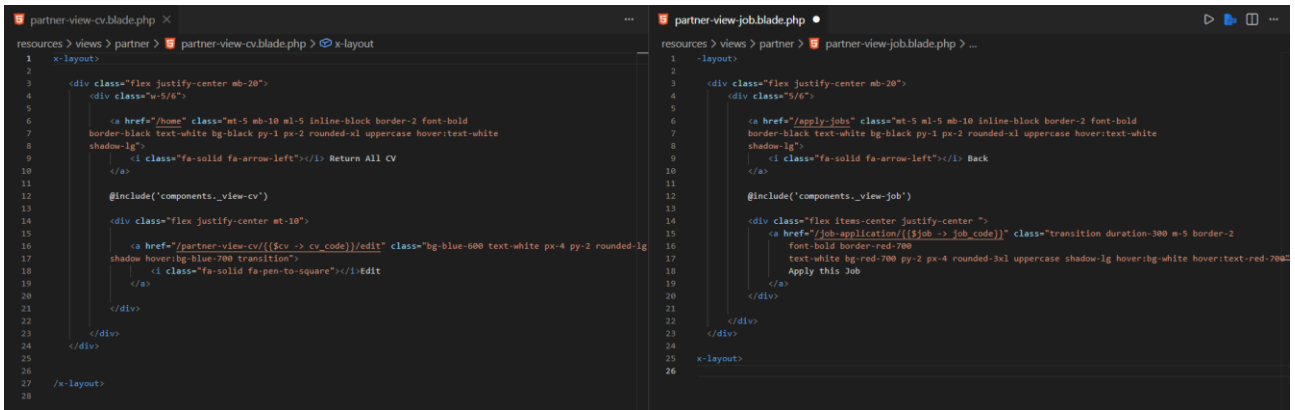


Figure: 11.8.14. partner-job-application.blade.php

Partner Side CV and Job View Page



```
partner-view-cv.blade.php
resources > views > partner > partner-view-cv.blade.php > x-layout
1 <x-layout>
2
3 <div class="flex justify-center mb-20">
4 <div class="w-5/6">
5
6 <a href="/home" class="mt-5 mb-10 w-1/5 inline-block border-2 font-bold
7 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
8 shadow-lg">
9 <i class="fa-solid fa-arrow-left"></i> Return All CV
10 </a>
11
12 @include('components._view-cv')
13
14 <div class="flex justify-center mt-10">
15
16 <a href="/partner-view-cv/{{ $cv -> cv_code }}/edit" class="bg-blue-600 text-white px-4 py-2 rounded-lg
17 shadow hover:bg-blue-700 transition">
18 <i class="fa-solid fa-pen-to-square"></i> Edit
19 </a>
20
21 </div>
22
23 </div>
24 </div>
25
26
27 </x-layout>
28

partner-view-job.blade.php
resources > views > partner > partner-view-job.blade.php > ...
1 <-layout>
2
3 <div class="flex justify-center mb-20">
4 <div class="w-5/6">
5
6 <a href="/apply-jobs" class="mt-5 mb-10 inline-block border-2 font-bold
7 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
8 shadow-lg">
9 <i class="fa-solid fa-arrow-left"></i> Back
10 </a>
11
12 @include('components._view-job')
13
14 <div class="flex items-center justify-center">
15 <a href="/job-application/{{ $job -> job_code }}" class="transition duration-300 w-5 border-2
16 font-bold border-red-700
17 text-white bg-red-700 py-2 px-4 rounded-3xl uppercase shadow-lg hover:bg-white hover:text-red-700">
18 Apply this Job
19 </a>
20 </div>
21
22 </div>
23 </div>
24
25 <x-layout>
26
```

Figure: 11.8.15. partner-view-cv.blade.php and partner-view-job.blade.php

11.9. Admin

The Company files (Admin Files) can be found in the Company Folder in Views Folder.

Admin Home Page

```
admin-home.blade.php < X
resources > views > company > admin-home.blade.php > x-layout-admin
1 layout-admin
2 <div class="flex justify-center mb-20">
3 </div>
4 <div class="m-5/6">
5
6 @include('components._message')
7
8 <div class="font-bold text-3xl mt-5 mb-10">
9 Admin Dashboard
10 </div>
11
12
13 </div>
14
15 <!-- STATISTICS SECTION 1 -->
16 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-5">
17 <div class="bg-gray-50 border border-gray-200 rounded p-5 text-center">
18 <h3 class="text-2xl font-bold mb-2">CVs Added This Month</h3>
19 <p id="monthlyCVs" class="text-3xl text-blue-600 font-bold">...</p>
20 <p class="text-sm text-gray-500">This Year: <span id="yearlyCVs">...</span></p>
21 </div>
22
23 <div class="bg-gray-50 border border-gray-200 rounded p-5 text-center">
24 <h3 class="text-2xl font-bold mb-2">Jobs Added This Month</h3>
25 <p id="monthlyJobs" class="text-3xl text-blue-600 font-bold">...</p>
26 <p class="text-sm text-gray-500">This Year: <span id="yearlyJobs">...</span></p>
27 </div>
28 </div>
29
30 <!-- STATISTICS SECTION 2 -->
31 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-5">
32 <div class="bg-gray-50 border border-gray-200 rounded p-5 text-center">
33 <h3 class="text-2xl font-bold mb-2">Partners Added This Month</h3>
34 <p id="monthlyPartners" class="text-3xl text-blue-600 font-bold">...</p>
35 <p class="text-sm text-gray-500">This Year: <span id="yearlyPartners">...</span></p>
36 </div>
37
38 <div class="bg-gray-50 border border-gray-200 rounded p-5 text-center">
39 <h3 class="text-2xl font-bold mb-2">Applications Made This Month</h3>
40 <p id="monthlyApplications" class="text-3xl text-blue-600 font-bold">...</p>
41 <p class="text-sm text-gray-500">This Year: <span id="yearlyApplications">...</span></p>
42 </div>
43 </div>
44
45 <!-- SEARCH DATE SECTION -->
46 <div class="flex flex-col justify-center mb-5 bg-gray-50 border border-gray-200 rounded p-5 text-center">
47 <form class="mb-10 action="/admin-home/date-filter" method="GET">
48 <h3 class="text-2xl font-bold mb-5">Filter by Date</h3>
49 <div class="grid grid-cols-1 mb-5">
50
51 <div class="m-5">
52 <label class="font-bold" for="start_date">Start Date </label>
53 <input type="date" name="start_date" id="">
54 </div>
55
56 <div class="m-5">
57 <label class="font-bold" for="end_date">End Date </label>
58 <input type="date" name="end_date" id="end_date">
59 </div>
60 </div>
61
62 <div>
63 <button type="submit" class="h-10 w-20 text-white rounded-lg bg-black hover:bg-gray-500">
64 Search
65 </button>
66 </div>
67 </form>
68
69 <div class="grid grid-cols-1 lg:grid-cols-2">
70 <div class="mb-5">
71 <h4 class="col-span-1 font-bold">Number of Total CVs Made from <span>{{start}}</span> to <span>{{end}}</span>
72 <span class="col-span-1 text-blue-600 font-bold">{{CVs}}</span>
73 </div>
74 <div class="mb-5">
75 <h4 class="col-span-1 font-bold">Number of Total Applications Made from <span>{{start}}</span> to <span>{{end}}</span>
76 <span class="col-span-1 text-blue-600 font-bold">{{Applications}}</span>
77 </div>
78 <div class="mb-5">
79 <h4 class="col-span-1 font-bold">Number of Total Partners Made from <span>{{start}}</span> to <span>{{end}}</span>
80 <span class="col-span-1 text-blue-600 font-bold">{{Partners}}</span>
81 </div>
82 <div class="mb-5">
83 <h4 class="col-span-1 font-bold">Number of Total Jobs Made from <span>{{start}}</span> to <span>{{end}}</span>
84 <span class="col-span-1 text-blue-600 font-bold">{{Jobs}}</span>
85 </div>
86 </div>
87 </div>
88
89 </div>
90
91 <!-- Jobs Bar Chart -->
92 <div class="flex row justify-left bg-gray-50 border border-gray-200 rounded p-5 mb-10">
93 <div class="flex items-center mb-10">
94 <div class="fa-solid fa-thumbtack fa-xl m-5"></div>
95 </div>
96
97 <div class="flex justify-center mb-20">
98 <div class="m-5/6">
99 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-10">
100 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
101 Total Jobs Created
102 </div>
103 </div>
104
105 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
106 <span>{{totalJobs}}</span>
107 </div>
108 </div>
109
110 <!-- Pie Chart -->
111 <div class="grid grid-cols-1 lg:grid-cols-2 gap-4 mb-10">
112 <div class="col-span-1 bg-gray-50 border border-gray-200 rounded p-5 mb-10">
113 <h2 class="text-2xl font-bold mb-10">Partner Chart
114 </h2>
115 <div class="mb-10 flex items-center justify-center text-xl font-bold text-blue-700">
116 <span>{{totalPartners}}</span>
117 </div>
118 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
119 <span>{{totalCVs}}</span>
120 </div>
121 </div>
122
123 <div class="flex">
124 <div class="flex flex-col justify-center text-xl font-bold text-blue-700">
125 <h3 class="text-xl font-bold m-5">Total Partners :
126 </h3>
127 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
128 <span>{{totalPartners}}</span>
129 </div>
130 </div>
131 </div>
132
133 <div class="flex">
134 <div class="flex flex-col justify-center text-xl font-bold text-blue-700">
135 <h3 class="text-xl font-bold m-5">Total CVs :
136 </h3>
137 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
138 <span>{{totalCVs}}</span>
139 </div>
140 </div>
141 </div>
142
143 <div class="flex">
144 <div class="flex flex-col justify-center text-xl font-bold text-blue-700">
145 <h3 class="text-xl font-bold m-5">Passed Applicants :
146 </h3>
147 <div class="flex items-center justify-center text-xl font-bold text-blue-700">
148 <span>{{totalPassed}}</span>
149 </div>
150 </div>
151 </div>
152
153 </div>
154
155 <!-- Calendar -->
156 <div class="flex justify-center mb-10">
157 <div class="relative overflow-y-auto" style="height: 600px;">
158 <div class="text-3xl font-bold m-5">Company Calendar</div>
159 <iframe
160 src="https://calendar.google.com/calendar/embed?src=phenomyintmas.g84@gmail.com&ctz=Asia%2FTokyo"
161 style="border: 0" width="1000" height="600" frameborder="0" scrolling="no">
162 </iframe>
163 </div>
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
810 </div>
811
812 </div>
813
814 </div>
815
816 </div>
817
818 </div>
819
820 </div>
821
822 </div>
823
824 </div>
825
826 </div>
827
828 </div>
829
830 </div>
831
832 </div>
833
834 </div>
835
836 </div>
837
838 </div>
839
840 </div>
841
842 </div>
843
844 </div>
845
846 </div>
847
848 </div>
849
850 </div>
851
852 </div>
853
854 </div>
855
856 </div>
857
858 </div>
859
860 </div>
861
862 </div>
863
864 </div>
865
866 </div>
867
868 </div>
869
870 </div>
871
872 </div>
873
874 </div>
875
876 </div>
877
878 </div>
879
880 </div>
881
882 </div>
883
884 </div>
885
886 </div>
887
888 </div>
889
890 </div>
891
892 </div>
893
894 </div>
895
896 </div>
897
898 </div>
899
900 </div>
901
902 </div>
903
904 </div>
905
906 </div>
907
908 </div>
909
910 </div>
911
912 </div>
913
914 </div>
915
916 </div>
917
918 </div>
919
920 </div>
921
922 </div>
923
924 </div>
925
926 </div>
927
928 </div>
929
930 </div>
931
932 </div>
933
934 </div>
935
936 </div>
937
938 </div>
939
940 </div>
941
942 </div>
943
944 </div>
945
946 </div>
947
948 </div>
949
950 </div>
951
952 </div>
953
954 </div>
955
956 </div>
957
958 </div>
959
960 </div>
961
962 </div>
963
964 </div>
965
966 </div>
967
968 </div>
969
970 </div>
971
972 </div>
973
974 </div>
975
976 </div>
977
978 </div>
979
980 </div>
981
982 </div>
983
984 </div>
985
986 </div>
987
988 </div>
989
990 </div>
991
992 </div>
993
994 </div>
995
996 </div>
997
998 </div>
999
1000 </div>
```

Figure: 11.9.1. admin-home.blade.php 1

```

admin-home.blade.php X
resources > views > company > admin-home.blade.php > x-layout-admin
1 layout-admin
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
174 <div class="flex justify-center mb-10">
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 <!-- Chart 35 Scripts -->
190 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
191 <script>
192 document.addEventListener('DOMContentLoaded', () => {
193 //Job Type BarChart
194 fetch('/chart/jobs-by-type')
195 .then(response => response.json())
196 .then(data => {
197 const labels = data.map(item => item.job_type);
198 const counts = data.map(item => item.count);
199
200 new Chart(document.getElementById('jobsByTypeChart'), {
201 type: 'bar',
202 data: {
203 labels: labels,
204 datasets: [{
205 label: 'Jobs by Type',
206 data: counts,
207 borderWidth: 1
208 }]}),
209 options: {
210 responsive: true,
211 plugins: {
212 legend: {
213 display: false
214 }},
215 scales: {
216 y: {
217 beginAtZero: true
218 }
219 }
220 }
221 });
222
223 //Partners by Nationality
224 fetch('/chart/partners-by-nationality')
225 .then(response => response.json())
226 .then(data => {
227 const labels = data.map(item => item.partner_nationality);
228 const counts = data.map(item => item.count);
229
230 new Chart(document.getElementById('partnersByNationalityChart'), {
231 type: 'pie',
232 data: {
233 labels: labels,
234 datasets: [{
235 label: 'Partners by Nationality',
236 data: counts,
237 backgroundColor: [
238 '#3882F6', '#10B991', '#F59E00', '#EF4444',
239 '#8B5CF6', '#E48999', '#14B8A6'
240 ]
241 }]}),
242 options: {
243 responsive: true,
244 plugins: {
245 legend: {
246 position: 'bottom'
247 }
248 }
249 }
250 });
251
252 //CVs by Nationality
253 fetch('/chart/cvs-by-nationality')
254 .then(response => response.json())
255 .then(data => {
256 const labels = data.map(item => item.cv_nationality);
257 const counts = data.map(item => item.count);
258
259 new Chart(document.getElementById('cvsByNationalityChart'), {
260 type: 'pie',
261 data: {
262 labels: labels,
263 datasets: [{

```

```

admin-home.blade.php X
resources > views > company > admin-home.blade.php > ...
1 <x-layout-admin>
184 <script>
195 document.addEventListener('DOMContentLoaded', () => {
265 .then(data => {
270 type: 'pie',
271 data: {
272 labels: labels,
273 datasets: [{
274 label: 'CVs by Nationality',
275 data: counts,
276 }]}),
277 options: {
278 responsive: true,
279 plugins: {
280 legend: {
281 position: 'bottom'
282 }
283 }
284 }
285 }
286 });
287
288 //Statistics
289 fetch('/graph/statistics')
290 .then(response => response.json())
291 .then(data => {
292 document.getElementById('monthlyCVs').textContent = data.cv.monthly;
293 document.getElementById('yearlyCVs').textContent = data.cv.yearly;
294 document.getElementById('monthlyJobs').textContent = data.job.monthly;
295 document.getElementById('yearlyJobs').textContent = data.job.yearly;
296 document.getElementById('monthlyPartners').textContent = data.partner.monthly;
297 document.getElementById('yearlyPartners').textContent = data.partner.yearly;
298 document.getElementById('monthlyApplications').textContent = data.application.monthly;
299 document.getElementById('yearlyApplications').textContent = data.application.yearly;
300 });
301 });
302 </script>
303 </x-layout-admin>
304 </x-layout-admin>
307

