

Machine Learning-Based Job-Hunting Web Application for CV Optimisation

Lam Wing Tung

2313983

Project Dissertation Document



Swansea University Prifysgol Abertawe

Department of Computer Science
Adran Gyfrifidureg

22nd February 2025

Declaration

Statement 1

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed



Lam Wing Tung (2313983)

Date

22nd February 2025

Lam Wing Tung (2313983)

Statement 2

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A bibliography is appended.

Signed



Lam Wing Tung (2313983)

Date

22nd February 2025

Lam Wing Tung (2313983)

Statement 3

The University's ethical procedures have been followed and, where appropriate, ethical approval has been granted.

Signed



Lam Wing Tung (2313983)

Date

22nd February 2025

Lam Wing Tung (2313983)

Abstract

This document is the dissertation for the project “Machine Learning-Based Job-Hunting Web Application for CV Optimisation”.

The document presents the full-stack development process of the final year project. The goal of the project is to support job-seeking candidates in the United Kingdom by addressing the mismatch between skills and job requirements.

The application was built using the Laravel framework for the frontend and integrates AI-based classification models for CV analysis. Key features were inspired by existing job platforms, and thorough research was conducted in technical, market, and literature areas. The development followed the SCRUM methodology, with all tasks managed through Jira and visualised using a Gantt chart.

The main AI technique used is the LSTM model, combined with natural language processing to analyse personal statements and improve CV-job matching. The application also demonstrates ethical AI use, including generating synthetic data to avoid consent issues and reduce bias.

While some deployment challenges occurred, they were resolved within the buffer time, and a functional production-level web application was delivered. The project was successfully completed, meeting all milestones and showing strong potential for future improvements if scaled with real-world data.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Zhang, whose guidance and invaluable advice were instrumental. I deeply appreciate the countless hours spent providing direction, feedback, and support, often extending beyond regular working hours and even into weekends.

My thanks also extend to the lecturers at Swansea University for their amazing teaching in AI, machine learning, software engineering, web development, and human-computer interaction. They equipped me with the essential skills and knowledge to undertake this project.

I am also incredibly grateful to my past colleagues – software developers working in a tech company – who contributed their time and feedback to test the application. Their professional insights and quality assurance testing enhance greatly for the project.

Finally, I would like to acknowledge the literature and past projects that inspired this work. The insights from previous research have greatly enriched the project's direction and depth.

Table of Contents

1	<i>A Brief Introduction to The Project</i>	1
1.1	Motivation: Addressing Skills Mismatch and Employment Issues in the UK	1
2	<i>Project Planning and Evaluation</i>	2
2.1	Project Aims	3
2.1.1	Personal Aim:	3
2.1.2	Broader Aim:	3
2.2	Key Terminologies and Definitions	3
2.3	Milestones and Achievements	4
2.3.1	Simple Job-Hunting Web Application Development	4
2.3.2	Artificial Intelligence Backend Analysis	4
2.3.3	Built-in CV Enhancements	5
2.3.4	Quality Assurance.....	5
2.4	Objectives	6
2.4.1	Measurable Outcome:	6
2.4.2	Potential Immeasurable Benefit:.....	6
2.5	Beneficial Group	7
3	<i>Background Research</i>	8
3.1	Market Research.....	8
3.1.1	Job-Hunting Web Applications – LinkedIn, Indeed & Glassdoor	8
3.1.2	CV Optimising Web Applications – Teal & Enhancv	9
3.2	Technical Research.....	9
3.2.1	Job-Hunting Web Applications – LinkedIn, Indeed & Glassdoor	9
3.2.2	CV Optimising Web Applications – Teal & Enhancv	10
3.2.3	Literature Review – Past Projects in Resume Optimization and Job Matching Website	11

3.2.4	Literature Review – AI implementation – benefits and ethical considerations.....	12
4	<i>Project Management</i>	13
4.1	Software Development Lifecycle	13
4.1.1	Agile Development.....	13
4.1.2	SCRUM	13
4.2	Methodology.....	14
4.2.1	Project Development Methods.....	14
4.2.2	Machine Learning Algorithm	16
4.2.3	Project Management Tools	18
4.2.4	Project Evaluation – App Testing	19
4.3	Project Schedule – Gantt Chart	20
4.4	Risk Analysis.....	22
5	<i>Implementation</i>	25
5.1	Web Application Development (Frontend & Backend)	25
5.1.1	Tech Stack	25
5.1.2	Web Application Design Documentation	31
5.1.3	App Portals – User Roles.....	33
5.1.4	Web Application Functionalities	33
5.1.5	Authentication.....	48
5.1.6	CRUD – Create, Read, Update, Delete	48
5.1.7	File Upload	48
5.1.8	Security and encryption	48
5.1.9	Debugging.....	50
5.1.10	Email.....	50
5.2	Database Design & Structure	51
5.2.1	Database Structure	51

5.2.2	Database Relationship.....	51
5.2.3	Database Relationship Visualization	52
5.2.4	Database Implementation	52
5.3	AI Model Integration.....	52
5.3.1	NLP.....	54
5.3.2	LSTM.....	55
5.4	Deployment	61
5.5	Version Control – GitHub	62
6	Evaluation	64
6.1	User Testing.....	64
6.2	Expert Evaluation	66
6.3	Web Content Accessibility Guidelines (WCAG) 2	68
6.4	Improvements	71
6.4.1	AI Functionalities Scaling Up With Ethical Data Collection Flow	71
6.4.2	Security	71
6.4.3	Deployment.....	71
7	Conclusion.....	72
8	Bibliography.....	73
A.	Appendix.....	79

1 A Brief Introduction to The Project

This report outlines the full stack development from planning to production to evaluation for the project titled “Machine Learning-Based Job-Hunting Web Application for CV Optimization.”

1.1 Motivation: Addressing Skills Mismatch and Employment Issues in the UK

Analysing the recent job market in the United Kingdom reveals significant issues that need to be addressed, including skills-job mismatching and an elevated unemployment rate among young job seekers. The UK job market has increased activity, with more people actively seeking employment. As suggested by the Office for National Statistics (ONS)¹, the economic inactivity rate has decreased to 21.2% in 2024 [1]. Economic inactivity refers to people who are neither employed nor actively seeking work for the previous four weeks, and they are not expected to work in the next two weeks [2], as defined by ONS. The declining trend in the working-age population not seeking employment shows the increase in job-hunting activity.

Despite this positive trend, job market challenges persist. There has been an increase in job vacancies, particularly in high-skill sectors such as technology, healthcare, and education [1]. This growing demand suggests that these industries are seeking more specialized expertise and higher qualifications. However, the unemployment rate has remained unchanged at 4.1% [1], with the service sector continuing to dominate overall employment [1]. This disparity suggests that, while more high-skilled jobs are becoming available, many individuals may lack the necessary qualifications, creating a gap between available jobs and qualified candidates. Nonetheless, statistics show that there is increasing government and non-government support for acquiring skills and further education [1]. The large amount of unemployed freshly graduated bachelor students, master students, vocational schools, and skilled immigrants contradicts the argument.

One possible factor for the static unemployment rate may be the mismatch between employee skills and job requirements. Despite the growth in employment opportunities, fresh graduates continue to face substantial challenges. According to ONS, younger job seekers face a higher unemployment rate of 14.2% [1], indicating that young specialists are struggling to enter the workforce, further contributing to the disconnection between job openings and qualified applicants. It is reported that many recent graduates are either unemployed or working in positions for which they are overqualified [3]. For example, fresh graduates in my personal network have encountered difficulties in securing employment, often submitting numerous applications without receiving responses. As a result, they are temporarily employed in lower-skilled roles that do not require prior work experience.

Another mismatched group is the immigrants. Numerous examples show immigrants facing challenges when job hunting due to a discrepancy between their previous roles and the

¹ The Office for National Statistics (ONS) is the largest independent organization for producing official statistics in the United Kingdom [54]. It is a non-governmental national-scaled institute, and the data published are unbiased, accurate, and up to date [54].

qualifications sought in the UK labour market [4]. These groups struggle to secure suitable employment, even in growing demand job sectors, immigrants generally earn lower wages and face more insecure employment than UK-born workers, as reported by The Migration Observatory at the University of Oxford² [5]. This aligns with ONS statistics, which both statistics reveal that a significant portion of labour force is either unemployed or working in lower-skilled jobs despite holding higher qualifications [1], [4], [5].

These underlying issues in the UK job market reveal the need for a tool that better aligns job seekers' qualifications with employers' expectations. For example, many immigrants from Hong Kong holding British National Overseas (BNO) passports are legally eligible to work in the UK without requiring a work visa [6]. However, they often face rejection as employers mistakenly perceive the hiring process as cumbersome. Statistics show that half of BNO migrants struggle to match their jobs with their qualifications [7]. Despite being qualified for employment, unclear or poorly structured CVs further complicate the process, making it more difficult for these candidates to convey their suitability for available roles.

Companies are already utilizing artificial intelligence (AI) for screening job applications; research suggests that AI integration in the recruitment process has a huge effect in reducing time-to-hire and uncovering hidden talent by identifying transferable skills that are overlooked in traditional hiring processes [8]. However, applicants lack the resources to tailor CVs to match job descriptions and to identify underlying skills that are required. The use of AI created a disadvantage for candidates who have poorly structured and misaligned CVs. The hidden concerns should be addressed carefully when integrating AI into the process, including systemic bias leading to unfair outcomes and ethical issues in collecting training data [9].

The motivation for this project arises from this imbalance. It aims to bridge the gap between job seekers' qualifications and employer expectations. This project is a proof of concept that AI can be implemented on the applicant's side of the hiring process, to match them with job descriptions and highlight their strengths. With machine learning's ability to analyse large amounts of data, this project aims to optimize CVs to better align with job requirements, thus reducing the likelihood of job-skills mismatch and increasing the employment rate. The project will also discuss the ethical issues related to using AI. In short, this project will develop an improved job-hunting tool that addresses the previously discussed job market challenges by optimizing CVs and matching job seekers' skills with employer expectations.

2 Project Planning and Evaluation

This section compares the project initial planning and final achievement, highlighting milestones that helped measure progress toward achieving the objectives and ultimately achieving the project's aim. These milestones are the measurable indicators of success and the guide of the development process. They helped the project remain aligned with its aim. Each milestone is closely tied to the objectives outlined, helping track the advancement of both technical and operational aspects of the project.

² The Migration Observatory at the University of Oxford is a research centre based at the leading academic institution, University of Oxford analysing migration issues specifically [55].

2.1 Project Aims

The aim of this project is to develop a proof-of-concept AI-powered web application on the job applicant's side that replicates existing job-hunting platform functionalities while integrating machine learning algorithms to optimize CVs and increase the chances of securing interviews.

2.1.1 Personal Aim:

Practice implementing machine learning algorithms and performing full-stack web application development.

2.1.2 Broader Aim:

By replicating existing job-hunting platforms and incorporating innovative features such as machine learning-driven CV analysis and reusable CV component builder, this project aims to create a tool to increase the applicants' success rate of getting interviews and making application process more efficient. In parallel, the project will explore the potential of AI integration, and the ethical considerations.

2.2 Key Terminologies and Definitions

To set the foundation of the project, key terms will be defined in this section.

1. **Machine Learning.** Under the branches of Artificial Intelligence, having an algorithm to perceive past data, make analysis, identify patterns, and make decisions and predictions [10]. This project will use machine learning to perceive CVs in the database, assess their success rates in securing interviews, and optimize users' CVs accordingly.
2. **Job-Hunting Platform.** A platform that enables users to find jobs, consisting of functions for signing up, creating profiles, generating CVs, browsing listed jobs, and other CRUD³ functionalities.
3. **Web Application.** A stand-alone network-centric software application designed to serve specific needs, which is accessible through web browsers and has data stored remotely allowing clients to interact with the application [11]. This project will create a software application that is hosted on remote servers that can be accessed through web browsers without the need to install the application on a local computer.
4. **Full Stack Development.** This project involves full-stack web application development, consisting of both the frontend (the user interface for job listings and CV creation) and the backend (the processes for analysing CVs for optimization).

³ CRUD. An acronym meaning Create, Read, Update, and Delete. These are operations that can be performed on datasets.

2.3 Milestones and Achievements

The main milestones of the project are divided into four categories and prioritized according to their importance, corresponding review for each category achievements are outlined below:

2.3.1 Simple Job-Hunting Web Application Development

No	Tasks	Priority	Descriptions	Completeness
1	Basic CRUD	HIGH	Includes essential functions for users to create accounts, log in, browse job listings, submit applications, and view application history.	Delivered to Production
2	Job Searching	MID	Allows users to search and filter jobs by categories, date added, salary, and location.	Delivered to Production
3	Job Application	LOW	Implement a quick applying and a reusable job application system for customizing CVs according to job requirements.	Delivered to Production
4	Company Rating	LOW	Users can comment the employer company to increase transparency.	Delivered to Production

Task Assessment: A fully developed web application is established, with a guest portal, a job hunter portal and a company portal. The application is deployed to Azure and can be accessed through: <https://ai-job-hunter-bcapbad4cgdwgbh.uksouth-01.azurewebsites.net/>. The full functionality and deployment process will be discussed further in Section 5.1.

2.3.2 Artificial Intelligence Backend Analysis

No	Tasks	Priority	Descriptions	Completeness
1	Machine Learning Algorithm	HIGH	Implementation of an algorithm to analyse resumes and job descriptions, providing a CV rating system	Delivered to Production with changes
2	Natural Learning Processing	HIGH	Extracts and analyses keywords from the CV database and newly created CVs for better matching and recommendations.	Delivered to Production with changes

Task Assessment: Machine learning and natural language processing (NLP) have been implemented. There have been changes in the application; Long Short-Term Memory (LSTM) was used to predict the constructiveness of the CV sentences compared to a database of labelled sentences, and NLP was used to extract keywords from job descriptions and CV input for matching. The AI implementations will be further discussed in the AI section 5.3.

2.3.3 Built-in CV Enhancements

No	Tasks	Priority	Descriptions	Completeness
1	Mandatory Fields	HIGH	A structured template requiring users to fill out all necessary fields to ensure CV completeness.	Delivered to Production
2	CV Feedback System	HIGH	Feedback provided based on algorithm analysis, with suggestions for improvement by comparing the CV to successful templates.	Delivered to Production
3	Categorising	MID	Tailors the CV scoring system based on job types and industries for more accurate matching.	Closed

Task Assessment: CV enhancement functionalities have been deployed to production site with minor changes of concept. Categorising was planned to be implemented; however, the feature was dismissed due to the method of obtaining training data in this project. The reason will be further discussed in the Improvement Section 6.4.

2.3.4 Quality Assurance

No	Tasks	Priority	Descriptions	Completeness
1	User-Involved Testing	HIGH	Gathering feedback from users on functionality, user experience, and bug identification.	PRODUCED
2	Expert Evaluation	HIGH	Conducting professional evaluations of the final product to ensure its quality and effectiveness.	PRODUCED
3	Mock-ups and Program Flow	MID	Mock-ups and program flow diagram was produced before the	PRODUCED

Task Assessment: Mock-ups and program flow diagram was produced and will be listed in Section 5.1.2. Testing and Evaluation was done to evaluate the application. Five users had been invited to be involved to improve the application. Results will be further discussed in Section 6.

The milestones outlined in the above section represent the measurable outcomes for some of the project's key objectives. At the end of this project, all milestones had been achieved, and a fully functional web application with integrated backend analytics was produced and deployed. The project's initial planning documents have mentioned some optional tasks for future expansion. Some tasks have been implemented while there are some potential areas of improvement.

These completed tasks indicate that the objectives mentioned in the next section (Section 2.4) have been successfully met. By testing and analysing the final product, the project's success will be measured in terms of functionality, user experience, and algorithm performance in Section 6.

2.4 Objectives

The measurable and immeasurable objectives are stated below targeting to match the core idea of the project discussed in Section 2.1.

2.4.1 Measurable Outcome:

By the end of this project, a fully functional job-hunting web application was developed and deployed, along with an algorithm designed to analyse and optimize CVs. This algorithm will help job seekers enhance their resumes to better match job requirements, increasing their attractiveness to potential employers. The web application developed included essential features such as job listings, account creation, job search filters, resume customization, and job experience matching. The machine learning backend implemented analyse CVs and provide personalized feedback, thereby improving users' chances of securing interviews. As the system collects data over time, it will continually refine its CV optimization recommendations.

2.4.2 Potential Immeasurable Benefit:

While many large corporations are already operating successful job-hunting web applications, this project recognises its own scale and resource limitations. As such, it does not aim to create a continuously operating platform. Instead, it intends to inspire enhancements in existing job-hunting platforms by demonstrating innovative features like CV optimisation, job experience matching, and machine learning analysis. These features will assist job hunters in navigating the job market more effectively. By aligning candidates' experiences with job requirements, the platform aims to help reduce unemployment rates among these job applicants.

Despite that, if this project or similar web applications successfully launch in the real world, they could provide a free and accessible platform for job seekers and employers to connect, eliminating subscription fees and making the job search process more accessible to everyone..

2.5 Beneficial Group

As discussed in section 2.4, whether this project is launched publicly or influences existing web applications, it has the potential to create significant impacts, and groups that will benefit including:

1. Fresh Graduates:

The platform offers tools to help inexperienced fresh graduates job hunters improve their CVs. By providing a direction to create CVs that are better aligned with job requirements, enhancing their chances of securing interviews, and reducing the mismatch between their qualifications and job expectations.

2. Immigrants:

For immigrants, particularly those unfamiliar with the UK job market, the platform provides tailored CV optimization to help them present their qualifications more effectively, improving their employability and reducing job-skills mismatch.

3. Employers:

Employers benefit from receiving more relevant and well-structured CVs with help of AI, making the hiring process faster and more efficient [8]. This is particularly helpful for startups that may lack the resources for extensive HR processes.

4. Job Seekers in High-Skill Industries:

Individuals in growing industries, such as technology and healthcare, will benefit from job search filters and resume customization, allowing them to match their qualifications to available positions more effectively.

5. Growing Demand Job Markets:

The platform helps employees to reach job vacancies in high-demand sectors. Relieving the situation of insufficient supply of high-skilled workers.

6. Underemployed Individuals:

Those working below their qualifications can use the CV optimization features to identify better job opportunities that match their skills, enabling career advancement.

7. Project Creator:

As the project creator, this initiative will offer substantial personal and professional development. The following skills and experiences have been gained and are crucial for future projects and offer potential pathways for further innovation in the field of web and AI-driven applications.

- **Project Management:** Managing this project from ideation to completion provided invaluable experience in coordinating tasks, setting deadlines, and maintaining project scope.

- **Full Stack Web Application Development:** Full stack development will inspire technical expertise in building web applications that are responsive, user-friendly, and efficient. Skills taught in lectures have been applied, while more advanced topics beyond lectures have been self-taught and researched thoroughly, for example, the AI and deployment process.
- **Machine Learning:** Implementing machine learning algorithms for CV optimization provided a deep understanding of artificial intelligence and data analysis. This experience enhanced knowledge of how machine learning can be used to analyse and optimize data, improving user experience in real-world applications.

3 Background Research

In this section, we will analyse the current overview of job-hunting and CV optimization web applications through market share and technical aspects. The market research in Section 3.1 will focus on analysing the user base and popularity of existing platforms to demonstrate the significance of these applications in the modern online job search process. Section 3.2, The technical research will explore the features that have contributed to the success of these platforms, identifying key functionalities that can be incorporated into this project.

3.1 Market Research

The job market is becoming increasingly dependent on digital platforms that streamline job hunting and CV optimization. This section will review three major job-hunting platforms and two chosen CV optimization tools to assess their user base, popularity, and market relevance.

3.1.1 Job-Hunting Web Applications – LinkedIn, Indeed & Glassdoor

The top three dominating job-hunting platforms in 2024 are LinkedIn, Indeed, and Glassdoor. These platforms have a massive user base according to their annual press.

- **LinkedIn**

With over 1 billion users worldwide [12], LinkedIn serves as a networking hub that connects professionals with job opportunities. According to their official statistics, LinkedIn has more than 40 million members in the United Kingdom, and 287 million members in Europe with more than 9 thousand applications happening every minute in Europe [12]. It is considered one of the most influential job-hunting web applications in both the United Kingdom and around the world.

- **Indeed**

Known for its simplicity and vast database, Indeed is another largest job search engines globally. According to their official website, they ranked the top in 2024 globally due to the large amount of unique monthly visitors of more than 350 million visitors [13].

- **Glassdoor**

Glassdoor is considered one of the top three job-hunting websites. With more than 212 million company reviews, salaries, and insights, Glassdoor stands out by combining job search with company reviews [14], allowing users to gain insights into potential employers and company cultures.

3.1.2 CV Optimising Web Applications – Teal & Enhancv

There are lots of CV-enhancing platforms accessible online, including both with and without the use of Artificial Intelligence. We will analyse two AI-involved cv optimisers in this section.

- **Teal**

Teal has a total of 678 thousand registered users and more than 934 thousand created resumes [15]. Functions such as AI resume builder and resume checker will be a good reference for this project.

- **Enhancv:**

This web application is marketed to enhance CVs for applying to major companies such as Tesla, Google, Facebook, Spotify, and Amazon [16]. Enhancv has around 2 million users and utilizes 20 thousand analysed training data points, along with other anonymized data from users [16], making it a highly valuable reference for optimizing CVs.

3.2 Technical Research

This section evaluates the functionalities and features of the selected job-hunting and CV optimization applications. While researching, the processes of creating accounts, uploading resumes, browsing job listings, and applying for positions have been tested from all the referenced applications to determine features that are worthy to be replicated in the proposed project.

3.2.1 Job-Hunting Web Applications – LinkedIn, Indeed & Glassdoor

- **LinkedIn:**

LinkedIn is well-known for its social networking features; research shows that these features allow employees to showcase their extraversion and protean career orientation [17]. The research concludes that personal traits lead to career success rather than the number of LinkedIn connections, and this feature does not directly address the issue of skills-job mismatches. Additionally, networking features may distract users and potentially expose personal information, such as nationality, race, and photos, which can lead to assumptions and stereotypes. Studies show that including racial information on resumes can lead to bias in hiring decisions and pay equity [18]. Therefore, this project will not include networking features.

The concept of the “Quick Apply” button is designed to allow applicants to apply for multiple jobs more efficiently, enabling users to submit applications to various job listings with a single mouse click. However, tailoring resumes to align with the specific requirements of each application is essential for enhancing the skills-job match [18]. To facilitate the application process and increase the likelihood of securing interviews, the proposed application can implement a feature that allows users to recall stored data—such as personal information, skills, and work experience.

Mandatory information will remain unchanged but blocks of related qualifications can be modified according to the job description. This function enhances the uniformity and relevance of submitted resumes, streamlining the hiring process for the hiring team.

- **Indeed:**

A notable feature of Indeed is its user-friendly interface and overall UI⁴ flow. The simple and clean UI allows users to focus on the application process instead of being distracted by networking, ads, or decoration.

In addition to the UI aspects, the filtering system is a valuable reference. The extensive filtering options allow applicants to search for jobs more effectively, such as location, salary, and job type.

- **Glassdoor:**

Studies show that Glassdoor is successful due to the feature that allows users to comment on and rate employers and provides insights into company culture and employee satisfaction [19]. This feedback helps job seekers understand the workplace environment before applying, reducing the likelihood of mismatches between employee expectations and company realities [19]. Given its effectiveness in helping users make informed decisions, this project will incorporate a similar feature into the job-hunting application. By enabling job seekers to review employers, the platform will promote transparency and improve job match quality.

3.2.2 CV Optimising Web Applications – Teal & Enhancv

Both applications have features such as an error detector, studies show that error-laden resumes significantly decrease chances of getting interviews, hiring offers and competitive salaries [18]. It is beneficial to implement spell detection features for the project.

- **Teal:**

Teal has a user-friendly design, providing features that assist users in resume creation and offering suggestions based on job descriptions. It is designed to align with applicant tracking systems (ATS), it identifies flaws in the resume by structure, keyword usage, measurable results, and overall quality [20]. Such functionalities can significantly enhance the quality of applicants’ submissions.

⁴ User interface (UI). The method allows users to interact with the system.

- **Enhancv:**

According to their official site, Enhancv uses a two-tier system that mimics applicant tracking systems (ATS) [21]. It analyses parsed content for keyword relevance, assesses writing quality, and quantifies achievements [21]. AI is employed to assist in optimizing resume content [21]. The resulting resume will increase the chance of passing ATS filters and having a better presentation.

Other remarkable features including the templates and export function. Enhancv is known for its visually appealing templates, which enhance resume presentation. Users can export their resumes directly as PDFs, making it convenient for job seekers who need a professional format. The platform also offers guidance and tips for customizing resumes, adding further value to job applications.

