



National University of Computer and Emerging Sciences, Lahore



AI-Powered Job Advisor

Muhammad Laraib Akhtar 21L-5294 BS(CS)

Wajeeh Ahmed Khan Niazi 21L-7565 BS(CS)

Sayyeda Fatima Masood 21L-7571 BS(CS)

Supervisor: Dr. Rana Asif Rehman

Final Year Project

January 5, 2025

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

Title: AI-Powered Job Advisor

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: 08/12/2024

Name: Muhammad Laraib Akhtar

Signature:

Name: Wajeeh Ahmed Khan Niazi

Signature:

Name: Sayyeda Fatima Masood

Signature:

Author's Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

Job Hunting is an important part of everyone's career, and it is crucial to get our dream job so we can grow professionally. Job seekers hence apply for any job post they consider as a viable option and often misinterpret the actual skill gap and what they perceive. This results in far too many rejections which not only lowers morale, but also makes you question your own capabilities. Thus, an accurate analysis of job posting, and proper advisory must be available to make educated and well-informed decisions. Similarly, a generic resume which is filled with several skills is not always a good fit for all job postings and often does not qualify you as a good fit for the job. In this project, we aim to automate job recommendations, customized CVs and provide skill gap analysis using CV ranking, compatibility index and providing relevant internships and job opportunities relevant to specific skills.

Executive Summary

Job hunting plays a vital role in advancing one's career, making it essential to secure our ideal job for professional growth. The project aims to revolutionize the job hunting process by developing an AI-powered platform that provides personalized job recommendations, tailors resumes, and performs skills gap analysis to enhance candidates' employability. This web-based application, particularly for job seekers in the computer science field, will offer these features using technologies like HTML, CSS, JavaScript, and nextjs. Constraints such as data quality, scalability, third-party integration, and AI limitations must be addressed to ensure success.

Functional Requirements of our project include features such as secure user registration, profile management, job search and filtering, CV generation, skill gap analysis, job scraping, and job recommendations based on user profiles. On the other hand, non functional requirements are concerned with the system's overall performance and qualities, and make sure the system is operating efficiently, securely and reliably.

To provide diversity and relevance of the dataset, the AI-Powered Job Advisor dataset is acquired from different sources. The CV dataset in our study is collected from the platforms like LinkedIn and Career Service Office (CSO) and covers a wide range of experiences for Computer Science students. To maintain up to date and diverse job postings, job descriptions are scraped from several platforms, and course and certification data is obtained from known platforms such as Coursera, EDX, and Credly. To improve both job recommendations and CV quality, the proposed methodology uses techniques like cosine similarity, LLM based systems, and GANs. To match resumes to job descriptions, cosine similarity is used to compare the embeddings of resumes and then LLMs are used to generate resumes from the user data and interaction history. Furthermore, GANs enhance low quality resumes by learning from high quality resumes and thereby enhancing user profiles for better job matching. The combination of this multi layered approach, using LaTeX for CV generation and a novel CV ranking system, offers a complete solution for job recommendations and skill gap analysis.

Key design considerations for the system include ensuring scalability, seamless integration with third-party platforms, and user-friendly interfaces. The system must handle large datasets efficiently while maintaining data integrity and offering secure user access. The system architecture is composed of high-level partitions that include a user interface, data processing, and AI-driven recommendation engines. The architecture supports modularity, where each component (e.g., job scraping, resume generation, and skill gap analysis) operates independently but collaborates to provide an integrated solution.

Table of Contents

List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Purpose of this Document	1
1.2 Intended Audience	1
1.3 Definitions, Acronyms, and Abbreviations	1
1.4 Conclusion	2
2 Project Vision	3
2.1 Problem Domain Overview	3
2.2 Problem Statement	4
2.3 Problem Elaboration	4
2.4 Goals and Objectives	4
2.5 Project Scope	5
2.6 Sustainable Development Goal (SDG)	5
2.7 Constraints	6
2.8 Business Opportunity	7
2.9 Stakeholders Description/ User Characteristics	7
2.9.1 Stakeholders Summary	7
2.9.2 Key High-Level Goals and Problems of Stakeholders	7
2.10 Conclusion	9
3 Literature Review / Related Work	10
3.1 Definitions, Acronyms, and Abbreviations	10
3.1.1 Definitions	10
3.1.2 Acronyms and Abbreviations	10
3.2 Detailed Literature Review	11

3.2.1	Enhancing job recommendation through llm-based generative adversarial networks	11
3.2.2	Resumeflow: An llm-facilitated pipeline for personalized resume generation and refinement	12
3.2.3	Resume Screening and Course Recommendation System	13
3.2.4	A survey on large language models for recommendation	14
3.2.5	Design and development of machine learning based resume ranking system	15
3.2.6	Resume ranking based on job description using SpaCy NER model	16
3.2.7	LlamaRec: Two-stage recommendation using large language models for ranking	16
3.2.8	An intelligent decision support system for recruitment: resumes screening and applicants ranking	17
3.2.9	Novel online recommendation algorithm for massive open online courses (NoR-MOOCs)	18
3.2.10	Course Recommendation Model Based on Layer Dropout Graph Differential Contrastive Learning	19
3.2.11	Resumonk	20
3.2.12	LinkedIn	21
3.2.13	ZipRecruiter	22
3.3	Literature Review Summary Table	22
3.4	Conclusion	24
4	Software Requirement Specifications	25
4.1	List of Features	25
4.2	Functional Requirements	25
4.3	Quality Attributes	26
4.4	Non-Functional Requirements	27
4.4.1	Performance	27
4.4.2	Security	27
4.4.3	Compatibility	27
4.4.4	Scalability	27
4.4.5	Usability	27
4.5	Assumptions	28
4.6	Use Cases	28
4.6.1	Register User	28
4.6.2	Login User	29
4.6.3	View Profile	29

4.6.4	Edit Profile	30
4.6.5	Delete Profile	30
4.6.6	Recommend Jobs	31
4.6.7	Filter Jobs	31
4.6.8	Generate CV	32
4.6.9	View Compatibility Scores	32
4.6.10	Generate Cover Letter	33
4.6.11	Assess Job Eligibility	33
4.6.12	Analyze Skill Gap	34
4.6.13	Give CV Ranking	34
4.6.14	Recommend Courses and Internships	35
4.6.15	Use Browser Extension	35
4.7	Hardware and Software Requirements	36
4.7.1	Hardware Requirements	36
4.7.2	Software Requirements	36
4.8	Graphical User Interface	36
4.8.1	GUI Screens and Functionality	36
4.8.2	Navigation Flow	39
4.8.3	GUI Guidelines	40
4.9	Database Design	40
4.9.1	ER Diagram	40
4.9.2	Data Dictionary	40
4.10	Risk Analysis	40
4.10.1	Technical Risks	42
4.10.2	Business Risks	43
4.11	Conclusion	43
5	Proposed Approach and Methodology	44
5.1	Dataset	44
5.1.1	CV Dataset	44
5.1.2	Job Description Dataset	44
5.1.3	Courses and Certifications	44
5.1.4	Dataset parsing	45
5.2	Proposed Methodology	45
5.2.1	Job Recommendation and Compatibility Score	45

5.2.2	CV Generation	47
5.2.3	CV Ranking	47
5.2.4	Skill Recommendation	47
5.3	Conclusion	48
6	High-Level and Low-Level Design	49
6.1	System Overview	49
6.1.1	Interactive User Interface	49
6.1.2	Job Recommendations	49
6.1.3	Skill Gap Analysis	49
6.1.4	Personalized CV and Cover Letter Generation	49
6.1.5	Job Scraping and External Integration	50
6.2	Design Considerations	50
6.2.1	Assumptions and Dependencies	50
6.2.2	General Constraints	51
6.2.3	Goals and Guidelines	51
6.2.4	Development Methods	52
6.3	System Architecture	52
6.3.1	High-Level Partitioning	52
6.3.2	Component Collaboration	52
6.3.3	Rationale for Decomposition	53
6.3.4	System Architecture Diagrams	53
6.3.5	Subsystem Architecture	56
6.3.6	Justification for Architectural Choices	59
6.4	Architectural Strategies	59
6.4.1	Use of Modern Technologies	59
6.4.2	Future Plans for Extension or Enhancement	59
6.4.3	User Interface Paradigms	60
6.4.4	Concurrency and Synchronization	60
6.4.5	Version Control	60
6.5	Domain Model/Class Diagram	60
6.6	Policies and Tactics	61
6.6.1	Coding Guidelines and Conventions	62
6.6.2	System Testing	62
6.6.3	Continuous Integration and Deployment	62

6.7 Conclusion	62
7 Implementation and Test Cases	63
7.1 Implementation	63
7.1.1 System Setup and Development Environment	63
7.1.2 Dataset Handling and Feature Extraction	63
7.1.3 CV and Cover Letter Generation	65
7.1.4 CV Ranking	65
7.2 Conclusion	65
8 Experimental Results and Discussion	66
8.1 CV Ranking	66
8.2 CV and Cover Letter Generation	66
8.3 Conclusion	67
9 Conclusion and Future Work	70

List of Figures

2.1	The figure illustrates SDG 8, “Decent Work and Economic Growth,” which is illustrated with an image of a briefcase alongside an upward pointing graph.	6
4.1	Login page	37
4.2	Registration page	37
4.3	Dashboard page	38
4.4	CV Generation Page	38
4.5	Profile page	39
4.6	Job Recommendation Feed	40
4.7	This figure represents all the entities and their relationships	41
6.1	Architecture Diagram	54
6.2	This figure represents all the entities and their relationships	61
8.1	This figure represents similarity scores by methods	67
8.2	This figure shows a template latex CV	68
8.3	This figure shows a template latex Cover Letter	69

List of Tables

3.1 Literature Review Summary Table	23
3.2 Table 3.1 continues	24
3.3 Application Review Summary Table	24
4.1 Data Dictionary	42

Chapter 1 Introduction

Due to the recent evolution of the job market, job seekers often find themselves in a tough spot when tailoring their resume and skills to specific job postings, which results in numerous rejections. This can be mentally challenging for job seekers. To resolve these issues we aim to create this AI-Powered Job Advisor which will help the users by automating job recommendations, generating personalized CVs, and conduct skill gap analysis by using ML/AI so as to maximize the chances of landing the desired job.

1.1 Purpose of this Document

This document aims to provide a clear and detailed representation of the design, implementation and general use cases of our FYP: AI-Powered Job Advisor .We aim to deliver a project which is capable of designing personalized CVs and Cover letter for specific jobs using NLP techniques, provide a recommendation system which will analyze the skills of the user to recommend relevant job postings, moreover perform skill gap analysis using CV ranking and compatibility score which well help us recommend courses and projects to improve the users skill set, and lastly to make an API so that we can Provide our services to applications outside our own to target a bigger audience. This document will elaborate the methodology, design implementation, testing, and evaluation of our project, aswell as the limitations and future work.

1.2 Intended Audience

The document is targeted towards multiple audiences which includes the FYP supervisors as this will help them oversee the progress of our project. Furthermore this document will prove useful for the Evaluation Panel as they can easily check the features and working of our project for grading purposes. Even our group will benefit from this document as it will provide a raodmap for our development process, and lastly, various investors can also read this document before they decide to invest in our project.

1.3 Definitions, Acronyms, and Abbreviations

ML: Machine Learning

FYP: Final Year Project

MVP: Minimum Viable Product

UI: User Interface

UX: User Experience

AI: Artificial Intelligence

CV: Curriculum Vitae

REST: Representational State Transfer

ORM: Object-Relational Mapping

CRUD: Create, Read, Update, Delete

API: Application Programming Interface

NLP: Natural Language Processing

LLM: Large Language Model

GAN: Generative Adversarial Network

1.4 Conclusion

In this report, we go through our Job-Advisor in great detail. Chapter 1 introduces the AI-Powered Job Advisor project, outlining its purpose and the challenges job seekers face in tailoring resumes and understanding skill requirements. It emphasizes the need for an intelligent system to enhance the job search process and improve applicant outcomes. Chapter 2 elaborates on the project vision, detailing the specific issues job seekers encounter, such as skill gaps and ineffective resume submissions. It also defines the project goals and objectives, focusing on personalized job recommendations and skill enhancement. Chapter 3 presents a literature review of existing research related to job recommendations and resume generation. It highlights the limitations of current systems and positions the proposed project as an innovative solution utilizing advanced AI techniques. Chapter 4 outlines the software requirement specifications for the AI-Powered Job Advisor, detailing key features, functional requirements, and quality attributes necessary for optimal performance and user satisfaction. It also discusses assumptions and provides use cases for system functionality. Chapter 5 describes the proposed approach and methodology, including the datasets utilized for job descriptions, resumes, and courses. It outlines the techniques for job recommendation, CV generation, and skill evaluation to ensure a comprehensive system. Chapter 6 focuses on the high-level and low-level design of the system, detailing its architecture and components. It ensures that the design is coherent and meets user needs, providing a solid foundation for developing an efficient and user-friendly platform.

Chapter 2 Project Vision

The changing landscape of the employment sector today has made job searching an integral part of every person's career path. While hunting for jobs, individuals have to deal with problems of estimating the skill requirements and articulating their skills in the right manner. This project intends to resolve these problems by designing an intelligent platform that recommends jobs and advises applicants on how to apply for jobs easily. In this regard, we plan to improve the job hunting experience, minimize job refusal chances, and facilitate proper career choices by automating the job recommendation process, tailoring resumes, and performing skills gap analysis.

During the course of research, we came across a few job recommendation system that helped in suggesting jobs based on particular skills and work experience of the user. But these systems tend to miss out on such things from an applicant's perspective – which is – assistance in writing resumes and honing the skills. Our project aims to address this problem by employing modern AI models, which will build easy-to-use and effective means of assistance for job-seekers.

To sum up, we consider this project as a breakthrough in this area of research and development of job advisory systems by improving the efficiency and effectiveness of job seekers. We want to strive to the creation of an equitable system that helps people looking for jobs and those who offer jobs in the same space by facilitating personalized assistance and improving individual skills, therefore enhancing the searching process for jobs.

2.1 Problem Domain Overview

Our system is designed to streamline the job hiring procedure for potential candidates, focusing on job seekers, particularly those in the computer science field. The main functionality of using artificial intelligence is to generate not only the jobs that are suited to every user's skills and experiences but also customized CVs and cover letters, and providing a thorough skill gap analysis. The system will employ advanced AI algorithms to assess job postings, rank CVs based on compatibility, and recommend relevant internships and job opportunities. Our platform will automatically extract data by using automated data scrapers which will, in turn, continuously search for job posts that have been updated and making sure that users have access to the latest opportunities.

Through this system, users will create profiles with information about their skills, and work experiences, the AI will then use this information to match the users with suitable job positions. Besides, the platform will be more than just an application tool and will also aid users in finding areas in which they can improve their applications. Then, the users will have the capability to enhance their skills and employability. Clearly, this approach is meant to be all-inclusive, meaning all job seekers will have the tools

they need in order to be successful in their professional lives.

2.2 Problem Statement

The core problem this project addresses is the significant difference between job seekers' qualifications and the specific requirements of job postings, which in turn lead to a very high rate of application rejections. A lot of job hunters send in generic resumes and they hardly have any idea on how to match their skills against the jobs they are applying for. This in turn results not only in decreased morale of the personnel but also in the wasted effort in finding a job.

2.3 Problem Elaboration

The primary problem has several sub-problems that we aim to address:

1. Skill Gap Analysis: A lot of applicants do not know what kind of skills are needed for a certain position they are applying for. Our system will identify those gaps and offer recommendation on how to fill them.
2. Personalized CV Generation: Writing personalized CVs for every application is very difficult. Our system will take care of this task and the users will be able to produce CVs that present appropriate skills for each position in no time.
3. Job Recommendations: There are job recommendations systems available today that do not take personalization into account. We will create a better recommendation system which will include not only the qualifications of the user, but also their career goals.
4. User Experience: There should be no doubts about the ease of use of the application to be developed. The application will be designed to accommodate users with various levels of technological knowledge and diverse mindsets.

2.4 Goals and Objectives

The major objectives of the project include:

- Job recommendations based on relevant skills and experience level.
- Generate personalized CVs and cover letters to target specific job openings.
- Provide skill gap analysis by using CV ranking and compatibility scores.
- Recommend relevant internships, projects, and certifications to enhance specific skills.

2.5 Project Scope

Specify the scope of this project is to develop a web-based application that helps job seekers manage their job applications more effectively. The application will allow users to:

- Create, view, modify and remove their profiles.
- Collect job postings from various online sources.
- Generate personalized CVs and cover letters using AI.
- Analyze their skills and provide recommendations for improvement.