```

Figure: 11.9.2. admin-home.blade.php 2

Admin side All Job Page

```
admin-job.blade.php X admin-add-new-job.blade.php admin-edit-job.blade.php admin-show-j... admin-job.blade.php X
resources > views > company > admin-job.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 >
1 <!-- layout-admin -->
2 <div class="flex justify-center mb-20">
3
4 <div class="w-5/6">
5
6 <!-- Top Section -->
7 <section class="grid grid-cols-1 items-center mb-5">
8
9 <div class="flex justify-left m-5">
10 <h1 class="text-3xl font-bold">
11 Jobs
12 </h1>
13 </div>
14 </section>
15
16 <!-- Job List -->
17 <section>
18
19 <!-- Search Bar -->
20 <div class="mb-10">
21 <p class="m-4">
22 Search with Company Name, Job Type, Job Title, Japanese Language Level, or
23 Applicable Nationality
24 </p>
25 <form action="/all-listed-jobs">
26
27 @include('components_search')
28 </form>
29 </div>
30 <div class="m-5 flex flex-col justify-center">
31 <div class="flex justify-between mb-5">
32 <div class="col-span-1">
33 <a href="/all-listed-jobs" class="transition duration-300 inline-block border-2
34 font-bold border-black text-black py-2 px-4 rounded-xl
35 hover:bg-black hover:text-white shadow-lg">
36 Refresh
37 </a>
38 </div>
39 <div class="col-span-1">
40 <a href="/add-new-job" class="transition duration-300 inline-block border-2
41 font-bold border-black text-black py-2 px-4 rounded-xl
42 hover:bg-green-700 hover:text-white hover:shadow-lg">
43 Add New Job
44 </a>
45
46
47
48
49 </div>
50
51
52
53 </div>
54
55 @include('components_message')
56
57 <!-- JOB SECTION -->
58 <div class="border-black border-4 rounded-2xl text-black pb-5">
59
60 <h1 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
61 Job List
62 </h1>
63
64 <div class="h-screen overflow-y-auto scrollbar">
65
66 <table class="w-full table-auto bg-white text-black">
67
68 <thead>
69
70 <tr>
71 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
72 Job Code
73 </th>
74 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
75 Company Name
76 </th>
77 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
78 Job Type
79 </th>
80 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
81 Japanese Language Level
82 </th>
83 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
84 Job Nationality
85 </th>
86 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
87 More
88 </th>
89 </tr>
90
91 </thead>
92
93 <tbody>
94
95 @foreach($jobs as $job)
96
97 <tr>
98 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
99 <div class="text-sm font-bold lg:text-lg">
100 {{ $job->job_code }}
101 </div>
102 </td>
103 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
104 {{ $job->company_name }}
105 </td>
106 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
107
108 <a class="text-sm font-bold lg:text-lg"
109 href="/all-listed-jobs/search/{{ $job->job_type }}" {{ $job->job_type }}</a>
110 </td>
111 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
112
113 <a class="text-sm font-bold lg:text-lg"
114 href="/all-listed-jobs/search/{{ $job->required_jp_level }}" {{ $job->required_jp_level }}</a>
115 </td>
116 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
117
118 <a class="text-sm font-bold lg:text-lg"
119 href="/all-listed-jobs/search/{{ $job->job_nationality }}" {{ $job->job_nationality }}</a>
120 </td>
121 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
122
123 <ul class="flex justify-center mb-1">
124 <li class="flex justify-center mb-1">
125 <a href="/all-listed-jobs/{{ $job->job_code }}"
126 class="items-center text-blue-700 hover:underline">
127 More Info
128 </a>
129 </li>
130 <li class="flex justify-center mb-1">
131 <a href="/all-listed-jobs/{{ $job->job_code }}/edit"
132 class="items-center text-blue-700 hover:underline">
133 Edit
134 </a>
135 </li>
136 </ul>
137
138 </td>
139 </tr>
140
141 </tbody>
142 </table>
143
144 @endforeach
145 </div>
146
147 </div>
148
149 </section>
150
151 </div>
152
153 </div>
154
155 </x-layout-admin>
156
157 </div>
158
159 </div>
160
161 </div>
162
163 </div>
164
165 </div>
166
167 </div>
168
169 </div>
170
171 </div>
172
173 </div>
174
175 </div>
176
177 </div>
178
179 </div>
180
181 </div>
182
183 </div>
184
185 </div>
186
187 </div>
188
189 </div>
190
191 </div>
192
193 </div>
194
195 </div>
196
197 </div>
198
199 </div>
200
201 </div>
202
203 </div>
204
205 </div>
206
207 </div>
208
209 </div>
210
211 </div>
212
213 </div>
214
215 </div>
216
217 </div>
218
219 </div>
220
221 </div>
222
223 </div>
224
225 </div>
226
227 </div>
228
229 </div>
230
231 </div>
232
233 </div>
234
235 </div>
236
237 </div>
238
239 </div>
240
241 </div>
242
243 </div>
244
245 </div>
246
247 </div>
248
249 </div>
250
251 </div>
252
253 </div>
254
255 </div>
256
257 </div>
258
259 </div>
260
261 </div>
262
263 </div>
264
265 </div>
266
267 </div>
268
269 </div>
270
271 </div>
272
273 </div>
274
275 </div>
276
277 </div>
278
279 </div>
280
281 </div>
282
283 </div>
284
285 </div>
286
287 </div>
288
289 </div>
290
291 </div>
292
293 </div>
294
295 </div>
296
297 </div>
298
299 </div>
300
301 </div>
302
303 </div>
304
305 </div>
306
307 </div>
308
309 </div>
310
311 </div>
312
313 </div>
314
315 </div>
316
317 </div>
318
319 </div>
320
321 </div>
322
323 </div>
324
325 </div>
326
327 </div>
328
329 </div>
330
331 </div>
332
333 </div>
334
335 </div>
336
337 </div>
338
339 </div>
340
341 </div>
342
343 </div>
344
345 </div>
346
347 </div>
348
349 </div>
350
351 </div>
352
353 </div>
354
355 </div>
356
357 </div>
358
359 </div>
360
361 </div>
362
363 </div>
364
365 </div>
366
367 </div>
368
369 </div>
370
371 </div>
372
373 </div>
374
375 </div>
376
377 </div>
378
379 </div>
380
381 </div>
382
383 </div>
384
385 </div>
386
387 </div>
388
389 </div>
390
391 </div>
392
393 </div>
394
395 </div>
396
397 </div>
398
399 </div>
400
401 </div>
402
403 </div>
404
405 </div>
406
407 </div>
408
409 </div>
410
411 </div>
412
413 </div>
414
415 </div>
416
417 </div>
418
419 </div>
420
421 </div>
422
423 </div>
424
425 </div>
426
427 </div>
428
429 </div>
430
431 </div>
432
433 </div>
434
435 </div>
436
437 </div>
438
439 </div>
440
441 </div>
442
443 </div>
444
445 </div>
446
447 </div>
448
449 </div>
450
451 </div>
452
453 </div>
454
455 </div>
456
457 </div>
458
459 </div>
460
461 </div>
462
463 </div>
464
465 </div>
466
467 </div>
468
469 </div>
470
471 </div>
472
473 </div>
474
475 </div>
476
477 </div>
478
479 </div>
480
481 </div>
482
483 </div>
484
485 </div>
486
487 </div>
488
489 </div>
490
491 </div>
492
493 </div>
494
495 </div>
496
497 </div>
498
499 </div>
500
501 </div>
502
503 </div>
504
505 </div>
506
507 </div>
508
509 </div>
510
511 </div>
512
513 </div>
514
515 </div>
516
517 </div>
518
519 </div>
520
521 </div>
522
523 </div>
524
525 </div>
526
527 </div>
528
529 </div>
530
531 </div>
532
533 </div>
534
535 </div>
536
537 </div>
538
539 </div>
540
541 </div>
542
543 </div>
544
545 </div>
546
547 </div>
548
549 </div>
550
551 </div>
552
553 </div>
554
555 </div>
556
557 </div>
558
559 </div>
560
561 </div>
562
563 </div>
564
565 </div>
566
567 </div>
568
569 </div>
570
571 </div>
572
573 </div>
574
575 </div>
576
577 </div>
578
579 </div>
580
581 </div>
582
583 </div>
584
585 </div>
586
587 </div>
588
589 </div>
590
591 </div>
592
593 </div>
594
595 </div>
596
597 </div>
598
599 </div>
600
601 </div>
602
603 </div>
604
605 </div>
606
607 </div>
608
609 </div>
610
611 </div>
612
613 </div>
614
615 </div>
616
617 </div>
618
619 </div>
620
621 </div>
622
623 </div>
624
625 </div>
626
627 </div>
628
629 </div>
630
631 </div>
632
633 </div>
634
635 </div>
636
637 </div>
638
639 </div>
640
641 </div>
642
643 </div>
644
645 </div>
646
647 </div>
648
649 </div>
650
651 </div>
652
653 </div>
654
655 </div>
656
657 </div>
658
659 </div>
660
661 </div>
662
663 </div>
664
665 </div>
666
667 </div>
668
669 </div>
670
671 </div>
672
673 </div>
674
675 </div>
676
677 </div>
678
679 </div>
680
681 </div>
682
683 </div>
684
685 </div>
686
687 </div>
688
689 </div>
690
691 </div>
692
693 </div>
694
695 </div>
696
697 </div>
698
699 </div>
700
701 </div>
702
703 </div>
704
705 </div>
706
707 </div>
708
709 </div>
710
711 </div>
712
713 </div>
714
715 </div>
716
717 </div>
718
719 </div>
720
721 </div>
722
723 </div>
724
725 </div>
726
727 </div>
728
729 </div>
730
731 </div>
732
733 </div>
734
735 </div>
736
737 </div>
738
739 </div>
740
741 </div>
742
743 </div>
744
745 </div>
746
747 </div>
748
749 </div>
750
751 </div>
752
753 </div>
754
755 </div>
756
757 </div>
758
759 </div>
760
761 </div>
762
763 </div>
764
765 </div>
766
767 </div>
768
769 </div>
770
771 </div>
772
773 </div>
774
775 </div>
776
777 </div>
778
779 </div>
780
781 </div>
782
783 </div>
784
785 </div>
786
787 </div>
788
789 </div>
790
791 </div>
792
793 </div>
794
795 </div>
796
797 </div>
798
799 </div>
800
801 </div>
802
803 </div>
804
805 </div>
806
807 </div>
808
809 </div>
810
811 </div>
812
813 </div>
814
815 </div>
816
817 </div>
818
819 </div>
820
821 </div>
822
823 </div>
824
825 </div>
826
827 </div>
828
829 </div>
830
831 </div>
832
833 </div>
834
835 </div>
836
837 </div>
838
839 </div>
840
841 </div>
842
843 </div>
844
845 </div>
846
847 </div>
848
849 </div>
850
851 </div>
852
853 </div>
854
855 </div>
856
857 </div>
858
859 </div>
860
861 </div>
862
863 </div>
864
865 </div>
866
867 </div>
868
869 </div>
870
871 </div>
872
873 </div>
874
875 </div>
876
877 </div>
878
879 </div>
880
881 </div>
882
883 </div>
884
885 </div>
886
887 </div>
888
889 </div>
890
891 </div>
892
893 </div>
894
895 </div>
896
897 </div>
898
899 </div>
900
901 </div>
902
903 </div>
904
905 </div>
906
907 </div>
908
909 </div>
909
```

Figure: 11.9.3. admin-job.blade.php

Admin Side Job Creation Page

```

resources > views > company > admin-add-new-job.blade.php @ x-layout-admin
1 <-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-4/5">
6
7 <a href="/all-listed-jobs" class="inline-block border-2 font-bold
8 border-black text-white bg-black py-3 px-4 rounded-xl uppercase hover:text-white
9 shadow-lg">
10 <i class="fa-solid fa-arrow-left"></i> Back to All Jobs
11 </a>
12
13 <main class="m-5">
14 <div class="mx-4">
15 <div class="max-w-9/10 mx-auto mt-10">
16
17 <header class="text-center mb-5">
18
19 <h1 class="text-3xl mb-10 font-bold uppercase">
20 | Job Information
21 </h1>
22
23 </header>
24
25
26 <!-- CV ADD FORM -->
27 <div class="flex justify-center">
28
29 <div class="lgw-4/5 p-10 border-2 border-black">
30
31 <form method="post" action="/all-listed-jobs" enctype="multipart/form-data">
32
33 @csrf
34
35 <div class="mb-6">
36
37 <label for="job_type" class="inline-block text-lg mb-2">
38 | Job Type
39 </label>
40
41 <select name="job_type" id="job_type"
42 class="border border-gray-200 rounded p-2 w-full" required>
43 <option value="" selected="">Select Job Type --</option>
44 <option value="RESTAURANT" @selected(old('job_type')== 'RESTAURANT')>
45 | RESTAURANT
46 </option>
47 <option value="NURSING" @selected(old('job_type')== 'NURSING')>NURSING
48 </option>
49 <option value="HOTEL" @selected(old('job_type')== 'HOTEL')>HOTEL</option>
50 <option value="CONSTRUCTION" @selected(old('job_type')== 'CONSTRUCTION')>
51 | CONSTRUCTION
52 </select>
53
54 <input type="text" class="border border-gray-200 rounded p-2 w-full"
55 name="recruit_number" value="{{old('recruit_number')}}" />
56
57 <input type="number" class="border border-gray-200 rounded p-2 w-full"
58 name="recruit_number" value="{{old('recruit_number')}}" />
59
60 <input type="text" class="border border-gray-200 rounded p-2 w-full"
61 name="application_deadline" value="{{old('application_deadline')}}" />
62
63 </div>
64
65 <div class="mb-6">
66
67 <label for="recruit_number" class="inline-block text-lg mb-2">
68 | Recruit Number
69 </label>
70
71 <input type="number" class="border border-gray-200 rounded p-2 w-full"
72 name="recruit_number" value="{{old('recruit_number')}}" />
73
74 <input type="text" class="border border-gray-200 rounded p-2 w-full"
75 name="application_deadline" value="{{old('application_deadline')}}" />
76
77 <input type="text" class="border border-gray-200 rounded p-2 w-full"
78 name="application_deadline" value="{{old('application_deadline')}}" />
79
80 </div>
81
82 <div class="mb-6">
83
84 <label for="application_deadline" class="inline-block text-lg mb-2">
85 | Application Deadline
86 </label>
87
88 <input type="date" class="border border-gray-200 rounded p-2 w-full"
89 name="application_deadline" value="{{old('application_deadline')}}" />
90
91 <input type="text" class="border border-gray-200 rounded p-2 w-full"
92 name="application_deadline" value="{{old('application_deadline')}}" />
93
94 </div>
95
96 </div>
97
98 <div class="mb-6">
99
100 <label for="company_name" class="inline-block text-lg mb-2">
101 | Company Name
102 </label>
103
104 <input type="text" class="border border-gray-200 rounded p-2 w-full"
105 name="company_name" value="{{old('company_name')}}" />
106
107 <input type="text" class="border border-gray-200 rounded p-2 w-full"
108 name="company_name" value="{{old('company_name')}}" />
109
110 </div>
111
112 <div class="mb-6">
113
114 <label for="company_website" class="inline-block text-lg mb-2">
115 | Company Website
116 </label>
117
118 <input type="text" class="border border-gray-200 rounded p-2 w-full"
119 name="company_website" value="{{old('company_website')}}" />
120
121 <input type="text" class="border border-gray-200 rounded p-2 w-full"
122 name="company_website" value="{{old('company_website')}}" />
123
124 </div>
125
126 <div class="mb-6">
127
128 <label for="company_location" class="inline-block text-lg mb-2">
129 | Company Location
130 </label>
131
132 <input type="text" class="border border-gray-200 rounded p-2 w-full"
133 name="company_location" value="{{old('company_location')}}" />
134
135 <input type="text" class="border border-gray-200 rounded p-2 w-full"
136 name="company_location" value="{{old('company_location')}}" />
137
138 </div>
139
140 </div>
141
142 <div class="mb-6">
143
144 <label for="job_title" class="inline-block text-lg mb-2">
145 | Job Title
146 </label>
147
148 <input type="text" class="border border-gray-200 rounded p-2 w-full"
149 name="job_title" value="{{old('job_title')}}" />
150
151 <input type="text" class="border border-gray-200 rounded p-2 w-full"
152 name="job_title" value="{{old('job_title')}}" />
153
154 </div>
155
156 <div class="mb-6">
157
158 <label for="job_description" class="inline-block text-lg mb-2">
159 | Job Description
160 </label>
161
162 <input class="border border-gray-200 rounded p-2 w-full" name="job_description"
163 rows="3" value="{{old('job_description')}}" />
164
165 <input class="border border-gray-200 rounded p-2 w-full" name="job_description"
166 rows="3" value="{{old('job_description')}}" />
167
168 </div>
169
170 <div class="mb-6">
171
172 <label for="work_time" class="inline-block text-lg mb-2">
173 | Work Time
174 </label>
175
176 <input type="text" class="border border-gray-200 rounded p-2 w-full"
177 name="work_time" value="{{old('work_time')}}" />
178
179 <input type="text" class="border border-gray-200 rounded p-2 w-full"
180 name="work_time" value="{{old('work_time')}}" />
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
810 </div>
811
812 </div>
813
814 </div>
815
816 </div>
817
818 </div>
819
820 </div>
821
822 </div>
823
824 </div>
825
826 </div>
827
828 </div>
829
830 </div>
831
832 </div>
833
834 </div>
835
836 </div>
837
838 </div>
839
840 </div>
841
842 </div>
843
844 </div>
845
846 </div>
847
848 </div>
849
850 </div>
851
852 </div>
853
854 </div>
855
856 </div>
857
858 </div>
859
860 </div>
861
862 </div>
863
864 </div>
865
866 </div>
867
868 </div>
869
870 </div>
871
872 </div>
873
874 </div>
875
876 </div>
877
878 </div>
879
880 </div>
881
882 </div>
883
884 </div>
885
886 </div>
887
888 </div>
889
890 </div>
891
892 </div>
893
894 </div>
895
896 </div>
897
898 </div>
899
900 </div>
901
902 </div>
903
904 </div>
905
906 </div>
907
908 </div>
909
910 </div>
911
912 </div>
913
914 </div>
915
916 </div>
917
918 </div>
919
920 </div>
921
922 </div>
923
924 </div>
925
926 </div>
927
928 </div>
929
930 </div>
931
932 </div>
933
934 </div>
935
936 </div>
937
938 </div>
939
940 </div>
941
942 </div>
943
944 </div>
945
946 </div>
947
948 </div>
949
950 </div>
951
952 </div>
953
954 </div>
955
956 </div>
957
958 </div>
959
960 </div>
961
962 </div>
963
964 </div>
965
966 </div>
967
968 </div>
969
970 </div>
971
972 </div>
973
974 </div>
975
976 </div>
977
978 </div>
979
980 </div>
981
982 </div>
983
984 </div>
985
986 </div>
987
988 </div>
989
990 </div>
991
992 </div>
993
994 </div>
995
996 </div>
997
998 </div>
999
1000 </div>