3.2.3 Literature Review – Past Projects in Resume Optimization and Job Matching Website

- **Resume Classification Using ML Techniques:**

A study conducted in 2023 explored different machine learning algorithms in classifying resumes. The study explored resume screening using decision tree, random forest, KNN, and Support Vector Machine [22].

The experiment is divided into two parts; data were extracted from unstructured resume data using natural language processing (NLP) and Machine Learning. The collected data was split into 80% training purposes and 20% testing sets. The results show that random forest and SVC have high accuracy of 91.38% and 90.62% while Decision Tree has 83.53% and 81.63% respectively [22].

This project will apply one of the algorithms according to the ease of implementation and accuracy.

- **Comparative Semantic Resume Analysis for Improving Candidate-Career Matching:**

The study conducted in 2022 explored CV optimization using machine learning and natural language processing (NLP) to implement a Support Vector Machine (SVM) [23]. The model achieved 92% accuracy in ranking resumes [23]. The high accuracy proves the significance of this ML model.

The experiment is divided into three parts: information extraction, relative and comparative labelling, and the SVM algorithm [23]. By implementing NLP tools such as NLTK and SpaCy, along with normalization using MinMaxScaler, it adjusts scores for comparison [23]. The SVM algorithm creates an ordered candidate list.

- **Designing a web-based career system using the Laravel framework:**

The project conducted in 2023 explored developing a web-based career system by implementing Laravel framework with MySQL and PHP [24]. This system includes

job listings and company information, displayed on a website. Its design and approach can serve as a helpful reference for this project.

3.2.4 Literature Review – AI implementation – benefits and ethical considerations

- **Efficiency in Recruitment:** AI integration in recruitment significantly reduces time-to-hire by automating the screening process and efficiently handling large volumes of applications [8].
- **Discovery of Transferable Skills:** AI tools can identify hidden or overlooked transferable skills that traditional hiring processes may miss, leading to more inclusive talent discovery [8].
- **Candidate Disadvantages:** Job seekers without access to AI tools are disadvantaged, especially if their CVs are poorly structured or misaligned with job descriptions [8].
- **Bias and Fairness Issues:** AI systems may perpetuate or even enlarge systemic bias if trained on biased historical data, leading to unfair hiring outcomes [9].
- **Ethical Data Use:** There are concerns about how training data is collected, stored, and used, raising questions about privacy, consent, and data governance [9], the UK Government mentioned that confidential data should be used with explicit consent [25].

This project addresses the imbalance by offering AI support to job seekers, enabling them to better tailor their CVs to job descriptions and highlight strengths. By matching candidate profiles with employer expectations, AI helps reduce mismatches, which can improve job placement success rates. The project serves as a demonstration of how AI implementation can be ethical and beneficial which can empower applicants, not just employers, by supporting fairer and more informed application strategies.

The analysis of existing products has provided valuable inspiration and techniques that can help with the design of this project. By recreating and enhancing effective features from these platforms, the application enables users to tailor their resumes more precisely to each job application.

Many HR teams currently utilize ATS tools to assess CVs systematically, allowing them to rate applications based on predefined criteria [8]. Incorporating similar functionalities explored by past projects will enhance the relevance and effectiveness of the resumes created through the app. This focus on customization and alignment with job specifications not only improve user experience but also increase the likelihood of securing interviews, thereby addressing the prevalent issues of skills-job mismatches in the current job market.

4 Project Management

4.1 Software Development Lifecycle

This section outlines the use of Agile and SCRUM methodologies to guide development. Given the project's large scope and the complexity of the tasks involved, this Software Development Lifecycle (SDLC) is well-suited for ensuring adaptability and continuous improvement throughout the development process.

4.1.1 Agile Development

Agile development has been utilized to manage this project due to its iterative and adaptable nature. Agile focuses on continuous development and delivery through short, defined cycles known as sprints, which allow for quick adjustments and feedback integration [26]. Given the large scope and challenging tasks of this solo project, Agile is a suitable methodology, enabling the developer to make rapid course corrections as needed. It supports regular interaction with stakeholders and ensures that the project evolves in alignment with the target audience's expectations [26]. These users have participated as testers throughout the development cycle, offering practical feedback that will guide the project's progress.

4.1.2 SCRUM

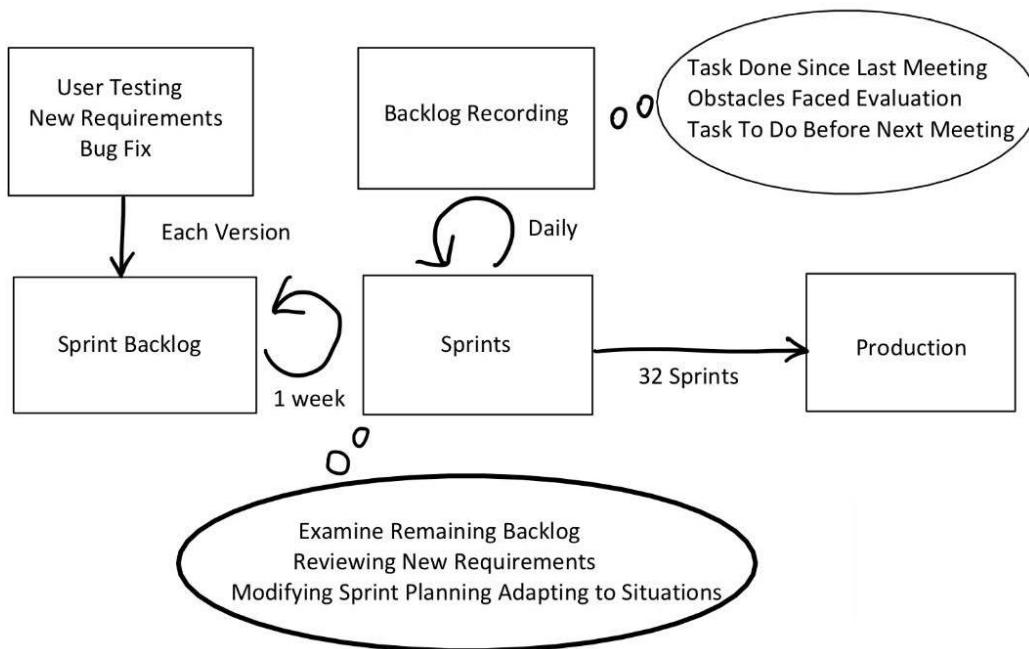


Figure 1 SCRUM Process Flow Designed to Visualize the Agile Process

SCRUM⁵ is the chosen Agile framework to manage this development process. It broke the project into smaller sprints, typically lasting between 2-4 weeks [27]; however, for this project, sprints were set to a shorter duration of one week due to the tight deadlines associated with

⁵ SCRUM is a specific framework, categorized as an agile framework [26].

final-year project requirements as shown in Figure 1. As this is a solo project, the roles were adapted to meet the constraints of solo full-stack development. The developer acted as both the Product Owner and the SCRUM Master. In this dual role, the Product Owner (developer) define the project vision [26] and gather feedback from the target audience (young job seekers and immigrants). The SCRUM Master (also the developer) ensured adherence to SCRUM principles and oversee progress [26].

A backlog was be established with prioritized tasks for each sprint, replacing standard SCRUM meetings [26] with a daily progress logbook that tracks development. This logbook documented progress, obstacles, and future steps, ensuring continuous forward momentum [26]. Regular user testing and feedback loops was be integrated, involving users who share characteristics with the target audience to ensure that the project reflects their needs and expectations.

User testing is key, allowing for the frequent adaptation of features based on practical feedback [26], [28]. This project incorporates user feedback into the development cycle regularly, aligning the software's development closely with the user base's needs. At the end of each sprint, completed tasks undergone evaluation, with adjustments made based on real-time user feedback. This approach ensures that the project remains agile and responsive, allowing room for refinements that improved the final product's relevance and usability.

4.2 Methodology

4.2.1 Project Development Methods

1. Laravel Framework

Laravel was used as the PHP framework for this project. PHP stands for "Hypertext Preprocessor." It is a server-side scripting language that can be embedded into HTML [29]. PHP was used with other technologies such as HTML⁶, CSS, Vite and Livewire. The choice of using PHP with the Laravel framework for this project is because of Laravel's suitability for beginners in full-stack web development. Compared to other frameworks, Laravel is progressive, provides extensive online documentation, and includes built-in features for authentication and routing [30], which are essential for this solo full-stack project.

Existing libraries such as PHP spellchecker [31] and Laravel-natural-language package [32] are well documented in the official Laravel site. While PHP has such libraries, it is recognized that Python remains a more widely adopted language for these tasks, due to its powerful libraries such as TensorFlow, Scikit-learn, and SpaCy [33]. The initial planning was to use PHP libraries to also implement the AI part. However, after researching, Python libraries are more suitable for developing NLP and LSTM. The developed Machine Learning model and other model was integrated to the main web application through API calls. This approach ensures that Laravel handles web development effectively while leveraging Python for machine learning tasks.

⁶ HTML means "Hypertext Markup Language," and it is used to structure webpages.

2. Backend Development Tools

Tools like Visual Studio Code (VS Code), Ubuntu, Windows Subsystem for Linux (WSL), and Docker was utilized to build the core web application. The project, developed using Laravel, ran in a Windows 11 environment. Ubuntu, WSL, and Docker are crucial for providing Linux-based functionalities needed for smooth server-side and backend development [30]. VS Code is recommended by Laravel due to its user-friendly interface and its broad support for debugging and development extensions [30], making it a suitable choice for a beginner in full-stack development. Jupyter Notebook was used to develop the AI code in Python.

3. Version Control

GitHub was used for version control in this project. GitHub offers continuous integration and version tracking, allowing seamless pushing, committing, and issue tracking. Since this is a solo full-stack development project, GitHub's collaboration features will not be used, and the repository will remain private until the project is completed.

4. Database Management

DBeaver was used for managing and analysing local database while MySQL was used for production site. DBeaver's interface provides powerful tools for database exploration and management, making it ideal for handling the data requirements of this project.

5. Deployment

To ensure accessibility of the web application, four deployment processes were performed using Azure. The first deployment was for the database, the second for the web application site, and the third and fourth for the APIs for LSTM rating model and the CV-job description keyword matching AI model. GitHub was used to deploy the application to the production site.

The discussed methodology can be summarized into the table below and will be further discussed in the later sections.

Tool	Web	Server	AI ML	DB	Host	Version	Usage
Visual Studio Code	v	v	x	v	x	x	Main code editor
Ubuntu	v	v	x	x	x	x	Via WSL/Docker for Linux environment
WSL	v	v	x	x	x	x	Enables Linux tools in Windows

Docker	v	v	x	x	x	x	Containerization for backend
Windows 11	v	v	v	v	v	v	Host OS
Laravel	v	v	x	x	x	x	PHP framework for web app
PHP	v	v	x	x	x	x	Used for web app, initial AI plan
HTML CSS Vite Livewire	v	x	x	x	x	x	Frontend technologies
Jupyter Notebook	x	x	v	x	x	x	Coding Environment for Python
Python + Python Libraries	x	x	v	x	x	x	AI/ML tasks
GitHub	v	v	v	v	v	v	Version control & deployment
DBeaver	x	x	x	v	x	x	Local database management
MySQL	x	x	x	v	v	x	Production database
Azure	x	x	v	v	v	x	Deployment/Hosting

Table: App Development Tools And Their Usage

4.2.2 Machine Learning Algorithm

The project draws inspiration from existing spam detection systems, particularly the method of using artificial intelligence (AI) to identify spam emails through keyword extraction and weighted scoring systems. However, it will extend far beyond mere keyword analysis. The final project implemented different natural language processing method and LSTM to evaluate CVs. Before implementation, some AI classification methods, such as Decision Tree, Random Forest, and K-Nearest Neighbours (KNN), and Support Vector Machine (SVM) was evaluated but was not used at the end.

Initially, the algorithm was designed to utilize natural language processing (NLP) techniques to extract keywords from both job descriptions and resumes. These extracted keywords will be matched to assess their relevance and alignment with the job requirements. A scoring system was planned to be established to quantify how well a candidate's resume aligns with the job description. Even though the method to achieve such goals were changed, this function was developed successfully at the end of the project.

The refined and final approach was to develop two models to optimize CV:

- **NLP-based Matching Model:** Assesses the alignment between job descriptions and CVs during the application process by comparing key terms and context.
- **LSTM Classification Model:** Evaluates each sentence within a CV component to determine whether it contributes constructively to the overall presentation and effectiveness.

1. Artificial Intelligence

To define the algorithm, there is an experience E, task T, performance measure P, and the formula of measuring if the performance of P against T has improved with E [10].

T (Task): optimizing CVs to increase the likelihood of securing interviews.

P (Performance Measure): The model's accuracy in predicting interview success, based on how well the CV matches job requirements.

E (Experience): A synthetic dataset containing 3,000 labelled entries of constructive and unconstructive CV sentences.

The project adopts a supervised machine learning approach to train classification algorithms that evaluate the effectiveness of CV content. The algorithm aims to improve its performance (P) in optimizing CVs (T) by learning from the labelled sentence data (E).

Since LSTM model's accuracy on predicting CV content based on the data being fed into the model. For future improvement of the project, LSTM allow the model to have continuous refinement.

2. Natural Language Processing

NLP relies on machine learning algorithms, specifically deep learning, to analyse natural human language, to do task such as ai generating articles, summaries, and AI response [34]. In this project, NLP libraries were implemented into the coding.

The specific NLP methodology used was Named Entity Recognition (NER) with SpaCy library for keyword extraction and matching, it will be discussed further in Section 5.3.

3. Long Short-Term Memory

LSTM was used for sentence classification. To determine if a CV is constructive, LSTM was used to classify the constructiveness of individual sentences in a CV. LSTM is an improved version of recurrent neural network (RNN) specifically for handling sequential data [35], which suits the project more than SVM. SVM was originally planned to be used as the classification method, however, LSTM was used in the end as it has better ability to capture long-range dependencies and contextual information over time, even better than RNN as LSTM utilizes gate function to improve the cells structure [35].

4. Data Collection

For training the AI model, large training and testing data were required. However, Resumes are confidential documents which are difficult to obtain. Only resources are from university machine learning repositories and government datasets.

Instead of relying on a CV database, which may raise privacy concerns, synthetic resume data was generated by language models to test the feasibility and functionality of the final product here raised issue regarding the application of AI, which will be discussed further in Section 5.3.

4.2.3 Project Management Tools

1. Project Management

Jira from Atlassian was utilized for managing the project. It is essentially free and offers project planning based on the SCRUM model. Jira facilitates the implementation of SCRUM features, including a scrum board, issue tracking, scheduling, and Gantt charts [36], which aid in visualizing project progress. Additionally, Jira supports the creation of epics, tasks, subtasks, and user stories [36], enhancing the overall project management process. This structured approach ensured effective scheduling, transparency, and efficiency throughout the project lifecycle. Another significant benefit of Jira is its ability to export and share project management documents [36], such as Gantt charts and issue lists, which are useful for documentation purposes.

2. Prototype Documentation

Figma was used to create mock-ups and program flow diagrams for the project. This tool provides functions for unifying styles and creating reusable widgets, facilitating a consistent design throughout the application [37]. Figma also allows for the development of interactive prototypes without the need for actual coding. This feature is particularly useful for visualizing the design and user experience before implementation, enabling better communication among stakeholders and ensuring that the final product aligns with user expectations.

4.2.4 Project Evaluation – App Testing

Software quality assurance is important to the software development process; it must ensure validation and verification. Validation means ensuring projects meet the needs of users while verification means products meet specifications [28]. The following approach was performed in the project for quality assurance.

1. Expert Evaluation

By using Ben Shneiderman's 8 golden rules for designing the user interface, an expert evaluation of the user interface was conducted to evaluate the effectiveness of interface design. These rules, which include striving for consistency, enabling frequent users to use shortcuts, and offering informative feedback [38], among others, was used as a framework to assess the interface and usability of the web application. This evaluation helped identify areas where the user experience could be enhanced and ensure the interface aligns with established usability principles.

2. User Involved Testing

An experiment involving 10 users was carried out to test the functionality of the system. The experiment involved the following tasks: Creating an account. 1. Building a resume using the CV editor. 2. Applying to a few jobs using the provided tools. 3. Browsing through different job listings using the filtering system. 4. Reporting any bugs or issues encountered during the process. 5. Providing feedback on the overall system experience, interface usability, and any recommended improvements. The detailed task list is attached in Section 6.1.

The user testing was conducted in person. The results from the experiment will be critical for identifying user pain points and making necessary adjustments to the system before final product delivery.

3. Use Case Manual Testing

Different use cases were created with different roles, scenarios, and targeted features to test for bugs and logic flow [28]. Top-down integration testing will be performed for each version of the application to identify flaws from high to low levels [39].

4.3 Project Schedule – Gantt Chart

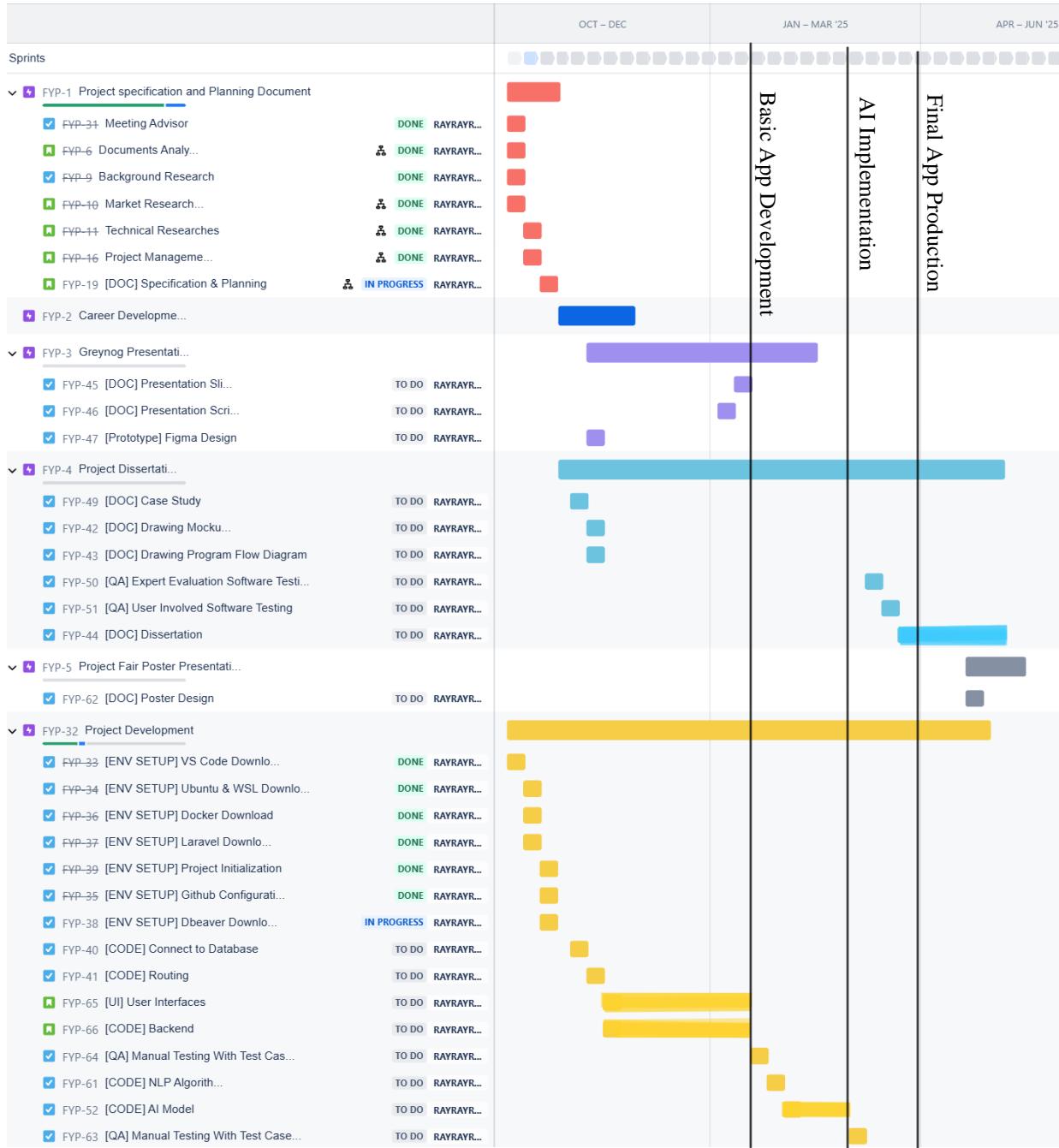


Figure 2. Gantt Chart Showing the Project Schedule Produced Using Jira on 28th Oct 2024

The project schedule presented above was the initial plan generated using Jira. It outlined the project deadlines and planned task deliveries throughout each sprint. This timeline was used to track progress and was updated and managed according to the advancements made.

The project ran from October until the end of May. Each small block on the chart represented a sprint, totalling 32 sprints, with each sprint lasting one week. Tasks were broken down and allocated across various sprints, ensuring they were scheduled effectively throughout the development timeline.

The first quality assurance testing was planned to take place in late January, by which time the initial version of the job-hunting application was planned to be developed. Before the Gregynog Presentation, the basic web app development was successfully developed locally following the schedule while testing was taken place. All the planned tasks were carried out successfully.

The second round of testing was planned to occur in early March, during which the remaining AI functionalities and additional features were planned to be implemented. Following this, bug fixes, code cleanup, and refinements were planned to be carried out. The final user-involved use case testing and expert evaluation were planned to be executed in early April, after which further modifications were planned to be made based on the quality assurance feedback. On the other hand, there was a slight delay in the development stage due to the incremented tasks, changed functionality and bugs. The web app and the AI part was completed at the end of March where second round of testing was carried out afterwards.

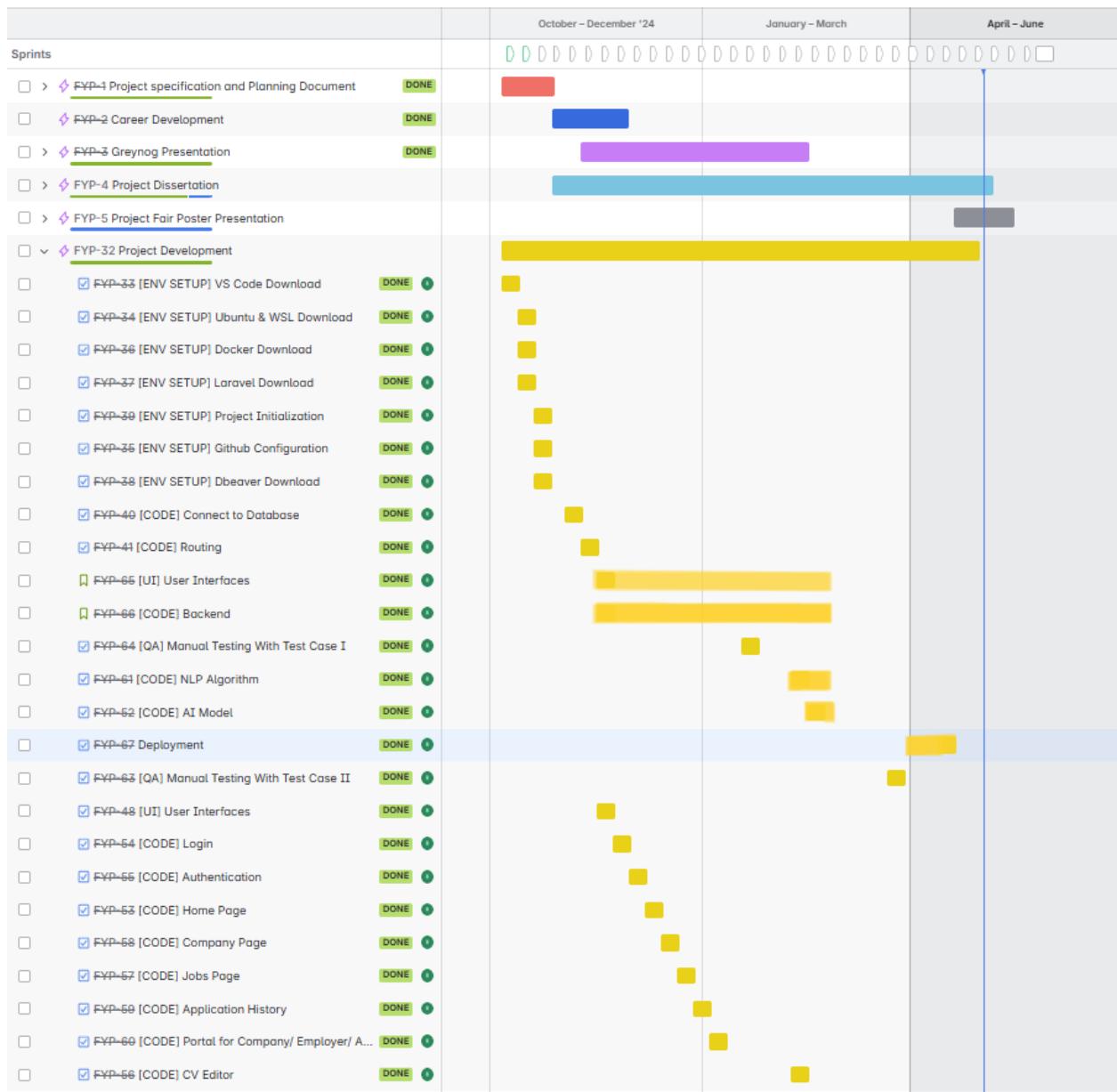


Figure 3Final Gantt Chart Produced At The End Of Project

Towards the end of the project, final user-involved testing was planned. However, due to participant's location limitations and the consideration of completing the challenge of full stack development, deploying the application to online server was carried out. The deployment process was extremely challenging; therefore, the final testing was postponed to late April. Bugs and improvement suggestion was gathered and fixed afterwards. Development process was completed on 29th April 2025. The final Gantt Chart was produced at the end of the project.