The project will be developed using a combination of HTML, CSS, JavaScript, and next.js. The application will use a RESTful API to communicate with a back-end database, which will be implemented using a ORM library. The project will be developed using Agile development principles, and project management will be done using SCRUM.

The project deliverables will include:

- A functional web-based application for job seekers.
- A user manual for the application
- A documentation of the design, implementation, and testing process.
- API for integrating the application with third-party services.

The project scope does not include the following:

- Providing hosting or server infrastructure for the application
- Developing mobile applications (iOS or Android)
- A platform for employers searching for job seekers.

2.6 Sustainable Development Goal (SDG)

This project focuses on Sustainable Development Goal 8: Decent Work and Economic Growth. Job-seekers are provided with custom job recommendations, skill gap analyses, and personalized CVs to cater to their needs so that their employability is improved and inclusive economic growth is maintained. Furthermore, the platform encourages creativity and economic growth by using Artificial Intelligence to ease the process of applying for jobs.

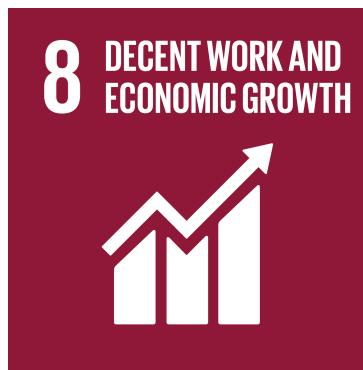


Figure 2.1: The figure illustrates SDG 8, “Decent Work and Economic Growth,” which is illustrated with an image of a briefcase alongside an upward pointing graph.

2.7 Constraints

While developing the intelligent job recommendation and advisory platform, several constraints must be considered to ensure the project’s success and feasibility:

- **Data Availability and Quality:** The effectiveness of AI-driven recommendations relies a lot on the availability of high-quality and up-to-date job postings and user data. Half-baked information or outdated data can lead to inaccurate recommendations and will affect user trust.
- **Scalability:** The system should also be able to handle gradually increasing amounts of data that will be processed and stored as the number of users grows over time. This is especially to make sure that the system can perform optimally despite the increase in the number of users.
- **Integration with Third-Party Services:** The application will also have capabilities to connect with some of the popular job boards, as well as educational resources. The integration of such services can be quite complex especially where different APIs and data structures are involved making it difficult to maintain seamless integration.
- **User Diversity:** Providing services to users of different technological skills makes it necessary for the design to be simple and straightforward. It is important to provide advanced features without over complicating the interface to meet the expectations of different types of customers.
- **AI Model Limitations:** It is possible to improve the functionality of the platform with the help of AI models, but they cannot be trusted completely. The platform’s continuing effectiveness is dependent on the models being kept up-to-date since older models will likely contain some inaccuracies and irrelevancies.
- **Resource Constraints:** Building an AI powered application comes with demanding technical resources and proficiency. Managing these resources efficiently within budget and time is also a

challenge.

2.8 Business Opportunity

The proposed intelligent job recommendation and advisory system offers a lot of business opportunities. With the current economic situation, more people are looking for work, there is a rise in demand for devices, wears and appliances that help lessen their chances of remaining unemployed. The platform addresses key issue points like skill gap analysis and personalized resume creation, which can be of great help. We can also offer subscription-based advanced features such as advanced coaching and priority job listings. Partnerships with universities and online course providers can create mutual growth opportunities by giving users customized learning routes. Revenue can also be generated through targeted advertising and sponsorships from companies that are promoting job openings or services. The platform has the potential to expand beyond the computer science field, increasing its user base.

2.9 Stakeholders Description/ User Characteristics

Identifying and understanding the stakeholders is very important for the successful development and deployment of the project. The primary stakeholders include job seekers, students, educational institutions and employers.

2.9.1 Stakeholders Summary

- **Job Seekers:** These are persons on the pursuit of available jobs, especially those related to computer science. They seek tools to enhance their job search efficiency, improve their resumes, and to gain knowledge of necessary skills required for their desired jobs.
- **Students:** The recommendation system can help students understand which courses they need to do and what skills they require to achieve their dream job in the future.
- **Educational Institutions and Training Providers:** Universities, colleges, coding academies, providers of different courses including online courses with certification. Our platform can be partnered with them to recommend relevant courses to job seekers.
- **Employers:** Seeking to recruit skilled people. This will allow them to get a good number of candidates suitable to their vacancies.

2.9.2 Key High-Level Goals and Problems of Stakeholders

The key high-level goals and problems of stakeholders are given below:

2.9.2.1 Job Seekers

- *Goals:*
 - Solve the problem of getting relevant employment opportunities efficiently.
 - To help enhance their resumes and cover letters to standout to employers.
 - Fill in the skill gap to enable better employability.
- *Current Problems to Address:*
 - Inability to tailor applications to meeting specified job requirements.
 - Lack of awareness of the skills required for a role that you desire.
 - Long and time consuming job searching and application processes.

2.9.2.2 Students

- *Goals:*
 - Finding the courses and skills needed to get to the career path they want.
 - So they can get personalized job recommendations based on their current skills and their academic background, in order to know where they are currently standing.
- *Current Problems to Address:*
 - They did not know which skills were most important for their target jobs.

2.9.2.3 Educational Institutions and Training Providers

- *Goals:*
 - They align their course offerings with current job market demands.
 - Make their programs seem relevant to increase enrollment.
- *Current Problems to Address:*
 - Understanding the changing skill requirements of employers.
 - The value and outcomes of their educational programs.

2.9.2.4 Employers

- *Goals:*

- Quickly attract and identify qualified candidates.
 - Streamline the hiring process to reduce time to fill positions.
 - Access diverse and skilled talent.
- *Current Problems to Address:*
 - Sifting through large volumes of generic applications.
 - Identifying candidates with the precise skills and experience needed.
 - Managing and maintaining an efficient recruitment pipeline.

2.10 Conclusion

This chapter has set forth an outstanding vision for the AI powered job advisor project by following a clear problem domain, what is the challenge and what are proposed solutions. AI powered features of the system — personalized job recommendations, custom CV generation and skill gap analysis — bridge the gap between the job seekers' qualifications and job requirements. Besides, it also makes sure that users get the access to the latest opportunities via automated data scrapers.

Goals of this project are to improve employability with actionable insights, provide individualized application materials, and suggest appropriate certifications and internships. HTML is used to develop a web based application to do all of the above that helps users maintain profiles, expertise analysis, and receives tailored job suggestions in Agile development principles within a defined scope.

For an effective design of an innovative career advisory tool, this chapter emphasizes refinement of the job application process by laying the groundwork for the project's full implementation.

Chapter 3 Literature Review / Related Work

To produce innovation in the field of technology, one must build on the shoulders of latest research work and projects others have done before. In this section, a deep analysis will be made to address similar works to our Job Advisor. We want to analyze what others have done, where the market lacks and where can our services fit in. We will try to understand the scope of each project to see with what solution they came up with, why they did it that way and how can we improve on that. This chapter will provide with detailed literature reviews of related work in the subject of job recommendations and resume creation.

3.1 Definitions, Acronyms, and Abbreviations

3.1.1 Definitions

- **Large Language Model:** A subset of artificial intelligence that has been developed to learn and also create human language from large data.
- **Generative Adversarial Network:** A machine learning framework with two neural networks competing to improve generation of data like images or text.
- **Cosine Similarity:** A measure that computes the cosine of measures of the angle between two vectors in several dimensions in order to evaluate similarity in between them in multi-dimensional space.
- **Named Entity Recognition:** A machine learning approach employed for the purpose of extracting features in text like a name and date.
- **Applicant Tracking System:** A software application that automates the hiring process by managing job applications and tracking candidates.

3.1.2 Acronyms and Abbreviations

- **LLM:** Large Language Models
- **GAN:** Generative Adversarial Network
- **CV:** Curriculum Vitae
- **ATS:** Applicant Tracking System
- **TF-IDF:** Term Frequency-Inverse Document Frequency
- **KNN:** K-Nearest Neighbors

- **NER:** Named Entity Recognition
- **MOOC:** Massive Open Online Course

3.2 Detailed Literature Review

This section gives a review of the existing literature in the relevant field. The research is summarized and clarified objectively in this section. Our main goal is to achieve clarity on what existing concepts, ideas and knowledge exists on our subject, along with their strength and weaknesses. The following describes detailed literature surveys of relevant research.

3.2.1 Enhancing job recommendation through llm-based generative adversarial networks

The paper [1] introduces a new job recommendation system leveraging LLMs and GANs. The system, called LLM-based GANs Interactive Recommendation (LGIR), addresses a very important problem of low-quality resumes by refining them through GANs. This is done by integrating both explicit and implicit user characteristics, such as skills and behavioral interactions with job platforms. LGIR enhances the accuracy of job recommendations and improves resume quality for both users with extensive interaction histories (many-shot users) and those with limited interactions (few-shot users). The system achieves better results over current job recommendation systems, proving their results through experiments across multiple datasets.

3.2.1.1 Critical analysis

The main research discussed in this paper is highly effective addressing two critical limitations in job recommendation systems, fabricated resume generation by the few-shot problem in LLMs. The combined use of LLMs and GANs for resume generation and refining low-quality resumes is a compelling approach to improving the overall quality of job recommendations. The model's design involving the mining of implicit user characteristics from interactions with the platform shows strong potential to create more personalized and meaningful job recommendations.

The empirical validation across three real-world recruitment datasets, where the LGIR method consistently outperforms baseline models, including content-based, behavior-based, and hybrid recommendation systems. However, one limitation lies in the assumption that users' interaction is sufficiently rich enough to infer implicit characteristics, which may not always be the case, as it might not span well with minimal or erratic interactions. Moreover, the GAN-based refinement strategy may add computational complexity, raising concerns about scalability for larger datasets.

3.2.1.2 Relationship to the proposed research work

The job recommendation system closely aligns with the objectives and methods we are targeting. Both approaches focus on automating job recommendations and improving resume quality using AI. Our system aims to provide personalized CV generation, skill gap analysis, and job recommendations based on user profiles, which is like the LGIR's approach of enhancing resumes and recommendations through LLMs and GANs.

Our project's compatibility scoring using Natural Language Processing (NLP) techniques like cosine similarity resonates well with the paper's focus on improving resume-to-job matching through textual embedding. Similar methods are used in both the approaches to handle the shared challenge of low quality resumes and skill gap analysis. For instance, transforming incomplete resumes into high quality ones with the help of AI models

This research can provide insights which can be incorporated into our project to potentially benefit from more advanced techniques like GANs for resume refinement and recommendation accuracy improvements based on user behavior to make our project more adaptive and user friendly.

3.2.2 Resumeflow: An llm-facilitated pipeline for personalized resume generation and refinement

The article [2] introduces ResumeFlow, a tool designed to automate the creation of job specific resumes using LLMs like GPT4 and Gemini. The system takes a generic resume and a specific job description and processes them to a personalized resume that matches job requirements. Problems related to manual resume customization such as time taking and errors increase. Resumes can be redefined to meet job specific needs by using LLMs to extract details from both resumes and job descriptions. Additionally, ResumeFlow features a ranking system that ranks resume based on job alignment and content preservation metrics, so that the resume is accurate and optimized for the job role.

3.2.2.1 Critical analysis

One of the big pain points of job seekers is how to create a resume for each job application, and ResumeFlow pipeline solves that. Using LLMs, resume personalization becomes easy by matching each user profile with particular job description. But you have to deal with risks of hallucination, where the AI generated content can fabricate details. The article mitigates this by introducing content preservation metrics, however, the risk still exists. Additionally, while the pipeline increases efficiency, it may over-rely on automated keyword matching, missing human nuances and or candidate experience. Furthermore, while LLMs make generating tailored resumes easier, the lack of fine-tuning in the LLMs as

stated in the paper might reduce precision in cases where detailed domain knowledge is required. Despite these drawbacks, the system provides significant advantages in reducing the manual labor required for job applications and improving chances in ATS systems.

3.2.2.2 Relationship to the proposed research work

Our research is also most relevant to the main concept of ResumeFlow which is the application of artificial intelligence based personalization of resumes and skill matching. However, while ResumeFlow targets resume refinement, our project extends to skill gap analysis, compatibility scoring, and providing suggestions for internships and certifications. ResumeFlow's methodology of using LLMs for job description analysis and resume customization can be directly applied to our research work. For example, the cosine similarity technique for job matching and the resume ranking system could be useful frameworks for our compatibility score feature. Moreover, the concern of LLM hallucinations discussed in ResumeFlow is carefully considered in our skill gap analysis feature to ensure accuracy. By addressing these risks and integrating skill improvement suggestions, our research work could advance ResumeFlow's ideas by providing more holistic support for job seekers beyond resume optimization and provide with fine tuning to better optimize our results.

3.2.3 Resume Screening and Course Recommendation System

Article [3] aims to automate the recruitment process by taking resumes and filtering them efficiently and recommending good candidates according to job descriptions. NLP techniques such as stemming, tokenization, and part of speech tagging are used in the system to extract key information (skills, education, experience) from unstructured resumes. The system compares resumes to job descriptions using algorithms such as TF-IDF and Cosine Similarity and outputs a ranked list of candidates. It also suggests the right courses to the candidates based on the skills mentioned in the job postings so that the candidates can enhance and fit their resumes to the job requirements. The system's architecture facilitates job description analysis as well as course recommendations to improve the hiring efficiency and candidate development.

3.2.3.1 Critical analysis

Detailed approach to solving the cause of the inefficient manual resume screening process is presented by the Resume Screening and Course Recommendation System. The use of NLP for extracting data and Cosine Similarity for comparing resumes with job descriptions ensures that the system can streamline the recruitment process. However, challenges such as handling poor-quality resumes and addressing gaps in resume-job matching remain unaddressed. The addition of the Affinity Propagation Algorithm

for clustering resumes based on similarities is an innovative approach, but the system's accuracy could further improve by incorporating semantic analysis, which considers context beyond keyword matching. Moreover, the use of course recommendations enhances the system's value by providing candidates with ways to bridge skill gaps, but the system could benefit from integrating real-time learning resources and certifications.

3.2.3.2 Relationship to the proposed research work

Our research is also most relevant to the main concept of ResumeFlow which is the application of artificial intelligence based personalization of resumes and skill matching. The paper also uses NLP and AI models for job matching and addressing skill gaps. Our project proposes features like CV ranking, compatibility score, and internship recommendations—concepts that are reflected in the Resume Screening System's use of TF-IDF, cosine similarity, and course recommendations. Where our system aims to provide a more personalized approach by generating personalized CVs and cover letters, the reviewed literature focuses the employer's side. Integrating insights from this system could help optimize our project's skill gap analysis and enhance the recommendations through improved clustering and similarity metrics.

3.2.4 A survey on large language models for recommendation

The paper [4] provides a comprehensive overview of the present role of LLMs in recommendation systems. The authors categorize LLM-based recommendation systems into two major paradigms: discriminative LLMs for recommendation known as DLLM4Rec and generative LLMs for recommendation known as GLLM4Rec. They classified these paradigms into various modeling paradigms such as LLM tokens + RS, LLM embeddings + RS, and LLM as RS. they also presented a review of existing methods and discusses the advantages and limitations of each paradigm.

3.2.4.1 Critical analysis

The paper gives a detailed analysis of LLM based recommendation systems. The authors' categorization of LLM based recommendation systems as discriminative and generative paradigms is useful to understand the various approaches in this domain. It discusses both strengths and weaknesses of each paradigm, in order for others to choose the most appropriate paradigm for their own particular use case. However, the paper has some limitations as the authors primarily focus on the technical aspects of LLM based RS and rarely discussed how these systems could be applied to other contexts. It could also benefit from more examples of how LLM based recommendation systems are effective in real world scenarios.

The paper lacks in discussion of the limitations of LLM based RS. As an example, the authors mention

that LLMs may propagate biases and stereotypes, but they don't go into much depth about this issue. They could go deeper and analyze the possible risks and challenges with its use like the possibility of manipulation or exploitation.

3.2.4.2 Relationship to the proposed research work

The following paper can help in our research work in multiple ways. Job postings and resumes can be analysed using LLMs to improve the accuracy of our job recommendation. LLMs can be useful for helping us identify relevant skills and experience, and recommend personalized jobs. With the help of LLMs, we can use our project to evaluate resumes and give feedback on where to improve. An LLM can also point out gaps in skills and experience and create resumes that emphasize relevant skills and experience.

3.2.5 Design and development of machine learning based resume ranking system

In the article [5], a machine learning based resume ranking system is explored. The main goal of the research is to automate the resume screening and ranking process so that recruiters can reduce the amount of time and manual effort. It matches resumes with job descriptions using natural language processing (NLP) techniques like cosine similarity and TF-IDF. It also implements a KNN algorithm to rank resumes in terms of their similarity to the job description. According to the authors, their system has an average parsing accuracy of 85% and a ranking accuracy of 92%.