```

Figure: 11.9.4. admin-add-new-job.blade.php 1

```

resources > views > company > admin-add-new-job.blade.php > x-layout-admin
1 <-layout-admin>
3 <div class="flex justify-center mb-20">
5 <div class="w-4/5">
13 <main class="m-5">
14 <div class="mx-4">
182 <p class="text-red-500 text-xs mt-1">{{message}}</p>
183 @enderror
184 </div>
185 </div>
186 <div class="mb-6">
187 <label for="off_days" class="inline-block text-lg mb-2">
188 Off Days
189 </label>
190 <input class="border border-gray-200 rounded p-2 w-full" name="off_days"
191 rows="3" value="{{old('off_days')}}"/>
192 @error('off_days')
193 <p class="text-red-500 text-xs mt-1">{{message}}</p>
194 @enderror
195 </div>
196 <div class="mb-6">
197 <label for="salary_benefits" class="inline-block text-lg mb-2">
198 A. Salary + Benefits
199 </label>
200 <input class="border border-gray-200 rounded p-2 w-full" name="salary_benefits"
201 rows="3" value="{{old('salary_benefits')}}"/>
202 @error('salary_benefits')
203 <p class="text-red-500 text-xs mt-1">{{message}}</p>
204 @enderror
205 </div>
206 <div class="mb-6">
207 <label for="deductables" class="inline-block text-lg mb-2">
208 B. Deductables
209 </label>
210 <input class="border border-gray-200 rounded p-2 w-full" name="deductables"
211 rows="3" value="{{old('deductables')}}"/>
212 @error('deductables')
213 <p class="text-red-500 text-xs mt-1">{{message}}</p>
214 @enderror
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 </div>
221 </div>
222 </div>
223 </div>
224 </div>
225 </div>
226 @error('deductables')

```

```

resources > views > company > admin-add-new-job.blade.php > x-layout-admin
1 admin>
3 <div class="flex justify-center mb-20">
5 <div class="w-4/5">
13 <main class="m-5">
14 <div class="mx-4">
270 <label for="required_jp_level" class="inline-block text-lg mb-2">
271 Required Japanese Level
272 </label>
273 <select name="required_jp_level" id="required_jp_level"
274 class="border border-gray-200 rounded p-2 w-full" required>
275 <option value="">-- Select Japanese Level --</option>
276 <option value="3LPT M4" @selected(old('required_jp_level')==>3LPT M4')>3LPT
277 M4
278 </option>
279 <option value="3LPT N3" @selected(old('required_jp_level')==>3LPT N3')>3LPT
280 N3
281 </option>
282 <option value="3LPT N2" @selected(old('required_jp_level')==>3LPT N2')>3LPT
283 N2
284 </option>
285 <option value="3LPT N1" @selected(old('required_jp_level')==>3LPT N1')>3LPT
286 N1
287 </option>
288 </select>
289 </div>
290 <div class="mb-6">
291 <label for="age_range" class="inline-block text-lg mb-2">
292 Age Range
293 </label>
294 <select name="age_range" id="age_range"
295 class="border border-gray-200 rounded p-2 w-full" required>
296 <option value="">-- Select Age Range --</option>
297 <option value="18-25" @selected(old('age_range')==>18-25')>18-25
298 </option>
299 <option value="Under 30" @selected(old('age_range')==>Under 30')>Under 30
300 </option>
301 <option value="Not limited" @selected(old('age_range')==>Not limited')>Not
302 limited
303 </option>
304 </select>
305 </div>
306 </div>
307 </div>
308 </div>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </div>
315 @error('age_range')
316 <p class="text-red-500 text-xs mt-1">{{message}}</p>
317 @enderror
318 </div>
319 <div class="mb-6">
320 <label for="job_gender" class="inline-block text-lg mb-2">
321 Gender
322 </label>
323 <select name="job_gender" id="job_gender"
324 class="border border-gray-200 rounded p-2 w-full" required>
325 <option value="">-- Select Gender --</option>
326 <option value="Male" @selected(old('job_gender')==>Male')>Male</option>
327 <option value="Female" @selected(old('job_gender')==>Female')>Female
328 </option>
329 <option value="Both" @selected(old('job_gender')==>Both')>Both</option>
330 </select>
331 </div>
332 <div class="mb-6">
333 <label for="job_nationality" class="inline-block text-lg mb-2">
334 Nationality
335 </label>
336 <select name="job_nationality" id="job_nationality"
337 class="border border-gray-200 rounded p-2 w-full" required>
338 <option value="">-- Select Nationality --</option>
339 <option value="Myanmar" @selected(old('job_nationality')==>Myanmar')>
340 Myanmar
341 </option>
342 <option value="Philippines" @selected(old('job_nationality')==>Philippines')>
343 Philippines</option>
344 <option value="Thailand" @selected(old('job_nationality')==>Thailand')>
345 Thailand
346 </select>
347 </div>
348 </div>
349 </div>
350 </div>
351 </div>
352 </div>
353 </div>
354 </div>
355 </div>
356 </div>
357 </div>
358 </div>

```

Figure: 11.9.5. admin-add-new-job.blade.php 2


```

admin-edit-job.blade.php X admin-show-job.blade.php
resources > views > company > admin-edit-job.blade.php > x-layout-admin
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <main class="m-5">
5 <div class="m-4">
6 <input type="text" class="border border-gray-200 rounded p-2 w-full"
7 name="company_name" value="{{ $job -> company_name }}" />
8
9 @error('company_name')
10 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
11 @enderror
12 </div>
13 <div class="mb-6">
14 <label for="company_website" class="inline-block text-lg mb-2">
15 Company Website
16 </label>
17
18 <input type="text" class="border border-gray-200 rounded p-2 w-full"
19 name="company_website" value="{{ $job -> company_website }}" />
20
21 @error('company_website')
22 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
23 @enderror
24 </div>
25 <div class="mb-6">
26 <label for="company_location" class="inline-block text-lg mb-2">
27 Company Location
28 </label>
29
30 <input type="text" class="border border-gray-200 rounded p-2 w-full"
31 name="company_location" value="{{ $job -> company_location }}" />
32
33 @error('company_location')
34 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
35 @enderror
36 </div>
37 <div class="mb-6">
38 <label for="job_title" class="inline-block text-lg mb-2">
39 Job Title
40 </label>
41
42 <input type="text" class="border border-gray-200 rounded p-2 w-full"
43 name="job_title" value="{{ $job -> job_title }}" />
44
45 @error('job_title')
46 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
47 @enderror
48 </div>
49 <div class="mb-6">
50 <label for="job_description" class="inline-block text-lg mb-2">
51 Job Description
52 </label>
53
54 <input class="border border-gray-200 rounded p-2 w-full" name="job_description"
55 rows="3" value="{{ $job -> job_description }}" />
56
57 @error('job_description')
58 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
59 @enderror
60 </div>
61 <div class="mb-6">
62 <label for="work_time" class="inline-block text-lg mb-2">
63 Work Time
64 </label>
65
66 <input class="border border-gray-200 rounded p-2 w-full" name="work_time" rows="
67 value="{{ $job -> work_time }}" />
68
69 @error('work_time')
70 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
71 @enderror
72 </div>
73 <div class="mb-6">
74 <label for="off_days" class="inline-block text-lg mb-2">
75 Off Days
76 </label>
77
78 <input type="text" class="border border-gray-200 rounded p-2 w-full"
79 name="off_days" value="{{ $job -> off_days }}" />
80
81 @error('off_days')
82 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
83 @enderror
84 </div>
85 </div>
86 </main>
87 </div>
88 </div>
89 </div>
90 </x-layout-admin>

```

```

admin-edit-job.blade.php X admin-show-job.blade.php
resources > views > company > admin-edit-job.blade.php > x-layout-admin
1 admin
2 class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <main class="m-5">
5 <div class="m-4">
6 <input type="text" class="border border-gray-200 rounded p-2 w-full" name="off_days"
7 value="{{ $job -> off_days }}" />
8
9 @error('off_days')
10 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
11 @enderror
12 </div>
13 <div class="mb-6">
14 <label for="salary_benefits" class="inline-block text-lg mb-2">
15 A. Salary + Benefits
16 </label>
17
18 <input class="border border-gray-200 rounded p-2 w-full" name="salary_benefits"
19 value="{{ $job -> salary_benefits }}" placeholder="Write Salary and Benefits" />
20
21 @error('salary_benefits')
22 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
23 @enderror
24 </div>
25 <div class="mb-6">
26 <label for="deductables" class="inline-block text-lg mb-2">
27 B. Deductables
28 </label>
29
30 <input class="border border-gray-200 rounded p-2 w-full" name="deductables" rows="3"
31 value="{{ $job -> deductables }}" placeholder="Write Salary and Benefits" />
32
33 @error('deductables')
34 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
35 @enderror
36 </div>
37 <div class="mb-6">
38 <label for="after_deduction" class="inline-block text-lg mb-2">
39 Salary After Deduction (A-B)
40 </label>
41
42 <input type="text" class="border border-gray-200 rounded p-2 w-full"
43 name="after_deduction" value="{{ $job -> after_deduction }}"
44 rows="3" value="{{ $job -> after_deduction }}"
45 placeholder="Salary + Benefits - Deductables" />
46
47 @error('after_deduction')
48 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
49 @enderror
50 </div>
51 <div class="mb-6">
52 <label for="airticket_exp" class="inline-block text-lg mb-2">
53 Airplane Ticket Expense
54 </label>
55
56 <select name="airticket_exp" id="airticket_exp"
57 class="border border-gray-200 rounded p-2 w-full" required>
58 <option value="{{ $job -> airticket_exp }}">{{ $job -> airticket_exp }}</option>
59 <option value="COMPANY_PAID" @selected(old('airticket_exp')=="COMPANY_PAID")>
60 COMPANY
61 PAID</option>
62 <option value="SELF_PAID" @selected(old('airticket_exp')=="SELF_PAID")>SELF
63 PAID
64 </option>
65 </select>
66 </div>
67 <div class="mb-6">
68 <label for="required_jp_level" class="inline-block text-lg mb-2">
69 Required Japanese Level
70 </label>
71
72 <select name="required_jp_level" id="required_jp_level"
73 class="border border-gray-200 rounded p-2 w-full" required>
74 <option value="{{ $job -> required_jp_level }}">{{ $job -> required_jp_level }}</option>
75 <option value="3LPT M4" @selected(old('required_jp_level')=="3LPT M4")>3LPT M4
76 </option>
77 </select>
78 </div>
79 </div>
80 </main>
81 </div>
82 </div>
83 </div>
84 </x-layout-admin>

```

Figure: 11.9.8. admin-edit-job.blade.php 2

```

admin-edit-job.blade.php X admin-show-job.blade.php
resources > views > company > admin-edit-job.blade.php > x-layout-admin
1 admin
2 class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <main class="m-5">
5 <div class="m-4">
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figure: 11.9.9. admin-edit-job.blade.php 3

Admin Side Job View Page

```
admin-show-job.blade.php X
resources > views > company > admin-show-job.blade.php > x-layout-admin
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <div class="flex justify-between mb-20">
8
9 <a href="/all-listed-jobs" class="mt-5 ml-5 inline-block border-2 font-bold
10 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
11 shadow-lg">
12 <i class="fa-solid fa-arrow-left"></i> All Job List
13 </a>
14
15 <a href="/all-listed-jobs/{{ $job -> job_code }}/edit" class="transition duration-200 mt-5 ml-5
16 inline-block font-bold text-white bg-blue-700 py-1 px-2
17 rounded-xl uppercase hover:bg-blue-400">
18 <i class="fa fa-pencil mr-2"></i>Edit
19 </a>
20
21 </div>
22
23 <!-- JOB SECTION -->
24 @include('components._view-job')
25
26 </div>
27
28 </div>
29
30 </x-layout-admin>
31
```

Figure: 11.9.10. admin-show-job.blade.php

Admin Side All Partners Page