Prototyping and documentation were planned and performed in parallel. Mock-ups, program flow diagrams, and data analysis documentation were generated within the specified timeframe.

The project schedule outlined above was designed to adapt to modifications in accordance with agile development principles, with buffer time planned. It was continuously reviewed and refined at the conclusion of each sprint to ensure it remained aligned with project goals and objectives. Evaluating the project management, considering the scale of the project, and the completeness, the slight delay was acceptable, and the project was considered a success.

Milestone	Planned Date	Actual Date	Status / Notes
Project Start	October 2024	October 2024	On Time
Initial Version Completed	Late January	Late January	On Time
1st QA Testing	Late January	Late January	On Time
2nd QA Testing + AI Feature Implementation	Early March	End of March	Delayed due to added tasks, changes, and bugs
Web App & AI Feature Completion	Early March	End of March	Slight Delay
Final User-Involved Testing	Early April	Late April	Postponed due to deployment and participant limitations
Deployment to Online Server	N/A	Mid to Late April	New Task
Final QA Fixes and Refinements	Early to Mid-April	Late April	On Time
Development Completed	Late April 2025	29th April	On Time
Final Gantt Chart Produced	End of Project	End of April 2025	On Time

4.4 Risk Analysis

Figure 3 shows the table describing the potential risks of the project associated with their alternatives, precaution and final evaluation. The likelihood of having the risks is rated from 1 to 10 with 10 having the highest chance of happening. Severity and affects are assessed from High to Low.

Risks Descriptions	Likelihood (1 ~ 10)	Severity & Affects	Alternatives/ Precautions
Technical Problems Solo development for a full-stack project as a beginner-level developer. The risk of encountering technical issues is high.	10	High (The whole project)	Sufficient resources are available, including software documentation, similar project examples, library documentation, and literature references on this topic. Increase the time allocated to the project to allow for trial and error.
Evaluation: Technical issues were countered in the part of deployment; other technical issues were resolved referenced from online tutorials and official documentations.			
Large Scale of Scope The scope of the project is too large, increasing the risk of underdeveloped features.	10	High (The whole project)	Tasks are sorted by priority, and varying levels of implementation are used for each feature, gradually increasing in complexity.
Evaluation: There were no underdeveloped features, they were split into small milestones and development followed tightly with the planning. Although there were incremented tasks, all the features planned initially were developed successfully with additional new features.			
Training Data Data may have bias, inaccuracy, and privacy concerns.	8	Medium (Accuracy of AI Algorithm)	Generating synthetic data with language models can provide balanced, unbiased, controlled training samples, and eliminate privacy concerns.
Evaluation: Since synthetic data was used for LSTM machine learning model, and LSTM model will perform according to data being fed to the model, when in real application, real data would be fed to the model and refine it. This issue can be dismissed.			
Technical Risks Integration of PHP with Python: When integrating different systems, there is a high risk of failure in	5 (There are previous studies)	High (Affect AI Algorithm integration)	If API cannot be integrated properly or fails at some point, PHP libraries should be utilized for data analysis instead of relying on Python integration.

data transfer or even in initializing the integration.	available for reference)		If PHP libraries are insufficient to perform the tasks, other integration method will be explored for bridging PHP with Python.
Evaluation: Implementing API was challenging, but it succeeded in local environment within the scheduled deadline, therefore integrating Python to Larval approach were used.			
Hardware Failure Loss of Progress due to hardware issues.	3 (The operation environment is setup in a recent hardware model)	High (The whole project)	Use a server-based GitHub repository for version control. In case of hardware failure, all project files and code can be retrieved online, ensuring that while the environment setup may need to be reset, the progress remains safely stored and accessible online.
Evaluation: No hardware failure experienced, version control was used consistently.			
Project Redesign Project redefined during production period.	1	High (The whole project)	The project is defined by the developer; no major misunderstanding of user requirements and communication issue should occur.
Evaluation: There were slight changes in the application of the machine learning part; however, the functional end goal met the initial documentation. Therefore, there was no major project redefinition.			
Delayed Feature Delivery The project is divided into small tasks across consecutive sprints. One delayed feature impacts the whole timeline.	1	Medium (Project Deadline)	SCRUM allows for frequent sprint reviews and adjustments to schedules and priorities. Buffer time has been allocated within the sprints. There are no customers involved, and the only unchangeable delivery date is the project completion date.
Evaluation: Since the above precautions were taken, the entire project was completed on time, with only slight delays in feature delivery for the internal deadline for production site.			

Figure 3. Table Describing The Potential Risk and Final Evaluation of The Project

5 Implementation

This section discusses the full stack development process for the whole project, including web app and AI development and their deployment respectively.

5.1 Web Application Development (Frontend & Backend)

The development of the web application will be discussed in this section, from concept to code application. All the technical applications will be documented to make sure this project is replicable.

5.1.1 Tech Stack

1. Laravel

Laravel is the core PHP framework selected for implementation. The code architecture was developed based on MVC architecture. MVC stands for Model-View-Controller, which means the code is separated into logic, user interface and data handling.

ApplicationDetail.php
ApplicationHistory.php
ApplicationPage.php
CertificationComponent.php
CheckStatement.php
CommentList.php
CompanyList.php
CompanyStats.php
CreateComment.php
CreateCVPage.php
CVComponent.php
CVHistory.php
CvMatcherComponent.php
CVPage.php
CvPreview.php
EditProfile.php

application-detail.blade.php
application-history.blade.php
application-page.blade.php
c-v-component.blade.php
c-v-history.blade.php
c-v-page.blade.php
certification-component.blade.php
check-statement.blade.php
comment-list.blade.php
company-list.blade.php
company-stats.blade.php
create-c-v-page.blade.php
create-comment.blade.php
create-job.blade.php
cv-matcher-component.blade.php
cv-preview.blade.php
edit-profile.blade.php
education-component.blade.php

Models
Application.php
Certification.php
Comment.php
ContactInformation.php
CV.php
Education.php
JobPost.php
PersonalStatement.php
ProfessionalExperience.php
Profile.php
Skill.php
User.php
Providers

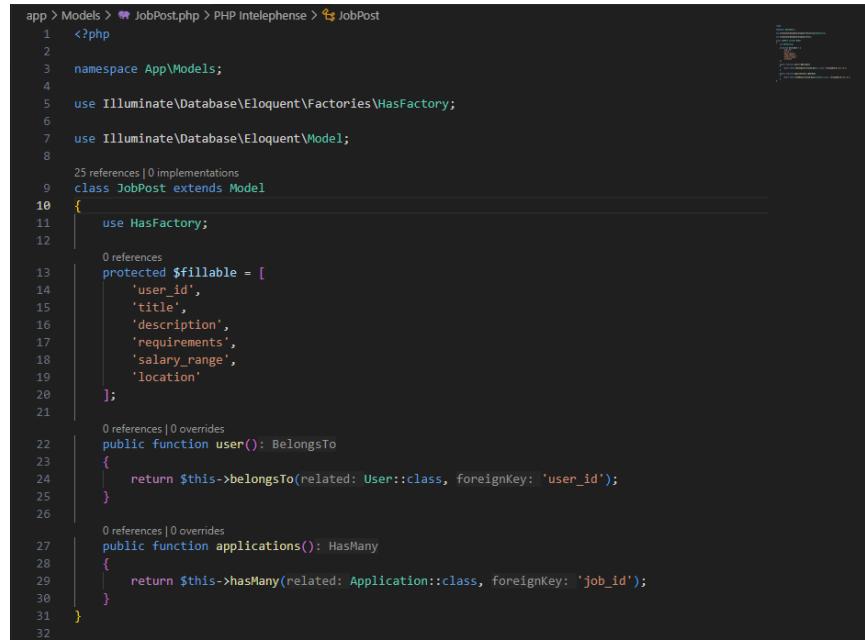
Figure 4 Some Controller Files

Figure 5 Some View Files

Figure 6 Model Files

The MVC architecture follows a request/response cycle. When a request is made, it goes through routing, which directs it to the respective controller [40]. The controller handles the logic and interacts with the model if needed, then renders a view to return a response [40]. This structure suits web application development well. The separation of concerns—UI, logic, and data—reduces code complexity and minimizes coupling issues. It also makes the code more reusable, easier to test, and simpler to debug [40].

Below are some codes of the project to illustrate the MVC pattern.



```

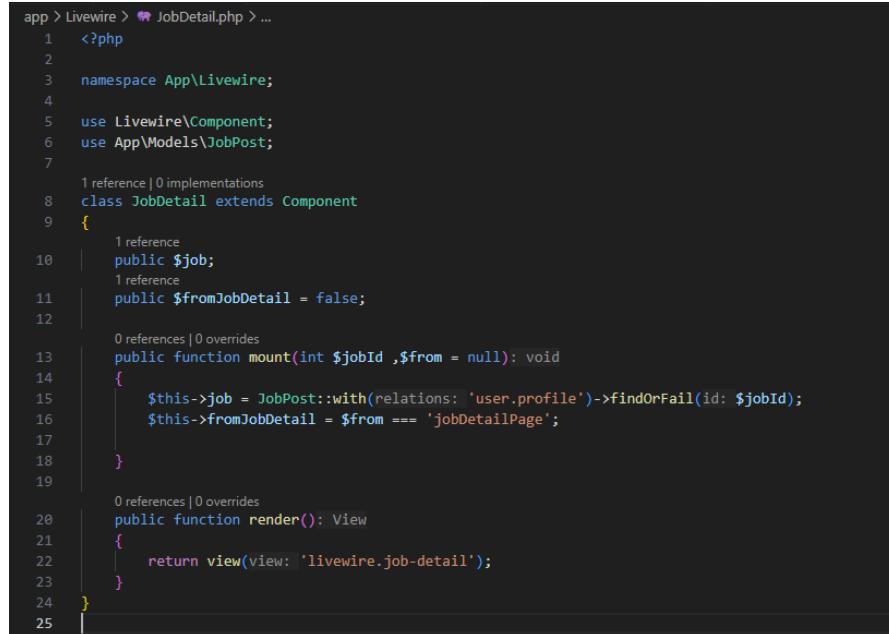
app > Models > JobPost.php > PHP Intelephense > JobPost
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6
7  use Illuminate\Database\Eloquent\Model;
8
9  class JobPost extends Model
10 {
11     use HasFactory;
12
13     protected $fillable = [
14         'user_id',
15         'title',
16         'description',
17         'requirements',
18         'salary_range',
19         'location'
20     ];
21
22     public function user(): BelongsTo
23     {
24         return $this->belongsTo(related: User::class, foreignKey: 'user_id');
25     }
26
27     public function applications(): HasMany
28     {
29         return $this->hasMany(related: Application::class, foreignKey: 'job_id');
30     }
31 }
32

```

Figure 7 Example of a Model File

The model maps the database table JobPost to the PHP class JobPost. It allows the data modification, such as saving, retrieving, editing and deleting the data from the database. Some code can be used to retrieve data using Tinker in the terminal: \$jobPost = JobPost::all(); will return all job posts.

The functions at the bottom define relationships between models, for example one-to-many relationship between jobPost-> application. It defines how models are related.



```

app > Livewire > JobDetail.php > ...
1  <?php
2
3  namespace App\Livewire;
4
5  use Livewire\Component;
6  use App\Models\JobPost;
7
8  class JobDetail extends Component
9  {
10     public $job;
11     public $fromJobDetail = false;
12
13     public function mount(int $jobId , $from = null): void
14     {
15         $this->job = JobPost::with(relations: 'user.profile')->findOrFail(id: $jobId);
16         $this->fromJobDetail = $from === 'jobDetailPage';
17     }
18
19     public function render(): View
20     {
21         return view(view: 'livewire.job-detail');
22     }
23 }
24
25

```

Figure 8 Example of A Controller File

Controller is in the middle between model and view. User input through HTTP request, the controller will interact with the model to fetch and send data to the view.

This controller handles the logic, interacts with the model, and passes data to the view. Mount() is where the job is fetched from the database using the JobPost model. Render () is the method that renders the view, which contains the data to be displayed.

```

1  <div>
2      @if ($session->has(key: 'success'))
3          <div class="alert alert-success">
4              {{ session(key: 'success') }}
5          </div>
6      @endif
7      <div class="element-container">
8
9          <div class="mb-6">
10             <h1 class="highlighted-headers text-2xl font-semibold leading-tight">{{ $job->title }}</h1>
11         </div>
12
13         <div class="flex items-center justify-between mb-6">
14             @if ($job->user->profile && $job->user->profile->profile_picture)
15                 <a href="{{ route(name: 'profile.detail', parameters: ['profileId' => $job->user->id]) }>
16                     <div class="mr-4">
17                         
19                     </div>
20                 </a>
21             @endif
22
23             <div class="flex-1">
24                 <p class="detail-subinfo">
25                     <a href="{{ route(name: 'profile.detail', parameters: ['profileId' => $job->user->id]) }>
26                         <strong>Posted by:</strong>
27                         @if ($job->user->profile)
28                             {{ $job->user->profile->user_name }}</a>
29             </div>

```

Figure 9 Example of a View File

The view file is in a HTML style; it is the frontend of the application that display the information fetched from the model. From the figure, it uses blade statement to pass output data to the view for example it gets the user's profile's user_name form \$job. \$job was defined in the controllers.

2. Livewire

In this project, Livewire was used heavily to create a single page application (SPA). Livewire is a package that enables real-time and reactive interfaces [41]. When there is form submission and UI updates, page reloads will not be triggered to achieve the modern SPA feature.

The structure of the project is making Livewire pages and calling their Livewire component. Since function codes are in the component Livewire scripts, it is highly reusable and coupling issue is reduced. For example, the job list was used in its job list page, and it was reused in the company profile page and the company job list under the tab of company dashboard.

```

resources > views > job-list.blade.php > x-app-layout
1 <x-app-layout>
2   <x-slot name="header">
3     <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4       {{ key: 'Job List' }}
5     </h2>
6   </x-slot>
7   <div class="page-container">
8     <livewire:job-list />
9   </div>
10 </x-app-layout>
11

```

Figure 10 Job List Page Using Job List Component

```

JobDetail.php X profile-detail.blade.php CvMatcherComponent.php EducationComponent.php education-component.blade.php
resources > views > profile-detail.blade.php > x-app-layout > div.page-container > div#comment-part.element-container-transparent >
1 <x-app-layout>
2   <div class="page-container">
3     <div class="element-container-transparent mb-4">
4       ...
5     </div>
6
7     <div class="element-container-transparent" id="job-list-part" style="display:;">
8       <h3 class="text-lg font-semibold mb-2">Jobs From This Company </h3>
9       <livewire:job-list :companyId="$profileId" />
10    </div>
11
12
13
14
15
16
17
18
19
20
21

```

Figure 11 Company Profile Page Using Job List Component

```

JobDetail.php X dashboard.blade.php CvMatcherComponent.php EducationComponent.php education-component.blade.php # app.css vite.css
resources > views > dashboard.blade.php > x-app-layout > div.page-container > div.p-6.text-gray-900 > div#jobListing.tabcontent > div.element-container > div.create-job-form
1 <x-app-layout>
2   <div class="page-container">
3     <div class="p-6 text-gray-900">
4       <div id="jobListing" class="tabcontent">
5         <div class="element-container">
6           ...
7         </div>
8
9         <div class="flex">
10           <div id="jobList" class="w-full transition-all">
11             <livewire:job-list :userId="auth()>user()>profile->id" :context="'company-dashboard'" />
12           </div>
13
14         <div id="createJobForm" class="hidden transition-all">
15           ...
16         </div>
17
18       </div>
19     </div>
20   </div>
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

```

Figure 12 Dashboard Page Using Job List Component

3. Blade

Blade is the template engine for building views with syntax for loops and conditionals, here is the application of blade code that keeps PHP logic out of raw HTML [41].

```

@foreach ($missedImportantKeywords as $keyword)
  <li class="keyword-badge keyword-badge--missing">{{ $keyword }}</li>
@endforeach

```

Figure 13 Blade Code That Uses For Loop

```

@livewire('name', 'checkStatement')
@if ($session()->has(key: 'error'))
    <div class="alert-error mt-4">
        {{ session(key: 'error') }}
    </div>
@endif

</div>

@error('newStatement')

```

Figure 14Blade Code That Uses Conditionals

4. CSS/Tailwind

The combination of CSS stylesheets and inline Tailwind CSS was used to style the application. CSS was used for reusable styling common component such as layout and buttons. It makes the code more readable and reusable. While the inline Tailwind CSS make styling quick and responsive, it was applied for custom or unique occurring components. The below Figures display how reusable the delete button class is, and the inline Tailwind CSS for the application status tag which only occurred once.

```

722 .delete-button {
723     background-color: #dc3545;
724     color: white;
725     padding: 8px 12px;
726     text-decoration: none;
727     border-radius: 4px;
728     cursor: pointer;
729     width: 80px;
730     height: 40px;
731     min-width: 100px;
732     transition: background-color 0.3s, transform 0.2s;
733 }
734
735 .delete-button:hover {
736     background-color: #be2e3c;
737     color: rgb(255, 255, 255);
738     transform: translateY(-2px);
739 }
740

```

Figure 15Part Of The Reusable Delete Button CSS Code

```

@if (!$creatingApplication)
    <button wire:click="deleteCV({{ $cv->id }})" class="delete-button">DELETE</button>
@endif

```

Figure 16Delete Button Style Class Used in CV Page

```

|   <button wire:click="cancelComment({{ $comment->id }})" class="cancel-button">
|     CANCEL
|   </button>
|   <button wire:click="confirmDelete({{ $comment->id }})" class="delete-button">
|     DELETE
|   </button>
| >

```

Figure 17 Delete Button Style Class Used In Comment List

```

<span
  class="px-3 py-1 rounded-full text-sm font-semibold text-white
@if ($application->status === 'pending') bg-yellow-500
@elseif($application->status === 'accepted') bg-green-600
@elseif($application->status === 'rejected') bg-red-600
@else bg-gray-400 @endif
">
  {{ ucfirst(string: $application->status) }}
</span>

```

Figure 18 Inline Tailwind Style For Application Status Tag

5. JavaScript & AJAX

JavaScript was used to enhance interactivity for handling dynamic behaviours, for example form toggles. The below figure shows the form toggle for company profile. It dispatches message ‘refreshCommentList’ when the tab is clicked.

```

<script>
  Livewire.on('toggleSection', (section) => {
    const jobListPart = document.getElementById("job-list-part");
    const commentPart = document.getElementById("comment-part");

    if (section === 'jobs') {
      jobListPart.style.display = "";
      commentPart.style.display = "none";
    } else if (section === 'comments') {
      commentPart.style.display = "";
      jobListPart.style.display = "none";
    }
  });

  function toggleCommentSection() {
    const commentSection = document.getElementById("commentSection");
    const button = document.getElementById("toggleCommentButton");
    const commentListOnly = document.getElementById("commentListOnly");
    Livewire.dispatch('refreshCommentList');

    if (commentSection.classList.contains("hidden")) {
      commentSection.classList.remove("hidden");
      button.textContent = "Close Comment Section";
      button.classList.remove("bg-[#36c73b]");
      button.classList.add("bg-gray-500");

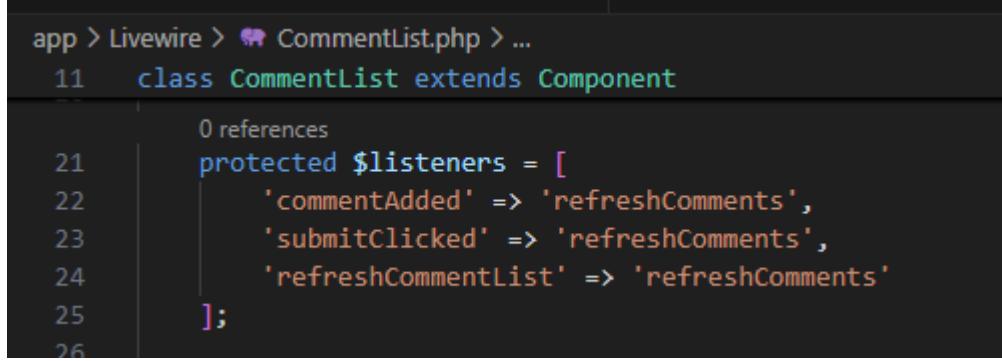
      commentListOnly.classList.add("hidden");
    } else {
      commentSection.classList.add("hidden");
      button.textContent = "Add Comment";
      button.classList.remove("bg-gray-500");
      button.classList.add("bg-[#36c73b]");

      commentListOnly.classList.remove("hidden");
    }
  }
</script>

```

Figure 19 JavaScript Toggle Section In Company Profile

When the comment list livewire component listens to the messages, it will refresh the component without refreshing the page. This behaviour is called Asynchronous JavaScript and XML (AJAX). AJAX means users could perform actions such as submitting forms, updating component or loading content without needing to refresh the page to make user experience smoother [42].



```

app > Livewire > 📄 CommentList.php > ...
11  class CommentList extends Component
-- 
21      0 references
22      protected $listeners = [
23          'commentAdded' => 'refreshComments',
24          'submitClicked' => 'refreshComments',
25          'refreshCommentList' => 'refreshComments'
26      ];

```

Figure 20 Comment List Livewire Controller Listeners

5.1.2 Web Application Design Documentation

Before code development, a program flow diagram and a mock-up was created to visualize the ideation and the functionalities. The documents usually act as prototypes for developer to better understand the design. In this solo project, the design documents were used as a guide for development.

1. Web Application Flow Diagram

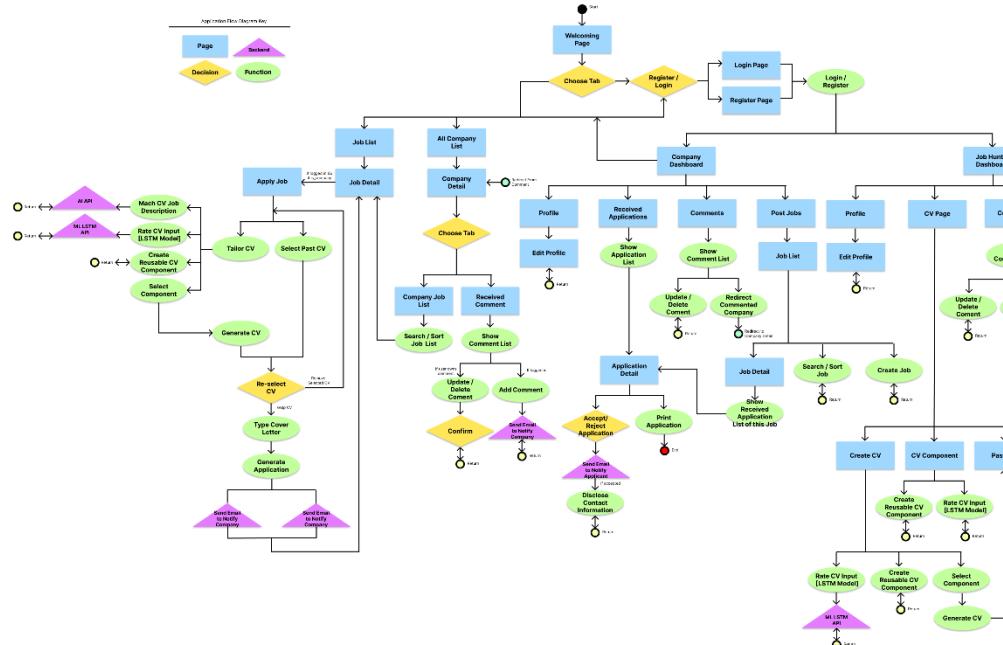


Figure 21 Web Application Flow Diagram Final Version

The above figure shows the complete web application flow. It was produced before the code implementation but refined throughout the process and a final version was produced.