3.2.5.1 Critical analysis

Machine learning techniques are used in the article to present a novel approach for automating resume screening process. Yet, the system has some limitations and biases that are to be addressed. For example, the resumes and job descriptions that the system uses can be variable. Moreover, the system could not capture the nuances of human judgment or miss important contextual information. Additionally, the system's use of TF-IDF and cosine similarity may not be enough to represent the intricacies of human language. Furthermore, the article does not present a complete assessment of the system's performance regarding its capacity to handle different resumes and job descriptions as well as the system's possible prejudice. Finally, the authors don't discuss what could happen if the resume screening relies solely on automated systems, and that could include errors and biases.

3.2.5.2 Relationship to the proposed research work

In several ways, the article is related to our proposed research work. In our proposed research work we attempt to use machine learning techniques to automate the resume screening process and expand upon

the article by proposing a more complete system including personalized job recommendations, skill gap analysis and resume enhancement suggestions. The project also seeks to employ more sophisticated methods, including generative adversarial networks, to enhance resume and job description quality. In general, the article is a good place to start for our project.

3.2.6 Resume ranking based on job description using SpaCy NER model

The article [6] introduces a resume ranking system based on job description using SpaCy NER model. It is a system that tries to automate the hiring process by taking resumes and extracting the required entities to create a graph with the score of each resume. To improve efficiency of the hiring process and reduce the cost of hiring, the authors propose a machine learning approach to rank candidates based on their resumes and job descriptions.

3.2.6.1 Critical analysis

This article introduces a new method of resume ranking using SpaCy NER model, which could increase hiring process efficiency. There are however some limitations to the study. The scope of the results cannot be evaluated without mentioning the dataset used for training the model, and thus, the first thing is to mention the dataset used for training the model. Thirdly, the authors do not give a full performance analysis of the model and provide metrics like precision, recall, F-score, which are important metrics to evaluate the accuracy of a machine learning model. The article, however, does not address the possibility of model biases, which could result in undesired treatment of some groups of candidates.

3.2.6.2 Relationship to the proposed research work

The proposed research work includes the development of a resume ranking system, which is discussed in this article. The goal of our project is to produce a resume compatibility score and rank resumes by job description. Similar to the article, we attempt to provide job recommendations based on skillset and experience level, which is what our research work is trying to achieve. Overall, the article serves as a foundation for our research work and the proposed system can be built upon the concepts and approaches discussed in the article to create a more complete AI powered job advisor.

3.2.7 LlamaRec: Two-stage recommendation using large language models for ranking

LlamaRec, a two stage recommendation system based on LLMs for ranking based recommendation, is proposed in the paper [7]. The framework comprises a retrieval stage with a small scale sequential recommender to retrieve candidate items, and a ranking stage with a LLM to rank retrieved items. It

is fine tuned with a verbalizer approach where output logits are translated to distributions over candidate items. The authors evaluate LlamaRec on three benchmark datasets and show its performance on recommendation in terms of the recommendation performance and efficiency.

3.2.7.1 Critical analysis

The paper provides a well structured and coherent framework for LLM based recommendation. A reasonable design choice is to use a two stage approach where a lightweight retriever is used to generate candidate items and a more powerful LLM is used to rank. Another interesting innovation is the verbalizer approach, which allows for ranking without producing long text. But the paper has some limitations and potential problems. There is no clear theoretical justification of using LLMs in recommendation in paper. Despite the demonstrated efficacy of LLMs for a range of NLP tasks, the application of LLMs to recommendation is not well motivated. The evaluation of LlamaRec in this paper is limited to three benchmark datasets which may not be representative of real world scenarios. The paper states that LlamaRec is efficient, but the evaluation of efficiency is quite limited to a simple comparison with a generation based approach. However, the paper offers no insights into the interpretability of the LLM-based ranking model. For real world applications, it is important to understand how the model makes recommendations and what features it is using to make decisions.

3.2.7.2 Relationship to the proposed research work

Product recommendation is the focus of the article, but concepts and techniques presented are applicable to job recommendation and CV ranking. The article proposes a two stage framework that can be adapted for job recommendation. The retrieval stage can be applied to efficiently locate a set of relevant job postings from user search query, skills or interests. After that, the ranking stage can be used to rank the retrieved job postings in relation to the user's preference, skills, and experience. The article suggests a way to rank CVs based on their relevance to a job post using the verbalizer approach. The LLM can be fine tuned to understand the job posting requirements and the skills and experience mentioned in the CV. Then, the output of the LLM can be transformed to a ranking score using the verbalizer, which makes CV ranking efficient and effective.

3.2.8 An intelligent decision support system for recruitment: resumes screening and applicants ranking

I-Recruiter, an intelligent decision support system developed in [8] to automate the recruitment process by screening and ranking resumes based on their semantic similarity to job descriptions, is discussed in the article. The system trains word embeddings using machine learning and natural language processing

techniques, and matches resumes to job descriptions. The system consists of three main blocks: matching and extracting, then training. For a set of resumes, domain trained word embeddings are generated in the training block. Semantic similarity between resumes and job descriptions is used for the matching block to find the top candidates. The extracting block takes basic info of top ranked candidates resumes. Execution time and accuracy of the system have been shown to be excellent.

3.2.8.1 Critical analysis

This article discusses a well structured method of automating the recruitment process using machine learning and natural language processing techniques. A strength of the system is that it uses word embeddings to capture the semantic meaning of words in resumes and job descriptions. The system, however, is very dependent on the quality of the training data, and the accuracy of the system is dependent on noisy or irrelevant data. In addition, the system only uses semantic similarity between resumes and job descriptions, and ignores other factors that might be relevant in the recruitment process, such as the candidate's personality or cultural fit.

3.2.8.2 Relationship to the proposed research work

In several ways, the proposed research work is related to the I-Recruiter system presented in the article. In the paper, the system provides job recommendations for the candidate based on the candidate's skills and experience. The I-Recruiter system ranks resumes using a matching block by their semantic similarity to the job description, just like the CV ranking feature of the research work. Although this research work is built on top of the I-Recruiter system, the proposed work adds additional features like personalized CV and cover letter generation, LinkedIn extension, and a feedback mechanism to iterate on the application until the application satisfies the job post requirements.

3.2.9 Novel online recommendation algorithm for massive open online courses (NoR-MOOCs)

The research paper [9] proposes a novel online recommendation algorithm for Massive Open Online Courses (MOOCs) called NoR-MOOCs. The algorithm addresses the information overload problem faced by MOOC learners due to the increasing number of available courses. Existing recommendation techniques often suffer from scalability, sparsity, and cold start problems, making them unsuitable for the dynamic MOOC environment. NoR-MOOCs aims to overcome these limitations by providing accurate recommendations while scaling efficiently with increasing data.

The algorithm utilizes a hyper-sphere-based clustering approach. Each hyper-sphere represents a group of similar learners, characterized by a center, density, and dynamic radius. The algorithm processes each

data point (learner's course ratings) only once. During training, new learners are either added to existing hyper-spheres if similar, or a new hyper-sphere is created. The radii of the hyper-spheres are dynamically adjusted after processing a predefined number of data points (a "generation"). Recommendations are generated by calculating the weighted average of similar hyper-spheres to a given learner. The algorithm is designed to be incremental, adapting to new data without requiring retraining of the entire model. The paper presents experimental results on the COCO dataset, claiming that NoR-MOOCs outperforms traditional KMeans and Collaborative Filtering algorithms.

3.2.9.1 Critical analysis

The proposed NoR-MOOCs algorithm presents some interesting ideas, however, the algorithm heavily relies on the concept of "similarity" between learners and hyper-spheres. The paper does not explicitly define the similarity measure used. The choice of similarity measure significantly impacts the clustering and recommendation results, and its omission makes it difficult to reproduce the results or assess the algorithm's effectiveness. Despite mentioning the cold-start problem as a limitation of existing techniques, the paper doesn't clearly explain how NoR-MOOCs addresses this issue.

3.2.9.2 Relationship to the proposed research work

The proposed research work and shares similarities with the AI-powered job advisor system. Both proposals aim to develop a recommendation system that provides personalized suggestions based on user preferences and behavior. However, the NoR-MOOCs proposal focuses on recommending online courses, whereas the AI-powered job advisor system focuses on recommending job postings and providing skill gap analysis. The NoR-MOOCs proposal uses a novel online recommendation algorithm that processes each data point only once, making it efficient and scalable for large datasets. The NoR-MOOCs proposal can be beneficial for the AI powered job advisor system by incorporating a similar online recommendation algorithm to recommend skills to users to improve their resumes.

3.2.10 Course Recommendation Model Based on Layer Dropout Graph Differential Contrastive Learning

Existing course recommendation systems that mostly hinge on bipartite graphs are not able to capture user–user and course–course relationships, and the paper [10] attempts to address these challenges. To address the problems of overfitting in multi layer graph convolutional networks (GCNs) and sparse recommendation data, the authors propose a novel course recommendation model which combines layer dropout and graph contrastive learning (DGCL). To address the problem of overfitting and to facilitate the contrastive learning process, the proposed model uses hybrid graph convolution networks along with

dropout techniques. The model is shown to perform better than existing models in terms of recommendation accuracy on the XuetangX and MOOCCube datasets. The paper also points out that differential contrastive learning is used to distinguish the similarity of nodes in the graph, and therefore the model can make more personalized recommendations.

3.2.10.1 Critical analysis

The research provides important improvements to the design of recommendation systems by proposing a method for dealing with both low order and high order relationships in the data and thus expanding the model. The DGDCL model is one of the strengths of the model because it can solve the problem of overfitting through the use of the layer dropout method, which is better than the traditional GCN. Moreover, the focus on addressing sparsity through contrastive learning is another key strength, as this directly tackles a common issue in recommendation systems. However, one limitation lies in the complexity of the model, which might require extensive computational resources and may not scale efficiently for very large datasets. Furthermore, the incorporation of auxiliary data, such as knowledge points or teacher data, as suggested for future work, could further refine the model but was not explored in the current iteration.

3.2.10.2 Relationship to the proposed research work

The job recommendation system proposed in our project shares a similar underlying goal with the course recommendation model in addressing the issue of matching users to relevant opportunities based on their profiles. The use of GCN-based models will be helpful to process complex relationships between users (or candidates) and the respective opportunities (jobs or courses). The focus of our project on providing skill gap analysis is comparable to the differential contrastive learning in the DGDCL model, which aims to fine-tune the similarity distances between nodes. The challenge of sparse data in course selection has parallels in the job recommendation process, where data related to candidate applications may be sparse. The dropout techniques used in the DGDCL model could be adapted to address similar issues in our job recommendation platform.

3.2.11 Resumonk

CV and cover letter making services offered by Resumonk [11] are quite popular among job seekers. Users who do not have much time or design experience will find it comes with easy tools for fast and efficient creation of professional resumes and cover letters. On the site, users have ready made templates to choose from, and can also change their documents as they please.

3.2.11.1 Critical analysis

The core strength of Resumonk is in the ability to create CVs and cover letters very quickly and easily. This is especially helpful for users who want a great looking resume but don't want to spend too much time customizing. Nevertheless, its primary drawback is the absence of a module for personalized CV creation that would take into account the particular criteria for the job description. The resume templates are nice looking, but the designs are pretty standard, so they don't go further with the unique abilities for the job. What is more, job seekers are not given any tools that could enable them to know their skills deficiency or how to improve their CVs other than beautifying them. This can also continue to miss out on opportunities for job seekers using Resumonk in competitive job markets where employers prefer candidates with a targeted resume.

3.2.11.2 Relationship to the proposed research work

Resumonk is trying to take on the challenge of streamlining and improving CVs for any job in consideration, and is currently not offering personalized CV creation. Resumonk's lack of ability to customize resumes to the relevant job postings is filled by the AI powered job advisor which seeks to use LLMs and machine learning to write CVs based on the job posting. The proposed research can also capitalize on the shortcomings that Resumonk suffers in the skill gap analysis and job recommendations, by incorporating these elements and improve the competitiveness of job seekers. In our study, Resumonk's simple framework can be designed to incorporate the posted jobs and modify CV's accordingly.

3.2.12 LinkedIn

LinkedIn [12] is a combination of features including job suggestions, endorsements of skills, CV making and connection, and it is a platform for its users, which makes it more effective for job seekers and employers.

3.2.12.1 Critical analysis

LinkedIn does not offer as much granular job recommendations or profile creation as it could. Its profile building is centered on generic skill lists and achievements, and the CVs are not tailored to particular job postings. The platform gives very little feedback on how users can work on their profile to fill a certain job role. Additionally, user generated data relies on incomplete candidate profiles and the candidate job matching of this approach is not as accurate as ML techniques that are more advanced.

3.2.12.2 Relationship to the proposed research work

The proposed AI based job advisor is consistent with LinkedIn's ability to connect users to job opportunities based on their skills and experience. However, our research aims to make this process better by targeting more targeted and job specific CV ranking and skill gap analysis. For instance, LinkedIn uses job recommendations based on user profiles but our system would take it a step further by analyzing each user's CV with respect to certain jobs and automatically crafting personalized resumes that reflect the most appropriate skills for a specific job.

3.2.13 ZipRecruiter

ZipRecruiter [13] is a platform where job seekers can reach jobs that are perfectly tailored to their skills, experiences and interests. The platform posts job advertisements on different sites, search CVs, match openings and applicants to simplify and improve the job search process.

3.2.13.1 Critical analysis

ZipRecruiter's strength is that it's able to rank candidates by qualifications and match them to the right job opportunities. Real time job recommendation and CV ranking is what the platform focuses on, making it useful to both job seekers and employers. Paid subscription is required for its advanced features. It also doesn't offer detailed feedback about candidates, skill gap analysis or personalized.

3.2.13.2 Relationship to the proposed research work

Our project's job recommendation system is similar to ZipRecruiter's job matching approach but we want to push it further by providing personalized CV and skill gap analysis. ZipRecruiter is more job matching than candidate development. Other than that it will match candidates to jobs and rank resumes, our system will also analyze job postings in depth and generate personalized CVs. And it will also suggest ways to improve skills, such as courses or internships, that ZipRecruiter doesn't currently provide. The proposed system is thus more complete tool for job seekers who can use it to improve not only their resumes but also their qualifications.

3.3 Literature Review Summary Table

To provide a summary of all the literature reviews above see table ?? for related research work and table 3.3 for related application.

Table 3.1: Literature Review Summary Table

Author	Method	Results	Limitations
Du et al. [1]	Combines LLMs and GANs to refine low-quality resumes by integrating explicit and implicit user characteristics.	Outperforms existing job recommendation systems across multiple datasets.	Assumes sufficient user interaction data for inferring implicit characteristics, and the GAN-based approach may add computational complexity.
Zinjad et al. [2]	Uses LLMs to automate the creation of job-specific resumes by processing generic resumes and job descriptions.	Efficiently generates tailored resumes, improving job alignment while preserving content accuracy.	Risks of hallucinations and over-reliance on keyword matching, with limited fine-tuning for domain-specific precision.
Jagtap et al. [3]	Employs NLP techniques like TF-IDF and Cosine Similarity for resume screening and course recommendations based on job descriptions.	Improves recruitment efficiency by ranking candidates and recommending skill improvement courses.	Does not fully address poor-quality resumes and may benefit from enhanced semantic analysis.
Wu et al. [4]	Categorizes LLM-based recommendation systems into discriminative and generative paradigms, reviewing different modeling techniques.	Provides a comprehensive overview of LLMs in recommendation systems and their applications.	Lacks discussion on broader applications, risks of biases, and real-world examples.
Tejaswini et al. [5]	Uses NLP (TF-IDF and Cosine Similarity) and KNN algorithm to automate resume screening and ranking.	Achieves 85% parsing accuracy and 92% ranking accuracy.	May overlook nuances in human judgment and relies on the quality of input resumes and job descriptions.
Satheesh et al. [6]	Uses SpaCy's NER model to extract required entities from resumes and match them with job descriptions for ranking.	Generates a graph displaying scores of resumes, aiding in efficient hiring	Lacks details on dataset, model performance metrics, and potential biases in the model.
Yue et al. [7]	Implements a two-stage system with a small retriever for item retrieval and an LLM for ranking using a verbalizer approach.	Shows improved recommendation performance on benchmark datasets.	Limited dataset evaluation, lacks a strong theoretical justification for LLM use, and offers minimal efficiency and interpretability analysis.
Najjar et al. [8]	Uses machine learning with word embeddings to match resumes with job descriptions for ranking.	Demonstrates high accuracy and fast execution times in ranking resumes based on semantic similarity.	Relies heavily on the quality of training data and ignores non-semantic factors like personality or cultural fit.