```
admin-partner.blade.php • admin-add-partner.blade.php • admin-edit-partner.blade.php • admin-partner.blade.php
1 <x-layout-admin>
2 <div flex justify-center mb-20>
3 <div w-5/6>
4 <div class="flex justify-center mb-20">
5 <div class="w-5/6">
6 <!-- Top Section -->
7 <section class="grid grid-cols-1 items-center mb-5">
8 <div class="col-span-3 flex justify-left m-5">
9 <h1 class="text-3xl font-bold">
10 Partners
11 </h1>
12 </div>
13 </section>
14 <!-- List Section -->
15 <section class="m-5 flex flex-col justify-center">
16 <!-- Search Bar -->
17 <div class="mb-10">
18 <p class="mb-4">
19 Search with Partner Name or Partner Nationality
20 </p>
21 <form action="/all-partners">
22 @include('components._search')
23 </form>
24 <div class="flex justify-between mb-5">
25 <div class="flex justify-center">
26 <a href="/all-partners" class="transition duration-300 inline-block border-2
27 font-bold border-black text-black py-2 px-4 rounded-xl
28 hover:bg-black hover:text-white shadow-lg">
29 Refresh
30 </a>
31 </div>
32 <div class="flex justify-center">
33 <a href="/register-new-partner" class="transition duration-200
34 inline-block border-2 font-bold border-black text-black py-2 px-4 rounded-xl
35 hover:bg-black hover:text-white shadow-lg">
36 <i class="fa-solid fa-plus"></i> Add New
37 </a>
38 </div>
39 </div>
40 </section>
41 </div>
42 </div>
43 </x-layout-admin>
44
45
46
47
48
49
50
51 <x-layout-admin>
52 <div class="flex justify-center mb-20">
53 <div class="w-5/6">
54 <section class="m-5 flex flex-col justify-center">
55 @include('components._message')
56 <!-- PARTNER SECTION -->
57 <div class="border-black border-4 rounded-2xl text-black pb-5">
58 <h1 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
59 Partner list
60 </h1>
61 <div class="h-full overflow-y-auto scrollbar">
62 <table class="w-full table-auto bg-white text-black">
63 <thead>
64 <tr>
65 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
66 Partner Name
67 </th>
68 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
69 Email
70 </th>
71 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
72 Nationality
73 </th>
74 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
75 More
76 </th>
77 </tr>
78 </thead>
79 <tbody>
80 <tr>
81 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
82 <a class="text-lg lg:text-2xl font-bold"
83 href="/all-partners?search={{partner -> partner_name}}">
84 {{partner -> partner_name}}
85 </a>
86 </td>
87 </tr>
88 </tbody>
89 </table>
90 @foreach($partners as $partner)
91 <tr>
92 <td class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
93 <a class="text-lg lg:text-2xl font-bold"
94 href="/all-partners?search={{partner -> partner_name}}">
95 {{partner -> partner_name}}
96 </a>
97 </td>
98 </tr>
99 </tbody>
100 </table>
101 </div>
102 </section>
103 </div>
104 </x-layout-admin>
```

```
admin-partner.blade.php • admin-add-partner.blade.php • admin-edit-partner.blade.php • admin-show-partner.blade.php
resources > views > company > admin-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > section.m-5.flex.flex-col.justify-center > div.mb-10 > p.mb-4
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <section class="m-5 flex flex-col justify-center">
5 <div class="border-black border-4 rounded-2xl text-black pb-5">
6 <h1 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
7 Partner list
8 </h1>
9 <div class="h-full overflow-y-auto scrollbar">
10 <table class="w-full table-auto bg-white text-black">
11 <thead>
12 <tr>
13 <th class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
14 <h3 class="text-lg lg:text-2xl font-bold">
15 {{partner -> email}}
16 </h3>
17 </th>
18 <th class="px-4 py-8 border-t border-b border-gray-300 text-center font-bold">
19 <a class="text-lg m-2 lg:text-xl w-24 hover:text-blue-500"
20 href="/all-partners?search={{partner -> partner_nationality}}">
21 {{partner -> partner_nationality}}
22 </a>
23 </th>
24 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
25 <a class="hover:text-blue-700 href="/all-partners/{{partner -> partner_id}}">
26 <i class="fa-solid fa-circle-info mr-1"></i> More Info
27 </a>
28 </th>
29 </tr>
30 </thead>
31 <tbody>
32 @foreach
33 </tbody>
34 </table>
35 </div>
36 </section>
37 </div>
38 </x-layout-admin>
```

Figure: 11.9.11. admin-partner.blade.php

Admin Side Partner Creation Page

```
resources > views > company > admin-add-partner.blade.php > x-layout-admin
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <a href="/all-partners" class="mt-5 ml-5 inline-block border-2 font-bold
8 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
9 shadow-lg">
10 <i class="fa-solid fa-arrow-left"/> Back to all Partners
11 </a>
12
13 <!-- PARTNER ADD FORM -->
14 <div class="m-5">
15
16 <header class="text-center mt-10 mb-5">
17
18 <h1 class="text-3xl font-bold uppercase">
19 Partner Information
20 </h1>
21 </header>
22
23 <div class="flex justify-center">
24
25 <div class="w-3/4 border-black border-4 rounded-2xl text-black overflow-hidden
26 flex flex-col mb-10 p-3">
27
28 <div class="text-2xl font-bold text-center border-b-2 text-center mb-10 p-3">
29
30
31 General Information
32 </div>
33 </div>
34
35 <form method="POST" action="/all-partners">
36
37 @csrf
38 <div>
39
40 <ul class="border border-gray-200 mb-5 p-5">
41
42 <li class="mb-5 p-3">
43 <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
44 Partner Name
45 </label>
46
47 <input type="text" class="border border-gray-200 rounded p-2 w-full"
48 name="partner_name" value="{{old('partner_name')}}" />
49
50 @error('partner_name')
```

Figure: 11.9.12. admin-add-partner.blade.php

Admin Side Partner Edit Page

```
admin-edit-partner.blade.php | admin-show-partner.blade.php | admin-edit-partner.blade.php | x-layout-admin
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <a href="/all-partners/{{ $partner -> partner_id }}" class="inline-block border-2 font-bold
8 border-black text-white bg-black py-3 px-4 rounded-xl uppercase hover:text-white
9 shadow-lg">
10 <i class="fa-solid fa-arrow-left"/> Back
11 </a>
12
13 <main class="w-5">
14
15 <div class="mx-4">
16
17 <header class="max-w-9/10 mx-auto mt-10">
18
19 <h1 class="text-3xl font-bold uppercase text-center mb-10">
20 Edit Partner Information
21 </h1>
22
23 </header>
24
25 <!-- PARTNER EDIT SECTION -->
26 <div class="flex justify-center">
27 <div class="p-20 border-2 border-black w-3/4">
28
29 @include('components._message')
30
31 <form method="post" action="/all-partners/{{ $partner -> partner_id }}"
32 enctype="multipart/form-data">
33
34 @csrf
35 @method('PUT')
36
37 <div>
38
39 <ul class="border border-gray-200 mb-5 p-5">
40
41
42
43
44 <li class="mb-5 p-3">
45 <label for="PartnerID" class="inline-block text-lg mb-2 font-bold">
46 Partner ID
47 </label>
48 <input type="text" class="border border-gray-200 rounded p-2 w-full"
49 value="{{ $partner -> partner_id }}" />
50 </li>
51
52
53
54
55 </ul>
56
57 <div class="mb-5 p-3">
58 <label for="partner_name" class="inline-block text-lg mb-2 font-bold">
59 Partner Name
60 </label>
61 <input type="text" class="border border-gray-200 rounded p-2 w-full"
62 name="partner_name" value="{{ $partner -> partner_name }}" />
63
64 @error('partner_name')
65 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
66 @enderror
67 </div>
68
69 <div class="mb-5 p-3">
70 <label for="partner_email" class="inline-block text-lg mb-2 font-bold">
71 Partner Email
72 </label>
73 <input type="email" class="border border-gray-200 rounded p-2 w-full"
74 name="partner_email" value="{{ $partner -> email }}" />
75
76 @error('email')
77 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
78 @enderror
79 </div>
80
81 <div class="mb-5 p-3">
82 <label for="partner_address" class="inline-block text-lg mb-2 font-bold">
83 Partner Address
84 </label>
85 <input type="text" class="border border-gray-200 rounded p-2 w-full"
86 name="partner_address" value="{{ $partner -> partner_address }}" />
87
88 @error('partner_address')
89 <p class="text-red-500 text-xs mt-1">{{ $message }}</p>
90 @enderror
91 </div>
92
93 <div class="mb-5 p-3">
94 <label for="password" class="inline-block text-lg mb-2 font-bold">
95 Partner Password (Leave if No Update is Needed)
96 </label>
97 <input type="password" class="border border-gray-200 rounded p-2 w-full"
98 name="password" />
99 </div>
100
101
102
103 </ul>
104
105 </div>
106
107 <div class="mb-6 flex justify-center">
108
109 <button type="submit" class="transition duration-300 bg-green-700 text-white rounded
110 py-2 px-4 hover:bg-green-500">
111 <i class="fa fa-pencil mr-2"/> Edit
112 </button>
113
114 </div>
115
116 </form>
117
118 </div>
119 </div>
120
121 </div>
122 </main>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </x-layout-admin>
129
130
```

Figure: 11.9.13. admin-edit-partner.blade.php

Admin Side Partner View Page

```

admin-show-partner.blade.php X
how-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > section.flex.justify-between.mb-20
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <!-- UPPER SECTION -->
5 <section class="flex justify-between mb-20">
6 <a href="/all-partners" class="text-5 ml-5 inline-block border-2 font-bold
7 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
8 shadow-lg">
9 <i class="fa-solid fa-arrow-left"/> Back to All Partner List
10 </a>
11 <a href="/all-partners/{{ $partner -> partner_id }}/edit" class="transition
12 duration-200 bg-green-700 mt-5 ml-5 inline-block border-2
13 font-bold border-green-700 text-white bg-black py-1 px-2 rounded-xl uppercase
14 hover:text-green-700 hover:bg-white">
15 <i class="fa-solid fa-pencil"/> Edit Partner
16 </a>
17 </section>
18 <!-- LOWER SECTION -->
19 <section class="m-5 flex flex-col justify-center">
20 <h2 class="text-center mb-5">
21 Partner Information
22 </h2>
23 <div class="flex justify-center">
24 <div class="w-3/4">
25 <!-- PARTNER SECTION -->
26 <div class="border-black border-4 rounded-2xl text-black overflow-hidden flex flex-col mb-10">
27 <div class="text-2xl font-bold text-center bg-gray-300 max-h border-b-2 p-3 text-center">
28 General Information
29 </div>
30 <div class="p-3">
31 <table border="1">
32 <thead>
33 <tr>
34 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
35 Language Level
36 </th>
37 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
38 Job Type
39 </th>
40 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
41 More Info
42 </th>
43 </thead>
44 <tbody>
45 <tr>
46 <td colspan="3">
47 <!-- PARTNER'S CV SECTION -->
48 <div class="border-black border-4 rounded-2xl text-black pb-5">
49 <h3 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
50 CV Created Candidate list
51 </h3>
52 <div class="h-96 overflow-y-auto scrollbar">
53 <table class="w-full table-auto bg-white text-black">
54 <thead>
55 <tr>
56 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
57 Photo
58 </th>
59 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
60 Name
61 </th>
62 </tr>
63 </thead>
64 <tbody>
65 <tr>
66 <td colspan="2">
67 Partner ID: <span class="font-bold">{{ $partner -> partner_id }}</span>
68 </td>
69 <td colspan="2">
70 Partner Name: <span class="font-bold">{{ $partner -> partner_name }}</span>
71 </td>
72 <td colspan="2">
73 Partner Email: <span class="font-bold">{{ $partner -> email }}</span>
74 </td>
75 <td colspan="2">
76 Partner Address: <span class="font-bold">{{ $partner -> partner_address }}</span>
77 </td>
78 <td colspan="2">
79 Partner Nation: <span
80 class="font-bold">{{ $partner -> partner_nationality }}</span>
81 </td>
82 </tr>
83 </tbody>
84 </table>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>

```

```

admin-show-partner.blade.php X
how-partner.blade.php > x-layout-admin > div.flex.justify-center.mb-20 > div.w-5/6 > section.flex.justify-between.mb-20
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <!-- UPPER SECTION -->
5 <section class="m-5 flex flex-col justify-center">
6 <div class="flex justify-center">
7 <div class="w-3/4">
8 <!-- PARTNER SECTION -->
9 <div class="border-black border-4 rounded-2xl text-black overflow-hidden flex flex-col mb-10">
10 <div class="text-2xl font-bold text-center bg-gray-300 max-h border-b-2 p-3 text-center">
11 General Information
12 </div>
13 <div class="p-3">
14 <table border="1">
15 <thead>
16 <tr>
17 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
18 Language Level
19 </th>
20 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
21 Job Type
22 </th>
23 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
24 More Info
25 </th>
26 </thead>
27 <tbody>
28 <tr>
29 <td colspan="3">
30 <!-- PARTNER'S CV SECTION -->
31 <div class="border-black border-4 rounded-2xl text-black pb-5">
32 <h3 class="text-center font-bold text-lg text-white bg-black border-b-black p-5">
33 CV Created Candidate list
34 </h3>
35 <div class="h-96 overflow-y-auto scrollbar">
36 <table class="w-full table-auto bg-white text-black">
37 <thead>
38 <tr>
39 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
40 Photo
41 </th>
42 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
43 Name
44 </th>
45 </tr>
46 </thead>
47 <tbody>
48 <tr>
49 <td colspan="2">
50 Partner ID: <span class="font-bold">{{ $partner -> partner_id }}</span>
51 </td>
52 <td colspan="2">
53 Partner Name: <span class="font-bold">{{ $partner -> partner_name }}</span>
54 </td>
55 <td colspan="2">
56 Partner Email: <span class="font-bold">{{ $partner -> email }}</span>
57 </td>
58 <td colspan="2">
59 Partner Address: <span class="font-bold">{{ $partner -> partner_address }}</span>
60 </td>
61 <td colspan="2">
62 Partner Nation: <span
63 class="font-bold">{{ $partner -> partner_nationality }}</span>
64 </td>
65 </tr>
66 </tbody>
67 </table>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>

```

Figure: 11.9.14. admin-show-partner.blade.php

Admin Side Applicant Evaluation Page

```

resources > views > company > admin-evaluate.blade.php > x-layout-admin
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <div class="font-bold text-3xl text-center w-5">
8 EVALUATION
9 </div>
10
11 <main class="grid grid-cols-1">
12
13 <!-- CV SECTION -->
14 <div class="col-span-1">
15 @include('components._view-cv')
16 </div>
17
18 <div class="col-span-1">
19
20 <!-- EVALUATION SECTION -->
21 <div class="flex flex-col items-center">
22
23 <div class="p-10 border-2 border-black rounded-2xl">
24
25 @include('components._message')
26
27 <form method="post" action="/admin-applications/{{ $cv->cv_code }}/evaluate"
28 enctype="multipart/form-data">
29 @csrf
30 @method('PUT')
31
32 <div class="font-bold text-xl text-blue-900 mb-10">
33 {{ $cv->cv_name }}
34 </div>
35
36 <div class="mb-10">
37 <label for="cv_status" class="inline-block text-lg mb-2 font-bold">
38 Interview Result Status
39 </label>
40
41 <select name="cv_status" id="cv_status"
42 class="border border-gray-200 rounded-p-2 w-full">
43
44 <option value="{{ $cv->cv_status }}" {{ $cv->cv_status }}</option>
45 <option value="Not Applied" @selected(old('cv_status')=='Applied')>Applied
46 <option value="Not Applied" @selected(old('cv_status')=='Not Applied')>Not
47 Applied
48 <option value="Pre-Interview Pass"
49 @selected(old('cv_status')=='Pre-Interview Pass')>
50 Pre-Interview Pass</option>
51 <option value="Final-Interview Pass"
52 @selected(old('cv_status')=='Final-Interview Pass')>
53 Final-Interview Pass</option>
54 <option value="Final Pass" @selected(old('cv_status')=='Left')>Left</option>
55
56
57
58
59
60
61
62
63
64 </div>
65
66 <div class="flex justify-center">
67
68 <button type="submit" class="border-black border-2 bg-black text-white
69 rounded py-2 px-4 hover:bg-white hover:text-black">
70 Evaluate
71 </button>
72
73 </div>
74
75 </div>
76
77 </div>
78
79 </div>
80
81 </div>
82
83 </div>
84
85 </div>
86
87 <a href="/admin-applications" class="transition duration-200 w-5
88 inline-block font-bold text-white bg-black py-2 px-4
89 rounded-xl uppercase">
90 Back to Application Page
91 </a>
92 </div>
93
94 </div>
95
96 </div>
97
98 </main>
99
100 </div>
101
102 </x-layout-admin>

```

Figure: 11.9.16. admin-evaluation.blade.php

Admin side Applicant CV View Page