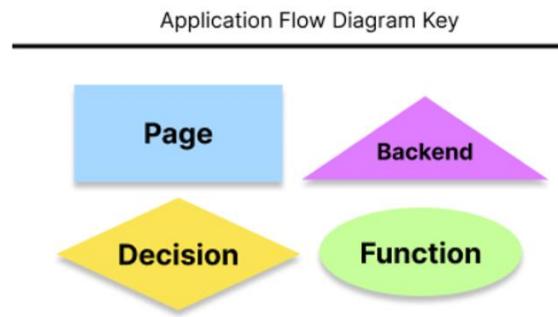


Figure 22 Program Flow Diagram Attribute Key

The attribute in blue rectangle represents different pages of the web application while the green oval attributes represent the functions that could be performed. The yellow diamond represents the decision-making point, and the purple triangle represents the backend functionalities. This program flow will be used below to explain the application flow. The diagram will be attached in the appendix and be submitted with the submission zip file.

2. Figma Mock-up

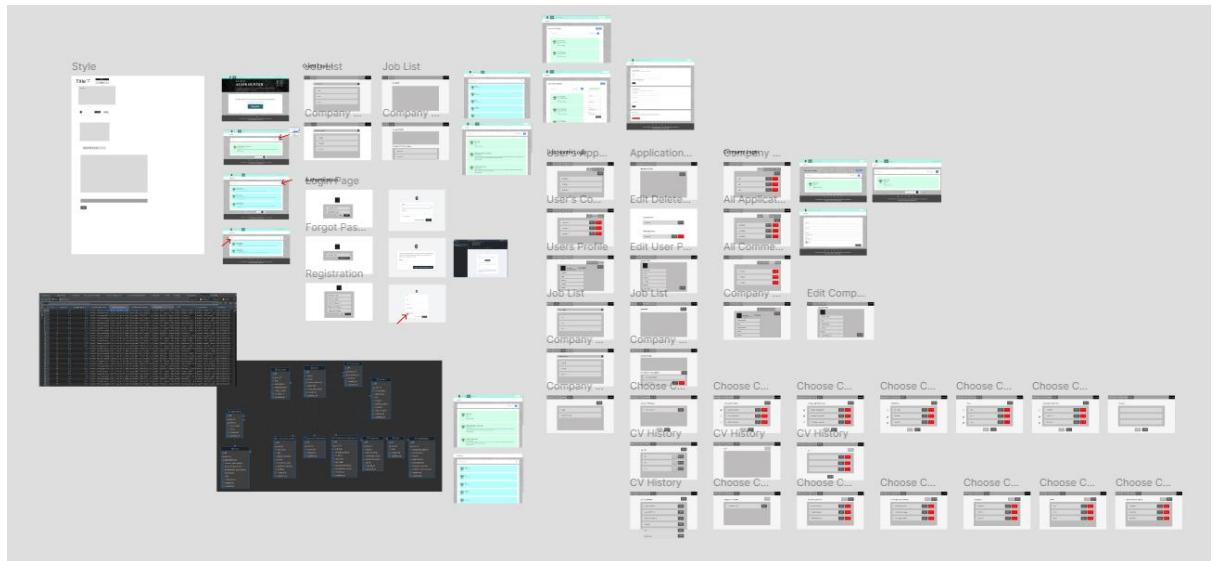


Figure 23 Web Application Figma Mock-up

A mock-up was produced before implementation using Figma online. It outlines how the application page's layout was designed. Throughout the implementation there were changes in the application flow, but it gave an insight into how the first draft of the application looked. The Figma could be accessed through this link:

<https://www.figma.com/design/085zgwDsMY4MQtXUCzuskA/Final-Year-Project?node-id=0-1&t=P4WMRrl9YwcOrSIh-1>

5.1.3 App Portals – User Roles

This application defines three user roles:

- **Guest User:** Have basic functionalities: browse public job listings and company listings but cannot interact with them. They can also register an account.
- **Job-Hunter User:** Add on to the basic functions, they can edit their profile, create and manage CVs, create and manage applications, create and manage their comments.
- **Company User:** Add on to the basic functions, they can edit their profile, post and manage job listings, view applicant CVs, create and manage their comments, profiles are visible to others.

Separating roles and permissions allow them to have a clearer user experience. Job-hunter and company users will only see relevant features to avoid confusion. It also acts as access control, role-based access reduces the risk of unauthorized actions.

5.1.4 Web Application Functionalities

This section describes the page rundown and their functionality for better understanding of the final product. (The production site for the web application can be accessed through the link: <https://ai-job-hunter-bcapbad4cgdwcbh.ualbany-01.azurewebsites.net/>)

Landing Page

When first get into the web application, the landing page is the welcoming page.

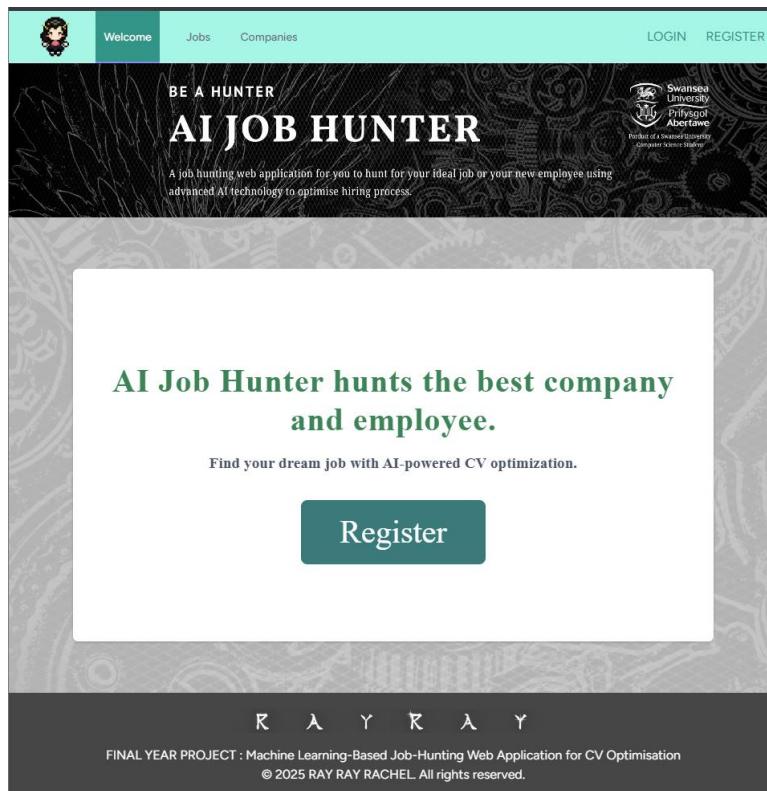


Figure 24Web Application Landing Page

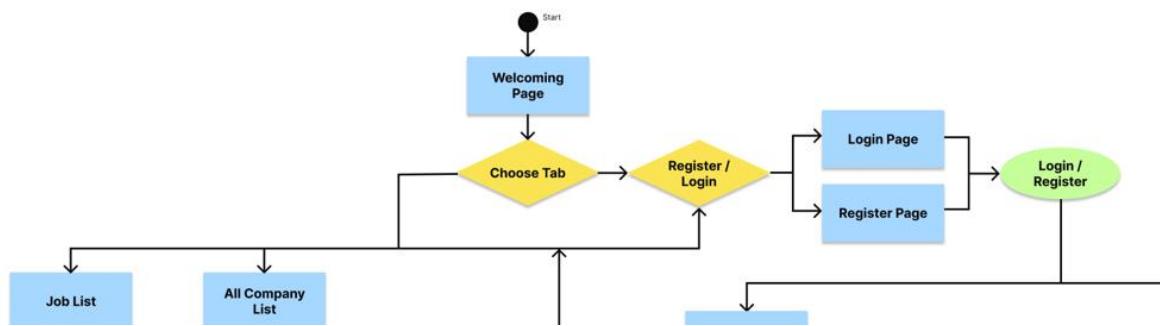


Figure 25 Flow Diagram From Welcoming Page Of Guest Login

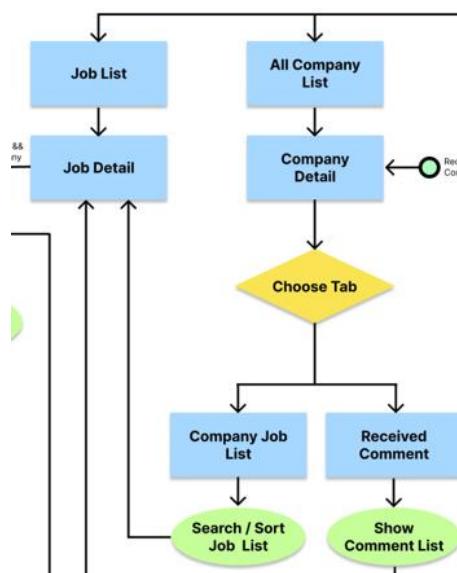


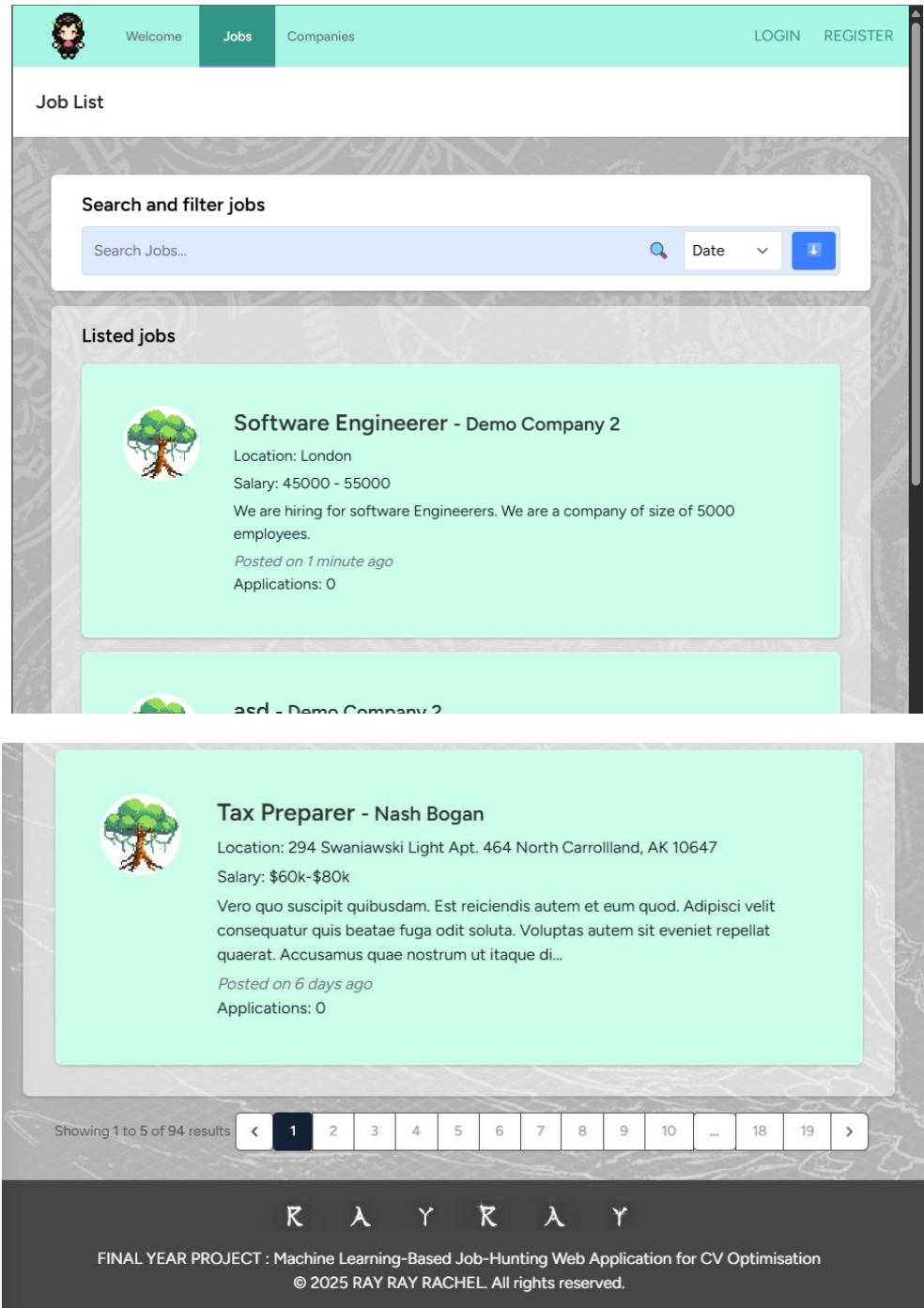
Figure 26 Flow Diagram For Permission For Everyone

The above figures are the flow diagram visualizing the pages available for guest login. Minimum permission is granted such as viewing job list page and company list page. They are also allowed to view company profiles. Guests are prompted to login or register.

GUEST PERMISSION

1. Job List Page

The Job List page can be viewed by all users, they display the jobs according to the date added. It also has functions to search jobs and sort jobs according to salary, date created and the alphabetical order of job title, ascending or descending ordered. At the bottom of the page, there is friendly pagination that allows only 5 jobs being displayed on each page.



The screenshot shows the 'Job List' page of a web application. At the top, there is a navigation bar with a user icon, 'Welcome', 'Jobs' (which is highlighted in blue), 'Companies', 'LOGIN', and 'REGISTER'. Below the navigation bar, the page title 'Job List' is displayed. A search bar labeled 'Search and filter jobs' contains a placeholder 'Search Jobs...' and a dropdown menu set to 'Date'. The main content area is titled 'Listed jobs' and features two job listings:

- Software Engineerer - Demo Company 2**
Location: London
Salary: 45000 - 55000
We are hiring for software Engineerers. We are a company of size of 5000 employees.
Posted on 1 minute ago
Applications: 0
- asd - Demo Company 2** (partially visible)

Below the job listings, a message states 'Showing 1 to 5 of 94 results' followed by a pagination bar with numbers 1 through 19 and arrows for navigating between pages. At the very bottom, there is a footer with the text 'FINAL YEAR PROJECT : Machine Learning-Based Job-Hunting Web Application for CV Optimisation' and '© 2025 RAY RAY RACHEL. All rights reserved.'

Figure 27 Job List Page

When the job is clicked, it will redirect to the job detail page. The job detail shows information about the job posting and when the company name is clicked, it will redirect to the company profile page.



The screenshot shows a job detail page for a position titled "Chef". The top navigation bar includes a user icon, "Welcome", "Jobs", "Companies", "LOGIN", and "REGISTER". The main title "Chef" is displayed prominently. Below the title, it says "Posted by: Dessie Tillman PhD". A green text box contains the "Salary Range: \$60k-\$80k". The "Location" is listed as "1283 Runolfsdottir Harbor Nienowberg, ND 47961-9171". The "Description" section contains a large amount of placeholder Latin text. The "Requirements" section also contains placeholder Latin text. At the bottom right, it says "Posted 6 days ago".

Figure 28Guest Login Job Detail Page.

2. Company List Page

The other tab in the top navigation bar is the company list that is accessible by all roles; there is also a search function and bottom pagination. It shows a list of five companies on each page and will redirect to the company profile when the item is clicked.

The screenshot displays the Company List Page of a web application. At the top, a navigation bar features a user icon, 'Welcome', 'Jobs', and 'Companies' (which is the active tab), along with 'LOGIN' and 'REGISTER' buttons. Below the navigation bar, the page title 'Company List' is visible. A search bar labeled 'Search and filter companies' contains the placeholder 'Search Companies...' and a magnifying glass icon. The main content area, titled 'Listed companies', contains five entries, each represented by a small tree icon and a company name. The entries are:

- Demo Company 2**
Location: N/A
- demo company**
Location: N/A
- Trycia Grimes**
Location: Denesikmouth
Odit est voluptas dolorum omnis enim qui consectetur at. Ut velit molestiae aut neque non enim reprehenderit.
- Rocio Weissnat**
Location: Port Johnnieberg
Dolorem ad laudantium explicabo quaerat commodi et repudiandae officiis. Nostrum laborum nemo facilis. Sed distinctio voluptas officia deleniti blanditiis consequatur.
- Abigayle Collins**
Location: Hegmannshire
Exercitationem et excepturi id vel quo et est. Ea voluptatum consequatur et.

At the bottom of the page, a footer bar includes the text 'Showing 1 to 5 of 12 results' and a set of navigation arrows (left, right). The footer also contains the text 'FINAL YEAR PROJECT : Machine Learning-Based Job-Hunting Web Application for CV Optimisation © 2025 RAY RAY RACHEL. All rights reserved.'

Figure 29 Company List Page

3. Company Profile Page

The company profile page shows the profile information for the company, the statistics (job posted and comment received) and the default showing job list of the company. When the tabs are clicked in the statistic card, it will toggle to the comment list of that company. As the component dispatches a message, the profile page listens to it and changes the displayed list without refreshing the whole page (AJAX implementation).

The screenshot displays the company profile page for 'Abigayle Collins'. At the top, there is a navigation bar with a user icon, 'Welcome', 'Jobs', 'Companies', 'LOGIN', and 'REGISTER'. Below the navigation bar, the title 'Abigayle Collins 's Information' is centered above a dark blue header. To the left of the header is a circular profile picture with a checkered pattern. To the right of the header are sections for 'About Company', 'Official Website', 'Location', 'Phone Number', 'Company Start Date', and 'Account Created'. Below this, two teal-colored boxes show 'Jobs Posted' (9) and 'Comments Received' (12). A section titled 'Jobs From This Company' follows, featuring a search bar with 'Search and filter jobs', a 'Search Jobs...' input field, and a date dropdown. Underneath is a 'Listed jobs' section with two items. The first item is 'Reporters OR Correspondent - Abigayle Collins', which includes a tree icon, location (3698 Jennifer Motorway Apt. 041 Lake Malloryborough, IL 27501-1978), salary (\$40k-\$60k), and a snippet of text. The second item is 'Network Admin OR Computer Systems Administrator - Abigayle Collins', also with a tree icon, location (51885 Damon Spurs Suite 018 East Bereniceville, NC 03810-7297), salary (\$80k-\$100k), and a snippet of text.

Figure 30 Company Profile Default Showing Job Posts

```

1 reference | 0 overrides
public function refreshStats(): void
{
    $this->jobCount = JobPost::where(column: 'user_id', operator: $this->companyId)->count();
    $this->commentCount = Comment::where(column: 'company_id', operator: $this->companyId)->count();
    $this->dispatch(event: 'toggleStatJobList');
}

0 references | 0 overrides
public function toggleJobList(): void
{
    $this->dispatch(event: 'toggleStatJobList');
}

0 references | 0 overrides
public function toggleCommentList(): void
{
    $this->dispatch(event: 'toggleStatCommentList');
}

```

Figure 31 Code Showing AJAX implementation

The screenshot shows a company profile page for 'Abigayle Collins'. At the top, there is a navigation bar with links for 'Welcome', 'Jobs', 'Companies', 'LOGIN', and 'REGISTER'. Below the navigation bar, the title 'Profile Details' is displayed. The main content area has a header 'Abigayle Collins's Information'. Underneath the header, there is a circular profile picture and some basic account information: 'About Company: Exercitationem et excepturi id vel quo et est. Ea voluptatum consequatur et.', 'Official Website: <https://www.kemmer.com/odit-officia-est-qui-vitae-consequuntur-non>', 'Location: Hegmannshire', 'Phone Number: +18163307519', 'Company Start Date: 1974-05-02', 'Years: 51 years', and 'Account Created: 6 days ago'. Below this, there are two green boxes: 'Jobs Posted' (9) and 'Comments Received' (12). The next section, 'Thoughts On This Company', contains five comments from different users (Paul Gor, Dr. Devin Herman V, Ms. Estel Rowe) with their respective profile pictures and timestamps (5 or 6 days ago). The comments are presented in a list format with blue backgrounds and white text.

Figure 32 Company Profile Page Showing The Toggled Comment List

The company job list reuses the component used in the job list page as mentioned in the last section. The controller fetches only the job with a specific company user ID because when calling the Livewire component, the company ID (profile ID) is also passed as a parameter.

```

3 | Route::get(uri: '/profile/{profileId}', action: function ($profileId): View {
4 |     return view(view: 'profile-detail', data: ['profileId' => $profileId]);
5 | })->name(name: 'profile.detail');
6 |
7 |

```

Figure 33Code Showing The Route For Company Profile

COMPANY PERMISSION

1. Company Portal

The following flow diagram shows the app flow of the company portal. After logging in, the company dashboard will default to showing the profile tab. One tab is for the company to see the received application. Another tab is for users to post jobs and show the job list. Both portals can view their comment and edit their profile.

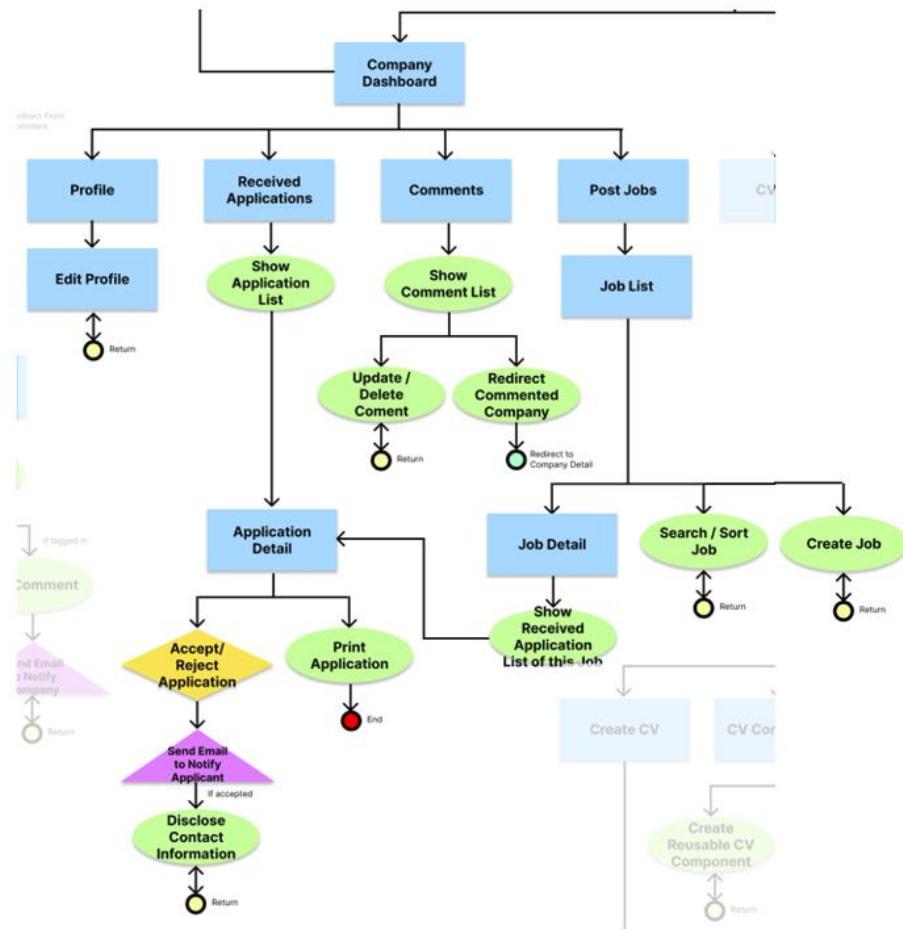


Figure 34Flow Diagram Of Company Portal

The Job Listing tab allows companies to post their jobs and view their past job listings. There is a toggle button to show and hide the section using JavaScript.

Figure 35 Company Job Listing Tab

In the received application page, company users can see the list of received applications for all their job postings with a status of pending. When clicking on the item, it will redirect to the application detail page. It will show a prompt asking if the user accepts the application. The user can only see the user information section when they accept the application. This way, the system will know if the CV attracts interview chances; it can be used for further AI development as labelled input.

Figure 36 Received Application Page Showing Accept Or Reject Prompt

```

if ($this->applicationId) {
    if (
        Auth::id() === $this->application->user_id ||
        (Auth::id() === $this->application->job->user_id && $this->application->status === 'accepted')
    ) {
        $this->canViewContactInfo = true;
    }
} else {
    if (Auth::id() === $this->cv->user_id) {
        $this->canViewContactInfo = true;
    }
}

```

Figure 37Code Showing Logic Of Displaying Contact Information

2. Job-hunter Portal

The following flow diagram shows the app flow of the job-hunter portal. Job-hunter can manage their CV and their reusable CV component in the MyCV tab.