Table 3.2: Table 3.1 continues

Author	Method	Results	Limitations
Khalid et al. [9]	Uses a hyper-sphere-based clustering approach for MOOC recommendations with incremental learning.	Outperforms traditional KMeans and Collaborative Filtering algorithms.	Does not clearly define the similarity measure used or fully explain how the cold-start problem is addressed.
Ouyang et al. [10]	Proposes a hybrid GCN model with layer dropout and contrastive learning to enhance course recommendation accuracy.	Outperforms existing models on benchmark datasets in terms of recommendation accuracy.	Complex model may require extensive computational resources, and further refinement using auxiliary data was not explored.

Table 3.3: Application Review Summary Table

Application	Features	Relevance	Limitations
Resumonk [11]	CV and Cover Letter Generation	Provides simple and effective CV creation and Cover Letter Generation	Lacks personalization according to job description and recommend skills
LinkedIn [12]	Job recommendation, CV generation, and skill recommendation	A comprehensive platform that integrates job recommendations, skill endorsements, and CV creation, offering a richer experience than traditional CV generators like Resumonk	Lack of personalized CV and skill recommendation related to specific job role
ZipRecruiter [13]	Job recommendation and CV ranking	Focuses on job recommendations and CV ranking, similar to LinkedIn but with a stronger emphasis on job matching algorithms	Lacks advanced skill recommendations and user-friendly CV customization features

3.4 Conclusion

This chapter highlights the use of LLMs and natural NLP techniques to enhance resume screening, job matching, and course recommendation systems. While these approaches have shown promising results, they also come with limitations, such as reliance on high-quality training data, potential biases, and computational complexity. Our Job-Advisor focuses on addressing these limitations and exploring more comprehensive and nuanced methods for improving the accuracy and fairness of these job recommendation system and develop resume generation to broaden to more use cases.

Chapter 4 Software Requirement Specifications

This Chapter will describe the complete and comprehensive description of the requirements for our **AI-Powered Job Advisor** by providing all the functional requirements, design constraints, and other factors necessary.

4.1 List of Features

The key features of the **AI-Powered Job Advisor** are as follows:

- **Job Recommendations**: Provides personalized job recommendations based on the user's skillset, experience, and preferences.
- **CV and Cover Letter Generation**: Automatically generates customized CVs and cover letters for specific job applications.
- **Skill Gap Analysis**: Compares the user's qualifications against job requirements and highlights skill gaps.
- **Job Scraping**: Scrapes job postings from various platforms (e.g., LinkedIn, job boards) and consolidates them on the platform.
- **LinkedIn Extension**: A browser extension to pull job posts directly from LinkedIn and suggest improvements for better matching.
- **Profile Management**: Users can create and update their profiles with detailed information, including skills, experience, and preferences.

4.2 Functional Requirements

The functional requirements fully describe the external behavior of the system. key requirement of our FYP include:

- Users should be able to register, create profiles, and log in securely.
- Users should be able to create, update or delete their accounts.
- Users should be able to enter relevant information when they are creating an account.
- Users should be able to search and filter job postings based on locations, industry, and other criteria.
- The system should be able to generate list of CVs and the Users should be able to select from the

list.

- The Users should be able to retrieve their compatibility score according to the job they have applied for.
- The Users should be able to see recommended courses and internships.
- The system should produce CVs based on the user's information and specific jobs.
- Users' profiles should be used to recommend relevant job postings on the system.
- Job postings from external platforms should be scraped and updated in the system, regularly.
- The system should be able to rank CVs among other candidates as well as other CVs and give a compatibility score to the user.
- Job postings should be able to provide key details about required skills and experience for the system to extract.
- We need the system to be able to extract key details such as skills and education from the user CV.
- The system should be capable of performing skill gap analysis with respect to the job requirement and the user's skill and suggest suitable courses and internships.
- We should have a browser extension to extend our services to users of other job platforms.

4.3 Quality Attributes

The system must meet the following quality attributes to ensure performance and user satisfaction:

- **Reliability:** However, data processing is a must and that the system should handle it consistently and return accurate results, specifically in job recommendation and skill gap analysis. In addition, all features of the application must be reliable, so that no data loss or compromise occurs.
- **Security:** The system must implement robust security measures to protect user data from unauthorized access or breaches and keep user data confidential and intact.
- **Usability:** A user friendly interface should be offered by the system in order to make navigation easy and a smooth user experience. It should be intuitive to use, and provide seamless interaction in response to user queries.
- **Performance:** The system has to operate in all scenarios with high performance even under time or resource constraints, and operate efficiently.
- **Efficiency:** The application should be able to run efficiently during peak recruitment periods.

Under heavy load, it must use available resources optimally in order to keep functioning smoothly.

- **Flexibility:** The system must be able to respond to uncertainty. It will easily adapt to future changes and will also enable the user to customize some criteria as per his need.

4.4 Non-Functional Requirements

The Non-functional requirements of our system are as follows:

4.4.1 Performance

- The system should be able to load and display data within 5 seconds.
- Job recommendations and CV generation should be able to be done in less than 1- seconds.
- The bounce rate should be between 20 to 35 percent.

4.4.2 Security

- There should be a process of user authentication in the system.
- The job scrapping should be from secure sites.
- The system should employ certain data encryption techniques to secure the user data.

4.4.3 Compatibility

- If integrated with other websites as extension it should work properly.

4.4.4 Scalability

- At least 250 users with different permissions and roles should be able to use the system.
- It should provide a unified and consistent environment for the developers and database administrators (DBAs) to efficiently manage the system.
- At least 3 CVs should be generated at a time for a single user.

4.4.5 Usability

- In less than 4 steps, the system should allow user to apply for a job.
- The system should be able to generate CV in less than 3 steps on request by the users.
- The system shall be responsive to different screen sizes.

- The user interface shall be simple and intuitive to the users of all technical expertise.

4.5 Assumptions

Following is a list of assumptions that help us define the flow of main functionalities of our system:

- The users know how to interact with web based platforms.
- Job posts scraped from third-party platforms are publicly available and accessible via APIs or web scraping techniques.
- Users provide authentic and genuine information while creating their profile.
- Users provide proper feedback for the CVs generated by the system.
- Users have a steady and consistent internet access for real-time platform functionalities.

4.6 Use Cases

The following use cases outline the primary interactions between users and the AI-Powered Job Advisor.

4.6.1 Register User

Name	Register User		
Actors	User		
Summary	A new user registers an account by providing personal details such as name, email, and password.		
Pre-Conditions	The user does not already have an account on the platform.		
Post-Conditions	A new user account is created, and the user is redirected to the login page.		
Special Requirements	<ul style="list-style-type: none"> - Valid email format. - Password strength verification. - CAPTCHA for bot protection. 		
Basic Flow			
Actor Action		System Response	
1	User navigates to the registration page.	2	The registration form is displayed, prompting the user to enter personal details.
3	User enters valid name, email, and password.	4	System validates the input, creates a new user account, and sends a confirmation email.
5	User receives confirmation and is redirected to the login page.	6	The user is informed that registration was successful and can now log in.
Alternative Flow			
3A	User enters an email that is already registered.	4A	System responds with an error message: "Email is already registered." and prompts the user to log in or use a different email.

4.6.2 Login User

Name	Login User		
Actors	User		
Summary	A registered user logs in using their email and password.		
Pre-Conditions	The user is already registered.		
Post-Conditions	The user is authenticated and granted access to their dashboard.		
Special Requirements	CAPTCHA for bot protection.		
Basic Flow			
Actor Action		System Response	
1	User navigates to the login page.	2	The login page is displayed asking for email and password.
3	User enters valid email and password.	4	The system verifies the email and password, establishes a session for the user and redirects the user to the home page.
Alternative Flow			
3A	User enters invalid email or password.	4A	The system responds with an error message: Incorrect email or password entered.

4.6.3 View Profile

Name	View Profile		
Actors	User		
Summary	The user views their detailed profile information, including skills, work experience, education, and preferences.		
Pre-Conditions	The user is logged in and has a profile set up.		
Post-Conditions	The user's profile details are displayed.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access to the dashboard.
3	User navigates to the profile section.	4	System retrieves and displays the user profile information.
Alternative Flow			
3A	If no profile exists, the system prompts the user to create one.	4A	System displays a message: "No profile found. Please create your profile."

4.6.4 Edit Profile

Name	Edit Profile		
Actors	User		
Summary	The user updates their profile information, such as adding new skills, updating work experience, or changing preferences.		
Pre-Conditions	The user is logged in and has an existing profile.		
Post-Conditions	The profile is updated and saved to the database.		
Special Requirements	Validation for input fields (e.g., proper date formats). Responsive design for various devices.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access to the dashboard.
3	User navigates to the profile section and selects "Edit Profile."	4	System displays the editable profile form with current information.
5	User updates the necessary fields and submits the changes.	6	System validates the input and updates the profile in the database.
7	User receives a confirmation message: "Profile updated successfully."	8	System redirects the user back to the profile view page with updated information.
Alternative Flow			
5A	User enters invalid data in one or more fields.	6A	System displays error messages indicating the invalid fields and prompts user to correct them.

4.6.5 Delete Profile

Name	Delete Profile		
Actors	User		
Summary	The user deletes their profile, removing all associated data from the platform.		
Pre-Conditions	The user is logged in and has an existing profile.		
Post-Conditions	The user's profile and all related data are permanently deleted.		
Special Requirements	Confirmation prompt before deletion. Ensure data is irreversibly removed.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access to the dashboard.
3	User navigates to the profile section and selects "Delete Profile."	4	System prompts the user to confirm deletion with a warning message.
5	User confirms the deletion.	6	System deletes the user's profile and all associated data from the database.
7	User receives a confirmation message: "Your profile has been successfully deleted."	8	System redirects the user to the homepage or logout.
Alternative Flow			
5A	User cancels the deletion process.	6A	System aborts the deletion and retains the user's profile.

4.6.6 Recommend Jobs

Name	Recommend Jobs		
Actors	User, Recommendation System		
Summary	The system provides job recommendations tailored to the user's skills, experience, and preferences.		
Pre-Conditions	The user is logged in and has a complete profile with skills and preferences.		
Post-Conditions	Relevant job recommendations are displayed to the user.		
Special Requirements	AI model trained for accurate job matching. Regular updates of job data sources.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies and grants access to the dashboard.
3	User navigates to job recommendations page.	4	System analyzes the user's profile and retrieves matching job postings.
5	System displays a list of personalized job recommendations.	6	User reviews the recommended jobs and can select to view more details or apply.
Alternative Flow			
5A	No relevant jobs are found for the user.	6A	System displays a message: "No personalized job recommendations available at the moment. Please update your profile or try again later."

4.6.7 Filter Jobs

Name	Filter Jobs		
Actors	User		
Summary	The user searches for job postings using filters such as location, job title etc.		
Pre-Conditions	The user is logged in.		
Post-Conditions	A list of job postings matching the search criteria is displayed.		
Special Requirements	Advanced filtering options. Responsive search functionality.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access to the dashboard.
3	User navigates to the job search page.	4	System displays the job search interface with filter options.
5	User enters keywords and selects filter criteria (e.g. location, industry).	6	System retrieves and displays job postings that match the search and filter criteria.
7	User reviews search results and can refine the search if necessary.	8	System updates the displayed job postings based on any additional refinements.
Alternative Flow			
5A	No job postings match the search criteria.	6A	System displays a message: "No jobs found matching your criteria. Please try different filters."

4.6.8 Generate CV

Name	Generate CV		
Actors	User		
Summary	The system automatically generates a personalized CV for the user based on their profile information and specific job applications.		
Pre-Conditions	The user is logged in and has a complete profile with necessary details.		
Post-Conditions	A personalized CV is generated and available for download.		
Special Requirements	Multiple CV templates available. Formatting consistency.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access to the dashboard.
3	User navigates to the CV generation.	4	System displays the CV generation interface.
5	User selects a job posting or initiates CV generation.	6	System analyzes job requirements and user's profile to generate a tailored CV.
7	System generates the personalized CV and presents it to the user for review.	8	User reviews the CV and downloads it in the preferred format (e.g., PDF, DOCX).
Alternative Flow			
5A	User initiates CV generation without selecting a specific job posting.	6A	System generates a generic CV based on the user's profile.
7A	User's profile is incomplete.	8A	System prompts the user to complete missing profile information before generating CV.

4.6.9 View Compatibility Scores

Name	View Compatibility Scores		
Actors	User, System		
Summary	The user can view their compatibility scores for jobs they applied to, allowing them to gauge their competitiveness for future applications.		
Pre-Conditions	User has applied to jobs through the platform. The system has calculated compatibility scores.		
Post-Conditions	Compatibility score is displayed to the user, along with feedback.		
Special Requirements	Accuracy in calculating compatibility scores. Display meaningful feedback.		
Basic Flow			
Actor Action		System Response	
1	User navigates to the application history page.	2	System lists all jobs the user applied to.
3	User selects a specific job.	4	System displays the compatibility score and feedback.
5	User reviews the score and feedback.	6	System suggests improvements for future job applications.
Alternative Flow			
3A	No score is available for the selected job.	4A	System informs the user that a score cannot be calculated.

4.6.10 Generate Cover Letter

Name	Generate Cover Letter		
Actors	User		
Summary	The system generates a customized cover letter for the user based on the job description and the user's profile.		
Pre-Conditions	The user is logged in and has a complete profile.		
Post-Conditions	A personalized cover letter is generated and available for download.		
Special Requirements	Customization fields for job title, company name, and role. Professional language standards.		
Basic Flow			
Actor Action		System Response	
1	User logs into their account.	2	System verifies the session and grants access.
3	User navigates to the cover letter generation feature.	4	System displays the cover letter generation interface with customization options.
5	User selects specific job posting or enters job details (e.g., job title, company).	6	System analyzes the job description and the user's profile to generate tailored cover letter.
		7	System generates the personalized cover letter and presents it to the user for review.
8	User reviews cover letter and downloads in the preferred format (e.g. PDF, DOC).		
Alternative Flow			
5A	User initiates cover letter generation without selecting a specific job posting.	6A	System generates a generic cover letter based on the user's profile.
7A	User profile lacks necessary information.	8A	System prompts the user to complete missing information before generating cover letter.

4.6.11 Assess Job Eligibility

Name	Assess Job Eligibility		
Actors	User		
Summary	The system assesses if the user is eligible for a job based on the job description and user profile.		
Pre-Conditions	User is logged in and has a complete profile.		
Post-Conditions	A job eligibility report is generated and displayed to the user.		
Special Requirements	<ul style="list-style-type: none"> - Job criteria analysis. - Eligibility threshold and result. 		
Basic Flow			
Actor Action		System Response	
1	User logs into the system.	2	System verifies the session and grants access.
3	User selects a job for eligibility check.	4	System compares user's profile with job criteria.
5	System generates an eligibility report.	6	User reviews the report and downloads it if desired.
Alternative Flow			
3A	User profile is incomplete.	4A	System prompts user to complete profile before proceeding with eligibility check.

4.6.12 Analyze Skill Gap

Name	Analyze Skill Gap		
Actors	User		
Summary	The system performs a skill gap analysis for a job based on the user's profile and job description.		
Pre-Conditions	User is logged in and has a complete profile.		
Post-Conditions	A detailed skill gap analysis report is generated and displayed to the user.		
Special Requirements	<ul style="list-style-type: none"> - Detailed skill mapping based on industry standards. - Suggestions for courses or certifications. 		
Basic Flow			
Actor Action		System Response	
1	User logs in to the system.	2	System verifies the user session and grants access.
3	User navigates to skill gap analysis feature.	4	System displays the interface to initiate skill gap analysis.
5	User selects a job description for analysis.	6	System compares user's skills with the job requirements.
7	User views the detailed skill gap report.	8	System suggests skills to improve and courses to take.
Alternative Flow			
5A	User profile lacks relevant skills.	6A	System prompts the user to update profile with missing information.

4.6.13 Give CV Ranking

Name	Give CV Ranking		
Actors	User, System		
Summary	The system ranks user's CV against other candidates and provides a compatibility score based on how well the user's qualifications match the job requirements.		
Pre-Conditions	The user has generated a CV and submitted a job application.		
Post-Conditions	A compatibility score is displayed to the user, indicating the strength of their application relative to other candidates.		
Special Requirements	AI/ML model for accurate ranking and scoring. Real-time data processing.		
Basic Flow			
Actor Action		System Response	
1	User submits a job application using their generated CV and cover letter.	2	The system records the application details and initiates the ranking process.
3	System analyzes the user's CV against job requirements and other applicants' CVs.	4	The system calculates a compatibility score based on the alignment of job criteria with the user's qualifications.
Alternative Flow			
3A	The system encounters an error during the ranking process.	4A	The system logs the error and notifies the user to try again later or contact support.