```

resources > views > company > admin-show-cv.blade.php > ...
1 <x-layout-admin>
2
3 <div class="flex justify-center mb-20">
4
5 <div class="w-5/6">
6
7 <div class="flex justify-between mb-10">
8
9 <a href="/all-cv" class="mt-5 ml-5 inline-block border-2 font-bold
10 border-black text-white bg-black py-1 px-2 rounded-xl uppercase hover:text-white
11 shadow-lg">
12 <i class="fa-solid fa-arrow-left"></i> Go to All CVs
13 </a>
14 </div>
15
16 <!-- CV SECTION -->
17 @include('components._view-cv')
18
19 </div>
20
21 </div>
22
23 </x-layout-admin>
24
25
26
27

```

Figure: 11.9.17. admin-show-cv.blade.php

Admin Side All Passed CV Page

```

resources > views > company > admin-completed.blade.php > x-layout-admin
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <div class="m-5 flex flex-col justify-center">
5 <div class="border-black border-4 rounded-2xl text-black pb-5">
6 <div class="grid grid-cols-1 lg:grid-cols-8 items-center mb-5">
7 <div class="col-span-3 flex justify-left m-5">
8 <div class="text-3xl font-bold">
9 Final Passed CV List
10 </div>
11 </div>
12 </div>
13 <div class="text-3xl font-bold">
14 Total Passed: <span class="text-blue-500">{{ $totalPassed }}</span>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 <div class="border-black border-4 rounded-2xl text-black pb-5">
25 <div class="text-center font-bold text-2xl text-white bg-black border-b-black p-5">
26 Interview Passed CV List
27 </div>
28 <div class="h-screen overflow-y-auto scrollbar">
29 <table class="w-full table-auto bg-white text-black">
30 <thead>
31 <tr>
32 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
33 Passed Job
34 </th>
35 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
36 Photo
37 </th>
38 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
39 Name
40 </th>
41 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
42 Job Type
43 </th>
44 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
45 School
46 </th>
47 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
48 Nationality
49 </th>
50 </tr>
51 <tbody>
52 <tr>
53 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
54 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
55 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
56 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
57 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
58 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
59 </tr>
60 </tbody>
61 </table>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </x-layout-admin>
132

```

```

resources > views > company > admin-completed.blade.php > x-layout-admin
1 <x-layout-admin>
2 <div class="flex justify-center mb-20">
3 <div class="w-5/6">
4 <div class="m-5 flex flex-col justify-center">
5 <div class="border-black border-4 rounded-2xl text-black pb-5">
6 <div class="grid grid-cols-1 lg:grid-cols-8 items-center mb-5">
7 <div class="col-span-3 flex justify-left m-5">
8 <div class="text-3xl font-bold">
9 Final Passed CV List
10 </div>
11 </div>
12 </div>
13 <div class="text-3xl font-bold">
14 Total Passed: <span class="text-blue-500">{{ $totalPassed }}</span>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 <div class="border-black border-4 rounded-2xl text-black pb-5">
25 <div class="text-center font-bold text-2xl text-white bg-black border-b-black p-5">
26 Interview Passed CV List
27 </div>
28 <div class="h-screen overflow-y-auto scrollbar">
29 <table class="w-full table-auto bg-white text-black">
30 <thead>
31 <tr>
32 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
33 Passed Job
34 </th>
35 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
36 Photo
37 </th>
38 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
39 Name
40 </th>
41 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
42 Job Type
43 </th>
44 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
45 School
46 </th>
47 <th class="px-4 py-8 border-t border-b border-gray-300 text-center">
48 Nationality
49 </th>
50 </tr>
51 <tbody>
52 <tr>
53 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
54 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
55 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
56 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
57 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
58 <td class="px-4 py-8 border-t border-b border-gray-300 text-center">
59 </tr>
60 </tbody>
61 </table>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </x-layout-admin>
132

```

Figure: 11.9.19. admin-completed.blade.php

11.10. Routes

The web.php file which contains routes can be found in the routes folder.

```
web.php X
routes > web.php > ...
1 <?php
2
3 use App\Http\Controllers\CVController;
4 use App\Http\Controllers\JoblistingController;
5 use App\Http\Controllers\DashboardController;
6 use App\Http\Controllers\AdminJoblistingController;
7 use App\Http\Controllers\AdminCVController;
8 use App\Http\Controllers\AdminUserController;
9 use App\Http\Controllers\PartnerController;
10 use App\Http\Controllers\ApplicationController;
11 use App\Http\Controllers\VPDFController;
12 use Illuminate\Support\Facades\Route;
13 use Illuminate\Support\Facades\Cache;
14 use App\Models\CVForas;
15 use App\Models\Joblisting;
16 use App\Models\Partner;
17
18 //Log In Log Out
19
20 // Log IN CHOOSE Page
21 Route::get('/', function () {
22     return view('login.login-choose');
23 });
24
25 // Log IN ADMIN Page
26 Route::get('/admin-login', [AdminUserController::class, 'login']);
27
28 // Log IN ADMIN Page ROUTE
29 Route::post('/admin/authenticate', [AdminUserController::class, 'adminlogin']);
30
31 // Log OUT ADMIN
32 Route::post('/admin/logout', [AdminUserController::class, 'logout']);
33
34 // Log IN PARTNER Page view
35 Route::get('/partner-login', [PartnerController::class, 'login']);
36
37 // Log IN PARTNER Page ROUTE
38 Route::post('/partner/authenticate', [PartnerController::class, 'partnerlogin']);
39
40 // Log OUT PARTNER
41 Route::post('/partner/logout', [PartnerController::class, 'logout']);
42
43
44
45
46
47 //PARTNER Side
48
49 // PARTNER CV
50
51 // Partner Full CV List
52 Route::get('/home', [CVController::class, 'index']);
53
54 // Partner CV View
55 Route::get('/partner-view-cv/{cv_code}', [CVController::class, 'viewcv']);
56
57 // Partner CV Edit FORM
58
59 Route::get('/partner-view-cv/{cv_code}/edit', [CVController::class, 'edit']);
60
61 // Partner CV Edit Submit and Update
62 Route::put('/partner-view-cv/{cv_code}', [CVController::class, 'update']);
63
64 // Partner CV Registration
65 Route::get('/new-cv-register', [CVController::class, 'create']);
66
67 // Partner New CV Storage
68 Route::post('/home', [CVController::class, 'store']);
69
70 //PARTNER JOB
71
72 // Partner Job Listing
73 Route::get('/apply-jobs', [JoblistingController::class, 'joblist']);
74
75 // Partner Show Job Info
76 Route::get('/job-info/{job_code}', [JoblistingController::class, 'viewjob']);
77
78 // PARTNER JOB APPLICATION
79
80 // Partner Job Application Page View
81 Route::get('/job-application/{job_code}', [ApplicationController::class, 'viewapplyjob']);
82
83 // Partner Job Application Add
84 Route::post('/job-application/{job_id}/add/{cv_id}', [ApplicationController::class, 'addCV']);
85
86 ->name('job.addCV');
87
88
89
90
91 //Partner Job Application Remove
92 Route::delete('/job-application/{job_id}/remove/{cv_id}', [ApplicationController::class, 'removeCV']);
93 ->name('job.removeCV');
94
95 // Partner Applied List
96 Route::get('/partner-applied-list', [ApplicationController::class, 'appliedCV']);
97
98
99 //Admin Side
100
101 // ADMIN DASHBOARD
102
103 Route::get('/admin-home', [DashboardController::class, 'home']);
104
105 Route::get('/admin-home/date-filter', [DashboardController::class, 'dateFilter']);
106
107 Route::get('/chart/jobs-by-type', [DashboardController::class, 'jobsbytype']);
108
109 Route::get('/chart/cvs-by-nationality', [DashboardController::class, 'cvsbynationality']);
110
111 Route::get('/chart/partners-by-nationality', [DashboardController::class, 'partnersbynationality']);
112
113 Route::get('/graph/statistics', [DashboardController::class, 'statistics']);
114
115
116 // ADMIN JOBS
117
118 // Admin All Jobs
119 Route::get('/all-listed-jobs', [AdminJoblistingController::class, 'joblistAdmin']);
120
121 // Admin Show Job
122 Route::get('/all-listed-jobs/{job_code}', [AdminJoblistingController::class, 'viewjobAdmin']);
123
124 // Admin Job Edit
125 Route::get('/all-listed-jobs/{job_code}/edit', [AdminJoblistingController::class, 'edit']);
126
127 // Admin Job Edit and SUBMIT -> UPDATE
128 Route::put('/all-listed-jobs/{job_code}', [AdminJoblistingController::class, 'update']);
129
130
131 // Admin Job Add New
132 Route::get('/add-new-job', [AdminJoblistingController::class, 'add']);
133
134 // Admin New Job Store
135 Route::post('/all-listed-jobs', [AdminJoblistingController::class, 'store']);
136
137 // ADMIN PARTNERS
138
139 // Admin All Partners Show
140 Route::get('/all-partners', [PartnerController::class, 'partner']);
141
142 // Admin Show Partner
143 Route::get('/all-partners/{partner_id}', [PartnerController::class, 'viewPartner']);
144
145 // Admin View Edit Partner
146 Route::get('/all-partners/{partner_id}/edit', [PartnerController::class, 'edit']);
147
148 // Admin Edit and Update Partner
149 Route::put('/all-partners/{partner_id}', [PartnerController::class, 'update']);
150
151 // Admin Add Partner
152 Route::get('/register-new-partner', [PartnerController::class, 'addNew']);
153
154 // Admin Store Partner
155 Route::post('/all-partners', [PartnerController::class, 'store']);
156
157 // ADMIN CV
158
159 // Admin all CV List
160 Route::get('/all-cv', [AdminCVController::class, 'allCV']);
161
162 // Admin Show CV
163 Route::get('/all-cv/{cv_code}', [AdminCVController::class, 'showCV']);
164
165 // Admin CV Completed
166 Route::get('/admin-completed', [AdminCVController::class, 'passedCV']);
167
168 // ADMIN CV/JOB APPLICATION
169
170 // Admin Application
171 Route::get('/admin-applications', [ApplicationController::class, 'AdminAppliedCV']);
172
173 // Admin Show CV Evaluate
174 Route::get('/admin-applications/{cv_code}/evaluate', [AdminCVController::class, 'seeEvaluate']);
175
176 // Admin EVALUATION UPDATE
177 Route::put('/admin-applications/{cv_code}/evaluate', [AdminCVController::class, 'evaluate']);
178
179 // CV/JOB PDF Download
180
181 Route::get('/cv/{cv_code}/pdf', [PDFController::class, 'cvPdf']);
182
183 Route::get('/job/{job_code}/pdf', [PDFController::class, 'jobPdf']);
184
185
186
187
188
189
190
```

Figure: 11.10.1. web.php

11.11. Controllers

The controller files are in the Http\Controllers folder in the app folder. The controller.php serves as the base abstract class for all controllers in the system. It defines the shared structure and namespace for the system's controllers. Other controllers extends this base class to inherit Laravel's default behaviours. Deleting or modifying it would cause all other controllers that depend on it to cease to work.

Controller File

```
Controller.php X
app > Http > Controllers > Controller.php > Controller
1 <?php
2
3 namespace App\Http\Controllers;
4
5 abstract class Controller
6 {
7     //
8 }
9
```

Figure: 11.11.1. Controller.php

Admin CV Controller

```
AdminCVController.php X
app > Http > Controllers > AdminCVController.php > AdminCVController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\CVForms;
6 use App\Models\JobListings;
7 use App\Models\Partner;
8 use Database\Factories\PartnerFactory;
9 use Illuminate\Http\Request;
10 use Illuminate\Validation\Rule;
11 use Illuminate\Support\Facades\Auth;
12
13 class AdminCVController extends Controller
14 {
15     //Admin -> Show All CVs
16     public function allCV()
17     {
18         $filters = request([
19             'cv_name',
20             'cv_ip_level',
21             'cv_job_type',
22             'cv_nationality',
23             'job_optionality',
24             'cv_status',
25             'search'
26         ]);
27
28         $cvcs = CVForms::with(['Partner'])
29             ->has('partner')->filter($filters)
30             ->get();
31
32         return view('company.admin-all-cv', [
33             'cvcs' => $cvcs
34         ]);
35     }
36
37     //Admin -> Show Selected CV
38     public function showCV($cv_code) {
39
40         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
41
42         return view('company.admin-show-cv', [
43             'cv' => $cv
44         ]);
45     }
46
47 }
48
49 //Admin -> Show all 'Final Interview Passed' CVs
50
51 class AdminCVController extends Controller
52 {
53     //Admin -> Show all 'Final Interview Passed' CVs
54     public function passedCV()
55     {
56         $jobs = JobListings::with(['application' => function ($query){
57             $query->whereHas('cv', function ($cvQuery) {
58                 $cvQuery->where('cv_status', 'Final-Interview Pass');
59             });
60             $application->where('cv_partner')->get();
61         }]);
62         $totalPassed = CVForms::where('cv_status', 'Final-Interview Pass')
63             ->count();
64
65         return view('company.admin-completed', [
66             'jobs' => $jobs,
67             'totalPassed' => $totalPassed
68         ]);
69     }
70
71     //Admin -> See Evaluate CV Page
72     public function seeEvaluate($cv_code){
73
74         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
75
76         return view('company.admin-evaluate', [
77             'cv' => $cv
78         ]);
79     }
80
81     //Admin -> Store Evaluated Message
82     public function evaluate(Request $request, $cv_code){
83
84         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
85
86         $formFields = $request->validate([
87             'cv_status' => 'required',
88             'cv_evaluation' => 'nullable|string|max:255'
89         ]);
90
91         $cv->update($formFields);
92
93         return back()
94             ->with('message', 'CV Evaluated!');
95     }
96 }
```

Figure: 11.11.2. AdminCVController.php

Admin Joblisting Controller