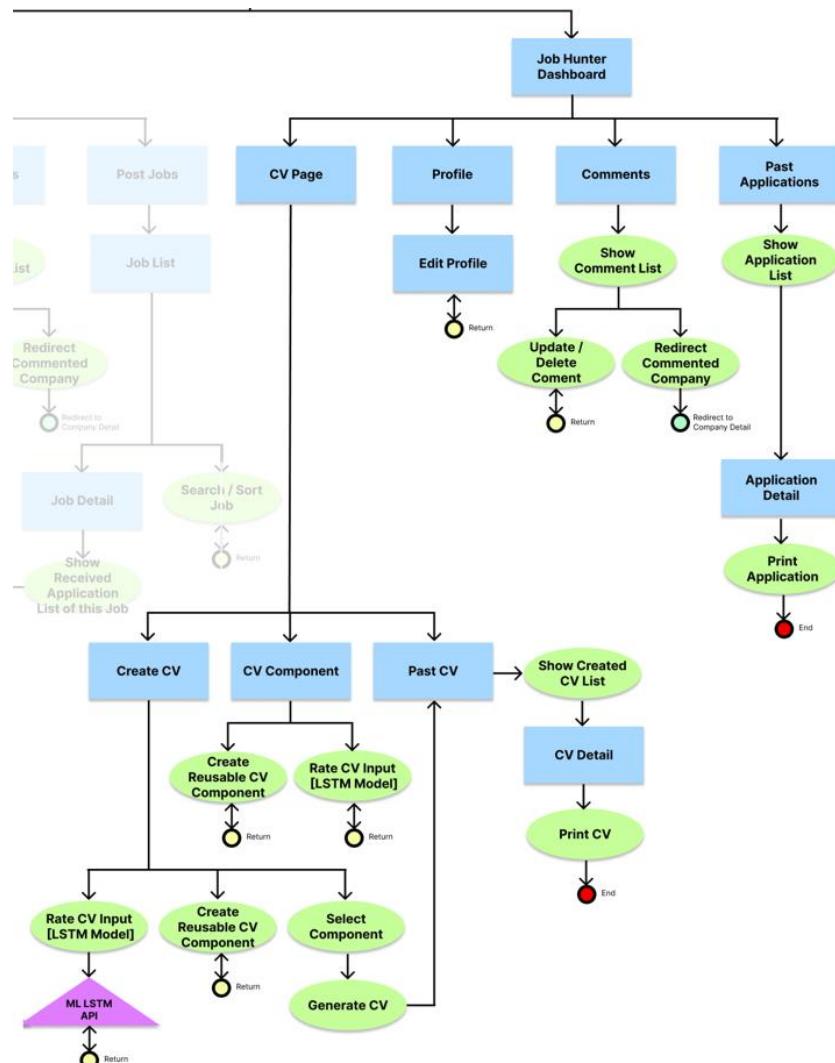


Figure 38Flow Diagram For Job-Hunter Portal

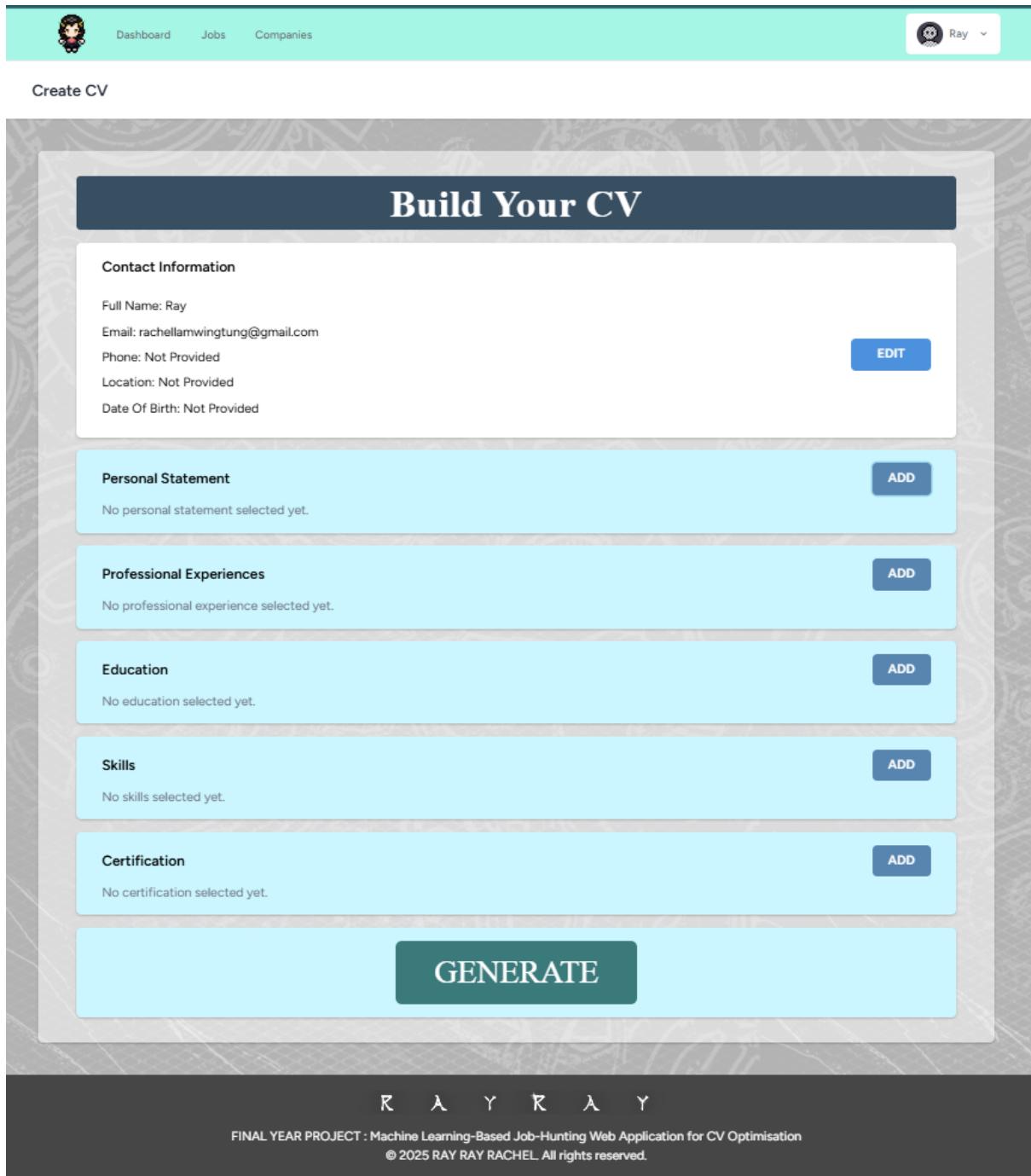


Figure 39 Make Reusable CV Component Page

Users can make a reusable CV component in the reusable CV component page or when they are creating a CV and application. This page acts as a place to manage the CV components for users to perform CRUD functions. When creating the component, there is an LSTM rate component button; the user can use it to rate the input to test if it is a constructive input with quantifiable results.

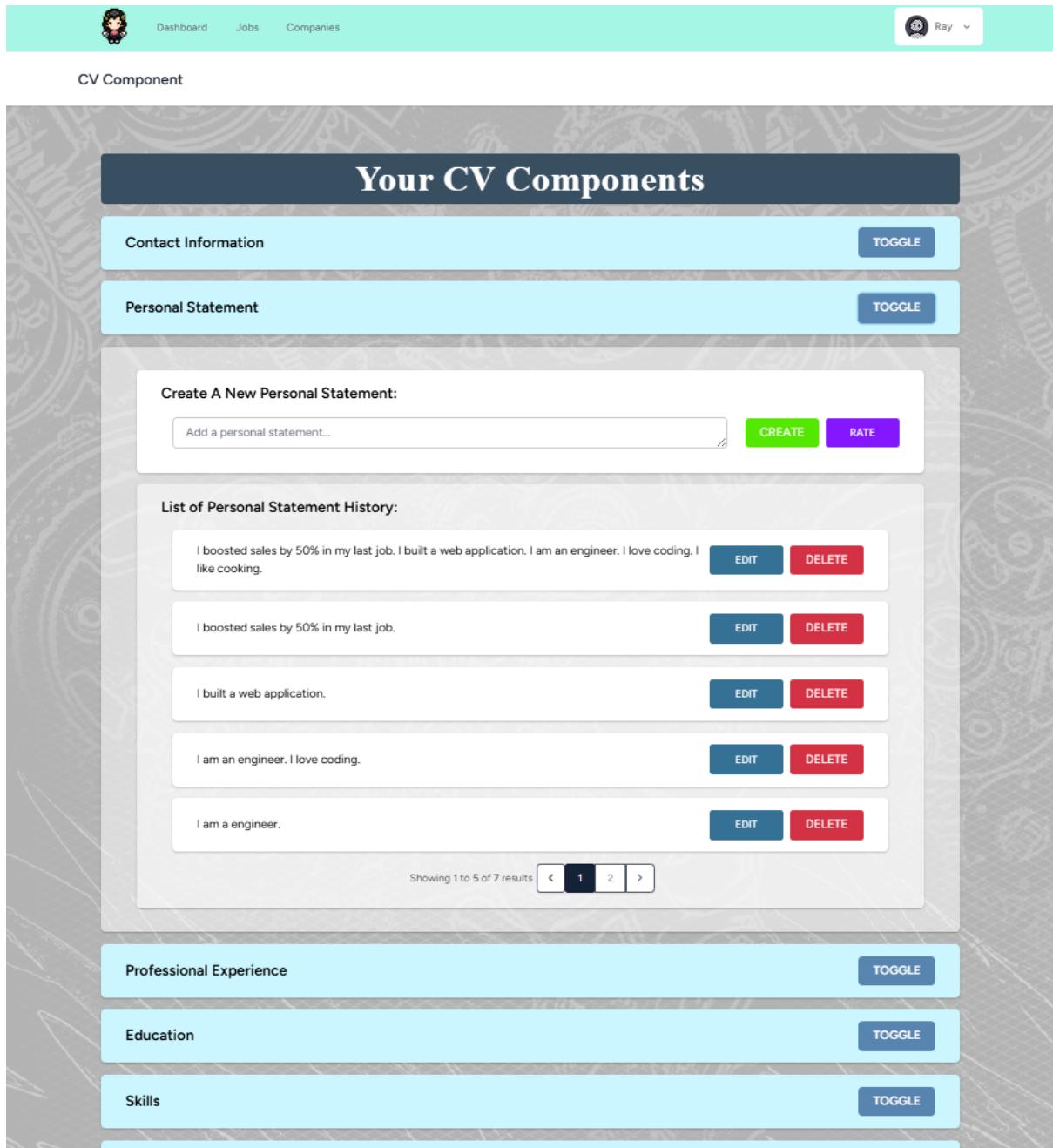


Figure 40 Personal Statement Component Toggled Out With CRUD and LSTM Rate Function

The pagination here will scroll to the top of the element to allow better user experience. It is caused by the following line of code.

```

<div class="pagination mt-4">
    {{ $educations->links(data: ['scrollTo' => false]) }}
</div>
...

```

Figure 41 Code Preventing Scroll To Top Of The Page When Pagination Is Clicked

When creating a CV, job-hunter can select past cv component or create a new one. Same Livewire component used here with the conditional logic of showing select button instead of delete button to avoid user from accidentally removing the component instead of selecting it.

The screenshot shows the 'Create CV' page with the following components:

- Contact Information:** Displays basic details like Full Name, Email, Phone, Location, and Date of Birth. An 'EDIT' button is present.
- Personal Statement:** Shows a message 'No personal statement selected yet.' with a 'CLOSE' button.
- Create A New Personal Statement:** Contains a text input field 'Add a personal statement...' and 'CREATE' and 'RATE' buttons.
- List of Personal Statement History:** A table-like list with five rows, each containing a statement and 'SELECT' and 'EDIT' buttons. The statements are:
 - I boosted sales by 50% in my last job. I built a web application. I am an engineer. I love coding. I like cooking.
 - I boosted sales by 50% in my last job.
 - I built a web application.
 - I am an engineer. I love coding.
 - I am a engineer.
 A pagination control 'Showing 1 to 5 of 7 results' is at the bottom of this section.
- Professional Experiences:** Shows a message 'No professional experience selected yet.' with an 'ADD' button.

Figure 42 Create CV Page

The selected component will show in the element container, removing it will go back to showing the cv component Livewire component.

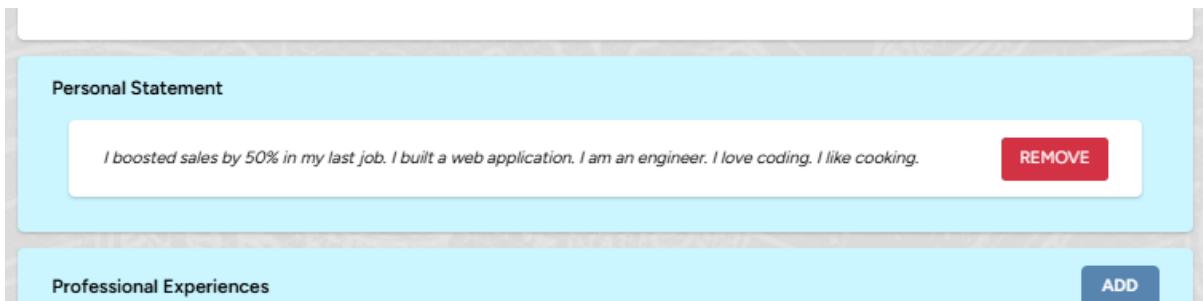


Figure 43 Selected Component

The following diagram shows the application flow of applying for jobs. When user is authorized, it shows a button to apply for job in the job detail page. Users can access the application page clicking on the apply button.

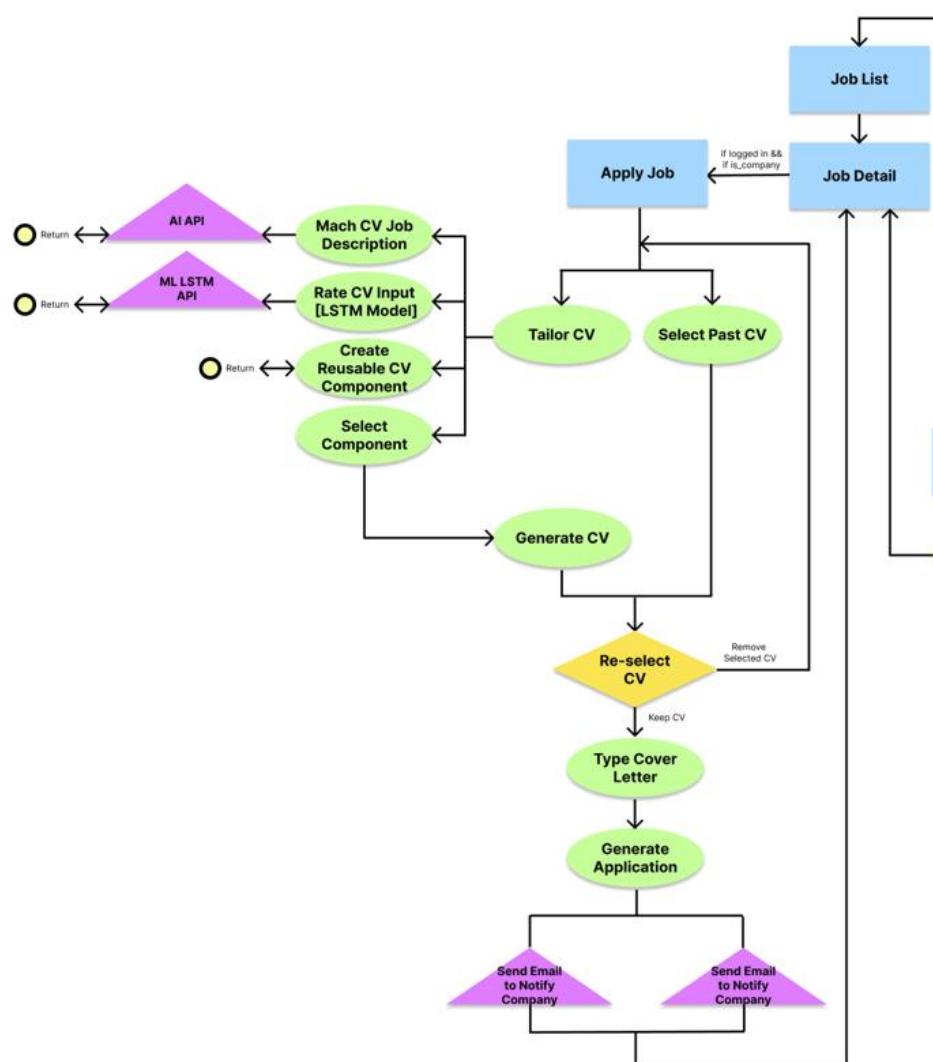


Figure 44 Flow Diagram For Application Process

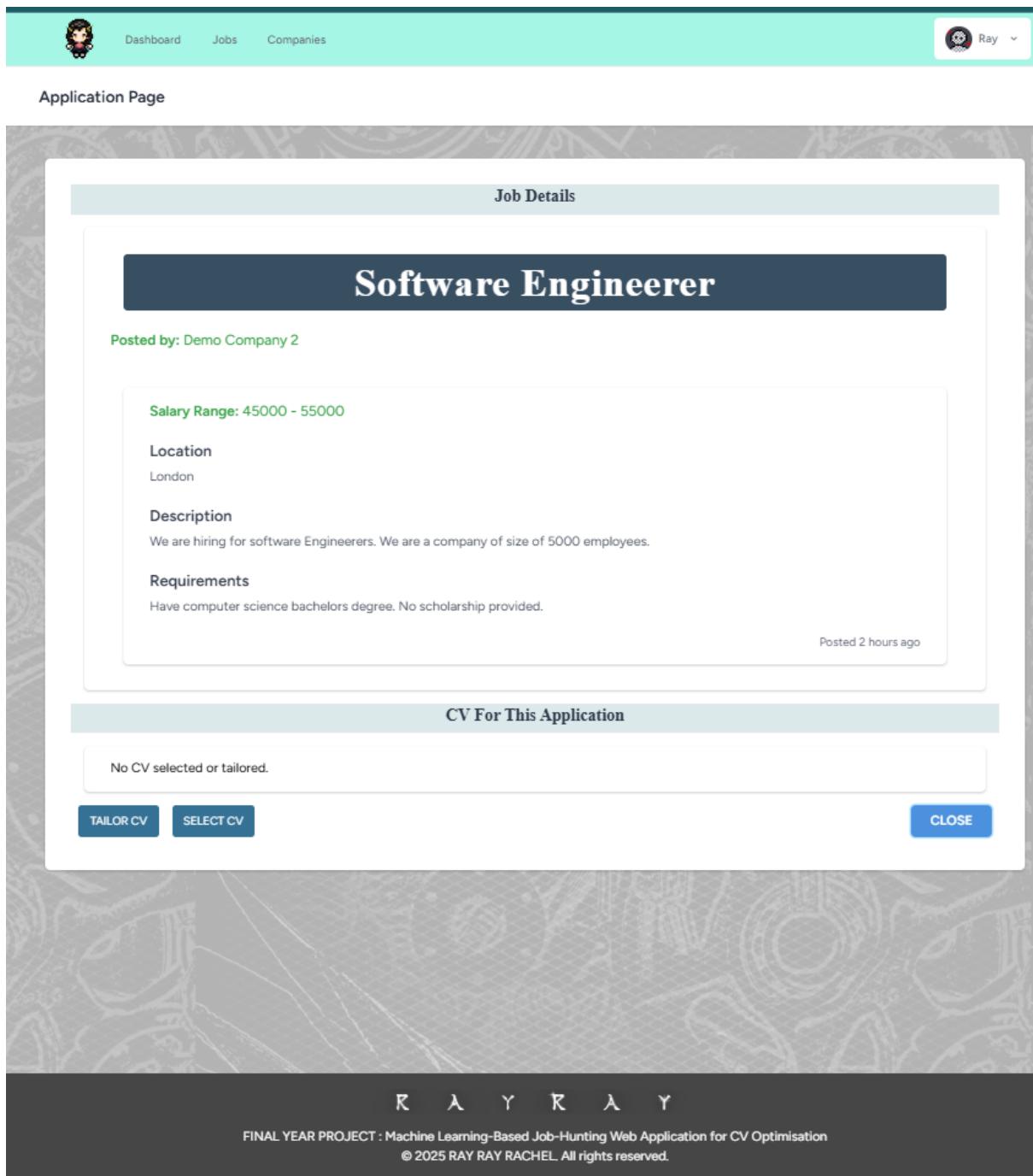


Figure 45Application Page

The application page allows users to tailor a CV according to the job description or select a past CV. When tailoring the CV, the match keyword with job description button appears and allows the user to use the AI model to match keywords. After a CV is selected or tailored, it will be displayed on the section. The cover letter section will then show up, and the apply button will be visible.



Figure 46AI Match Keyword Button Showing In Application Page When Tailoring A CV

5.1.5 Authentication

Laravel Breeze was implemented in this project. It provides routes, controllers, and views for authentication-related features. The reason for using Breeze is that it has the benefit of being beginner-friendly; it has inbuilt functions for registration, login, logout and password management. Implementing authentication manually is not recommended for inexperienced developers, as it involves complex logic related to password hashing, session management, and secure data encryption. By using Laravel Breeze, the project ensures a secure authentication system.

5.1.6 CRUD – Create, Read, Update, Delete

In the web application, there are CRUD operations implemented extensively. For example, CV components, company comments and CVs. However, certain models, such as job postings and applications, are intentionally limited and do not include update functionality. This design choice ensures data integrity and preserves the accuracy of submitted job posts and applications, which are treated as finalised once created.

5.1.7 File Upload

The system supports file upload functionality to enhance user personalisation and company branding. Users can upload profile pictures, which are automatically stored in Laravel's public storage directory. These uploaded images are then displayed across the platform, for example, in the comment section next to user posts or as a company logo on job listings and company profiles. The file upload process is handled using Laravel's built-in storage system.

```
@if ($job->user->profile && $job->user->profile->profile_picture)
    <a href="{{ route(name: 'profile.detail', parameters: ['profileId' => $job->user->profile->id]) }}" wire:navigate>
        <div class="mr-4">
            
        </div>
    </a>
@endif
```

Figure 47Code Showing Fetching Profile Picture From Storage Link

5.1.8 Security and encryption

1. Form Data Validation

Each form submission is validated on the server side using Laravel's built-in validation rules.

```
0 references | 0 overrides
public function saveEditedExperience(): void
{
    $this->validate([
        'editedExperience.job_title' => 'required|string|max:255',
        'editedExperience.company_name' => 'required|string|max:255',
        'editedExperience.location' => 'required|string|max:255',
        'editedExperience.start_date' => 'required|date',
        'editedExperience.end_date' => 'nullable|date',
        'editedExperience.key_achievements' => 'nullable|string',
        'editedExperience.quantifiable_results' => 'nullable|string',
    ]);

    $experience = ProfessionalExperience::find(id: $this->editingExperienceId);
    $experience->update($this->editedExperience);
}
```

Figure 48Code Showing Input Validation

* Company Name:

The company name field is required.

* Location: sf at sf sf | Started: 2025-04-02 | Ended: 2025-04-12 Key Achievements: N/A Quantifiable Results: N/A

The location field must not be greater than 255 characters.

* Start Date: mm/dd/yyyy

Figure 49UI Showing Input Validation

2. Access Control – route guard

There are different permissions for the three roles, guest, company login or job-hunter login. Route guard ensures that only authenticated users can access specific routes.

```
Route::middleware('auth')->get(uri: '/edit-profile', action: function (): View {
    return view(view: 'edit-profile');
})->name(name: 'edit-profile');
```

Figure 50Code Showing Route Guard For Authenticated User Only

3. Access Control – check authentication

The system verifies user identity before granting access to sensitive pages or operations. It uses Laravel's authentication mechanisms to ensure that only logged-in users can perform actions such as editing profiles, posting jobs, or submitting applications.

Additionally, the database model includes an `is_company` field to distinguish between company users and job hunters. This value is frequently checked throughout the application to determine what content, and actions should be displayed to the user.

```

@if ($fromJobDetail && auth()->check() && !auth()->user()->profile->is_company)
    <div>
        <a href="{{ route(name: 'application-page', parameters: ['jobId' => $job->id]) }}" wire:navigate>
            <button class="editing-button">APPLY</button>
        </a>
    </div>
@endif

```

Figure 51Code Showing The Authentication Check For Showing Apply Button

Furthermore, the system performs strict ID matching, validating that the requested item belongs to the authenticated user, before allowing access or modification. This prevents unauthorized access and enforces data ownership.

```

    </div>
@elseif (Auth::id() === $comment->hunter_id)
    <div class="flex gap-2 mt-2 goto">
        <button wire:click="toggleEditForm({{ $comment->id }})" class="edit-button">
            {{ $editingCommentId === $comment->id ? 'QUIT' : 'EDIT' }}
        </button>
        <button wire:click="confirmDelete({{ $comment->id }})" class="delete-button">
            DELETE
        </button>
    </div>
@endif

```

Figure 52Code Showing Authentication ID Check For Showing Edit and Delete Button

5.1.9 Debugging

During development, bugs were constantly encountered. Laravel's built-in error handling system displays detailed error messages, including stack traces. Another debugging method is the dump and die function. It logs the data flow and database queries for troubleshooting. Reading the error message in the browser using inspect mode also gives an idea of the bug.