4.6.14 Recommend Courses and Internships

Name	Recommended Courses and Internships		
Actors	User, System		
Summary	Based on the skill gap analysis, the system recommends relevant courses and internships to help users bridge their skill gaps and enhance their employability.		
Pre-Conditions	The user has completed a skill gap analysis.		
Post-Conditions	The user receives a list of recommended courses and internships tailored to their skill gaps and career goals.		
Special Requirements	Integration with external learning platforms and internship databases, real-time recommendation updates.		
Basic Flow			
Actor Action		System Response	
1	User completes a skill gap analysis.	2	The system processes analysis results to identify skill gaps and eligibility for various roles.
3	System generates a list of recommended courses and internships based on identified skill gaps.	4	The system displays the recommendations to the user, including links to enroll in courses or apply for internships.
Alternative Flow			
3A	No relevant courses or internships are found for the identified skill gaps.	4A	The system suggests alternative resources such as articles, guides, or webinars to help the user improve their skills.

4.6.15 Use Browser Extension

Name	Use Browser Extension		
Actors	User		
Summary	Users install a browser extension that pulls job postings from external platforms and allows them to apply for jobs directly by streamlining the application process.		
Pre-Conditions	User has access to external job platforms (e.g., LinkedIn). User has installed the browser extension.		
Post-Conditions	Job data is pulled from external sites, and the user is redirected to the platform for CV and cover letter generation.		
Special Requirements	Smooth integration with external job platforms. Secure handling of user and job data.		
Basic Flow			
Actor Action		System Response	
1	User installs the browser extension.	2	System confirms successful installation.
3	User browses job postings on external sites.	4	System pulls job data from external platform.
5	User selects a job posting and clicks "Apply."	6	System redirects the user to the platform for CV and cover letter generation.
Alternative Flow			
3A	Job data is not available or fails to load.	4A	System notifies the user and suggests retrying or checking connection.

4.7 Hardware and Software Requirements

To ensure that clients have the best user experience with our system we have listed below certain hardware and software requirements which must be fulfilled before using our system

4.7.1 Hardware Requirements

- **Client:** The client should have a device with a processor with a speed of 1.5 GHz or faster, a minimum of 6 GB RAM, and 500 MB of free space for cache and temporary files. The device should support an Internet connection (wired or wireless).
- **Server:** The system requires a server with sufficient storage and computational power to host the AI models and manage user requests. Minimum requirements include a Quad-core 2.5 GHz or Higher CPU, a RAM of at least 16 GB, and a storage capacity of 500 GB SSD.

4.7.2 Software Requirements

- **Client:** The clients should use the latest up-to-date version of the Google chrome browser.
- **Server:** The application must support multiple operating systems such as Windows, Mac-OS, and . The front-end will be developed using Next.js and for back-end will utilize Node.js along with Flask to integrate AI models developed in Python.

4.8 Graphical User Interface

The Graphical User Interface (GUI) section outlines the visual elements and user interaction flows of the system. The following elements should be captured:

4.8.1 GUI Screens and Functionality

1. Login Screen

- **Elements:** Fields for username, password, and a login button.
- **Functionality:** Allows the user to enter credentials and log into the system. Error messages will appear if the input is incorrect.

Figure 4.1: Login page

2. Registration Screen

- **Elements:** Fields for personal details such as full name, email, and password.
- **Functionality:** Registers a new user by creating a profile with valid credentials.

Signup

Personal Information

Name:

Email:

Password:

[Next](#)

Signup

Professional Information

Skills:

Experience:

Education:

[Back](#) [Next](#)

(a) Sign Up Page 1**(b) Sign Up Page 2****Figure 4.2: Registration page**

3. Dashboard

- **Elements:** Personalized job recommendations, shortcuts to CV and Cover Letter generation.
- **Functionality:** Displays a summary of user profile information, job applications, and rec-

ommended actions.

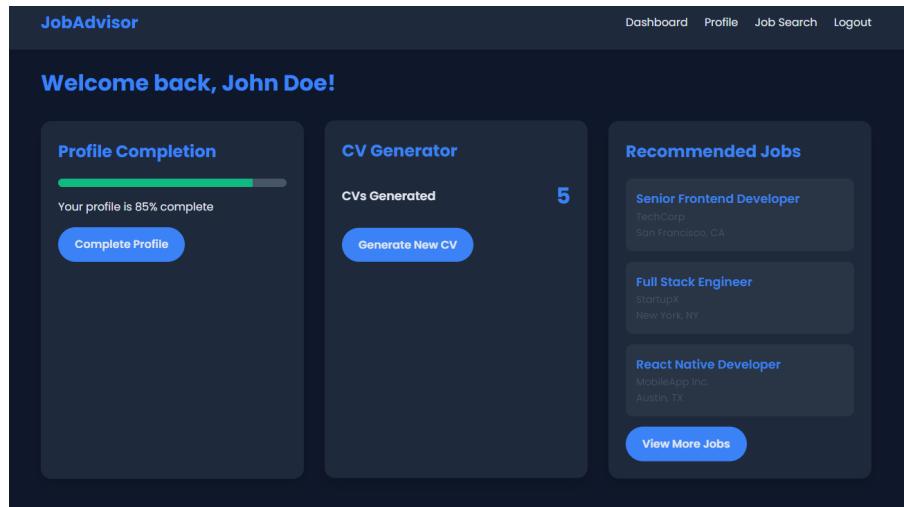


Figure 4.3: Dashboard page

4. CV and Cover Letter Generation

- **Elements:** Dropdowns for template selection and preview buttons.
- **Functionality:** Generates downloadable files (PDF, DOCX) based on user input and job description data.

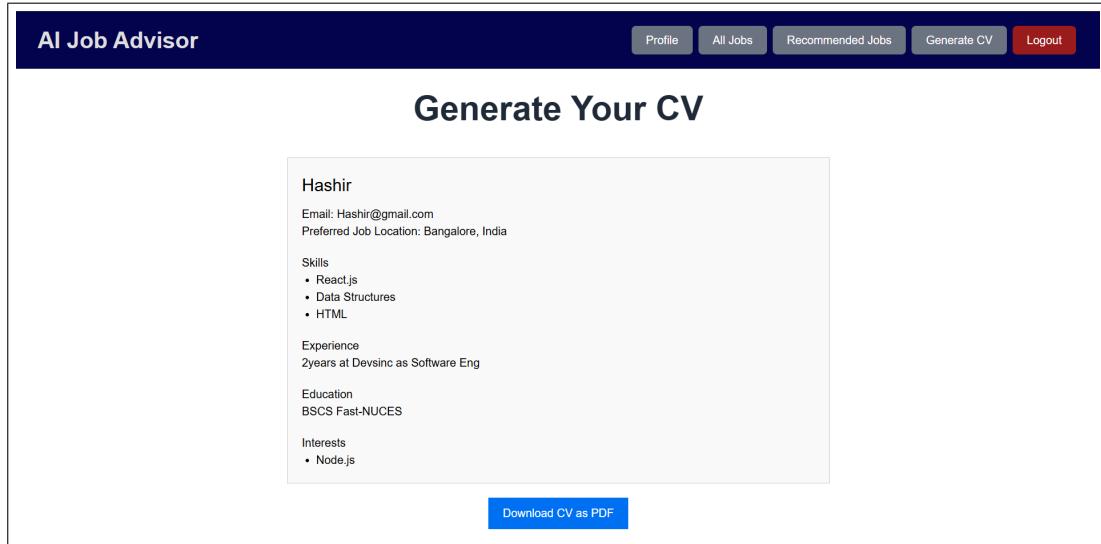


Figure 4.4: CV Generation Page

5. Profile Management

- **Elements:** Profile editing options, fields for skills, education, experience.
- **Functionality:** Allows users to update or edit their profiles with new information.

The screenshot shows a profile editing interface for 'Hashir's Profile'. At the top right is a blue 'Edit Profile' button. Below it are several input fields:

- Name:** Hashir
- Email:** Hashir@gmail.com
- Skills:** React.js, Data Structures, HTML
- Experience:** 2years at Devsinc as Software Eng
- Education:** BSCS Fast-NUCES
- Preferred Job Location:** Bangalore, India
- Interests:** Node.js

At the bottom center is a green 'Change Password' button.

Figure 4.5: Profile page

6. Job Search and Filtering

- **Elements:** Filters for location, industry, job title, and search bar.
- **Functionality:** Allows users to search and filter job listings based on entered criteria.

4.8.2 Navigation Flow

- **Login → Dashboard:** Once logged in, users are directed to their dashboard.
- **Dashboard → Profile Management:** Users can click on their profile picture to edit details.
- **Dashboard → CV/Letter Generation:** Options available to generate or download CVs and cover letters. Users can manually enter job details.
- **Job Search → CV/Letter:** Users can filter jobs and click “Apply” on job listings on LinkedIn.

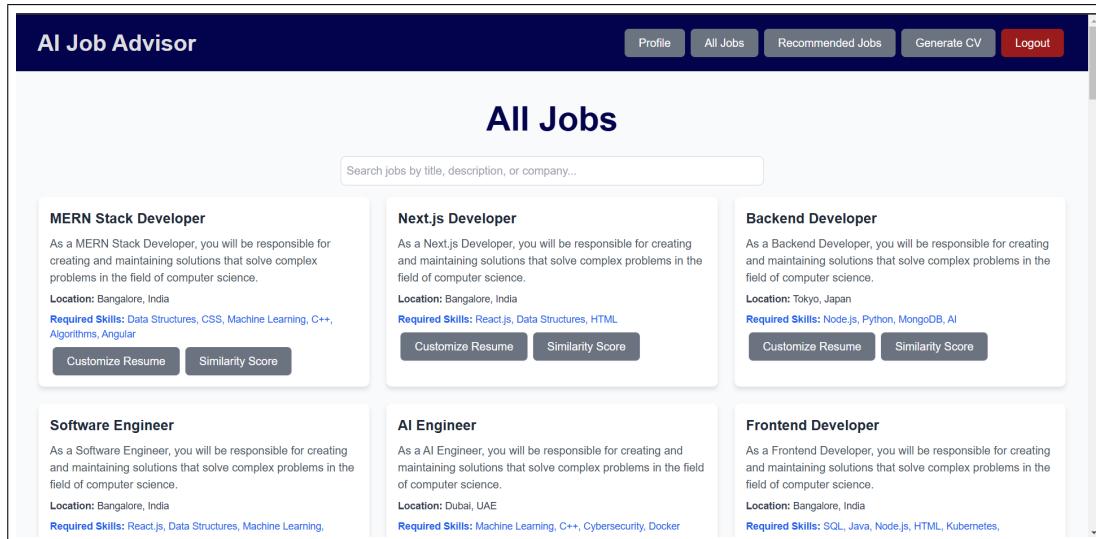


Figure 4.6: Job Recommendation Feed

4.8.3 GUI Guidelines

- **Responsiveness:** The interface must scale across various devices, including desktops, tablets, and smartphones.
- **Accessibility:** The GUI should be accessible to users with disabilities, supporting screen readers and high-contrast modes.

4.9 Database Design

The following provide details on the design of the database

4.9.1 ER Diagram

See figure 6.2 for ER diagram.

4.9.2 Data Dictionary

See Table 4.1 for Data Dictionary.

4.10 Risk Analysis

A thorough risk assessment of the risks presents themselves with respect to the operation of the system above is very important. Here is a list of risks associated with the project and the plans to cover these risks:

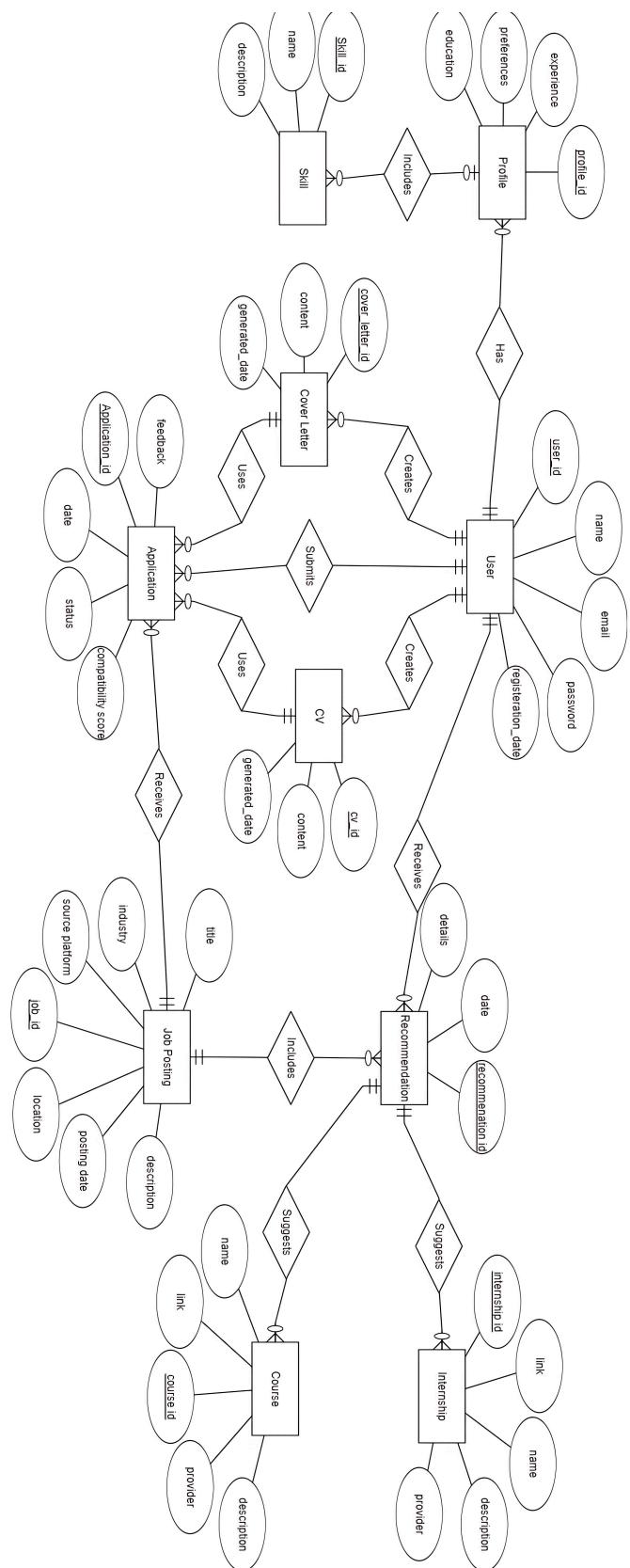


Figure 4.7: This figure represents all the entities and their relationships

Table 4.1: Data Dictionary

Entity	Attribute	Data Type	Null-able	Description
User	user_id	Integer	No	Unique identifier for each user.
	name	String	No	Full name of the user.
	email	String	No	User's email address.
	password	String	No	User's encrypted password.
	date	Date	No	Date of user registration.
Profile	profile_id	Integer	No	Unique identifier for user profile.
	experience	String	Yes	Summary of professional experience.
	preferences	String	Yes	User's job preferences.
	education	String	Yes	Educational background of the user.
Skill	skill_id	Integer	No	Unique identifier for each skill.
	name	String	No	Name of the skill.
	description	String	Yes	Description of the skill.
Cover Letter	cover_letter_id	Integer	No	Unique identifier for cover letters.
	content	Text	No	Text of the cover letter.
	generated_date	Date	No	Date when cover letter was created.
CV	cv_id	Integer	No	Unique identifier for CVs.
	content	Text	No	Text of the CV.
	generated_date	Date	No	Date when the CV was created.
Application	id	Integer	No	Unique id for applications.
	date	Date	No	Date of application submission.
	status	String	No	Current application status.
	compat_score	Integer	Yes	Score showing user-job match.
Recommendation	id	Integer	No	Unique id for recommendations.
	date	Date	No	Date when the recommendation was created.
	details	Text	Yes	Information about recommendation.
Job Posting	job_id	Integer	No	Unique identifier for job postings.
	title	String	No	Title of the job position.
	description	Text	Yes	Detailed job description.
	industry	String	No	Job's industry category.
	location	String	Yes	Job geographical location.
	source platform	String	Yes	Origin platform of the job posting.
	posting date	Date	Yes	Date when the job was posted.
Course	course_id	Integer	No	Unique identifier for courses.
	name	String	No	Name of the course.
	description	String	Yes	Overview of the course content.
	provider	String	Yes	Institution offering the course.
	link	URL	Yes	URL for course details or registration.
Internship	internship_id	Integer	No	Unique identifier for internships.
	name	String	No	Title of the internship position.
	description	Text	Yes	Overview of the internship.
	provider	String	Yes	Organization offering internship.
	link	URL	Yes	URL for internship details.