```
AdminJoblistingController.php X
app > Http > Controllers > AdminJoblistingController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Joblisting;
6 use Illuminate\Http\Request;
7 use Illuminate\Validation\Rule;
8 use Illuminate\Support\Facades\Auth;
9
10 class AdminJoblistingController extends Controller
11 {
12     //Admin -> Show All Jobs
13     public function joblistAdmin(){
14
15         $filters = request(
16             [
17                 'company_name',
18                 'job_type',
19                 'job_title',
20                 'required_ip_level',
21                 'job_nationality',
22                 'search'
23             ]
24         );
25
26         //Filters -> Check Joblisting Model
27         $jobs = Joblisting::latest()->filter($filters)->get();
28
29         return view('company.admin-job', [
30             'jobs' => $jobs,
31         ]);
32     }
33
34
35     //Admin -> Show Selected Job
36     public function viewJobAdmin($job_code){
37
38         $job = Joblisting::where('job_code', $job_code)->firstOrFail();
39
40
41         return view('company.admin-show-job', [
42             'job' => $job
43         ]);
44     }
45
46
47
48
49     //Admin -> Edit Selected Job
50     public function edit($job_code){
51
52         $job = Joblisting::where('job_code', $job_code)->firstOrFail();
53
54
55         return view('company.admin-edit-job', [
56             'job' => $job
57         ]);
58     }
59
60
61     //Admin -> Update Edited Job
62     public function update(Request $request, $job_code){
63
64         $job = Joblisting::where('job_code', $job_code)->firstOrFail();
65
66         $formFields = $request->validate([
67             'recruit_number' => 'required',
68             'application_deadline' => 'required',
69             'company_name' => 'required',
70             'company_website' => 'required',
71             'company_location' => 'required',
72             'job_title' => 'required',
73             'job_description' => 'required',
74             'work_time' => 'required',
75             'off_days' => 'required',
76             'salary_benefits' => 'required',
77             'deductables' => 'required',
78             'after_deduction' => 'required',
79             'airticket_exp' => 'required',
80             'required_ip_level' => 'required',
81             'age_range' => 'required',
82             'job_gender' => 'required',
83             'job_nationality' => 'required',
84             'other_requirements' => 'required',
85             'job_availability' => 'required'
86         ]);
87
88         $job->update($formFields);
89
90         return back();
91     }
92
93     //Admin -> View Add New Job Page
94     public function add(){
95
96         return view('company.admin-add-new-job');
97     }
98
99
100     //Admin -> Store Added New Job Data
101     public function store(Request $request){
102
103         $formFields = $request->validate([
104             'job_type' => 'required',
105             'recruit_number' => 'required',
106             'application_deadline' => 'required',
107             'company_name' => 'required',
108             'company_website' => 'required',
109             'company_location' => 'required',
110             'job_title' => 'required',
111             'job_description' => 'required',
112             'work_time' => 'required',
113             'off_days' => 'required',
114             'salary_benefits' => 'required',
115             'deductables' => 'required',
116             'after_deduction' => 'required',
117             'airticket_exp' => 'required',
118             'required_ip_level' => 'required',
119             'age_range' => 'required',
120             'job_gender' => 'required',
121             'job_nationality' => 'required',
122             'other_requirements' => 'required',
123         ]);
124
125         $jobs = Joblisting::create($formFields);
126         $prefix = strtoupper(substr($jobs->job_type, 0, 3));
127
128         $jobs->job_code = 'JOB - ' .
129             str_pad($jobs->id, 4, '0', STR_PAD_LEFT) . '-' . $prefix;
130
131         $jobs->save();
132
133         return redirect('/all-listed-jobs')
134             ->with('message', 'Job Created!');
135     }
136
137
138     public function store(Request $request){
139
140         $formFields = $request->validate([
141             'required_ip_level' => 'required',
142             'age_range' => 'required',
143             'job_gender' => 'required',
144             'job_nationality' => 'required',
145             'other_requirements' => 'required',
146         ]);
147
148         $jobs = Joblisting::create($formFields);
149         $prefix = strtoupper(substr($jobs->job_type, 0, 3));
150
151         $jobs->job_code = 'JOB - ' .
152             str_pad($jobs->id, 4, '0', STR_PAD_LEFT) . '-' . $prefix;
153
154         $jobs->save();
155
156         return redirect('/all-listed-jobs')
157             ->with('message', 'Job Created!');
158     }
159
160
161     }
162 }
```

```
AdminJoblistingController.php X
app > Http > Controllers > AdminJoblistingController.php > ...
10 class AdminJoblistingController extends Controller
61 public function update(Request $request, $job code){
86
87     $job->update($formFields);
88
89     return back();
90     ->with('message', 'Job Updated!');
91 }
92
93 //Admin -> View Add New Job Page
94 public function add(){
95
96     return view('company.admin-add-new-job');
97 }
98
99
100 //Admin -> Store Added New Job Data
101 public function store(Request $request){
102
103     $formFields = $request->validate([
104         'job_type' => 'required',
105         'recruit_number' => 'required',
106         'application_deadline' => 'required',
107         'company_name' => 'required',
108         'company_website' => 'required',
109         'company_location' => 'required',
110         'job_title' => 'required',
111         'job_description' => 'required',
112         'work_time' => 'required',
113         'off_days' => 'required',
114         'salary_benefits' => 'required',
115         'deductables' => 'required',
116         'after_deduction' => 'required',
117         'airticket_exp' => 'required',
118         'required_ip_level' => 'required',
119         'age_range' => 'required',
120         'job_gender' => 'required',
121         'job_nationality' => 'required',
122         'other_requirements' => 'required',
123     ]);
124
125     $jobs = Joblisting::create($formFields);
126     $prefix = strtoupper(substr($jobs->job_type, 0, 3));
127
128     $jobs->job_code = 'JOB - ' .
129         str_pad($jobs->id, 4, '0', STR_PAD_LEFT) . '-' . $prefix;
130
131     $jobs->save();
132
133     return redirect('/all-listed-jobs')
134         ->with('message', 'Job Created!');
135 }
136
137
138     public function store(Request $request){
139
140         $formFields = $request->validate([
141             'required_ip_level' => 'required',
142             'age_range' => 'required',
143             'job_gender' => 'required',
144             'job_nationality' => 'required',
145             'other_requirements' => 'required',
146         ]);
147
148         $jobs = Joblisting::create($formFields);
149         $prefix = strtoupper(substr($jobs->job_type, 0, 3));
150
151         $jobs->job_code = 'JOB - ' .
152             str_pad($jobs->id, 4, '0', STR_PAD_LEFT) . '-' . $prefix;
153
154         $jobs->save();
155
156         return redirect('/all-listed-jobs')
157             ->with('message', 'Job Created!');
158     }
159
160
161     }
162 }
```

Figure: 11.11.3. AdminJoblistingController.php

Admin User Controller

```
AdminUserController.php X
app > Http > Controllers > AdminUserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\User;
6  use Illuminate\Http\Request;
7  use Illuminate\Validation\Rule;
8  use Illuminate\Support\Facades\Auth;
9
10 class AdminUserController extends Controller
11 {
12     //View Admin Log In Page
13     public function login(){
14         return view('login.login-admin');
15     }
16
17     //Authenticate Admin
18     public function adminLogin(Request $request){
19
20         $formFields = $request->validate([
21             'email' => ['required', 'email'],
22             'password' => 'required',
23         ]);
24
25         if (Auth::guard('web')->attempt($formFields, $request->filled('remember'))) {
26
27             $request->session()->regenerate();
28             return redirect('/admin-home') -> with('message', 'You have logged in!');
29         }
30
31         return back()->withErrors(['email' => 'Invalid Credentials']) -> onlyInput('email');
32     }
33
34
35     //Admin Log Out
36     public function logout(Request $request){
37
38         Auth::guard('web')->logout();
39
40         $request->session()->invalidate();
41
42         $request->session()->regenerateToken();
43
44         return redirect('/')->with('message', 'You Have Been Logged Out');
45     }
46 }
47
48 }
49
```

Figure: 11.11.4. AdminUserController.php

Application Controller

```

ApplicationController.php X
app > Http > Controllers > ApplicationController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5
6 use App\Models\Application;
7 use App\Models\CVForms;
8 use App\Models\JobListing;
9 use Illuminate\Http\Request;
10
11 class ApplicationController extends Controller
12 {
13
14     //Partner -> View Apply Job Page
15     public function viewapplyjob($job_code){
16
17         $partnerId = auth('partner')->id();
18
19         $job = JobListing::where('job_code', $job_code)->firstOrFail();
20
21         $availableCvs = CVForms::where('partner_id', $partnerId)
22             ->where('cv_status', 'Not Applied')
23             ->where('cv_job_type', $job->job_type)
24             ->get();
25
26         $appliedCvs = $job->cvs()->where('partner_id', $partnerId)->get();
27
28         return view('partner.partner-job-application', [
29             'job' => $job,
30             'availableCvs' => $availableCvs,
31             'appliedCvs' => $appliedCvs
32         ]);
33     }
34
35     //Partner -> Add CV in Apply Job Page
36     public function addCV($job_id, $cv_id){
37
38         $cv = CVForms::where('id', $cv_id)->where('cv_status', 'Not Applied')
39             ->first();
40
41         Application::create([
42             'job_id' => $job_id,
43             'cv_id' => $cv_id,
44         ]);
45
46         $cv->update(['cv_status' => 'Applied']);
47
48         return back()->with('message', $cv->cv_name . ' added to job successfully!');
49     }
50
51
52
53     //Partner -> Remove CV in Apply Job Page
54     public function removeCV($job_id, $cv_id){
55
56         $application = Application::where('job_id', $job_id)
57             ->where('cv_id', $cv_id)
58             ->first();
59
60         if (!$application) {
61             return back()->with('error', 'Application not found. ');
62         }
63
64         $application->delete();
65
66         $cv = CVForms::find($cv_id);
67         $cv->update(['cv_status' => 'Not Applied']);
68
69         return back()->with('message', $cv->cv_name . ' removed from job successfully!');
70     }
71
72
73     //Partner -> View Applied CVs and Job
74     public function appliedCV(){
75
76         $partnerId = auth('partner')->id();
77
78         $filters = request([
79             'job_code',
80             'company_name',
81             'search'
82         ]);
83
84         $jobs = JobListing::whereHas('application.cv', function ($query) use ($partnerId)
85             {
86                 $query->where('partner_id', $partnerId);
87             })
88             ->with(['application' => function ($query) use ($partnerId) {
89                 $query->whereHas('cv', function ($q) use ($partnerId)
90                     {
91                         $q->where('partner_id', $partnerId);
92                     })
93                     ->with('cv');
94             })
95             ->latest()->filter($filters)->paginate(5);
96
97         return view('partner.partner-applied-job', [
98
ApplicationController.php X
app > Http > Controllers > ApplicationController.php > ApplicationController > appliedCV
11 class ApplicationController extends Controller
74 public function appliedCV(){
94     })
95     ->latest()->filter($filters)->paginate(5);
96
97     return view('partner.partner-applied-job', [
98         'jobs' => $jobs
99     ]);
100
101 }
102
103 //Admin -> View Applied CVs and Jobs
104 public function AdminAppliedCV(){
105
106     $filters = request([
107         'job_code',
108         'company_name',
109         'search'
110     ]);
111
112     $jobs = JobListing::with(['application.cv'])
113         ->has('application')->latest()
114         ->filter($filters)->paginate(5);
115
116     return view('company.admin-application', [
117         'jobs' => $jobs
118     ]);
119
120 }
121
122 }
123

```

Figure: 11.11.5. ApplicationController.php

CV Controller

```
CVController.php X
app > Http > Controllers > CVController.php > ...
1 c.php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Auth;
5 use Illuminate\Support\Facades\Hash;
6 use Illuminate\Support\Facades\RateLimiter;
7 use Illuminate\Support\Facades\Redis;
8 use Illuminate\Support\Facades\URL;
9 use Illuminate\Support\Facades\Validator;
10
11 class CVController extends Controller
12 {
13     //Partner -> Default Home Page -> Show CV Forms Created by Partner
14     public function index()
15     {
16         $filters = request();
17         $partnerID = auth('partner')->id();
18         $cv = CVForms::where('partner_id', $partnerID)->latest()
19             ->filter($filters)->get();
20         $spoused = CVForms::where('partner_id', $partnerID)->where('cv_status',
21             'Final-Interview Pass')->count();
22         $total = $cv->count();
23         return view('partner.partner-home', [
24             'cv' => $cv,
25             'total' => $total,
26             'spoused' => $spoused,
27         ]);
28     }
29
30     //Partner -> View Single CV
31     public function view($cv_code)
32     {
33         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
34         return view('partner.partner-view-cv', [
35             'cv' => $cv
36         ]);
37     }
38
39     //Partner -> Edit Single CV
40     public function edit($cv_code)
41     {
42         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
43         return view('partner.partner-edit-cv', [
44             'cv' => $cv
45         ]);
46     }
47
48     //Partner -> Update New Edited CV
49     public function update(Request $request, $cv_code)
50     {
51         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
52         $formFields = $request->validate([
53             'cv_profile' => 'nullable|file|mimes:jpg,svg,png|max:2048',
54             'cv_name' => 'required|string|max:255',
55             'cv_age' => 'required|integer|min:15|max:100',
56             'cv_dob' => 'required|date',
57             'cv_address' => 'required|string|max:255',
58             'cv_religion' => 'required|string|max:100',
59             'cv_marriage' => 'required|string|max:50',
60             'cv_job_type' => 'required|string|max:255',
61             'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
62             'cv_ip_level' => 'required|string|max:50',
63             'cv_ip_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
64             'cv_ip_study_year' => 'required|integer|min:0|max:50',
65             'cv_ip_study_month' => 'required|integer|min:0|max:11',
66             'cv_desired_work_length' => 'required|string|max:100',
67         ]);
68         //SCHOOL HISTORY
69         'cv_schoolhistory1_name' => 'required|string|max:255',
70         'cv_schoolhistory1_subject' => 'required|string|max:255',
71         'cv_schoolhistory1_start' => 'required|date',
72         'cv_schoolhistory1_end' => 'required|date|after_or_equal:cv_schoolhistory1_start',
73         'cv_schoolhistory1_status' => 'required|string|max:100',
74         'cv_schoolhistory2_name' => 'nullable|string|max:255',
75         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
76         'cv_schoolhistory2_start' => 'nullable|date',
77         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal:cv_schoolhistory2_start',
78         'cv_schoolhistory2_status' => 'nullable|string|max:100',
79         'cv_schoolhistory3_name' => 'nullable|string|max:255',
80         'cv_schoolhistory3_subject' => 'nullable|string|max:255',
81         'cv_schoolhistory3_start' => 'nullable|date',
82         'cv_schoolhistory3_end' => 'nullable|date|after_or_equal:cv_schoolhistory3_start',
83         'cv_schoolhistory3_status' => 'nullable|string|max:100',
84         //JOB HISTORY
85         'cv_jobhistory1_name' => 'nullable|string|max:255',
86         'cv_jobhistory1_position' => 'nullable|string|max:255',
87         'cv_jobhistory1_start' => 'nullable|date',
88         'cv_jobhistory1_end' => 'nullable|date|after_or_equal:cv_jobhistory1_start',
89         'cv_jobhistory1_status' => 'nullable|string|max:100',
90         'cv_jobhistory2_name' => 'nullable|string|max:255',
91         'cv_jobhistory2_position' => 'nullable|string|max:255',
92     }
93
94     //Partner -> Create New CV
95     public function create()
96     {
97         return view('partner.partner-new-cv-register');
98     }
99
100     //Partner -> Store New CV
101     public function store(Request $request)
102     {
103         $formFields = $request->validate([
104             'cv_profile' => 'nullable|file|mimes:jpg,svg,png|max:2048',
105             'cv_name' => 'required|string|max:255',
106             'cv_gender' => 'required|string|max:10',
107             'cv_age' => 'required|integer|min:15|max:100',
108             'cv_dob' => 'required|date',
109             'cv_nationality' => 'required|string|max:100',
110             'cv_address' => 'required|string|max:255',
111             'cv_religion' => 'required|string|max:100',
112             'cv_marriage' => 'required|string|max:50',
113             'cv_job_type' => 'required|string|max:255',
114             'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
115             'cv_ip_level' => 'required|string|max:50',
116             'cv_ip_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
117             'cv_ip_study_year' => 'required|integer|min:0|max:50',
118             'cv_ip_study_month' => 'required|integer|min:0|max:11',
119             'cv_desired_work_length' => 'required|string|max:100',
120         ]);
121         //SCHOOL HISTORY
122         'cv_schoolhistory1_name' => 'required|string|max:255',
123         'cv_schoolhistory1_subject' => 'required|string|max:255',
124         'cv_schoolhistory1_start' => 'required|date',
125         'cv_schoolhistory1_end' => 'required|date|after_or_equal:cv_schoolhistory1_start',
126         'cv_schoolhistory1_status' => 'required|string|max:100',
127         'cv_schoolhistory2_name' => 'nullable|string|max:255',
128         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
129         'cv_schoolhistory2_start' => 'nullable|date',
130         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal:cv_schoolhistory2_start',
131         'cv_schoolhistory2_status' => 'nullable|string|max:100',
132         'cv_schoolhistory3_name' => 'nullable|string|max:255',
133         'cv_schoolhistory3_subject' => 'nullable|string|max:255',
134         'cv_schoolhistory3_start' => 'nullable|date',
135         'cv_schoolhistory3_end' => 'nullable|date|after_or_equal:cv_schoolhistory3_start',
136         'cv_schoolhistory3_status' => 'nullable|string|max:100',
137         //JOB HISTORY
138         'cv_jobhistory1_name' => 'nullable|string|max:255',
139         'cv_jobhistory1_position' => 'nullable|string|max:255',
140         'cv_jobhistory1_start' => 'nullable|date',
141         'cv_jobhistory1_end' => 'nullable|date|after_or_equal:cv_jobhistory1_start',
142         'cv_jobhistory1_status' => 'nullable|string|max:100',
143         'cv_jobhistory2_name' => 'nullable|string|max:255',
144         'cv_jobhistory2_position' => 'nullable|string|max:255',
145     }
146
147     //Partner -> View Create New CV Page
148     public function create()
149     {
150     }
151 }
152
153 }
```