5.1.10 Email

Emails can be sent along with operations in this web application for confirmation or notification purposes. For when a comment is made, the commented company will receive an email; when an application is made, the applicant will receive a confirmation email, and the company will receive a notification email. An email will be sent also to the applicant when the status of the application has been changed.

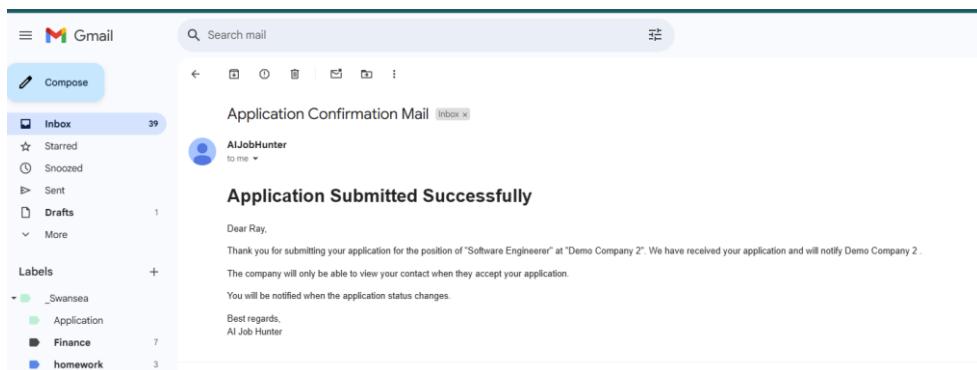


Figure 53Example Email Received In Gmail Inbox

5.2 Database Design & Structure

5.2.1 Database Structure

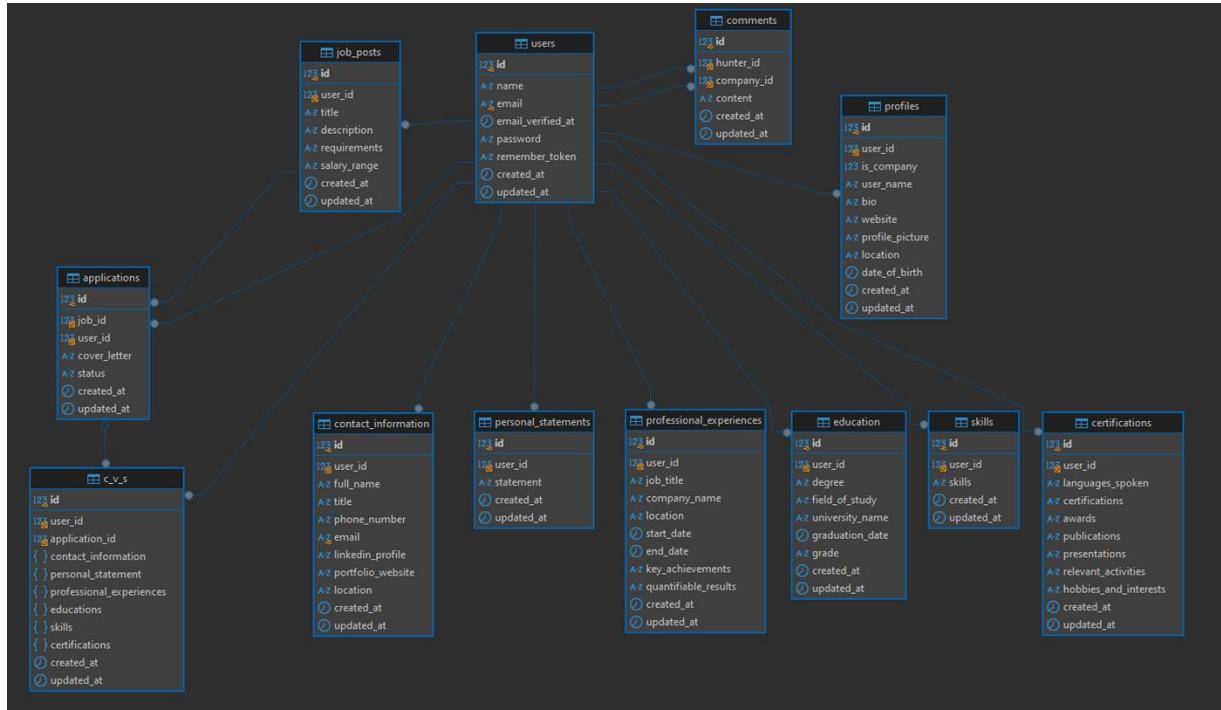


Figure 54 Database Structure Showed Using DBeaver

5.2.2 Database Relationship

This Laravel application uses models, migrations, seeders, and factories to define and manage the database structure.

- The core of the system is the User model. When a user is created, a related Profile is automatically created. This profile includes a field called `is_company` to distinguish between job hunters and companies.
- The Comment model stores comments created by all users.
- The JobPost model stores job listings created by users who are companies.
- The Application model stores job applications submitted by job hunters for specific job posts.
- There are multiple CV component models (Personal Statement, Education, Work Experience, Skills, Certifications) that allow users to store reusable CV components.
- The CV model stores the user's selected CV components in a JSON format, representing a complete CV that can be submitted with an application.

5.2.3 Database Relationship Visualization

Entity 1	Relationship Type	Entity 2	Description
User	One-to-One	Profile	Each user has one profile; each profile belongs to one user
User	Many-to-Many	User (Company)	Many users can create multiple comments on the company user.
User (Company)	One-to-Many	JobPost	A company user can create multiple job posts
JobPost	One-to-Many	Application	Each job post can receive multiple applications
User (Job Hunter)	One-to-Many	Application	A job-hunting user can submit multiple applications
User (Job Hunter)	One-to-Many	CV Components	A user can have many CV components of each type (e.g., education, exp.)
User (Job Hunter)	One-to-Many	CV	A user can create multiple CVs

Table: Database Relationship Table

5.2.4 Database Implementation

Laravel uses model files to represent database tables and relationships, such as a user has one profile. Migrations define the structure of table. Seeders populate the tables using factories, while factories are used to generate fake data.

Tinker was used for making SQL queries to test the database for debugging issue.

5.3 AI Model Integration

Two API was developed for the NLP AI job description match feature and the LSTM Rating CV component feature. In create new cv component, the input can match with the job description and rate the constructiveness with LSTM. They use FastAPI to connect Laravel

PHP with Python coding by post method. Laravel will uses POST request to the deployed API link and get the good sentences, bad sentences and results.

```
$response = Http::post(url: 'https://check-statement-api-ggffauf4byexc9a2.azuresouth-01.azurewebsites.net/predict', data: [
    'paragraph' => $this->paragraph,
]);
```

Figure 55 Laravel Code Posting To API

```
63     class TextRequest(BaseModel):
64         paragraph: str
65
66     @app.post("/predict")
67     def predict_paragraph(req: TextRequest):
68         sentences = sent_tokenize(req.paragraph)
69
70         good_sentences = []
71         bad_sentences = []
72         results = []
73
74         for sent in sentences:
75             score = predict_statement(sent)
```

Figure 56 Code Demonstrate How FastAPI Works

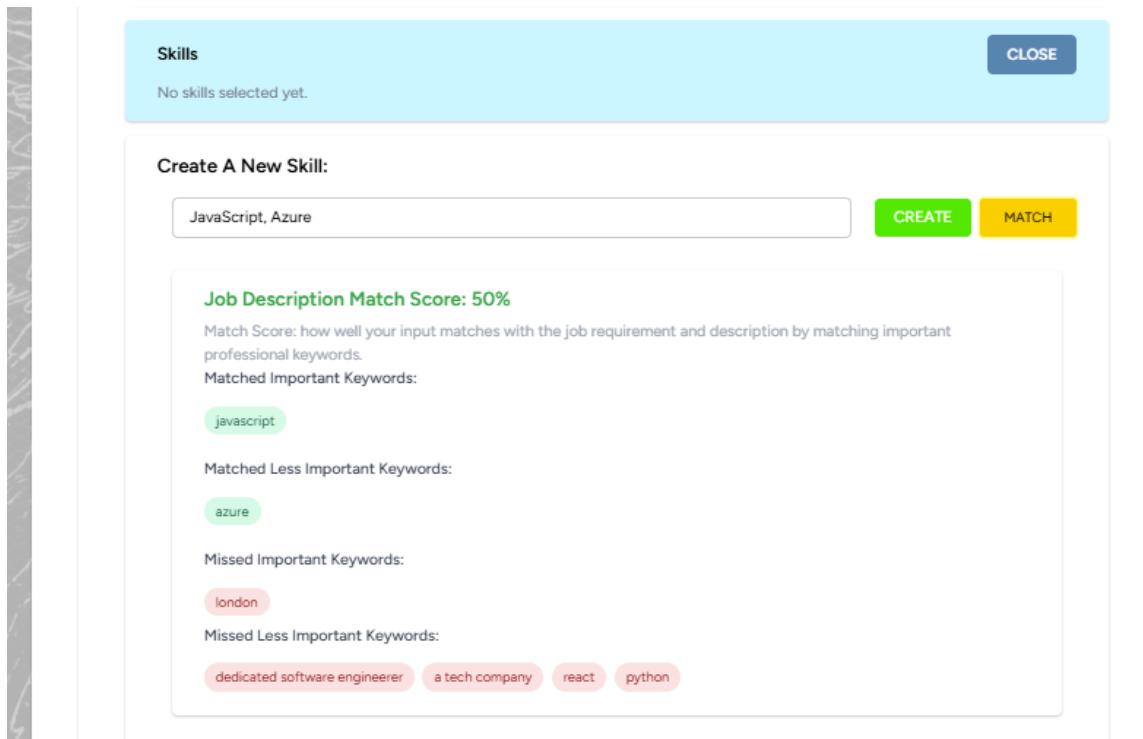


Figure 57 Example Usage Of AI Job Matching For CV Optimisation.

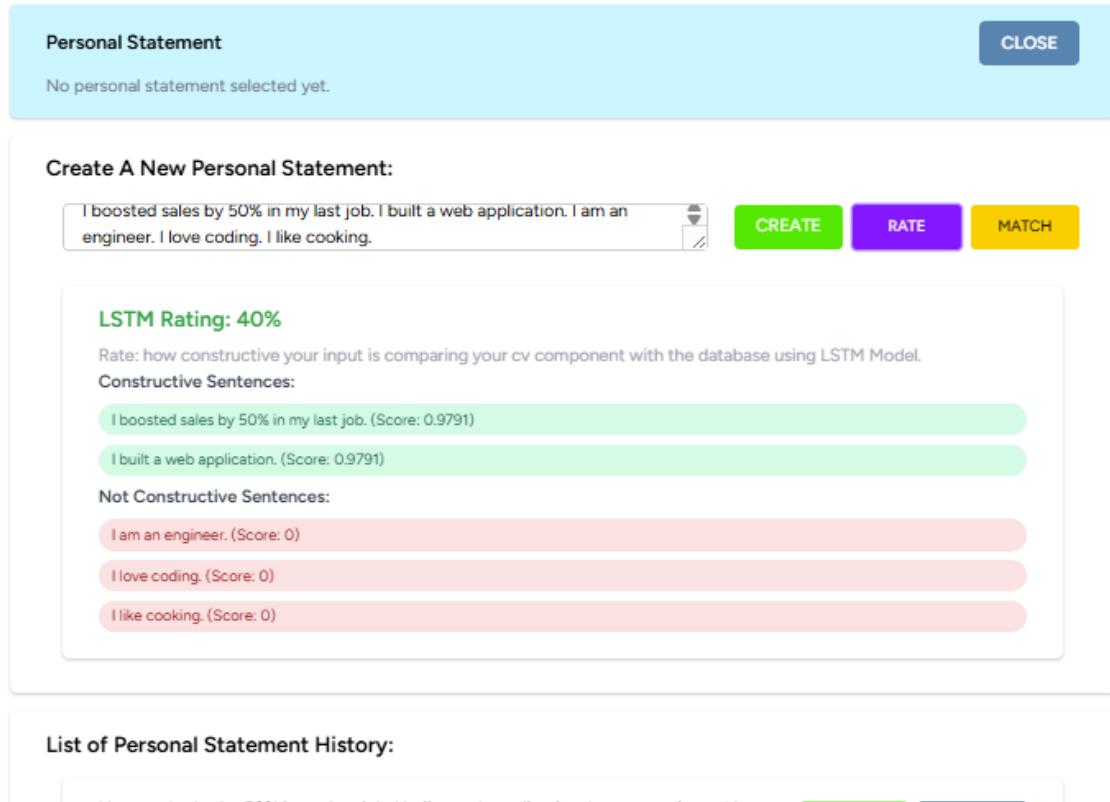


Figure 58 Example Usage LSTM Model For CV Optimisation.

The deployed API can be tested using the following links.

keyword match API link:

<https://my-cv-match-api.azurewebsites.net/docs>

This is the LSTM API check link:

<https://check-statement-api-ggffauf4byexc9a2.eksouth-01.azurewebsites.net/docs>

5.3.1 NLP

The keyword match system uses NLP library SpaCy and the pre trained model “en_core_web_sm” to compare the input text and the job description, requirement text. The implementation referenced the SpaCy documentation [43]. The goal is to help job-hunter to understand how well their CV aligns with the job posting and check if there are missing areas.

NLP SCRIPT

The method used was keyword extraction with SpaCy, it scans for named entities, for example, “Microsoft”, “Python”, “London”. It also scans for proper nouns and nouns that are not stopwords such as, “developer”, “engineer”. These are important keywords that used for scoring.

The remaining multi-word noun phrases are considered as the less important keywords it will be shown as reminding the job-hunter of some missing area.

Laravel POST to the API link, with the job description and cv input string, it will return the score, matched important keywords, missed important keywords and other missed and matched keywords. The score is calculated through the formula:

$$\text{Score} = \frac{\text{matched important keywords}}{\text{total important keywords in job description}} \times 100$$

5.3.2 LSTM

LSTM CONCEPT

LSTM works by maintaining a cell state, which is memory that carries relevant information through time steps in sequence [44]. It uses three types of gates (input, forget and output gate) to regulate the flow of information [44].

The input gate decides which new information should be stored in the cell state [44]. In this case, LSTM will carry useful information from the CV input across the whole sentence without forgotten them in the middle of the process.

The forget gate determines which information should be discarded from the previous cell state [44]. The model here will forget vague, generic and uninformative CV input phrase.

Finally, the output gate controls which information from the current cell state should be passed to the next hidden state [44].

These gates are activated using sigmoid functions, while new candidate values for the cell state are created using a tanh function [44]. The sigmoid functions determine if the final sentence is considered “constructive” or “unconstructive”. This allows LSTM to retain long-term dependencies and avoid issues like vanishing gradients [44].

An example sentence is “Increased sales by 35%”, the cell state will keep useful information like “increased” and “by 35%” without forgetting them. While for “I worked hard”, “work” would be forgotten as it is too generic and unconstructive. This way, CV sentences will filter unimportant vague phrases and boost the quantifiable results.

TRAINING DATA COLLECTION

3000 entry of labelled synthetic data was generated using a Large Language Model (LLM) for training data. 1500 “constructive” sentences, 1000 “unconstructive” sentences and 500 “vague unconstructive” sentences were generated.

Constructive sentences example:

641	Optimized customer service workflows, reducing turnaround time by 64%.	1
642	Played a key role in improving the quality of deliverables.	1
643	Presented research findings at the International AI Symposium.	1

Figure 59 Constructive Sentences Data In Training Database

Unconstructive sentences example:

652	Did various tasks without proper supervision or guidance.	0
653	Preferred tasks that required minimal effort.	0

Figure 60 Unconstructive Sentences Data In Training Database

Vague Unconstructive sentences example:

2966	Supported projects but not sure what my exact role was.	0
2967	Assisted on projects but didn't contribute significantly.	0
2968	Contributed to meetings but rarely led discussions.	0
2969	Tried helping out with team activities occasionally.	0

Figure 6157 Vague Unconstructive Sentences Data In Training Database

Reason of using Large Language Model for generating synthetic training data:

One of the key benefits of using a LLM for generating training data is the ability to respect data privacy and ethical boundaries. CV inputs often contain highly confidential and personally identifiable information, making it unethical and illegal to use real resumes from online sources without explicit consent according to the UK government [25]. Therefore, data scraping from using online resources is not a valid and ethical approach for obtaining training data. Additionally, manually crafting and labelling thousands of synthetic examples is both time-consuming and unrealistic. In contrast, LLM allows for efficient generation of diverse and contextually appropriate examples that reflect both constructive and unconstructive CV statements.

This way it ensures no violation of individual privacy, no unauthorized reuse of creative content from real individuals and it stays with ethical standards mentioned in the government document for AI data usage [25]. Thus, using LLMs for data generation provides a practical and ethically responsible solution to training CV evaluation models.

This approach demonstrates a valid and meaningful application of AI.

PYTHON SCRIPT

The python model code highly references the PyTorch LTSM tutorial documentation [45] and the past GitHub project [46].

In the beginning of the code, the Punkt Tokenizer was downloaded and loaded into the Python code from the Natural Language Toolkit (NLTK). It is used to split a paragraph of text into sentences and split a sentence into tokens.

Then it loads the generated dataset into the model. For preparation for data analysis, library NLTK was used to tokenise the text into all lowercasing word, flatten into a list: all_tokens. Then as a good practice, the occurrence of the tokenised words is counted, and order them based on their occurrence. It is for larger scale later when the database grew too large, we keep the most appeared words for keeping the model computable. The last part is to store the most occurred vocabulary into a dictionary.

```

def tokenize(text):
    return word_tokenize(text.lower())

df['tokens'] = df['statement'].apply(tokenize)

all_tokens = [token for tokens in df['tokens'] for token in tokens]
token_counts = Counter(all_tokens)
vocab = {token: idx+2 for idx, (token, _) in enumerate(token_counts.items())}

# Handel when tokens not found uses id 1 for LSTM
vocab['<PAD>'] = 0
vocab['<UNK>'] = 1

def encode(tokens):
    return [vocab.get(token, vocab['<UNK>']) for token in tokens]

df['input_ids'] = df['tokens'].apply(encode)
MAX_LEN = max(df['input_ids'].apply(len))

def pad_sequence(seq, max_len=MAX_LEN):
    return seq + [vocab['<PAD>']] * (max_len - len(seq))

df['input_ids'] = df['input_ids'].apply(pad_sequence)

```

Figure 62 Preparation Code For LSTM

The tokens 0 and 1 are reserved for when 0 for interpolate padding and 1 for tokens not found [47].

```

# Hyperparameters
BATCH_SIZE = 32
EPOCHS = 100
LEARNING_RATE = 0.0005
EMBEDDING_DIM = 110
HIDDEN_DIM = 256

```

Figure 63 Hyperparameters For LSTM

The above are the hyperparameters for LSTM. Batch size is the number of training samples used in one forward or backward propagation [48]; here, 32 is a common choice. The epochs mean the model will loop over the database data 100 times [48]; here it is increased it to 100, where 50 is a common amount; it is to boost the opportunity to learn the data. Learning rate is the step size to make gradual adjustments to the model weight [48]. It was decreased here to reduce the chance of overshooting the optimal value. The embedding dimension means each word is represented as a vector with 110 dimensions [48]; it is increased from normal to allow

the model to increase recognition performance. Hidden dimension means the hidden layer [48]; it is increased to allow it to learn complex patterns.

The data is then split into 80% training data and 20% testing data [49]. Using the above hyperparameters, the model is trained using the following code.

```
# LSTM Model

class LSTMClassifier(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim):
        super(LSTMClassifier, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx=vocab['<PAD>'])
        self.lstm = nn.LSTM(embedding_dim, hidden_dim, batch_first=True, bidirectional=True)
        self.dropout = nn.Dropout(0.5)
        self.fc = nn.Linear(hidden_dim * 2, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        embedded = self.embedding(x)
        _, (hidden, _) = self.lstm(embedded)
        hidden_cat = torch.cat((hidden[-2,:,:], hidden[-1,:,:]), dim=1)
        dropped = self.dropout(hidden_cat)
        output = self.fc(dropped)
        return self.sigmoid(output)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = LSTMClassifier(vocab_size=len(vocab), embedding_dim=EMBEDDING_DIM, hidden_dim=HIDDEN_DIM).to(device)
criterion = nn.BCELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE)
```



```
# Training the Data

for epoch in range(EPOCHS):
    model.train()
    total_loss = 0
    for inputs, labels in train_loader:
        inputs, labels = inputs.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(inputs).squeeze()
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    print(f"Epoch {epoch+1}/{EPOCHS} - Loss: {total_loss/len(train_loader):.4f}")
```

Figure 64 Code For LSTM Model Referencing PyTorch Documentation [45]

```

...
Epoch 1/100 - Loss: 0.2065
Epoch 2/100 - Loss: 0.0003
Epoch 3/100 - Loss: 0.0001
Epoch 4/100 - Loss: 0.0001
Epoch 5/100 - Loss: 0.0001
Epoch 6/100 - Loss: 0.0001
Epoch 7/100 - Loss: 0.0000
Epoch 8/100 - Loss: 0.0000
Epoch 9/100 - Loss: 0.0000
Epoch 10/100 - Loss: 0.0000
Epoch 11/100 - Loss: 0.0000
Epoch 12/100 - Loss: 0.0000
Epoch 13/100 - Loss: 0.0000
Epoch 14/100 - Loss: 0.0000
Epoch 15/100 - Loss: 0.0000
Epoch 16/100 - Loss: 0.0000
Epoch 17/100 - Loss: 0.0000
Epoch 18/100 - Loss: 0.0000
Epoch 19/100 - Loss: 0.0000
Epoch 20/100 - Loss: 0.0000
Epoch 21/100 - Loss: 0.0000
Epoch 22/100 - Loss: 0.0000
Epoch 23/100 - Loss: 0.0000
Epoch 24/100 - Loss: 0.0000
Epoch 25/100 - Loss: 0.0000
...
Epoch 97/100 - Loss: 0.0000
Epoch 98/100 - Loss: 0.0000
Epoch 99/100 - Loss: 0.0000
Epoch 100/100 - Loss: 0.0000

```

Output is truncated. View as a [scrollable element](#)

Figure 65 Training Loss Function

The above figure shows when it loops to the 100 times, the loss function is 0 which indicates there is no overfitting. The below figure shows the confusion matrix for training and testing data. It shows a 100% correctly classified sentences in both the constructive and unconstructive categories, demonstrating the model's strong performance.



Figure 66 Confusion Matrix For Training Dataset

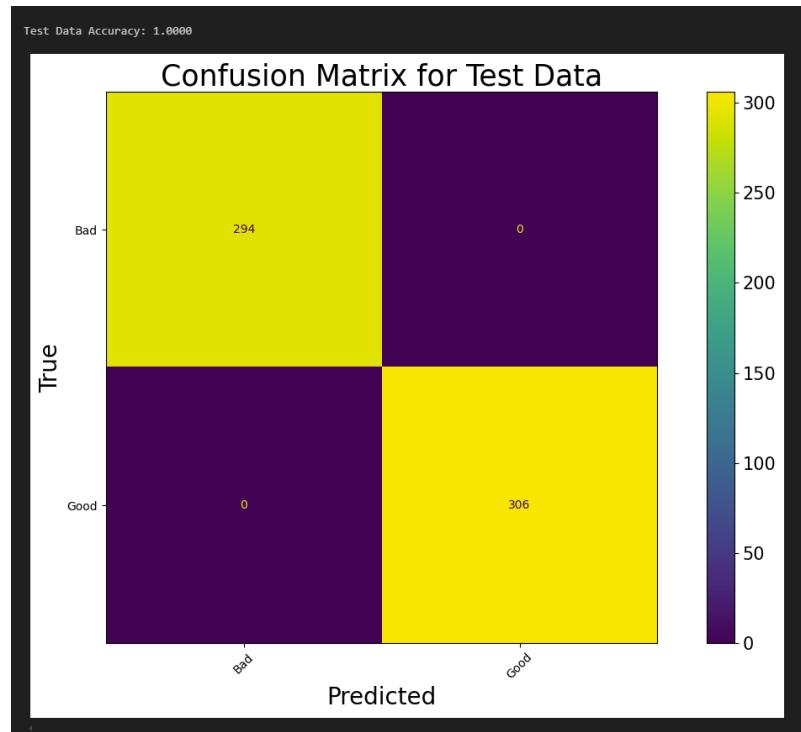


Figure 67Confusion Matrix For Testing Dataset

The below Figure visualizes the output.

		statement	score	true_label
0	Led a cross-functional team to deliver a proj...	1.0000		Good
1	Developed an AI-based system.	0.9811		Good
2	Managed a \$1M budget while reducing costs by 15%.	1.0000		Good
3	Presented research findings at an internationa...	1.0000		Good
4	Designed and implemented a secure database sys...	1.0000		Good
5	Achieved a 30% increase in sales through strat...	1.0000		Good
6	Authored several technical papers published in...	1.0000		Good
7	Organized and led weekly team meetings to impr...	0.9775		Good
8	Optimized website performance, reducing load t...	1.0000		Good
9	Created detailed project documentation and use...	1.0000		Good
10	I like working sometimes if the team is nice.	0.0000		Bad
11	I have some knowledge about computers maybe.	0.0000		Bad
12	Just looking for any opportunity, not picky.	0.0000		Bad
13	I'm not really sure what I want to do yet.	0.0000		Bad
14	Helped out with various things but no big proj...	0.0000		Bad
15	Prefer a job where I don't have to do too much.	0.0006		Bad
16	Did many things in previous job but don't reme...	0.0000		Bad
17	Looking for an easy job with flexible hours.I ...	0.0000		Bad
18	I love animals.	0.0000		Bad
		predicted		
0		Good		
1		Good		
2		Good		
3		Good		
4		Good		
5		Good		
6		Good		
7		Good		
8		Good		
9		Good		
10		Bad		
11		Bad		
12		Bad		
13		Bad		
14		Bad		
15		Bad		
16		Bad		
17		Bad		
18		Bad		

Figure 68LSTM Model Visualization

5.4 Deployment

Azure and GitHub was used for the deployment. GitHub inbuilt with a feature to link to the Azure portal. When the GitHub directory is configured in Azure, for every commit, it will automatically deploy to the production site. There are four deployments in total: one for the SQL database, one for the main web application, and two for APIs, which can be accessed through the links below.