4.10.1 Technical Risks

- **Advancements in AI Models:** As the project is built around recent AI technologies like prompt engineering, there's a high chance that future advancements in AI can make current models or

techniques outdated.

- **API Deprecation:** The system relies on third-party APIs for scraping job data and integrating with external platforms. If any of these APIs are removed or updated, the system may require modifications or even a complete updation to maintain functionality.
- **Changing Requirements:** The job market and user needs are constantly evolving. If the project does not adapt to new requirements or market changes, it will become irrelevant.
- **System Performance Scalability:** As the user base grows, the system may face performance bottlenecks in terms of job recommendations, real-time data processing etc. Ensuring the system can scale without degradation in performance could require additional technical resources.

4.10.2 Business Risks

- **Market Competition:** The job recommendation and career advisory space is highly competitive. Larger, more established players may introduce similar features or enhancements, reducing the potential user base and making it difficult for the project to sustain its market presence.
- **Legal Risks:** There is always a risk of not complying with evolving data privacy regulations like GDPR, which could result in legal challenges, especially in storing and processing user data related to job applications and personal information.

4.11 Conclusion

This chapter explains the features, requirements and use cases to make sure that the system is meets users' needs. The goal is to design a reliable and user-friendly platform that makes job searching easy and also support skill development. The system is made to help users to achieve their career goals in an easy and efficient way.

Chapter 5 Proposed Approach and Methodology

In this chapter, we dive into the details of the methodology devised to solve given problem. It will primarily focus on how we will acquire datasets related to CVs, job description, courses, and how data scrapers and data parsers will be made to acquire new data and extract features to implement our proposed methodology. We will discuss how we will make use of cosine similarity, LLM and GANs to implement Job recommendation. Make use of Latex to create CVs and do prompt engineering to generate an ideal cover letter. We will also dive into the method used to generate a CV ranking.

5.1 Dataset

In this section, we will discuss how we will go through the process of collecting datasets and make use of data scrapers for accessing latest job description, CV and courses. We elaborate over each dataset item and define their usability within our research.

5.1.1 CV Dataset

A variety of sources were considered during the dataset's collecting process. The Career Service Office (CSO), online repositories and LinkedIn were the main sources of the dataset. The datasets gathered from internet ensured diversity in Resumes and reduce the need of data augmentation. The resumes collected from Career Service Office (CSO) were diverse in domains related to Computer Science Students, who are at the heart of our research work. LinkedIn is a well-known professional networking site which is a valuable resource to collect modern and up to date resumes. Making use of the platform's accessibility we can acquire a range of experience and backgrounds.

5.1.2 Job Description Dataset

Job Description is at heart of our problem domain since we are more focused on job recommendation system. In order to target latest and up-to-date job posts a web scraper will be made to scrape the latest jobs. Already scraped datasets are also used to train models on diverse dataset.

5.1.3 Courses and Certifications

To enhance the applicants knowledge with trusted and known platform. We have decided to scrape data from Known-platforms like Coursera and EDX whose courses and certifications are recognized. Also we have collected data from Credly for more certifications.

5.1.4 Dataset parsing

Manually labeling resume words is a gigantic task so a utility based tool was used to annotate each resume and job description one at a time. The Named Entity Recognition (NER) annotator will be used to make labels. These labels can be used to categorize information of the loaded data. Once the categorization is done a JSON file will be saved of the resume.

5.2 Proposed Methodology

In this section, a comprehensive overview of the proposed methodology of each feature is provided. Our Main goal is to create a sophisticated system which will work as intended to and meticulously connect each feature to provide with a seamless experience.

5.2.1 Job Recommendation and Compatibility Score

5.2.1.1 Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors in an inner product space. We can enhance job Recommendation by comparing the similarity between the job description and a candidate profile. To match a user's resume with a job description, calculate the cosine similarity between the resume embedding and each job description embedding. Compute the cosine similarity between the user resume embedding and each job description embedding. Then rank job descriptions based on their cosine similarity scores, with the highest scores indicating the most relevant matches.

5.2.1.2 LLM-based recommendation system

LLM model can be used to process natural language text, such as job description and candidate profiles. They can extract key information, understand context, and generate relevant representations. The Simple Resume Completion method uses LLMs to directly complete the user's resume based on their self-provided information (e.g., work experience, education, skills). The prompt fed to the LLM contains details about the user's self-description. This method may suffer from fabrication or hallucination by the LLMs, where the model might generate content that doesn't accurately reflect the user's true qualifications or experience.

Interactive Resume Completion is introduced to mitigate the limitations of the simple approach. By incorporating the user's interaction behaviors with job systems (i.e., jobs the user has viewed, applied for, or interacted with), this method improves the LLM understanding. When these interactions are considered, the LLMs are able to create a more personalized and fitting resume.

The model then uses text embeddings and a similarity based matching system to make job recommendations based on the refined resumes. We convert the user’s resume and the job descriptions into text embeddings using a BERT based model (SIM-BERT). The semantic meaning of both resumes and job requirements are captured in these embeddings. Next, the system uses a multi-layer perceptron (MLP) to further refine the representations of the user embeddings and job description embeddings. We measure similarity between the user’s profile and the job requirements using the refined embeddings of the user resume and job descriptions. It helps to match the user with the most appropriate job positions. In the case of the system, it is able to better predict what kinds of jobs the user might be interested in or qualified for by including interaction history.

5.2.1.3 GAN (Multiple Shot vs Few Shot Generation)

Despite the ability of LLMs to assist in user profiling, the few shot problem remains. To address this challenge, we propose to refine low-quality resumes of few-shot users. We achieve this by using a classifier and GAN.

To detect low-quality resume for alignment we define a user with very few interactions as low-quality user. We decide a thresholds separating few-shot and many-shot users. The generator aims to refine low quality resume identified by the classifier. It takes in the low-quality resume representation and tries to transform it into a high-quality version by learning the mapping between these two distributions. The discriminator’s job is to determine whether the resume is a real high-quality one or a refined version created by the generator.

The generator and the discriminator are trained together in a competitive process. The generator tries to improve its outputs to ”fool” the discriminator into thinking the refined resume is high quality. Meanwhile, the discriminator becomes better at identifying whether a resume is a genuine high-quality one or a refined low-quality version.

Let’s say a user submits a resume with a few details (e.g., 2 years of work experience, no technical skills listed, and vague job descriptions). The classifier identifies it as a low-quality resume because of its lack of detailed information and poor formatting. The generator takes this low-quality resume and uses learned patterns from high-quality resumes to add more relevant information. It may expand on the vague job descriptions, adding relevant technical skills, improving formatting, and enhancing the language used. The discriminator is then shown both a high-quality resume and the refined resume generated by. It tries to determine which one is real. The generator is trained to make its refined resume look as close as possible to the real high-quality example, so over time, it becomes harder for the discriminator to tell the difference. After many iterations of adversarial training, the generator gets

better at refining low-quality resumes, eventually producing resumes that are high-quality and ready for real-world job applications.

5.2.2 CV Generation

CV generation is an important feature of our project and displays the results of the proposed methodology presented above. We have decided to use Latex to generate CVs and Cover Letters.

Using LaTeX to generate CVs (Curriculum Vitae) is highly effective for producing professional, structured, and customizable documents with consistent formatting. LaTeX is a typesetting system widely used in academic and technical fields, and it provides full control over the layout and appearance of documents.

Generating a LaTeX document programmatically can be done in various ways, depending on the programming language you' being used. We will use python that can handle string manipulations to generate LaTeX content dynamically. Similarly Cover Letter can also be generated

5.2.3 CV Ranking

A Majority of CV ranking approaches use already ranked resumes and conduct some form of similarity scores to achieve the ranking of current resume. However, this approach is very short-sighted as it is very labour intensive to scale and biases can occur too. With the advent of LLM, we can extract more features and can give accurate compatibility scores according to each job description. However, this does not paint the whole picture of how this profile performs against each job description. To give a holistic perspective we have decided to come up with our own CV ranking approach which will sum all the compatibility scores and give the user a CV ranking.

5.2.4 Skill Recommendation

Skill ranking is another important part of our proposed approach which makes our project unique. Telling the applicant your compatibility score is low and their CV ranking is not high enough. To gamify this process and to enhance user experience, we will recommend courses, certifications and internships to the user to increase their scores. This will not only help them identify where they lack but also encourage them to learn courses which are relevant in the current industry

Below each job description. The system will identify the skills the user lacks. The user can add the skill if he already has it but did not add to his profile or learn it from our recommended courses and certification

5.3 Conclusion

In Conclusion, the suggested methodology offers a unique solution to multiple challenges which are connected to each other. The collection and pre-processing of data will ensure efficient implementation of proposed methodology. The use of LLM and GAN is relatively a new approach to recommending jobs, however, it's use will help us avoid cold start problems and enable rich feature extraction. By tightly-winding Job Recommendation, CV Generation, CV Ranking and Skill Recommendation, our goal is to bring a new innovative approach to the industry which can revolutionize it.

Chapter 6 High-Level and Low-Level Design

This chapter presents the high-level and low-level design of the **AI-Powered Job Advisor**. It provides a system overview, design considerations, and architectural strategies, followed by detailed descriptions of subsystems, components, and their interactions. Both the internal architecture and external relationships of the system are explored, with a focus on the design strategies and methodologies adopted for development.

6.1 System Overview

The **AI-Powered Job Advisor** is a web-based platform designed to help users improve their job application by providing them with the following features :

6.1.1 Interactive User Interface

The application will provide an interactive and responsive user dashboard where users can view job recommendations and track their application progress. Users can also input and update their skills, experience, and preferences to tailor job recommendations to their unique profile.

6.1.2 Job Recommendations

This module utilizes LLMs to analyze the user's skills and experience and matches them with relevant job postings scraped from external sources. Then it will display these recommendations to the user on their dashboard.

6.1.3 Skill Gap Analysis

Similar to the job recommendation module LLMs are utilized to extract the skills and experience from the users profile and is matched with the job description and then if needed it offers personalized suggestions on how to improve their profile to better align with desired positions.

6.1.4 Personalized CV and Cover Letter Generation

Our applications will generate personalized CVs as well as Cover letters for specific jobs tailored to the skills and experience of the user so that their skills and expertise are highlighted, optimizing the user's chances of being selected for interviews. Multiple templates are available to ensure flexibility and customization.

6.1.5 Job Scraping and External Integration

The system employs automated web scraping techniques to gather job postings from various external platforms, including LinkedIn and job boards. This ensures that users have access to the latest job opportunities. The platform also integrates with third-party websites (through Chrome Extension) to enhance the range of job postings available and ensure they are relevant to the user's profile.

6.2 Design Considerations

This section describes many issues that need to be addressed or resolved before attempting to devise a complete design solution. These are crucial to ensure that the end user experiences no issues while using the application.

6.2.1 Assumptions and Dependencies

The design of our Job Advisor is based on the following assumptions and dependencies:

6.2.1.1 Related Software and Hardware

To ensure a seamless experience with our web-application it is assumed that users will have access to laptops or computers that meet the minimum requirement for our application. In addition to this, a stable Internet connection with adequate speed would also be required.

6.2.1.2 Operating Systems

The application will support major operating systems such as Windows, macOS, and Linux. Therefore the user must use a device which has the above mentioned operating system(s) installed on it.

6.2.1.3 End-user Characteristics

The users are expected to have basic knowledge of web browsing and understand the process of creating a CV and applying for jobs. Moreover, the users are expected to be honest in regards to listing their skills while creating a profile.

6.2.1.4 Possible and/or Probable Changes in Functionality

As the job market evolves, the system assumes that additional features and updates will be required, particularly to support changes in job posting formats and data sources. Moreover we will also consider the development in the field of ML/AI and will change the back-end mechanism accordingly.

6.2.2 General Constraints

There are certain limitations and guidelines which need to be kept in mind while creating this project which include the following:

6.2.2.1 Hardware or Software Environment

The performance of our system is dependant on a stable Internet connection aswell as a good machine which works efficiently and meets our minimum requirements. Any fluctuations in these could affect the system's functionality and negatively impact the user experience.

6.2.2.2 End-user Environment

The user must know how to navigate websites on the internet and they should have basic knowledge about the internet and web browsers to make their experience even better.

6.2.2.3 Interoperability Requirements

The system will need to work smoothly with other system software through API and also ensure seamless integration with the database and AI models.

6.2.2.4 Verification and Validation Requirements

The system must include mechanisms for validation and verification of data input aswell as the system response to ensure that a certain standard of accuracy and reliability is maintained.

6.2.2.5 Network Communication

The system will use hyper text transfer protocol secure (https) for network communication which will ensure consistent connectivity for uninterrupted services.

6.2.2.6 Performance Requirements

The system must ensure quick responses without delay, handle ample amount of users at once and it should not crash even under high traffic of users and increased data processing tasks.

6.2.3 Goals and Guidelines

The design of our job advisor is influenced by several principles and guidelines. Firstly, we ensure modularity by dividing our system into components each of which is responsible for a specific function such job recommendation, CV generation, and etc. Secondly we will utilize the KISS principle (Keep It

Simple Stupid) ensuring simplicity in design to make the system as user friendly as possible. Moreover, the system will be designed to operate at high speeds and will not require additional space in memory. Lastly, we will implement user-centric designs. This would involve creating an interactive interface that is understandable and accessible to all users.

6.2.4 Development Methods

Our development approach is inspired by agile methods. We will employ the scrum framework in which the project will be divided into sprints so that we can work on each of the modules efficiently. After each sprint a review meeting will be held in which we will track the progress of our project. This will help us adhere to the deadlines of the project deliverables. Adopting this approach will ensure a seamless development process and a user-friendly product.

6.3 System Architecture

This section will cover the both the internal module architecture and external system interfaces. A diagrammatic representation is provided to help in understanding the system's structure.

6.3.1 High-Level Partitioning

The system will follow the layering architecture and is divided into the following 3 parts:

- **Presentation Layer:** In Front-end we would use Next.js, for user interaction and session maintenance.
- **Business Logic Layer:** The back-end will also utilize next and all our AI-models will be integrated in this layer using Python libraries such as PyTorch or TensorFlow.
- **Data Persistence Layer:** MongoDB is used for data storage.

6.3.2 Component Collaboration

All the components interact in a well defined flow. The Presentation Layer communicates with the Back-end via Restful APIs, sending and receiving data as required. Business logic Layer interacts with the Database for storage/retrieval and utilizes AI-models for recommendations and generations. The Database would be the central repository for all system data, maintaining integrity and security.

6.3.3 Rationale for Decomposition

The layering architecture provides us scalability, maintainability, and efficient handling of various functionalities.

6.3.4 System Architecture Diagrams

The following diagrams are included for a better understanding of our system and the inter linkage between components.

6.3.4.1 High-Level System Architecture Diagram

The major layers and their primary functions are outlined in the high-level architecture diagram 6.1. The inner and other flow of data and control is also highlighted.

6.3.4.2 Detailed Back-end Services

The back-end services are responsible for executing the system's core functionalities, such as data processing, system logic, and communication with external APIs. These services ensure that the data is properly managed, secure, and consistently delivered to the front-end.

- **Job Recommendation Service:** Implements the recommendation algorithms for jobs and interacts with the Job Scraper to process job postings. This service also manages user profile data, ensuring job recommendations are relevant to the user's preferences and qualifications.
- **Skill Gap Analysis Service:** Manages the logic behind comparing job descriptions with user profiles to identify missing or underdeveloped skills. It interacts with both the Skill Analyzer and the Course Recommendation Engine to provide personalized improvement suggestions.
- **CV Generation Service:** It automates CV and cover letter generation from user data, job descriptions, and historical interactions using large language models and GAN based enhancement techniques.
- **Course and Project Recommendation Service:** It suggests personalized course and project based on identified skill gaps. It is a service that connects to other platforms like Coursera, EDX, and GitHub and fetches relevant learning opportunities and development projects.
- **External Integration Service:** It will help to communicate with external job boards and educational platforms. It ensures that job postings and learning resources in the system are always up to date.