```
CVController.php X
app > Http > Controllers > CVController.php > ...
10 class CVController extends Controller
11 {
12     //Partner -> Update New Edited CV
13     public function update(Request $request, $cv_code)
14     {
15         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
16         $formFields = $request->validate([
17             'cv_profile' => 'nullable|file|mimes:jpg,svg,png|max:2048',
18             'cv_name' => 'required|string|max:255',
19             'cv_age' => 'required|integer|min:15|max:100',
20             'cv_dob' => 'required|date',
21             'cv_address' => 'required|string|max:255',
22             'cv_religion' => 'required|string|max:100',
23             'cv_marriage' => 'required|string|max:50',
24             'cv_job_type' => 'required|string|max:255',
25             'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
26             'cv_ip_level' => 'required|string|max:50',
27             'cv_ip_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
28             'cv_ip_study_year' => 'required|integer|min:0|max:50',
29             'cv_ip_study_month' => 'required|integer|min:0|max:11',
30             'cv_desired_work_length' => 'required|string|max:100',
31         ]);
32         //SCHOOL HISTORY
33         'cv_schoolhistory1_name' => 'required|string|max:255',
34         'cv_schoolhistory1_subject' => 'required|string|max:255',
35         'cv_schoolhistory1_start' => 'required|date',
36         'cv_schoolhistory1_end' => 'required|date|after_or_equal:cv_schoolhistory1_start',
37         'cv_schoolhistory1_status' => 'required|string|max:100',
38         'cv_schoolhistory2_name' => 'nullable|string|max:255',
39         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
40         'cv_schoolhistory2_start' => 'nullable|date',
41         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal:cv_schoolhistory2_start',
42         'cv_schoolhistory2_status' => 'nullable|string|max:100',
43         'cv_schoolhistory3_name' => 'nullable|string|max:255',
44         'cv_schoolhistory3_subject' => 'nullable|string|max:255',
45         'cv_schoolhistory3_start' => 'nullable|date',
46         'cv_schoolhistory3_end' => 'nullable|date|after_or_equal:cv_schoolhistory3_start',
47         'cv_schoolhistory3_status' => 'nullable|string|max:100',
48         //JOB HISTORY
49         'cv_jobhistory1_name' => 'nullable|string|max:255',
50         'cv_jobhistory1_position' => 'nullable|string|max:255',
51         'cv_jobhistory1_start' => 'nullable|date',
52         'cv_jobhistory1_end' => 'nullable|date|after_or_equal:cv_jobhistory1_start',
53         'cv_jobhistory1_status' => 'nullable|string|max:100',
54         'cv_jobhistory2_name' => 'nullable|string|max:255',
55         'cv_jobhistory2_position' => 'nullable|string|max:255',
56     }
57
58     //Partner -> Create New CV
59     public function create()
60     {
61         return view('partner.partner-new-cv-register');
62     }
63
64     //Partner -> Store New CV
65     public function store(Request $request)
66     {
67         $formFields = $request->validate([
68             'cv_profile' => 'nullable|file|mimes:jpg,svg,png|max:2048',
69             'cv_name' => 'required|string|max:255',
70             'cv_gender' => 'required|string|max:10',
71             'cv_age' => 'required|integer|min:15|max:100',
72             'cv_dob' => 'required|date',
73             'cv_nationality' => 'required|string|max:100',
74             'cv_address' => 'required|string|max:255',
75             'cv_religion' => 'required|string|max:100',
76             'cv_marriage' => 'required|string|max:50',
77             'cv_job_type' => 'required|string|max:255',
78             'cv_job_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
79             'cv_ip_level' => 'required|string|max:50',
80             'cv_ip_certificate' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
81             'cv_ip_study_year' => 'required|integer|min:0|max:50',
82             'cv_ip_study_month' => 'required|integer|min:0|max:11',
83             'cv_desired_work_length' => 'required|string|max:100',
84         ]);
85         //SCHOOL HISTORY
86         'cv_schoolhistory1_name' => 'required|string|max:255',
87         'cv_schoolhistory1_subject' => 'required|string|max:255',
88         'cv_schoolhistory1_start' => 'required|date',
89         'cv_schoolhistory1_end' => 'required|date|after_or_equal:cv_schoolhistory1_start',
90         'cv_schoolhistory1_status' => 'required|string|max:100',
91         'cv_schoolhistory2_name' => 'nullable|string|max:255',
92         'cv_schoolhistory2_subject' => 'nullable|string|max:255',
93         'cv_schoolhistory2_start' => 'nullable|date',
94         'cv_schoolhistory2_end' => 'nullable|date|after_or_equal:cv_schoolhistory2_start',
95         'cv_schoolhistory2_status' => 'nullable|string|max:100',
96         'cv_schoolhistory3_name' => 'nullable|string|max:255',
97         'cv_schoolhistory3_subject' => 'nullable|string|max:255',
98         'cv_schoolhistory3_start' => 'nullable|date',
99         'cv_schoolhistory3_end' => 'nullable|date|after_or_equal:cv_schoolhistory3_start',
100         'cv_schoolhistory3_status' => 'nullable|string|max:100',
101         //JOB HISTORY
102         'cv_jobhistory1_name' => 'nullable|string|max:255',
103         'cv_jobhistory1_position' => 'nullable|string|max:255',
104         'cv_jobhistory1_start' => 'nullable|date',
105         'cv_jobhistory1_end' => 'nullable|date|after_or_equal:cv_jobhistory1_start',
106         'cv_jobhistory1_status' => 'nullable|string|max:100',
107         'cv_jobhistory2_name' => 'nullable|string|max:255',
108         'cv_jobhistory2_position' => 'nullable|string|max:255',
109     }
110
111     //Partner -> View Create New CV Page
112     public function create()
113     {
114     }
115 }
116
117 }
```

Figure: 11.11.6. CVController.php 1

```

CVController.php X
app > Http > Controllers > CVController.php > CVController > store
10 class CVController extends Controller
175 public function store(Request $request) {
224     'cv_jobhistory2_start' => 'nullable|date',
225     'cv_jobhistory2_end' => 'nullable|date|after_or_equal:cv_jobhistory2_start',
226     'cv_jobhistory2_status' => 'nullable|string|max:100',
227
228     'cv_jobhistory3_name' => 'nullable|string|max:255',
229     'cv_jobhistory3_position' => 'nullable|string|max:255',
230     'cv_jobhistory3_start' => 'nullable|date',
231     'cv_jobhistory3_end' => 'nullable|date|after_or_equal:cv_jobhistory3_start',
232     'cv_jobhistory3_status' => 'nullable|string|max:100',
233
234     //OTHERS
235     'cv_email' => 'nullable|email|max:255',
236     'cv_phone' => 'nullable|string|max:20',
237
238     'cv_passport_number' => 'nullable|string|max:50',
239     'cv_passport_date' => 'nullable|date',
240     'cv_passport' => 'nullable|file|mimes:pdf,doc,docx|max:5120',
241     'cv_jp_visit_time' => 'nullable|integer|min:0|max:50',
242     'cv_COE_Received' => 'nullable|integer|min:0|max:50',
243     'cv_COE_Rejected' => 'nullable|integer|min:0|max:50',
244
245     });
246
247     if ($request->hasFile('cv_profile')) {
248         $formFields['cv_profile'] = $request->file('cv_profile')
249         ->store('cv/cv_profile', 'public');
250     }
251
252     if ($request->hasFile('cv_job_certificate')) {
253         $formFields['cv_job_certificate'] = $request->file('cv_job_certificate')
254         ->store('cv/job_certificates', 'public');
255     }
256
257     if ($request->hasFile('cv_jp_certificate')) {
258         $formFields['cv_jp_certificate'] = $request->file('cv_jp_certificate')
259         ->store('cv/jp_certificates', 'public');
260     }
261
262     if ($request->hasFile('cv_passport')) {
263         $formFields['cv_passport'] = $request->file('cv_passport')
264         ->store('cv/passports', 'public');
265     }
266
267     $formFields['partner_id'] = auth('partner')->id();
268
269     $cv = CVForms::create($formFields);
270
271     $cv->cv_code = 'CV-' . str_pad($cv->id, 4, '0', STR_PAD_LEFT);
272     $cv->save();
273
274     return redirect('/home')->with('message', 'CV Created!');
275
276
277
278 }

```

Figure: 11.11.7. CVController.php 2

Dashboard Controller

```

DashboardController.php X
app > Http > Controllers > DashboardController.php > DashboardController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\CvForms;
6 use App\Models\Joblisting;
7 use App\Models\Partner;
8 use App\Models\Application;
9 use Illuminate\Support\Facades\Cache;
10 use Carbon\Carbon;
11 use Illuminate\Http\Request;
12 use Illuminate\Validation\Rule;
13 use Illuminate\Support\Facades\Auth;
14
15 class DashboardController extends Controller
16 {
17     //Admin -> Default Home Page -> Dashboard
18     public function home(){
19
20         $totalCVs = CvForms::count();
21         $totalJobs = Joblisting::count();
22         $totalPartners = Partner::count();
23         $totalApplications = Application::count();
24         $totalPassed = CvForms::where('cv_status', 'Final-Interview Pass')->count();
25
26         $start = null;
27         $end = null;
28         $cvs = null;
29         $partners = null;
30         $applications = null;
31         $jobs = null;
32
33         return view('company.admin-home', [
34             'totalCVs' => $totalCVs,
35             'totalPartners' => $totalPartners,
36             'totalJobs' => $totalJobs,
37             'totalApplications' => $totalApplications,
38             'totalPassed' => $totalPassed,
39             'start' => $start,
40             'end' => $end,
41             'cvs' => $cvs,
42             'jobs' => $jobs,
43             'partners' => $partners,
44             'applications' => $applications,
45         ]);
46     }
47
48     //Admin -> Default Home Page -> Barchart
49     public function jobsbytype(){
50
51         //Admin -> Default Home Page -> Barchart
52         public function jobsbytype(){
53
54             return Cache::remember('chart_jobs_by_type', 60, function () {
55                 return Joblisting::selectRaw('job_type, COUNT(*) as count')
56                 ->groupBy('job_type')
57                 ->get();
58             });
59
60             //Admin -> Default Home Page -> Partner Pie Chart
61             public function partnersbynationality(){
62
63                 return Cache::remember('chart_partners_by_nationality', 60, function () {
64                     return Partner::selectRaw('partner_nationality, COUNT(*) as count')
65                     ->groupBy('partner_nationality')
66                     ->get();
67                 });
68
69                 //Admin -> Default Home Page -> CV Pie Chart
70                 public function cvsbynationality(){
71
72                     return CvForms::selectRaw('cv_nationality, COUNT(*) as count')
73                     ->groupBy('cv_nationality')
74                     ->get();
75                 }
76
77                 //Admin -> Default Home Page -> Statistics
78                 public function statistics(){
79
80                     $currentMonth = Carbon::now()->month;
81                     $currentYear = Carbon::now()->year;
82
83                     $stats = [
84
85                         'cv' => [
86                             'monthly' => CvForms::whereMonth('created_at', $currentMonth)
87                             ->whereYear('created_at', $currentYear)
88                             ->count(),
89                             'yearly' => CvForms::whereYear('created_at', $currentYear)
90                             ->count(),
91                         ],
92                         'job' => [

```

Figure: 11.11.8. DashboardController.php 1

```

DashboardController.php X
app > Http > Controllers > DashboardController.php > DashboardController > jobstype
15 class DashboardController extends Controller
16 public function statistics(){
17     job => [
18         'monthly' => Joblisting::whereMonth('created_at', $currentMonth)
19             ->whereYear('created_at', $currentYear)
20             ->count(),
21         'yearly' => Joblisting::whereYear('created_at', $currentYear)
22             ->count(),
23     ],
24     'partner' => [
25         'monthly' => Partner::whereMonth('created_at', $currentMonth)
26             ->whereYear('created_at', $currentYear)
27             ->count(),
28         'yearly' => Partner::whereYear('created_at', $currentYear)
29             ->count(),
30     ],
31     'application' => [
32         'monthly' => Application::whereMonth('created_at', $currentMonth)
33             ->whereYear('created_at', $currentYear)
34             ->count(),
35         'yearly' => Application::whereYear('created_at', $currentYear)
36             ->count(),
37     ],
38 ];
39 return response()->json($stats);
40 }
41 //Admin -> Date Filter
42 public function dateFilter(Request $request){
43     $request -> validate([
44         'start_date' => 'required|date',
45         'end_date' => 'required|date|after_or_equal:start_date'
46     ]);
47     $start = $request -> start_date;
48     $end = $request -> end_date;
49     $cvs = CVForms::whereBetween('created_at', [$start, $end])->count();
50     $jobs = Joblisting::whereBetween('created_at', [$start, $end])->count();
51     $partners = Partner::whereBetween('created_at', [$start, $end])->count();
52     $applications = Application::whereBetween('created_at', [$start, $end])->count();
53     $totalCvs = CVForms::count();
54     $totalJobs = Joblisting::count();
55     $totalPartners = Partner::count();
56     $totalApplications = Application::count();
57     $totalPassed = CVForms::where('cv_status', 'Final-Interview Pass')->count();
58     return view('company.admin-home', [
59         'totalCvs' => $totalCvs,
60         'totalPartners' => $totalPartners,
61         'totalJobs' => $totalJobs,
62         'totalApplications' => $totalApplications,
63         'totalPassed' => $totalPassed,
64         'start' => $start,
65         'end' => $end,
66         'cvs' => $cvs,
67         'jobs' => $jobs,
68         'partners' => $partners,
69         'applications' => $applications,
70     ]);
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }

```

Figure: 11.11.9. DashboardController.php 2

Joblisting Controller

```

JoblistingController.php X
app > Http > Controllers > JoblistingController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Joblisting;
6 use Illuminate\Http\Request;
7 use Illuminate\Validation\Rule;
8 use Illuminate\Support\Facades\Auth;
9
10 class JoblistingController extends Controller
11 {
12     //Partner -> View All Jobs
13     public function joblist(){
14
15         $partner = auth('partner')->user();
16
17         $filters = request(
18             [
19                 'job_type',
20                 'job_title',
21                 'required_jp_level',
22                 'search'];
23
24         //forPartner, filter -> check joblisting Model
25         $jobs = Joblisting::query()->forPartner($partner)
26             ->where('job_availability', 'Available')
27             ->latest()->filter($filters)->paginate(6);
28
29         return view('partner.partner-all-jobs', [
30             'jobs' => $jobs
31         ]);
32     }
33 }
34 //Partner -> View Job
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