Main web application production site:

<https://ai-job-hunter-bcapbad4cgdwcbh.eksouth-01.azurewebsites.net/>

keyword match API link:

<https://my-cv-match-api.azurewebsites.net/docs>

This is the LSTM API check link:

<https://check-statement-api-ggffauf4byexc9a2.eksouth-01.azurewebsites.net/docs>

The decision to deploy the application was made during the sprint after the Gregynog presentation, when most of the web application features had been successfully developed on localhost. However, the deployment process took a long time to debug, which led to a slight delay in later tasks.

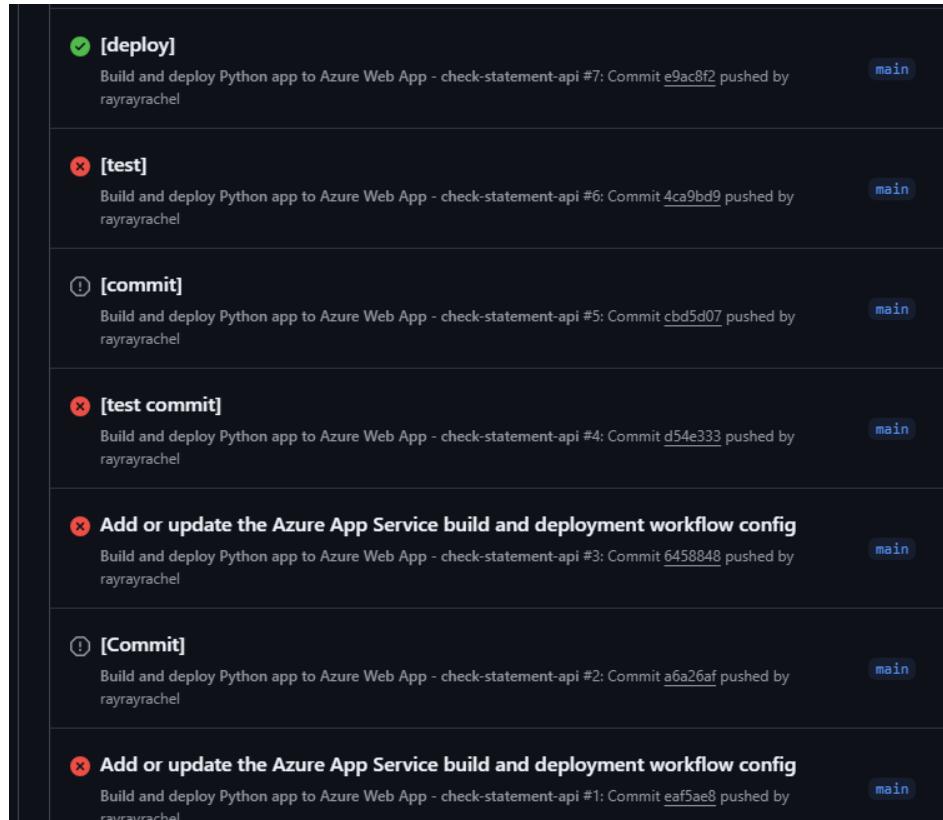


Figure 69 Deployment History In GitHub Showing Errors

Reading Azure documentation gave me insights; however the issue was solved by following a past project tutorial on YouTube. With the help of the tutorial, the issue of the application not starting correctly after deployment was resolved using SSH commands (implemented a startup.sh file) and the creation of a few additional files [50]. Since deployment is not part of the assessment rubric, it will not be discussed in detail.

The reason for choosing to deploy the application was due to the users being in different locations and time zones. Additionally, to challenge myself in full-stack development, deploying the project to a production site was essential.

5.5 Version Control – GitHub

Below is some of the git commit history. As it is stored online, version control was used as a precaution for hardware failure and to preserve the last successful commit. If there was an issue with the hardware, the complete project could be downloaded from the GitHub account into another device. With version control, more bold coding could be implemented between versions, as it allows reverting the code to the last version.

```
• [UI] Dashboard Tabs rayrayrachel
• [LIVEWIRE] Company Profile rayrayrachel
• [UI] Create Job Listing in Company Portal rayrayrachel
• [LIVEWIRE] [VOLT] create job rayrayrachel
• [CODE] Comany dashboard job list rayrayrachel
• [UI] Job list and sorting rayrayrachel
• [CODE] create profile when register rayrayrachel
• [UI] Minor Changes rayrayrachel
• [UI] Navigation rayrayrachel
• [LOGO] App logo rayrayrachel
• [UI] [NAVIGATION] [LIVEWIRE] added landing page rayrayrachel
• [UI] Added default profile picture, added footer, some other minor changes rayrayrachel
• [UI] Change Company List from table to card rayrayrachel
• [LIVEWIRE] [NAVIGATION] Company List with search feature, Navigation to display guest layout and user login rayrayrachel
• [DATABASE] [UI] Job List with search feature and sorting feature with ui and navigation rayrayrachel
• [UI] Stylesheet Initialise rayrayrachel
• [DATABASE] CV Model Migration Factory Seeder rayrayrachel
• [DATABASE] Certifications Model Migration Factory Seeder rayrayrachel
• [DATABASE] Skill Model Migration Factory Seeder rayrayrachel
```

Figure 70Partial Git Commit History

Version control also allows easier debugging. For example, there were issue with the API link in later versions after the following section of code was committed. It helps debugging what changes was made before the version to minimize the debugging area.

The screenshot shows a code editor interface with a dark theme. The file being edited is `CvMatcherComponent.php` located in the `app/Livewire` directory. The code is part of the `CvMatcherComponent` class, specifically within the `checkMatch()` method. The code block is as follows:

```
47     47
48     48     $jobText = $job->description . "\n" . $job->requirements;
49     49
50 -     $response = Http::post('http://host.docker.internal:8000/cv-match', [
50+     $response = Http::post('https://my-cv-match-api.azurewebsites.net/cv-match', [
51     51         'cv_text' -> $this->cvText,
52     52         'job_description' -> $jobText,
53     53     ]);

```

Annotations highlight the URL in the second line of the code block. The code editor also displays statistics at the top: 46 hidden lines and 19 hidden lines.

Figure 71 API Code History

It also boosts coding speed by searching the history of implementing a function of code when some feature was to be replicated.

6 Evaluation

6.1 User Testing

Throughout the development process, user-involved testing was performed twice. One was when the web application was developed, and another was when the AI feature worked locally. The final user evaluation took place after the deployment to production was completed and hosted on the live site.

1. First User-involved Testing

For the two local user testing sessions, a group of five app users was invited to test the site with a list of functions and were asked to provide evaluation. The usability heuristic form received relatively high scores. However, three out of five users felt that the application and CV creation process was not user-friendly, as they did not have control over the order of adding CV components. This was due to the initial use of a wizard-style approach. Therefore, the layout was changed to a long-page format with toggle sections.

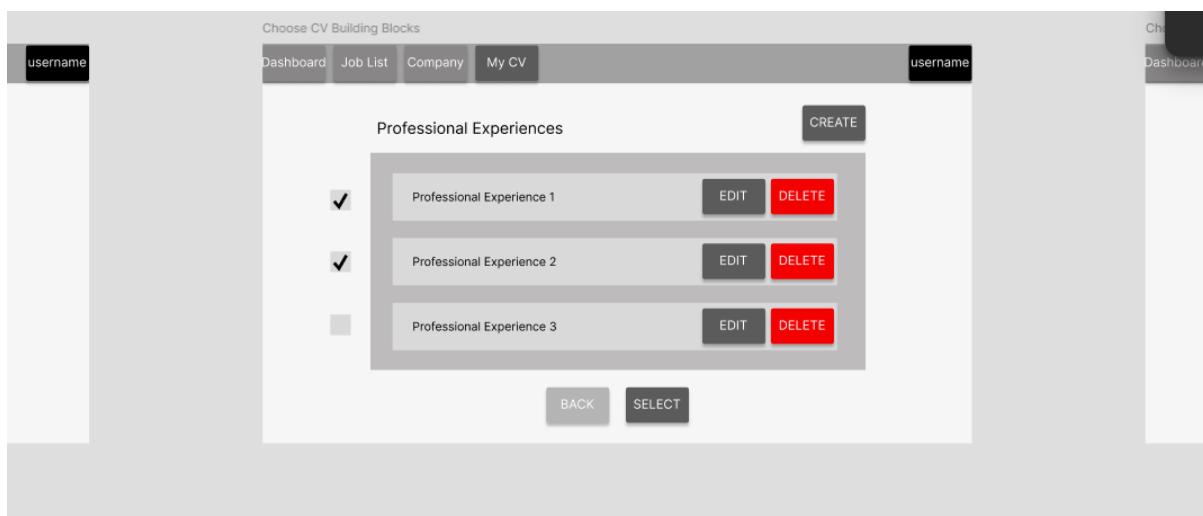


Figure 72 Mock-up Of Create CV Wizard Style

2. Second User-involved Testing

After the toggle section layout was implemented in the second user testing, the AI and LSTM functionalities were also completed. All five users reported that the entire application was user-friendly, and that the new layout improved the user experience. The reported time for creating a CV and applying for a job with prepared information was under two minutes.

3. Final User-involved Testing

The final user-involved testing on the production site included five users and five developers. The developers, who work at an IT software company. Tester helped identify several bugs, such as the email not being sent on the production site, which were later fixed. All test cases

passed in the finalized function list, and the expert testing evaluation is reconfirmed in a later section.

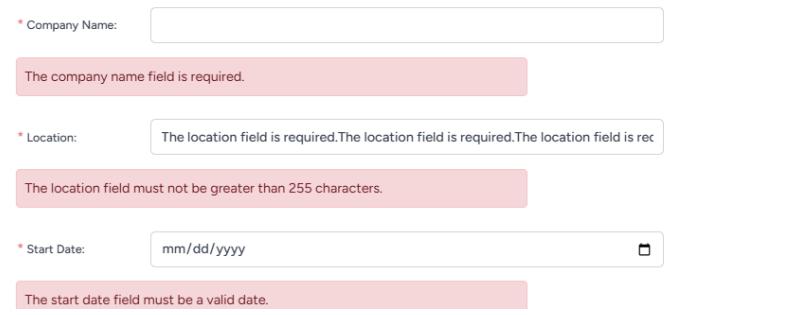
No.	Page Name	Functionality
1	Guest Job List	Search / Sort Job List With Different Conditions
2	Guest Company List	Search / Sort Company List
3	Guest Company Profile	View Company Profile/ Toggle Division
4	Register Page	Register Company Account
5	Register Page	Register Job-Hunter Account
6	Company Dashboard	Edit Profile
7	Company Dashboard	Post Job
8	Company Dashboard	Search / Sort Job
9	Company List	Add Comment
10	Company List	Update / Delete Comment
11	Job Hunter Dashboard	Update / Delete Comment
12	Job Hunter Dashboard	Redirect Commented Company
13	N/A	Logout
14	Welcoming Page	Login Job Hunter Account
15	Job Hunter Dashboard	Edit Profile
16	CV Component	Create Reusable CV Component
17	CV Component	Rate CV Input [LSTM Model]
18	Create CV	Rate CV Input [LSTM Model]
19	Create CV	Create Reusable CV Component
20	Create CV	Select Component
21	Create CV	Generate CV
22	Past CV	Show Created CV List
23	Past CV	Print CV
24	Job Detail	Click Apply Job
25	Application	Tailor CV
26	Application	Mach CV Job Description
27	Application	Rate CV Input [LSTM Model]
28	Application	Create Reusable CV Component
29	Application	Select Component
30	Application	Generate CV
31	Application	Re-select CV
32	Application	Select Past CV
33	Application	Type Cover Letter
34	Application	Generate Application
35	Personal Email	Check Email Inbox
36	Company List	Add Comment
37	Company List	Update / Delete Comment
38	Job Hunter Dashboard	Update / Delete Comment
39	Job Hunter Dashboard	Redirect Commented Company
40	Job Hunter Dashboard	Show Application List
41	Past Application	Print Application
42	N/A	Logout
43	Welcoming Page	Login Company Account

44	Company Dashboard	Received Applications
45	Company Dashboard	Show Application List
46	Application Detail	Accept/ Reject Application
47	Application Detail	Check Disclose Contact Information
48	Personal Email	Check Company Email Inbox
49	Personal Email	Check Job Hunter Email Inbox
50	Application Detail	Print Application
51	Job Detail	Check Show Received Application List of this Job
52	N/A	Logout
53	Welcoming Page	Login Job Hunter Account
54	Past Applications	Check Show Application List Status
55	Application Detail	Print Application

Table: Finalized Function List For Quality Assurance

6.2 Expert Evaluation

No	Usability Heuristics	P/F	Description
1	Visibility of System Status	Pass	<p>The system provides immediate feedback for user actions, such as loading indicators and success/error messages after form submissions.</p> <p><i>Figure 73Success Messages</i></p> <p><i>Figure 74Error Messages</i></p>
2	Match Between System and The Real World	Pass	<p>Terminology used throughout the app reflects real-world job-seeking and CV creation language, while common colour coding is applied for when delete button is red and a progress button is green, which matches the real world.</p>

3	User Control and Freedom	Pass	Full control in reusing CV components CRUD functionalities, with comments CRUD. Toggle button can be hide, while select button has a remove selected button after selected. Allow user going back to last process.
4	Consistency and Standards	Pass	Consistent button style, colour, placement, text style, consistent page layout, reused cv, job list, company list component. Ensures user familiar with using the system.
5	Error Prevention	Pass	<p style="text-align: center;">Your Comments On Other Company</p> 
6	Recognition Rather Than Recall	Pass	Dropdowns for filters and cv component selectors help user recognize options instead of remembering their id or their name.
7	Flexibility and Efficiency of Use	Pass	Experienced users can quickly complete repetitive actions using saved CV components, improving efficiency. Average application and creating CV time is under a minure.
8	Aesthetic and Minimalist Design	Pass	The interface uses a clean layout with clearly grouped sections, avoiding unnecessary information and clutter. Applied similar minimalistic design as Indeed.
9	Help Users Recognize, Diagnose and Recover from Errors	Pass	<p>Clear error messages guide users to correct their mistakes, such as explaining missing required fields and max input limitations.</p> 
10	Help and Documentation	Pass	Clear and informative instructions are implemented within forms.

6.3 Web Content Accessibility Guidelines (WCAG) 2

The web application was developed with reference to the Web Content Accessibility Guidelines (WCAG) 2 to ensure usability for individuals with diverse abilities. The design adheres to the four main principles: perceivable, operable, understandable, and robust [51].

Perceivable:

1. All text content maintains sufficient colour contrast against background elements [51].

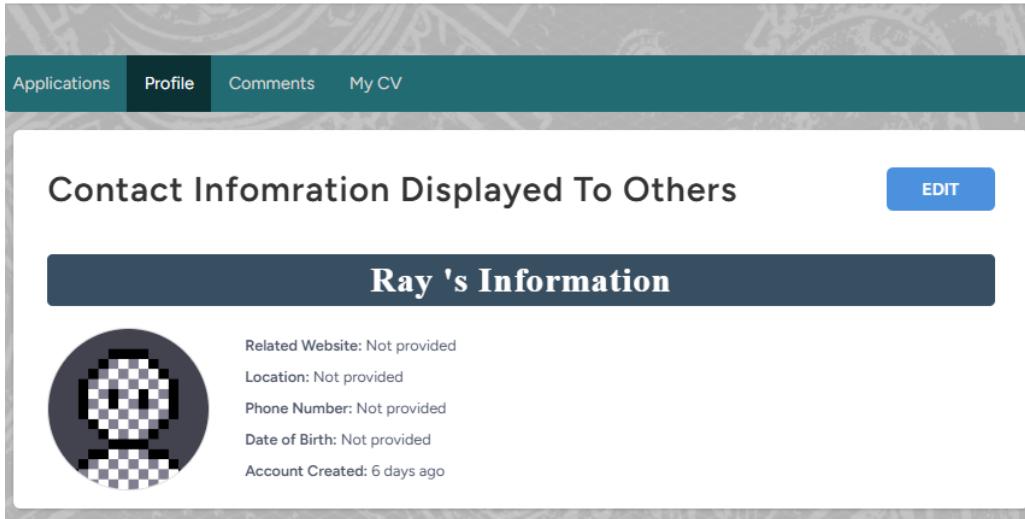


Figure 77 Contrasted Text With Background

2. Semantic HTML tags such as <label>, <header>, and <section> are used to improve screen reader compatibility [51].

```
<body class="font-sans antialiased" style="background-image: url('{{ asset(path: 'images/background.jpeg') }}'); ">
  <div
    |   style="position: fixed; top: 0; left: 0; width: 100%; height: 100%; background-color: #rgba(255, 255, 255, 0.7); z-index: -1;">
  @livewireScripts
  </div>
  <livewire:layout.navigation />

  <!-- Page Heading -->
  @if (isset($header))
    <header class="bg-white shadow">
      <div class="max-w-7xl mx-auto py-6 px-4 sm:px-6 lg:px-8">
        |   {{ $header }}
      </div>
    </header>
  @endif

  <!-- Page Content -->
  <main>
    |   {{ $slot }}
  </main>
  </div>
</body>

<footer class = "font-sans">
  <div class="container mx-auto px-4 text-center">
    <div class="mx-auto mb-4"
        |   style="max-width: 400px; height: 30px; background-image: url('{{ asset(path: 'images/footer.png') }}'); background-size: cover; background-position: center;">
    </div>
    <p>FINAL YEAR PROJECT : Machine Learning-Based Job-Hunting Web Application for CV Optimisation </p>
    <p>© 2025 RAY RAY RACHEL. All rights reserved.</p>
  </div>
</footer>
</html>
```

Figure 78 Tagged HTML

3. Alt text is provided for meaningful images [51].

```
mg src="https://ai-job-hunter-bcapbad4cgdwgbh.ksouth-01.azurewebsites.net/images/Logo.gif" alt="App Logo" class="logo">> == $0
```

Figure 79 Alt Text For App Logo

Operable:

1. All interactive elements, such as buttons and form inputs, are fully keyboard navigable [51].
2. Focus outlines are preserved to aid users relying on keyboard navigation [51].

The screenshot shows a modal window titled "Edit Your Profile". Inside, there are several input fields with blue outlines: "Website" (a text input field containing a single character), "Profile Picture" (a file input field showing "Choose File No file chosen"), "Location" (a text input field), "Phone Number" (a text input field), and "Date of Birth" (a date input field with a placeholder "mm/dd/yyyy"). Below these is a blue "Update Profile" button.

Figure 80 Keyboard Accessible Fields With Outlines

Understandable:

1. Forms provide clear input instructions and use real-time validation to guide users [51].

The screenshot shows a user interface for managing personal statements. At the top, a light blue box displays "Personal Statement" and "No personal statement selected yet.", with a "CLOSE" button. Below is a white box for creating a new statement, containing a text input field with the placeholder "Add a personal statement..." and two buttons: "CREATE" (green) and "RATE" (purple). At the bottom, a grey box lists a personal statement history entry: "I boosted sales by 50% in my last job. I built a web application. I am" followed by a "SELECT" button (green) and an "EDIT" button (blue).

Figure 81 Clear Input Instructions

- Buttons and navigation items use descriptive labels [51] (e.g., “Generate” instead of “Submit”) to reduce ambiguity.

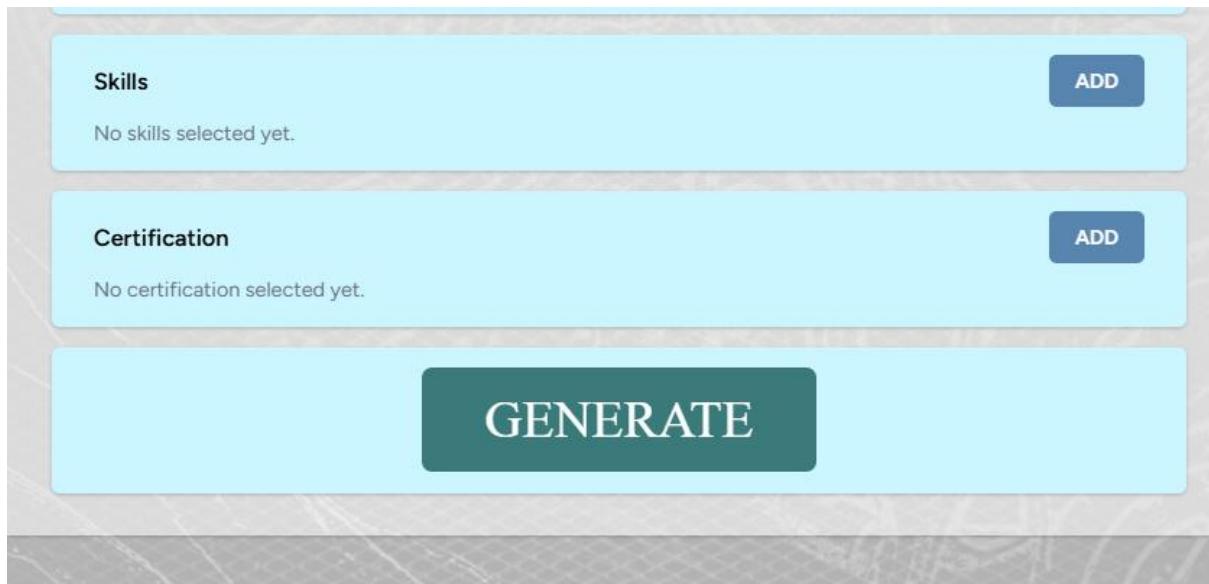


Figure 82 Descriptive labelled Buttons

Robust:

Responsive layout was used to support browser in different dimensions.

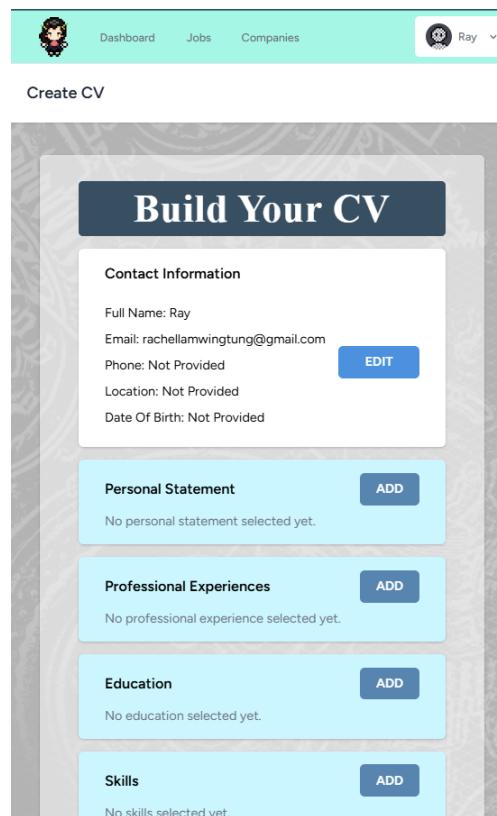


Figure 83 Responsive Layout

6.4 Improvements

There are several improvements that could be made to this project. For example, adding an admin role, integrating external APIs from larger companies for more advanced NLP, and including statistical analysis of stored data.