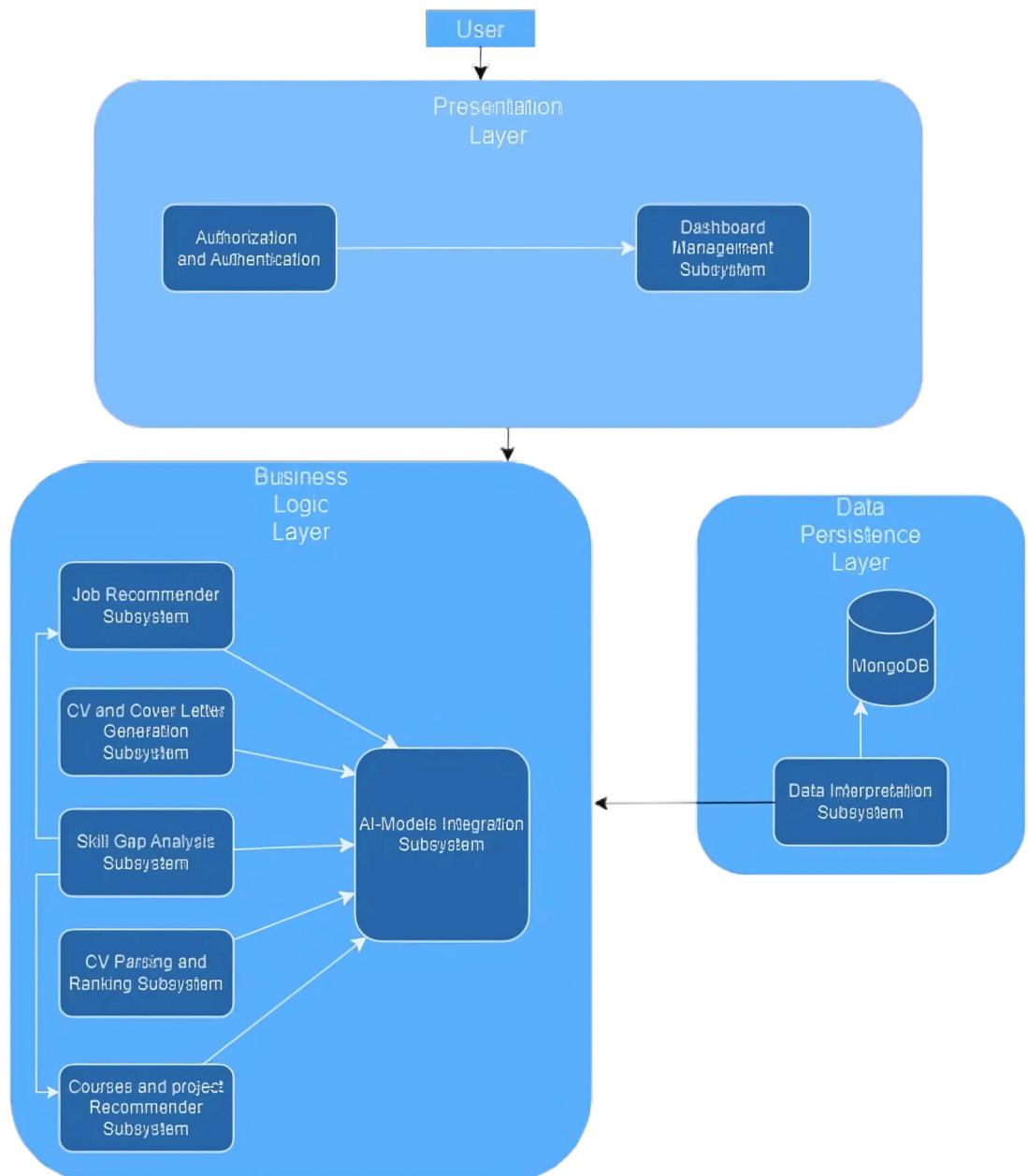


Figure 6.1: Architecture Diagram

6.3.4.3 Detailed CRUD Operations

Data handling in the system is based on CRUD (Create, Read, Update, Delete) operations. These operations provide efficient management of user profiles, job data, CVs, as well as system related entities.

- **Create Operations:**

- Users can create their profile, so they need to input their personal info, education, experience and skill set.
- Integrating with external job boards allows admins and external data scrapers to create job postings, or admins can manually add new postings directly.
- The CV Generation Subsystem allows users to create new CVs and cover letters.

- **Read Operations:**

- Users can read and view their profiles, job recommendations and generated CVs.
- The Job Recommendation Engine reads job postings from external job boards and processes them for analysis.
- The Skill Gap Analysis and CV Parser subsystems use and read user data in order to complete their respective functionalities.

- **Update Operations:**

- Users can update their profile by changing the personal data, skills and experiences.
- When users apply to positions, or when new data is retrieved from external sources, job postings are updated.
- Generating new versions of CVs and cover letters based on user interactions or updated job descriptions can update CVs and cover letters.

- **Delete Operations:**

- Users can delete their profiles and delete all the data that comes along with it including job applications, CVs and recommendations.
- When job postings and recommendations are no longer relevant or no longer valid, they can be deleted.
- This allows users to delete earlier generated CVs and cover letters, thus deleting old versions.

6.3.5 Subsystem Architecture

The components of each Layer is described in the following.

6.3.5.1 Presentation Layer

It is in charge of the system's user interface and interaction. It is the primary gateway through which all user activities occur and with which all the features of the system can be interacted with.

- **Authorization and Authentication Subsystem**

User Management is a part of the system which is responsible for registering or logging in a user, and authenticating user to the platform. In addition to that, it offers OAuth integration support, so users can login via social platforms like Google and LinkedIn.

- **User Interface Subsystem**

This subsystem provides a user friendly interface for the users of this system. Users can manage their profiles, search for jobs, get recommendations and create a customized CV.

It has the following components:

- Login and Registration Module: It provides secure user authentication and registration.
- Dashboard: It presents job recommendations, skill gap analysis and profile insights.
- CV and Cover Letter Generator: It provides options to generate personalised CVs and cover letters according to job descriptions.
- Interaction with Other Subsystems: Business logic layer gets processed data and it sends the data to the user interface to display the data in real time.

6.3.5.2 Business Logic Layer

- **Job Recommender subsystem**

It is a subsystem that processes scraped job postings from different platforms, analyses user profiles and gives personalized job recommendations.

It has the following components:

- Job Scraper: Get job postings from job boards like LinkedIn, or from other external sources.
- Recommendation Algorithm: Uses machine learning models (cosine similarity, LLM-based systems) to match users' skills and preferences in order to recommend relevant jobs to the user.
- Interaction with Other Subsystems: This engine interacts with the User Profile subsystem

to access user information and with the Skill Gap Analysis subsystem to ensure accurate recommendations.

- **CV and Cover Letter Generation Subsystem**

Automates the generation of personalized CVs and cover letters for particular job applications.

It has the following components:

- CV Generator: Uses large language models (LLMs) for automatic creation of CVs with respect to user's profile, job description, and interaction history.
- GAN-based CV Enhancer: Improves the quality of generated CVs by learning from high-quality examples.
- Interaction with Other Subsystems: This subsystem retrieves user data from the profile manager and job details from the recommendation engine.

- **Skill Gap Analysis Subsystem**

This subsystem performs a comparison between user profiles and job descriptions to highlight missing or underdeveloped skills.

It has the following components:

- Skill Analyzer: Uses cosine similarity to compare user skills and job requirements.
- Course Recommendation Engine: Recommending relevant courses and certifications (e.g. from online portals Coursera, EDX etc) to the users based on their profiles to bridge skill gap.
- Interaction with Other Subsystems: It accesses user profile data and Job Recommendation Engine to improve the relevance of the jobs by including the possible skills that the user can develop through the job.

- **CV Parsing and Ranking Subsystem**

This subsystem parses user-uploaded CVs, extracts relevant data, and ranks them based on how well they match job descriptions.

It has the following components:

- CV Parser: Extracts essential details such as skills, experiences, and education from user CVs.
- Ranking Algorithm: It uses machine learning techniques (cosine similarity, keyword extraction, etc.) to rank the CV based on how it matches with job descriptions.
- Interaction with Other Subsystems: The information processed from the CVs is used by the

Job recommendation Engine to have more accurate match with the jobs. It is also used in the Skill Gap Analysis Subsystem to suggest improvements according to the CV's shortcomings using the ranking system.

- **Courses and Project Recommender Subsystem**

This subsystem provides relevant courses, certifications, and projects to fill in the identified skills gaps that will make users more suitable for job postings.

It has the following components:

- Course Recommendation Engine: Based on the skills users need to develop, it recommends courses and certifications from platforms like Coursera, EDX, and Udemy.
- Project Recommendation Engine: It suggests open source or personal project ideas from the platforms like GitHub, fit to the user's development needs.
- Interaction with Other Subsystems: The Skill Analyzer provides information about missing skills, and recommends targeted courses and projects, and learning paths to improve user profiles to refine job recommendations based on possible skill development.

- **External Integration Subsystem**

This subsystem is in charge of interplay between the AI Powered Job Advisor and external job boards or educational platforms.

It has the following components:

- Job Board API Integration: Scraps relevant job data from job boards like LinkedIn, and other platforms.
- Course API Integration: Brings in recommendations from online educational resources like Coursera and EDX.
- Interaction with Other Subsystems: This subsystem supplies the Job Recommendation Engine with job information, and the Skill Gap Analysis Subsystem with course materials.

6.3.5.3 Data Persistence Layer

This layer is responsible for storing and retrieving data:

- **MongoDB:** The database used to store user profiles, job data, and system-generated results.
- **Data Interpretation Subsystem:** Analyzes and interprets stored data to support decision-making in the Business Logic Layer.

6.3.6 Justification for Architectural Choices

The architectural decisions behind the system follow principles of modularity, scalability, and maintainability:

- **Separation of Modules:** Each subsystem is designed to handle a specific functionality which makes the system easier to maintain, test, and extend in the future.
- **Scalability:** The architecture supports adding new features by introducing new subsystems or handling more users or job postings.
- **Integration Capabilities:** By isolating external integrations into a dedicated subsystem, the system can interact with external job boards and educational platforms without overburdening the basic functionalities.

6.4 Architectural Strategies

The design of the system employs several architectural strategies to ensure robustness and scalability:

6.4.1 Use of Modern Technologies

- **NextJS for front-end and back-end:** This choice has been made because of the numerous features that next provides such as component based programming and file routing. Moreover nextjs performs server-side rendering which reduces the load on client side therefore ensuring a smooth experience.
- **Python for AI-models:** Python is chosen due to its versatility. It offers a wide range of AI-focused libraries such as TensorFlow, PyTorch, Keras, and Scikit-learn, which simplify the development of machine learning and deep learning models.
- **MongoDB for Database:** MongoDB is a No SQL database which will help us in handling large amount of data. Moreover it can store different types of CV information which help us in training our models more easily. it can also handle a lot of data and is strong, making sure the data is correct and we can get it quickly.

6.4.2 Future Plans for Extension or Enhancement

The architecture is built with a modular and scalable approach, ensuring it can accommodate future enhancements and adapt to changing recruitment practices. Software design principles will guide the development of the web application, with a particular focus on SOLID principles. Specifically, the Open-

Closed Principle will be applied, allowing for extensions without requiring modifications to existing code

6.4.3 User Interface Paradigms

The application will prioritize a responsive and intuitive user interface, utilizing NextJS to ensure accessibility and ease of use for a diverse range of users. We will adhere to Shneiderman's Eight Golden Rules of Interface Design, which include:

- Ensuring consistency,
- Promoting universal usability,
- Providing informative feedback,
- Designing interactions that lead to closure,
- Preventing errors,
- Allowing easy reversal of actions,
- Keeping users in control,
- Minimizing the demand on users' short-term memory.

6.4.4 Concurrency and Synchronization

On front-end our application will utilize the `async await` functions which will hell synchronize the API request with the front-end. In addition to this parallel processing will be used at the back-end to make sure that the system can handle multiple user simultaneously and does not crash under increased traffic.

6.4.5 Version Control

To ensure that the development of the application is kept in check without conflicts we will use a collection of repositories on GitHub. Members will make pull requests and the changes will be incorporated upon approval of two other team members. This will not only clean our code but also aligns with the best industry practices.

6.5 Domain Model/Class Diagram

See Figure ?? below for Class Diagram.

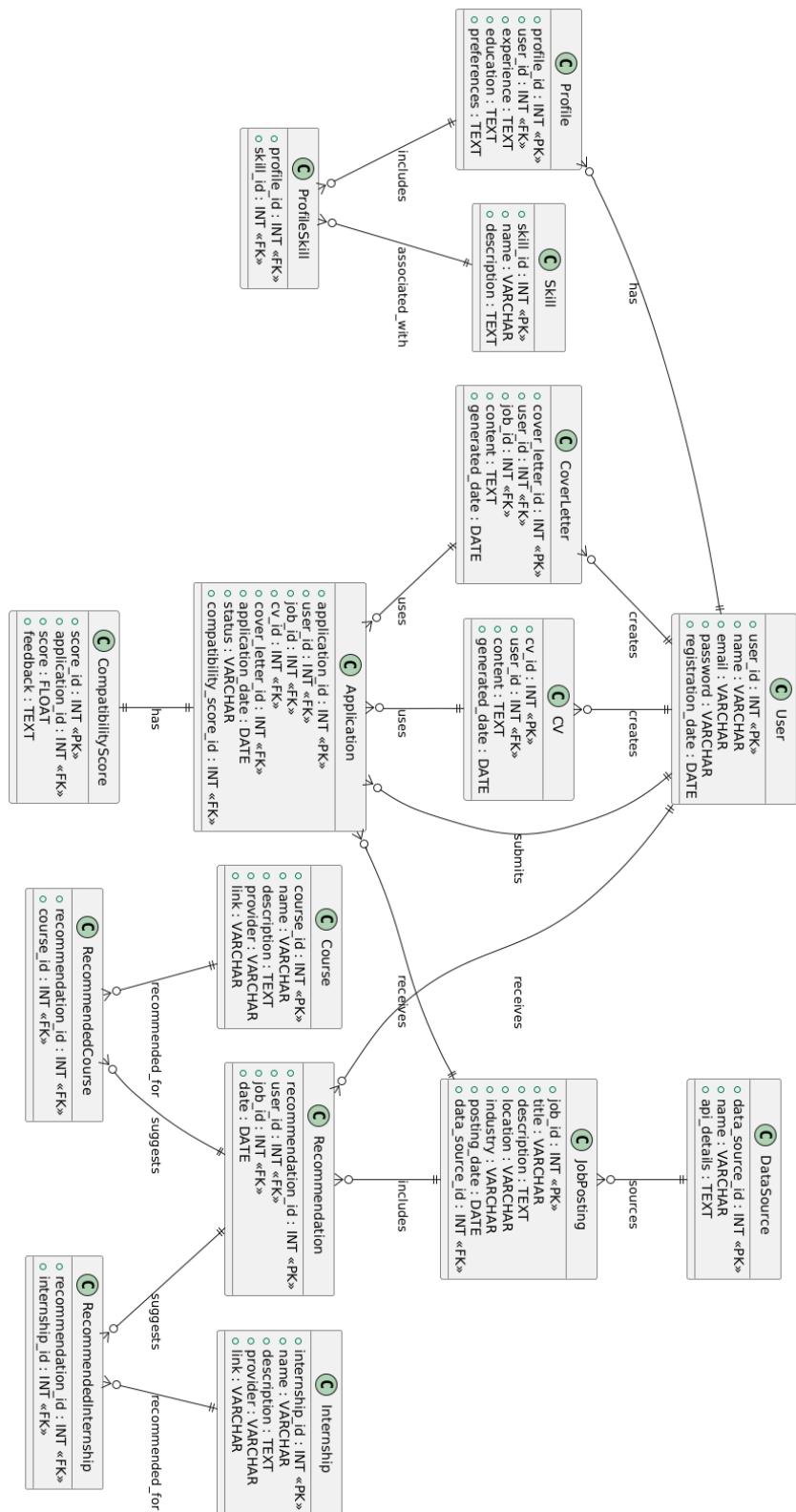


Figure 6.2: This figure represents all the entities and their relationships

6.6 Policies and Tactics

The following design policies and tactics were applied to ensure effective implementation:

6.6.1 Coding Guidelines and Conventions

The system follows modern coding standards, with a focus on clean, modular code. The front-end and back-end will be built using NextJS and the will follow JavaScript ES6 standards whereas Python PEP 8 standards for the Model training part. These guidelines will ensure consistency, readability, and maintainability of the code. Industry Conventions and best practices are employed to ensure that the code written is well documented, quality and clean code. The IDE choice shall be VS Code due to the cross-compatibility and lightweight run time.

6.6.2 System Testing

Automated unit and integration tests are employed to ensure system reliability. Test-driven development (TDD) is used during implementation to minimize bugs and ensure feature completeness. In addition to this we will also include unit testing, integration testing, and end-to-end testing. This would ensure complete testing coverage of the application at different levels. We will use both white and black box testing techniques such as decision tables for black box and white box techniques such as UI, API and Unit testing.