Figure: 11.11.10. JoblistingController.php

Partner Controller

```

PartnerController.php X
app > Http > Controllers > PartnerController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\CVForms;
6 use App\Models\partner;
7 use Illuminate\Http\Request;
8 use Illuminate\Validation\Rule;
9 use Illuminate\Support\Facades\Auth;
10 use Illuminate\Support\Facades\Hash;
11 use Illuminate\Support\Facades\Crypt;
12
13 class PartnerController extends Controller
14 {
15     //View Partner Log In Page
16     public function login()
17     {
18         return view('login.login-partner');
19     }
20
21     //Authenticate Partner
22     public function partnerLogin(Request $request){
23
24         $formFields = $request->validate([
25             'email' => ['required', 'email'],
26             'password' => 'required',
27         ]);
28
29         if (Auth::guard('partner')->attempt($formFields, $request->filled('remember'))) {
30
31             $request->session()->regenerateToken();
32             return redirect('/home')->with('message', 'You have logged in!');
33         }
34
35         return back()->withErrors(['email' => 'Invalid Credentials'])->onlyInput('email');
36     }
37
38     //Partner Log Out
39     public function logout(Request $request){
40
41         Auth::guard('partner')->logout();
42     }
43
44
45
46
47
48
49
50
51
52
53
54 //Admin -> Partner Add Page
55 public function addNew(){
56
57     return view('company.admin-add-partner');
58 }
59
60 //Admin -> New Partner Store
61 public function store(Request $request){
62
63     $formFields = $request->validate([
64         'partner_name' => 'required',
65         'email' => ['required', 'email', 'unique:partners,email'],
66         'partner_address' => 'required',
67         'partner_nationality' => 'required',
68         'password' => 'required|min:6',
69     ]);
70
71     $formFields['password'] = Hash::make($formFields['password']);
72
73     $partner = partner::create($formFields);
74
75     $prefix = strtoupper(substr($partner->partner_nationality, 0, 2));
76     $partner->partner_id = 'PI' . $prefix . '-';
77     str_pad($partner->id, 3, '0', STR_PAD_LEFT);
78     $partner->save();
79
80     return redirect('/all-partners')->with('message', 'Partner Created!');
81 }
82
83 //Admin -> Partner Edit Page View
84
85
86
87
88
89
90
91
92
93
94
95 //Admin -> Partner Edit and Update
96 public function update(Request $request, $partner_id){
97
98     $partner = partner::where('partner_id', $partner_id)->firstOrFail();
99
100     $formFields = $request->validate([
101         'partner_name' => 'required',
102         'email' => ['required', 'email'],
103         'partner_address' => 'required',
104         'password' => 'nullable|string|min:6',
105     ]);
106
107     //Dont Overwrite with Null Password
108     if (empty($formFields['password'])) {
109         $formFields['password'] = Hash::make($formFields['password']);
110     } else {
111         unset($formFields['password']);
112     }
113
114     $partner->update($formFields);
115
116     return back()
117         ->with('message', 'Partner Information Updated!');
118 }
119
120 //Admin -> View Partner Info
121 public function viewPartner($partner_id){
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137 //Admin -> View Partner Info
138 public function viewPartner($partner_id){
139
140     $partner = partner::where('partner_id', $partner_id)->firstOrFail();
141     $cvs = $partner->cvs;
142
143     return view('company.admin-show-partner', [
144         'partner' => $partner,
145         'cvs' => $cvs
146     ]);
147 }
148
149 //Admin -> Show All Partners
150 public function partner(){
151
152     $filters = request([
153         'partner_name',
154         'partner_nationality',
155         'search'
156     ]);
157
158     //filter -> Check Partner Model
159     $partners = partner::latest()->filter($filters)->get();
160
161     return view('company.admin-partner', [
162         'partners' => $partners,
163     ]);
164 }
165 }

```

Figure: 11.11.11. PartnerController.php

PDF Controller

```
PDFController.php X
app > Http > Controllers > PDFController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\CVForms;
6  use App\Models\joblisting;
7  use Barryvdh\DomPDF\Facade\Pdf;
8  use Illuminate\Http\Request;
9
10 class PDFController extends Controller
11 {
12     //Generate CV PDF
13     public function cvPdf($cv_code){
14         $cv = CVForms::where('cv_code', $cv_code)->firstOrFail();
15         $pdf = Pdf::setPaper('A4', 'portrait')
16             ->loadView('view-as-pdf.cv', compact('cv'));
17
18         return $pdf->stream("CV-{$cv->cv_code}.pdf");
19     }
20 }
21
22 //Generate JOB PDF
23 public function jobPdf($job_code){
24     $job = joblisting::where('job_code', $job_code)->firstOrFail();
25     $pdf = Pdf::setPaper('A4', 'portrait')
26         ->loadView('view-as-pdf.joblisting', compact('job'));
27
28     return $pdf->stream("Job-{$job->job_code}.pdf");
29 }
30 }
31
32 }
33
```

Figure: 11.11.12. PDFController.php

11.12. PDF View

The pdf file view codes are in the view-as-pdf folder in the views folder, which is in the resources folder.

CV PDF View

```
resources > views > view-as-pdf > cv.blade.php > html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5
6 <meta charset="utf-8">
7 <title>{{ $cv->cv_name }} - Curriculum Vitae</title>
8 <style>
9
10 font-family: DejaVu Sans, sans-serif;
11 background-color: #ffff;
12 display: flex;
13 justify-content: center;
14 font-size: 12px;
15 color: #000;
16
17 h2,
18 h3 {
19 text-align: center;
20 }
21
22 h2 {
23 font-size: 18px;
24 font-weight: bold;
25 margin-top: 40px;
26 margin-bottom: 50px;
27 }
28
29 h3 {
30 font-size: 14px;
31 font-weight: bold;
32 margin-top: 25px;
33 margin-bottom: 10px;
34 }
35
36 table {
37 width: 100%;
38 border-collapse: collapse;
39 margin-bottom: 10px;
40 }
41
42 th,
43 td {
44 border: 1px solid #d1d5db;
```

```
resources > views > view-as-pdf > cv.blade.php > html
2 <html>
4 <head>
7 <style>
41
42 th,
43 td {
44 border: 1px solid #d1d5db;
45 padding: 6px 8px;
46 text-align: left;
47 vertical-align: top;
48 }
49
50 th {
51 background-color: #e5e7eb;
52 font-weight: bold;
53 }
54
55 .font-bold {
56 font-weight: bold;
57 }
58
59 .text-center {
60 text-align: center;
61 }
62
63 .photo {
64 width: 100px;
65 height: 120px;
66 object-fit: cover;
67 border: 1px solid #d1d5db;
68 }
69
70 .photo-cell {
71 text-align: center;
72 vertical-align: top;
73 width: 120px;
74 }
75
76 .evaluation-box {
77 border: 1px solid #d1d5db;
78 height: 200px;
79 padding: 10px;
80 }
81
```

```
resources > views > view-as-pdf > cv.blade.php > html
2 <html>
4 <head>
7 <style>
81
82 </style>
83 </head>
84
85 <body>
86
87 <div>
88
89 <h2>Oshigoto Japan CV Form</h2>
90
91 <!-- Basic Info -->
92 <table>
93
94 <tr>
95 <th>CV Code</th>
96 <td class="font-bold">{{ $cv->cv_code }}</td>
97 <th>Name</th>
98 <td>{{ $cv->cv_name }}</td>
99 <td class="photo-cell" rowspan="6">
100 @if($cv->cv_profile)
101 Photo</p>
103 @else
104 <p><small>No Photo</small></p>
105 @endif
106 </td>
107 </tr>
108 <tr>
109 <th>Registered</th>
110 <td>{{ $cv->created_at }}</td>
111 <th>Gender</th>
112 <td>{{ $cv->cv_gender }}</td>
113 </tr>
114 <tr>
115 <th>Nationality</th>
116 <td>{{ $cv->cv_nationality }}</td>
117 <th>Address</th>
118 <td>{{ $cv->cv_address }}</td>
119 </tr>
120 <tr>
121 <th>Date of Birth</th>
122 <td>{{ $cv->cv_dob }}</td>
123 <th>Age</th>
124 <td>{{ $cv->cv_age }}</td>
125 </tr>
126 <tr>
127 <th>Religion</th>
128 <td>{{ $cv->cv_religion }}</td>
129 <th>Marital Status</th>
130 <td>{{ $cv->cv_marriage }}</td>
131 </tr>
132 <tr>
133 <th>Language Level</th>
134 <td>{{ $cv->cv_jp_level }}</td>
135 <th>Job Type</th>
136 <td>{{ $cv->cv_job_type }}</td>
137 </tr>
138 </table>
139
140 <!-- School History -->
141 <h3>School History</h3>
142 <table>
143 <thead>
144 <tr>
145 <th>School Name</th>
146 <th>Subject</th>
147 <th>Start Date</th>
148 <th>End Date</th>
149 <th>Status</th>
150 </tr>
151 </thead>
152 <tbody>
153 @foreach([1,2,3] as $i)
154 <tr>
155 <td>{{ $cv->["cv_schoolhistory($i)_name"] }}</td>
156 <td>{{ $cv->["cv_schoolhistory($i)_subject"] }}</td>
157 <td>{{ $cv->["cv_schoolhistory($i)_start"] }}</td>
158 <td>{{ $cv->["cv_schoolhistory($i)_end"] }}</td>
159 <td>{{ $cv->["cv_schoolhistory($i)_status"] }}</td>
160 </tr>
161 </tbody>
162 </table>
```

Figure: 11.12.1. cv.php 1

```

resources > views > view-as-pdf > cv.blade.php > html
2 <html>
85 <body>
87 <div>
141 <table>
151 <tbody>
159 <td>{{ $cv->["cv_schoolhistory($i)_end"] }}</td>
160 <td>{{ $cv->["cv_schoolhistory($i)_status"] }}</td>
161 </tr>
162 </tbody>
163 @endforeach
164 </table>
166 </div>
167 <!-- Job History -->
168 <h3>Job History</h3>
169 <table>
170 <thead>
171 <tr>
172 <th>Company Name</th>
173 <th>Position</th>
174 <th>Start Date</th>
175 <th>End Date</th>
176 <th>Status</th>
177 </tr>
178 </thead>
179 <tbody>
181 @foreach ([1,2,3] as $i)
182 <tr>
183 <td>{{ $cv->["cv_jobhistory($i)_name"] }}</td>
184 <td>{{ $cv->["cv_jobhistory($i)_position"] }}</td>
185 <td>{{ $cv->["cv_jobhistory($i)_start"] }}</td>
186 <td>{{ $cv->["cv_jobhistory($i)_end"] }}</td>
187 <td>{{ $cv->["cv_jobhistory($i)_status"] }}</td>
188 </tr>
189 @endforeach
190 </tbody>
191 </table>
192 </div>
193 </body>
194 </html>
195
resources > views > view-as-pdf > cv.blade.php > html
2 <html>
85 <body>
87 <div>
195 </table>
196 <!-- Evaluation -->
197 <h3>Evaluation</h3>
198 <div class="evaluation-box">
199 <div class="evaluation-box">
200 <div class="evaluation-box">
201 <div class="evaluation-box">
202 <div class="evaluation-box">
203 <div class="evaluation-box">
204 </div>
205 </body>
206 </html>
207

```

Figure: 11.12.2. cv.php 2

Joblisting PDF View

```

resources > views > view-as-pdf > joblisting.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <title>{{ $job->job_code }} - Job Information</title>
7
8 <style>
9 @page {
10 size: A4 portrait;
11 margin: 1.5cm;
12 }
13
14 body {
15 font-family: DejaVu Sans, sans-serif;
16 font-size: 12px;
17 margin: 0;
18 padding: 0;
19 color: #111827;
20 }
21
22 /* Card container */
23 .container {
24 max-width: 800px;
25 margin: 0 auto;
26 background: #ffffff;
27 border-radius: 12px;
28 box-shadow: 0 4px 8px #000000;
29 padding: 24px;
30 }
31
32 /* Headings */
33 h1,
34 h2 {
35 text-align: center;
36 font-weight: 700;
37 margin: 0;
38 padding: 0;
39 }
40
41 h2 {
42 font-size: 20px;
43 text-transform: uppercase;
44 margin-top: 40px;
45 }
46
47
48
49
50
51
52
53 /* Table styling */
54 table {
55 width: 100%;
56 border-collapse: collapse;
57 margin-bottom: 20px;
58 font-size: 12px;
59 }
60
61 th,
62 td {
63 border: 1px solid #374151;
64 background-color: #e9e7eb;
65 padding: 8px 18px;
66 vertical-align: top;
67 text-align: left;
68 }
69
70 /* Info table header cells */
71 td.bg-gray {
72 background-color: #e9e7eb;
73 font-weight: 700;
74 text-align: center;
75 width: 30%;
76 font-size: 14px;
77 }
78
79 td.text-blue {
80 color: #1d4ed8;
81 font-weight: 700;
82 }
83

```

Figure: 11.12.3. joblisting.php

```
resources > views > view-as-pdf > joblisting.blade.php > ...
2 <html lang="en">
4 <head>
6 <style>
83 }
84
85 ul {
86     list-style: none;
87     margin: 0;
88     padding-left: 0;
89 }
90
91 li {
92     padding: 2px 0;
93 }
94
95 .bold {
96     font-weight: 700;
97 }
98
99 .section {
100     page-break-inside: avoid;
101     margin-bottom: 30px;
102 }
103
104 table,
105 tr,
106 td,
107 th {
108     page-break-inside: avoid;
109 }
110
111 </style>
112 </head>
113
114 <body>
115 <h1>
116     Oshigoto Japan
117 </h1>
118
119 <div class="container">
120 <div class="section">
121 <div class="section">
122 <div class="section">
123 <div class="section">
124 <h2>Job Information</h2>
125 <table>
126 <tbody>
127 <tr>
128 <td>
129 <tr>
```

```
resources > views > view-as-pdf > joblisting.blade.php > ...
2 <html lang="en">
114 <body>
120 <div class="container">
122 <div class="section">
124 <div class="section">
174 <tr>
175 <td class="bg-gray">Off Days</td>
176 <td>{{ $job->off_days }}</td>
177 </tr>
178
179 <tr>
180 <td class="bg-gray">A. Salary + Benefits</td>
181 <td>{{ $job->salary_benefits }}</td>
182 </tr>
183
184 <tr>
185 <td class="bg-gray">B. Deductables</td>
186 <td>{{ $job->deductables }}</td>
187 </tr>
188
189 <tr>
190 <td class="bg-gray">Salary after Deduction (A - B)</td>
191 <td>{{ $job->after_deduction }}</td>
192 </tr>
193
194 <tr>
195 <td class="bg-gray">Airplane Ticket Expense</td>
196 <td>{{ $job->airticket_exp }}</td>
197 </tr>
198
199 </tbody>
200 </table>
201 </div>
202
203 </div>
204
205 <div class="section">
206 <h2>Requirements</h2>
207 <table class="rounded">
208 <tbody>
209 <tr>
210 <td class="bg-gray">Japanese Level</td>
211 <td>{{ $job->required_jp_level }}</td>
212 </tr>
213
214 <tr>
215 <td class="bg-gray">Age</td>
216 <td>{{ $job->age_range }}</td>
217 </tr>
218 <tr>
```

Figure: 11.12.4. joblisting.php 2