6.4.1 AI Functionalities Scaling Up With Ethical Data Collection Flow

However, the most significant potential improvement lies in scaling up the project. It could support ethical data collection from users' input. For instance, the system can explicitly ask for permission to collect CV data for training the model. Instead of using long texts, descriptive images and figures can be provided to help users understand how their data contributes to system improvement. The database can collect constructive CV components (from accepted applications) and unconstructive CV components (from rejected applications). As the database grows, the training dataset will increase, and the LSTM model will become more effective in identifying constructive sentences with varied structures.

6.4.2 Security

Since this project involves handling sensitive data, security methods should be explored more deeply if the project is expanded. Although input validation has been implemented, different hacking methods could severely harm the system. Simple Laravel authentication was used, but for further improvement, adding bot detection during login could be beneficial. This also represents another valid application for AI integration.

6.4.3 Deployment

There were issues during the deployment process which caused delays in some of the later project tasks. However, these issues were handled effectively within the buffer time that was planned in the project timeline. Deployment was not a mandatory requirement but rather an optional challenge added to enhance the project and provide a more realistic experience of full-stack development.

For improvement, future projects could benefit from preparing a deployment pipeline earlier in the development process. Automating the deployment using continuous integration from GitHub, to prevent last-minute technical issues.

7 Conclusion

To conclude this document, the dissertation has outlined the full-stack development process for the final year project, “Machine Learning-Based Job-Hunting Web Application for CV Optimization.” The primary aim of this project is to achieve a proof of concept in making a job-hunting application that supports job-seeking candidates by addressing the mismatch between skills and job requirements in the United Kingdom.

The project environment is based on the Laravel framework for web development, combined with classification models to analyse CV data. It seeks to apply advanced natural language processing and machine learning methods to improve CV-job matching. Thorough market, technical, and literature research has been conducted. Key features are drawn from existing job platforms and CV tools, while methodologies such as AI models and software tools have been selected to support the implementation phase.

The project follows the SCRUM methodology, with task management handled in Jira to support agile development and tracking. Milestones are broken into smaller tasks and visualised through a Gantt chart to ensure progress is clear and manageable. As the project evolved, adaptations were made based on user testing, technical challenges, and available time. Continuous research also influenced methodological changes during the development. By the end of the project, a completed, deployed production-level web application was created, and all milestones had been achieved.

The document also discusses the methodology used, implementation, and the deployment process. With different functionalities and the design of the application, the web application has achieved the expected outcome. Evaluation and further improvement were discussed, with several issues arising.

The first area is the integration of AI. Although, as discussed, AI has major drawbacks when implemented wrongly, such as bias issues and ethical problems regarding data collection. This project demonstrates several applications of AI that are ethical and beneficial, such as using AI for generating synthetic data that reduces production time and bypasses the ethical consent issue. Another example is the main feature of using AI to analyse CVs to allow a fairer environment for job applicants with the same quality. LSTM and NLP were used in the project, where the application promotes fairness.

The main AI method used was the LSTM model and NLP. Under the type of recurrent neural network, it fulfils the project title, “Machine Learning-Based Job-Hunting Web Application for CV Optimization.” Throughout the implementation, tasks were met positively, according to schedule, with only slight postponements but stayed within the deadline. The project is limited due to the scale of the development as a solo developer. If it has the chance to scale up, prediction accuracy will improve with a large user base and real-world data. To conclude, this project is a success with unlimited potential.

8 Bibliography

- [1] Office for National Statistics, “Vacancies and jobs in the UK: September 2024,” Office for National Statistics, 15 October 2024. [Online]. Available: <https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/bulletins/jobsandvacanciesintheuk/october2024>. [Accessed 15 October 2024].
- [2] Office for National Statistics, “Economic inactivity,” Office for National Statistics, 2021. [Online]. Available: <https://www.ons.gov.uk/employmentandlabourmarket/peoplenotinwork/economicinactivity#:~:text=People%20not%20in%20employment%20who,within%20the%20next%202%20weeks..> [Accessed 15 October 2024].
- [3] M. Salas-Velasco, “Mapping the (mis)match of university degrees in the graduate labor market,” *Journal for Labour Market Research volume*, vol. 55, p. 14, 07 May 2021.
- [4] Tijana Prokic-Breuer and Patricia A. McManus, “Immigrant Educational Mismatch in Western Europe, Apparent or Real?,” *European Sociological Review*, vol. 32, no. 3, p. 411–438, June 2016.
- [5] M. Fernández-Reino and B. Brindle, “Migrants in the UK Labour Market: An Overview,” 10 June 2024. [Online]. Available: <https://migrationobservatory.ox.ac.uk/resources/briefings/migrants-in-the-uk-labour-market-an-overview/>. [Accessed 10 October 2024].
- [6] UK Government, “ British National Overseas (BNO) visa,” [Online]. Available: <https://www.gov.uk/british-national-overseas-bno-visa>. [Accessed 10 October 2024].
- [7] L. Hamer, “Half of BN(O) migrants struggle to find jobs that match their skills, but 99% have no plans to return to Hong Kong, study finds,” South China Morning Post, 8 December 2023. [Online]. Available: https://www.scmp.com/news/hong-kong/society/article/3242378/half-bno-migrants-are-jobless-99-have-no-plans-returning-hong-kong-study-finds?module=perpetual_scroll_0&pgtype=article. [Accessed 15 October 2024].

- [8] Patrick van Escha, J. Stewart Black and Joseph Ferolie, “Marketing AI recruitment: The next phase in job application and selection,” *Computers in Human Behavior*, vol. 90, no. 01, p. 215–222, 2019.
- [9] K. Nikolenko, “Artificial Intelligence and Society: Pros and Cons of the Present,” *Future Prospects. Futurity Philosophy*, vol. 1, no. 2, pp. 54-67.
- [10] M. Tom Michael, Machine learning, New York: McGraw-Hill, 1997.
- [11] Roger S. Pressman and Brue R. Maxim, “The Nature of Software,” in *Software Engineering A Practitioner's Approach*, New York, McGraw-Hill Education, 2015, pp. 1-13.
- [12] LinkedIn Corporation, “LinkedIn Pressroom About Us Statistics,” LinkedIn Corporation , 2024. [Online]. Available: <https://news.linkedin.com/about-us#Statistics>. [Accessed 11 October 2024].
- [13] Indeed, “About Indeed,” Indeed, 2024. [Online]. Available: <https://uk.indeed.com/about?hl=en>. [Accessed 19 October 2024].
- [14] Glassdoor, “Glassdoor About Us / Press,” Glassdoor , 2024. [Online]. Available: <https://www.glassdoor.com/about/>. [Accessed 19 October 2024].
- [15] Teal Labs, Inc, “Teal About Us,” Teal Labs, Inc, 2024. [Online]. Available: <https://www.tealhq.com/about>. [Accessed 20 October 2024].
- [16] Enhancv, “Enhancv About Us Company,” Enhancv, 2024. [Online]. Available: https://enhancv.com/about/?_gl=1*lb0gxl*_up*MQ..*_ga*MzYzMjcwMTU5LjE3Mjk1NDU1OTc.*_ga_4KPKJHBDT0*MTcyOTU0NTU5Ni4xLjAuMTcyOTU0NTU5Ni4wLjAuMTg0ODIzNTMx. [Accessed 20 October 2024].
- [17] Joanna Davis, Hans-Georg Wolff, Monica L. Forret and Sherry E. Sullivan, “Networking via LinkedIn: An examination of usage and career benefits,” *Journal of Vocational Behavior*, vol. 118, p. 103396, 2020.
- [18] T. Shore, A. Tashchian and W. R. Forrester, “The influence of resume quality and ethnicity cues on employment decisions,” *JBEM*, vol. 22, no. 1, pp. 61-76, 2021.

- [19] A. Gupta, “Company, Job, and Perceived Advantage Influence on Job Hunting: Evaluating Glassdoor Job Hunting platform,” *Journal of Digitovation and Information System*, vol. 2, no. 1, p. 31 – 40, 2022.
- [20] Teal, “Teal Resume Checker,” Teal, [Online]. Available: <https://www.tealhq.com/tool/resume-checker>. [Accessed 23 October 2024].
- [21] Enhancv, “Enhancv Resume Checker,” Enhancv, [Online]. Available: <https://enhancv.com/resources/resume-checker/>. [Accessed 23 October 2024].
- [22] B. Surendiran, Tejas Paturu, Harsha Vardhan Chirumamilla and Maruprolu Naga Raju Reddy, “Resume Classification Using ML Techniques,” *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, pp. 1-5, 2023.
- [23] Asrar Hussain Alderham and Emad Sami Jaha, “Comparative Semantic Resume Analysis for Improving Candidate-Career Matching,” *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 313-321, 2022.
- [24] Abdul Khaliq, Supina Batubara, Maya Syaula and Sahrial, “Designing a web-based career system using the laravel framework,” *Jurnal Mantik*, vol. 7, no. 1, pp. 30-38, 2023.
- [25] GOV.UK, “Guidance Ethics, Transparency and Accountability Framework for Automated Decision-Making,” 29 November 2023. [Online]. Available: <https://www.gov.uk/government/publications/ethics-transparency-and-accountability-framework-for-automated-decision-making/ethics-transparency-and-accountability-framework-for-automated-decision-making>. [Accessed 2025].
- [26] I. Sommerville, “Agile software developmetn,” in *Software engineering*, 10th ed., Boston, Mass, Pearson, 2016, pp. 73-100.
- [27] Pressman, Roger S. and Maxim, Bruce R., “Agility And Process,” in *Software engineering: a practitioners approach*, 9th ed., New York, NY, McGraw-Hill Education, 2020, pp. 27-50.
- [28] Pressman, Roger S. and Maxim, Bruce R., “Quality and security,” in *Software engineering: a practitioners approach*, 9th ed., New York, NY, McGraw-Hill Education, 2020, pp. 339-355.

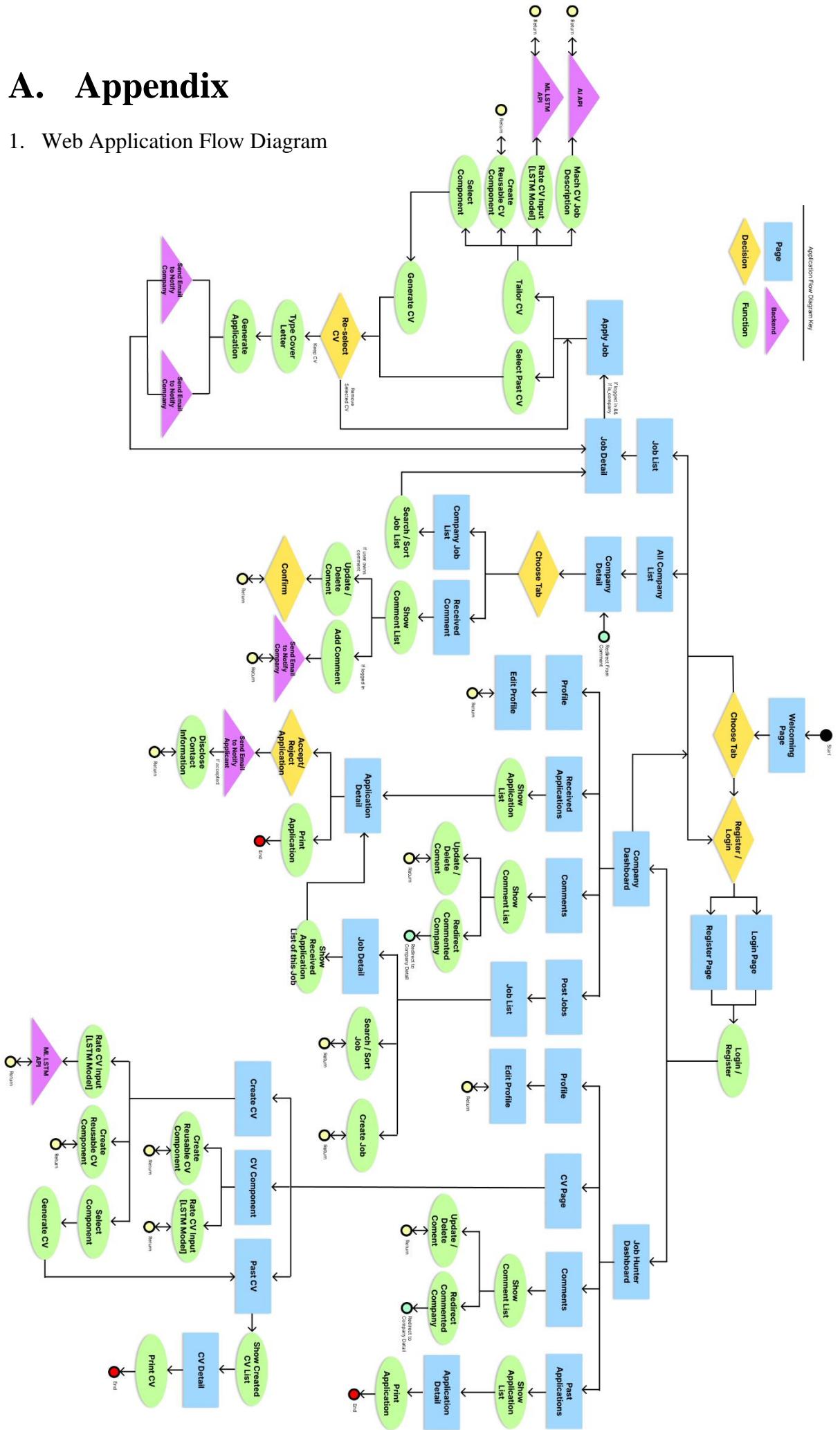
- [29] The PHP Group , “PHP Mnual,” [Online]. Available: <https://www.php.net/manual/en/introduction.php>. [Accessed 10 October 2024].
- [30] Laravel Holdings Inc., “Laravel Documentation V11.x,” [Online]. Available: <https://laravel.com/docs/11.x>. [Accessed 9 October 2024].
- [31] P. Redmond, “PHP Spellchecker Library,” Laravel, 2020. [Online]. Available: <https://laravel-news.com/php-spellchecker>. [Accessed 24 October 2024].
- [32] P. Redmond, “Google Natural Language API for Laravel,” Laravel, 26 May 2021. [Online]. Available: <https://laravel-news.com/google-natural-language-api-for-laravel>. [Accessed 24 October 2024].
- [33] Python Software Foundation, “The Python Standard Library,” 21 October 2024. [Online]. Available: <https://docs.python.org/3/library/index.html>. [Accessed 21 Oct 2024].
- [34] Z. Liu, Y. Lin and M. Sun, Representation Learning for Natural Language Processing, Singapore: Springer Nature Singapore : Imprint: Springer, 2020.
- [35] Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Computation*, vol. 31, no. 7, p. 1235–1270., 2019.
- [36] Atlassian, “Jira Cloud Support,” Atlassian, [Online]. Available: <https://support.atlassian.com/jira-software-cloud/docs/create-and-configure-your-issues/>. [Accessed 17 October 2024].
- [37] Figma, “Figma Best Practice Guides,” Figma, [Online]. Available: <https://www.figma.com/best-practices/guides/>. [Accessed 20 October 2024].
- [38] Shneiderman, Ben and Nicholls, Paul, Designing the user interface: strategies for effective human-computer interaction., 3rd ed., London: Pelican Island Information, 1997.
- [39] Roger S. Pressman and Bruce R. Maxim, “Software Testing Strategies,” in *Software Engineering A Practitioner’s Approach*, 8th ed., New York, McGraw-Hill Education, 2015, pp. 466-495.

- [40] L. H. L. L. a. J. L. W. Cui, “The Research of PHP Development Framework Based on MVC Pattern,” *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, pp. 947-949,, 2009.
- [41] Laravel Document, “Blade with Livewire,” [Online]. Available: <https://laravel.com/docs/12.x/blade#supercharging-blade-with-livewire>. [Accessed April 2025].
- [42] Anthony T. Holdener, Ajax: the definitive guide, CA 95472: O'Reilly Media. Inc., 2008.
- [43] spaCy, “Linguistic Features Guides,” [Online]. Available: <https://spacy.io/usage/linguistic-features>. [Accessed 2025].
- [44] R. C. S. a. E. R. Morris, “Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks,” *ArXiv*, vol. abs/1909.09586, 2019.
- [45] PyTorch, “Sequence Models and Long Short-Term Memory Networks,” 2017. [Online]. Available: https://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html. [Accessed 2025].
- [46] bentrevett, “PyTorch Seq2Seq,” [Online]. Available: <https://github.com/bentrevett/pytorch-seq2seq/blob/main/1%20-%20Sequence%20to%20Sequence%20Learning%20with%20Neural%20Networks.ipynb>. [Accessed 2025].
- [47] KDnuggets, “How to Create a Vocabulary for NLP Tasks in Python,” 2019. [Online]. Available: <https://www.kdnuggets.com/2019/11/create-vocabulary-nlp-tasks-python.html>. [Accessed 2025].
- [48] H. Z. Tong Yu, “Hyper-Parameter Optimization: A Review of Algorithms and Applications,” 2020.
- [49] Tensorflow, “Long short-term memory (LSTM) RNN in Tensorflow,” [Online]. Available: <https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow/>. [Accessed 2025].

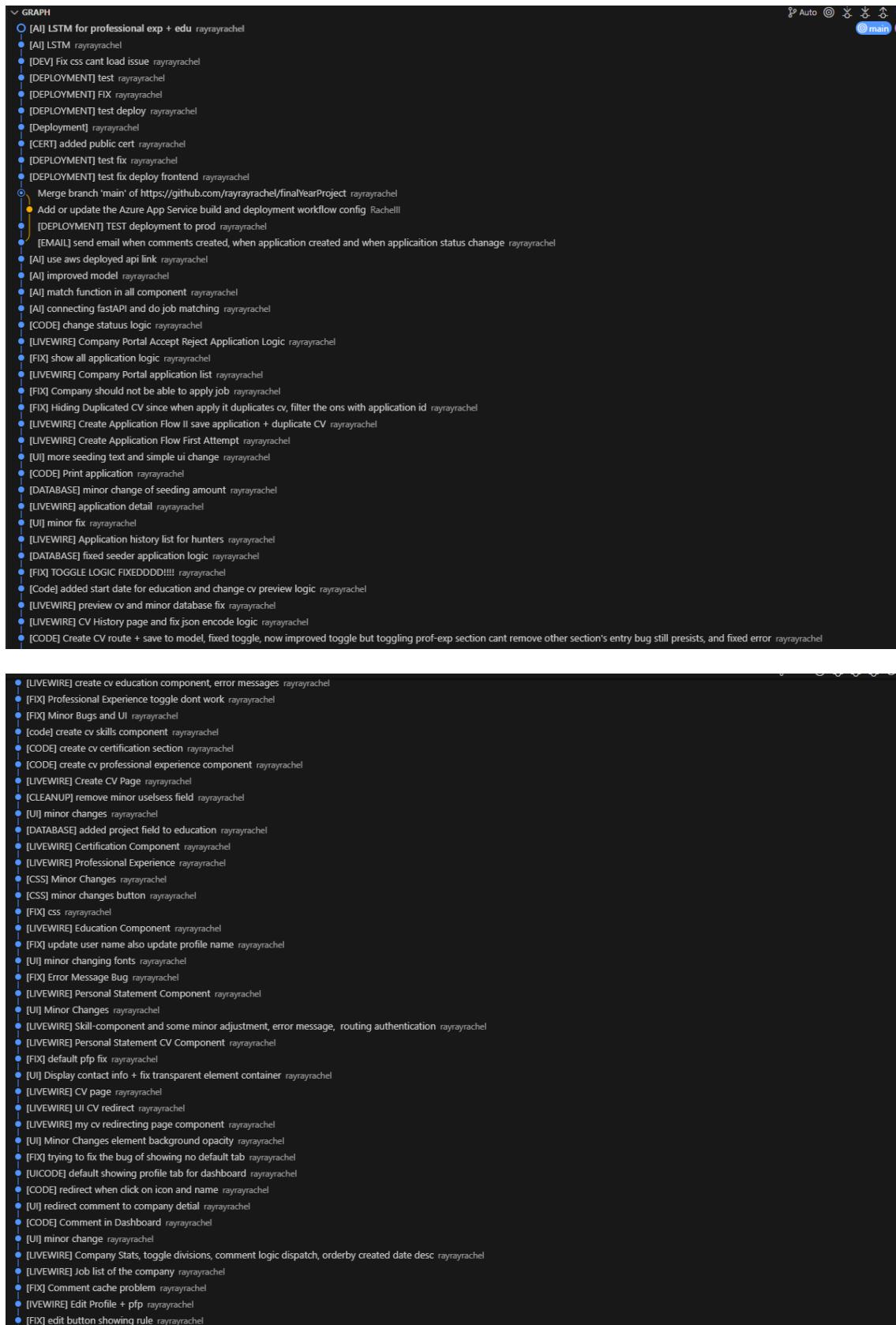
- [50] DevPointed, “Laravel on Azure Web App,” 2024. [Online]. Available: https://www.youtube.com/watch?v=PGhVFsOnGA&t=798s&ab_channel=DevPointed. [Accessed 2025].
- [51] World Wide Web Consortium (W3C), “Web Content Accessibility Guidelines (WCAG) 2.0,” 11 December 2008. [Online]. Available: <https://www.w3.org/TR/WCAG20/>. [Accessed 2025].
- [52] Johan A. K. Suykens, Marco Signoretto and Andreas Argyriou, “Multi-Layer Support Vector Machines,” in *Regularization, Optimization, Kernels, and Support Vector Machines*, 1st ed., Boca Raton, Taylor & Francis, 2015, pp. 457-472.
- [53] B. Scholkopf and A. J. Smola, “Concept And Tools,” in *Learning with kernels : support vector machines, regularization, optimization, and beyond*, 1st ed., Cambridge, Mass. : MIT Press, 2002, pp. 23-184.
- [54] Office for National Statistics, “Office for National Statistics, About Us,” Office for National Statistics, [Online]. Available: <https://www.ons.gov.uk/aboutus>. [Accessed October 2024].
- [55] The Migration Observatory, “The Migration Observatory About,” The Migration Observatory, [Online]. Available: <https://migrationobservatory.ox.ac.uk/about/>. [Accessed October 2024].
- [56] Medium, “Explained: Hyperparameters in Deep Learning,” [Online]. Available: <https://medium.com/the-research-nest/explained-hyperparameters-in-deep-learning-9b1e0f3b9029>. [Accessed 2025].

A. Appendix

1. Web Application Flow Diagram



2. Full Git Commit History



- [LIVEWIRE] update comment, delete comment, restrict unauthorised user to check user detail and allow check company detail rayrayrachel
- [LIVEWIRE] Display company comment rayrayrachel
- [UI] detail page, page min height, spa link rayrayrachel
- [FIX] Fixing CV factory personal statement rayrayrachel
- [LIVEWIRE] Company Profile Detail rayrayrachel
- [LIVEWIRE] Job Detail rayrayrachel
- [FIX] Trying to fix default tab but fail rayrayrachel
- [UI] Dashboard Tabs rayrayrachel
- [LIVEWIRE] Company Profile rayrayrachel
- [UI] Create Job Listing in Company Portal rayrayrachel
- [LIVEWIRE] [VOLT] create job rayrayrachel
- [CODE] Comany dashboard job list rayrayrachel
- [UI] Job list and sorting rayrayrachel
- [CODE] create profile when register rayrayrachel
- [UI] Minor Changes rayrayrachel
- [UI] Navigation rayrayrachel
- [LOGO] App logo rayrayrachel
- [UI] [NAVIGATION] [LIVEWIRE] added landing page rayrayrachel
- [UI] Added default profile picture, added footer, some other minor changes rayrayrachel
- [UI] Change Company List from table to card rayrayrachel
- [LIVEWIRE] [NAVIGATION] Company List with search feature, Navigation to display guest layout and user login rayrayrachel
- [DATABASE] [UI] Job List with search feature and sorting feature with ui and navigation rayrayrachel
- [UI] Stylesheet Initialise rayrayrachel
- [DATABASE] CV Model Migration Factory Seeder rayrayrachel
- [DATABASE] Certifications Model Migration Factory Seeder rayrayrachel
- [DATABASE] Skill Model Migration Factory Seeder rayrayrachel
- [DATABASE] Education Model Migration Factory Seeder rayrayrachel
- [DATABASE] Professional Experience Model Migration Factory Seeder rayrayrachel
- [DATABASE] PersonalStatement Model Migration Factory Seeder rayrayrachel
- [DATABASE] Contact Information Model Migration Factory Seeder rayrayrachel
- [DATABASE] Comment Model Migration Factory Seeder rayrayrachel
- [DATABASE] Application Model Migration Factory Seeder rayrayrachel
- [DATABASE] Job Posts Model Migration Seeder Factory rayrayrachel
- [DATABASE] Profile Factory Profile Seeder and User Seeder rayrayrachel
- [DATABASE] Profile Model and Migration file rayrayrachel
- [ENV] [AUTH] Setup laravel authentication, npm, breeze rayrayrachel
- [TEST] Commit rayrayrachel
- first commit rayrayrachel

3. Database Structure