6.6.3 Continuous Integration and Deployment

CI/CD pipelines will be implemented to streamline code integration, automate testing, and manage deployments, improving the overall efficiency of the development process. GitHub will serve as the code repository, with regular code additions and merges into the main branch after code reviews, ensuring continuous integration into the primary code-base.

6.7 Conclusion

The AI-Powered Job Advisor is designed to be simple and adaptable. The system focuses on features like job matching, skill gap analysis, CV ranking and generation. Its layered design shows connections among all modules clearly from front-end to back-end. Overall, this design provides a good base for developing a practical and user-friendly system ready for successful implementation.

Chapter 7 Implementation and Test Cases

This chapter explains how we implemented our proposed approach. It provides a detail of the methods, technologies and tools used to develop the AI-Powered Job Advisor system.

7.1 Implementation

The key parts of our job advisor that are job recommendation engine, CV ranking system, skill gap analysis module and user interface are explained in detail.

7.1.1 System Setup and Development Environment

The following technologies were employed:

- Programming Languages: Python for the logic outside of the UI, AI/ML models; JavaScript (Next.js) for the front-end.
- Frameworks: Django REST Framework for building backend API.
- Database: MongoDB to store resumes and job descriptions and data of the users.
- Libraries:
 - NLP: SpaCy and Transformers for data parsing and the semantic analyzing.
 - AI Models: SIM-BERT for embedding generation, GANs for enhancing resumes.
 - LaTeX: For adaptive generation of the CV and the cover letter.
- Development Tools: Visual Studio Code, Git, and Jupyter Notebooks.

7.1.2 Dataset Handling and Feature Extraction

In order to provide very high quality recommendations and analysis, the system embraces CVs, job descriptions and certifications. Sophisticated methods were used to acquire significant attributes, which could be matched and analyzed properly. The process for data acquisition and feature extraction is outlined below:

CV Dataset:

CVs were collected from sources like Career Service Offices (CSOs), LinkedIn and online repositories. The first step involved in the analysis process involved data cleaning to help manage the raw data. For the purpose of text comprehension, the tool word embeddings was used to transform the resumes to a set of numerical representations with regards to word relationship between them. This step built a good

base for the next analytical procedures. After that, a Convolutional Neural Network (CNN) layer was implemented to help the model find the relations within the local structures of the resumes, for example, the patterns or details of specific sections including skills or work experience. Based on this, more Bidirectional Long Short-Term Memory (Bi-LSTM) layers were employed to capture the contextual information by capturing the contextual relationship of the words. This was made possible given that the system got to look at both an initial part of the text and a final part in order to arrive at a complete analysis of each resume. Finally the feature maps obtained from convolutional layers was fed to the dense layer with activate function known as the softmax to come up with probabilities. This way the system was able to prioritize and differentiate such information as job titles, degrees or technical skill, for the purpose of compatibility score or job skill deltas. It also provided for a multilevel analysis of resumes which provided very specific and reliable foundation for building recommendation and analysis tools for the system.

Jobs Dataset:

The process began with using crawlers on job sites to obtain updated postings and make sure the data was current with the tilts and turns within the industry. Every job advertisement was analyzed to identify relevant features such as job name, description of tasks, desired skills and experience. As a result of such analysis, word embeddings were used to convert the jobs descriptions into numerical feature vectors in which semantic meanings of the texts are represented. These embeddings were very important that serve to match the requirements of the jobs with the user profiles. CNNs were then utilized to identify localized features in the text, including finding regions of closely related keywords or skill sets. Based on this, Bidirectional Long Short-Term Memory (Bi-LSTM) layers further investigated context within the job description and how it helped better understand relations between requirements and responsibilities. It enabled the system to correctly place identified listed skills in relation to the sequence and dependency which made accurate matching possible. After the processing of the data we forwarded it through a density layer using softmax activation for classification of and prioritizing features. This constructed description of occupations helped achieve compatibility scores and matching within the recommendation system efficiently.

Courses and Certifications Dataset:

Data was collected from legitimate website sources like Coursera, EDX, and Credly only, specifically the courses and certifications offered bear more industry relevance. I was able to pull name, provider, duration and skills taught. The courses and certifications dataset was processed in the same fashion as the job description dataset with the purpose of offering users possible paths toward improving their skill set.

7.1.3 CV and Cover Letter Generation

The process of generating CVs and cover letters was automated using LaTeX with the professional layout of the latter in accordance with a specific theme. Templates saved in LaTeX were used so as to ensure that the documents were professional and they had a standard format. In order to populate these templates, scripts written in Python were employed to feed such content directly from a user-profiling system. Also, prompt engineering was applied to help the large language model (LLM) produce context-specific (job-specific) cover letters. This approach minimised the possibility of producing incorrect content by corroborating the generated text with the user input in order to ensure that the final produced text was not only correct but also topical.

7.1.4 CV Ranking

The cv ranking system works by analyzing resumes against the job description in order to provide an evaluation of the documents. Compatibility scores are obtained through cosine similarity algorithm that involves a sophisticated natural language processing of the job description and the user's profile. Furthermore, the system provides users with an overall score and ranks CVs according to the compatibility scores obtained by summing all the scores in multiple CVs.

7.2 Conclusion

Thus, job advisor system employed in our work incorporates SIM-BERT, GANs and NLP technologies to enhance services for job seekers. In general, this system reveals that the application of AI and automation in job search enhances efficiency and individuality.

Job recommendation, skill gap analysis and course recommendation are the features we will be covering in FYP-II.

Chapter 8 Experimental Results and Discussion

In the following chapter, we dive deeper into the experimentation that we carried out not only to propose a methodology but also to implement them. In that process, we discovered and developed experiments mentioned in the proposed methodologies, as well as some new ways we found along the way. We experimented with two of our modules, CV ranking and CV generation.

8.1 CV Ranking

In order to rank resumes we calculated similarity scores using cosine similarity. This helped us match the keywords used in both the job postings and the resumes.

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

For experimentation we worked on a resume dataset of 2482 resumes. For job descriptions we worked on the dataset of LinkedIn job posts between 2023-2024. They were more than 24000 industries in total. This took longer time. To test our code we reduced the industries to IT Services, Financial Services, Real Estate and Banking. This reduced the total count to 3800. From these filtered industries we worked on more than 400 job descriptions.

In order to match the resumes with job descriptions we tested 3 methods, TF-IDF (Term Frequency-Inverse Document Frequency), LDA (Latent Dirichlet Allocation) and Word Embeddings, all followed by Cosine Similarity. Figure 8.1 shows the following results.

TF-IDF although scored very low, we found it was very useful for exact keyword matching. This can be useful in domain-specific words like "TensorFlow" or "React". LDA, scoring the highest was best for thematic analysis providing valuable scores across all job descriptions showing better results on generic terms like "Data Science". Word2Vec provided with a better semantic understanding, showing correlation between related keywords and terminologies.

8.2 CV and Cover Letter Generation

In this section we will discover ways to generate resumes and cover letter programmatically. In order to make dynamic CVs we made use of latex templates and python. Since each template has its own unique set of input data fields, each template will have its own python code. LLM will be used to input descriptive data fields. The rest will be fetched from user profile. Figure 8.2 and 8.3 shows latex templates for CVs and resume.

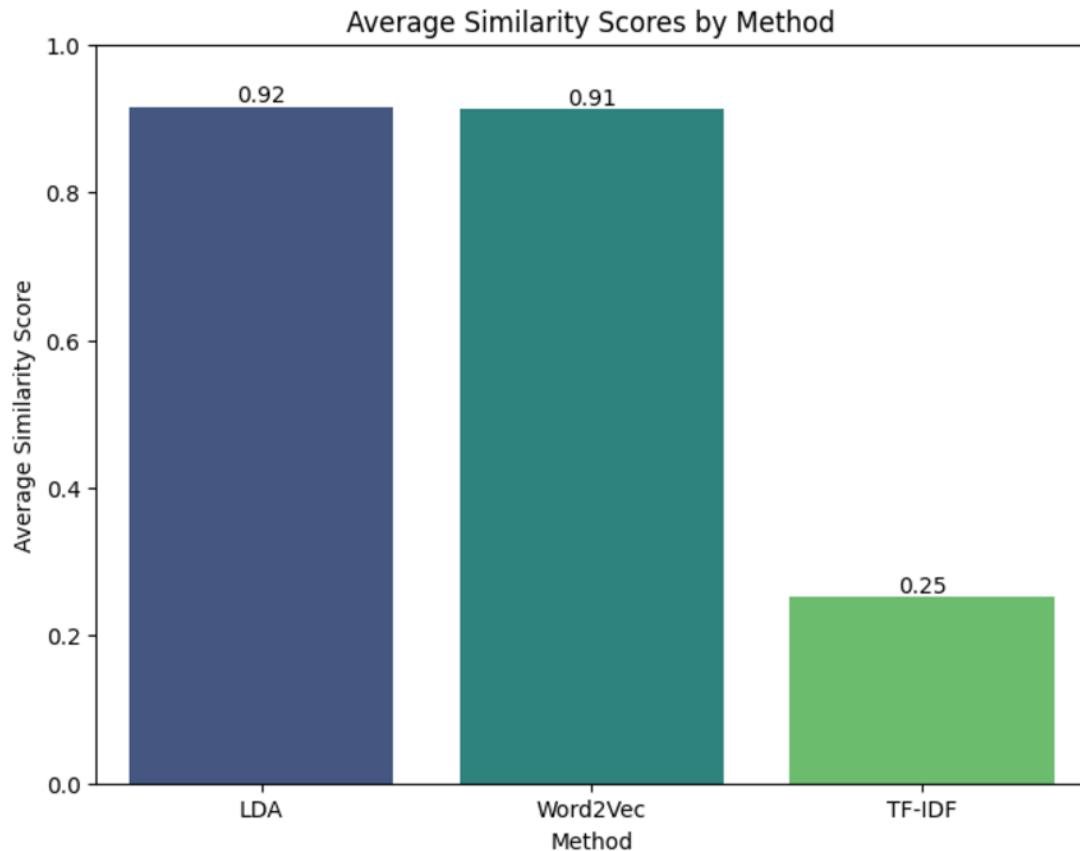


Figure 8.1: This figure represents similarity scores by methods

8.3 Conclusion

To conclude, we conducted multiple experiments for our ranking model in order to get the best results as well as ways in which we can generate resumes and cover letters dynamically. For resume ranking we found out LDA provides with a better thematic analysis for resumes based on general topics. Why TF-IDF scored so low, yet it can be useful for domain specific key-word matching. Word2Vec provided wit semantic understanding providing with diverse yet related terms. For cover letters and resumes latex templates along with separate python scripts were tried to generate CVs and cover letter. We also found that there needs to be a seperate script for each template for dynamic content. LLMs will be used for descriptive

Jane Doe

Anti Curriculum Vitae

+1 (123) 456 7898 jane.doe@gmail.com
 website janedoe.xyz github github.com/jane

KEY

- ♥✗ Heart-stab. Professional rejections and things I messed up.
- ~~> Squigly arrow. What did I learn? What were the consequences?

EDUCATION

- ♥✗ High-school: Never took German class seriously. To this day I don't speak German.
- ~~> I think I learned my lesson. I regret not having learned German, I wish I could speak to my German colleagues in their mother tongue now.

WORK EXPERIANCE

- ♥✗ Summer 2021 Rejected from XYZ.
- ♥✗ Summer 2021, didn't participate in the final round of the Alibaba math competition.
- ♥✗ Spring 2021 University research scholarship, my sloppy last minute application was rejected ~~> Don't make a last minute sloppy application. Write multiple drafts days in advance.
- ♥✗ DEF, rejected ~~> they replied and were cordial, and told me they would get back to me if they needed me in the future.
- ♥✗ Lorum Ipsum rejected me ~~> twitter DMs work better than cold emails.

SKILLS

French	Never really worked on my written french. My reading speed in French is abysmal.
German	Never took German class seriously. My speaking skills are abysmal.
Software	I struggle to get the hang of JS for a long time.

ACHIEVEMENTS AND INTERESTS

- ♥✗ Dropped out of University Club summer 2021 as the team lead. ~~> Don't bite off more than you can chew. Don't accept something just because you can.
- ♥✗ Didn't complete the 2021 DEFG international math competition final round because of self-esteem issues. ~~> Don't be scared of losing at stuff. Nothing ever comes out of self pity.
- ♥✗ Never applied to University entrance scholarship ~~> this would have been good to have, as a credential, and for money, even though the fees are not too dear here it's still easy money.
- ♥✗ I haven't played any music in years ~~> sometimes you need to make sacrifices to get what you want. You can't have everything you want at the same time. Though sad I think it was a good decision.

Figure 8.2: This figure shows a template latex CV

Firstname Lastname
Streetname 11, 0000 City
example@hotmail.com
+47 000 00 000

Company /v Person
Streetname 14, 1111 City

02.10.2022

JOB APPLICATION LETTER

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Attachments: CV, grades, certificate (2019), Id

Sincerely,

Firstname Lastname

Figure 8.3: This figure shows a template latex Cover Letter

Chapter 9 Conclusion and Future Work

A considerable amount of effort has been invested in developing the AI-powered Job Advisor system, which aims to help job seekers align their resume and skills according to the Job market and land their desired Job. The system will utilize Large Language Models and advanced NLP techniques to optimize various aspects including CV and cover letter generation, CV ranking ,skill gap analysis, and career recommendations. This document contains a detailed overview of all these techniques, their efficiency and results, including a description of the projects main modules.In addition to this the document also specifies on all the technical requirements such as hardware and software requirements, and detailed overview of the systems high-level and low-level designs using Er diagrams and Architecture diagrams.

In FYP-I the team has successfully designed and implemented a prototype featuring the two main modules of the system which are CV ranking and CV and Cover letter generation, and a web-based application with a user-friendly interface using Next.js, MongoDB and Fast API. The users will enter the relevant information during the sign-up process such as personal info, skills, education, interests, and etc. Then they can navigate the app to use the system's functionalities. This prototype will generate a CV and a Cover letter for the user which will factor in the target job and the users experience and skills. The users will also be able to see the ranking of their CVs according to the Job that they want to apply for.

Looking forward, in FYP-II the team will aim to implement the remaining two modules which are job recommendation and skill gap analysis. In addition, to this we will also improve the front-end of the application based on the feedback and ensure a smooth user-experience. The system will also go through rigorous testing phases to ensure a bug-free system ready to deployed as a refined product for our users.

In conclusion, the AI-powered Job Advisor system lays a solid foundation for creating a comprehensive and intelligent career refinement and a job-seeking tool. The development so far has provided valuable insights into the challenges and opportunities in building AI-driven career advisory systems. Future phases will focus on transforming this prototype into a fully functional and deployable solution, offering job seekers a reliable and efficient platform to enhance their career prospects.

Bibliography

- [1] Y. Du, D. Luo, R. Yan, X. Wang, H. Liu, H. Zhu, Y. Song, and J. Zhang, “Enhancing job recommendation through llm-based generative adversarial networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 8363–8371, 2024.
- [2] S. B. Zinjad, A. Bhattacharjee, A. Bhilegaonkar, and H. Liu, “Resumeflow: An llm-facilitated pipeline for personalized resume generation and refinement,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2781–2785, 2024.
- [3] V. G. Jagtap, P. Bhosale, S. Londhe, P. Meher, and P. Raut, “Resume screening and course recommendation system,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, May 2023.
- [4] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, *et al.*, “A survey on large language models for recommendation,” *World Wide Web*, vol. 27, no. 5, p. 60, 2024.
- [5] K. Tejaswini, V. Umadevi, S. M. Kadiwal, and S. Revanna, “Design and development of machine learning based resume ranking system,” *Global Transitions Proceedings*, vol. 3, no. 2, pp. 371–375, 2022.
- [6] K. Satheesh, A. Jahnavi, L. Iswarya, K. Ayesha, G. Bhanusekhar, and K. Hanisha, “Resume ranking based on job description using spacy ner model,” *International Research Journal of Engineering and Technology*, vol. 7, no. 05, pp. 74–77, 2020.
- [7] Z. Yue, S. Rabhi, G. d. S. P. Moreira, D. Wang, and E. Oldridge, “Llamarec: Two-stage recommendation using large language models for ranking,” *arXiv preprint arXiv:2311.02089*, 2023.
- [8] A. Najjar, B. Amro, and M. Macedo, “An intelligent decision support system for recruitment: resumes screening and applicants ranking,” *Informatica*, vol. 45, no. 4, 2021.

- [9] A. Khalid, K. Lundqvist, A. Yates, and M. A. Ghzanfar, “Novel online recommendation algorithm for massive open online courses (nor-moocs),” *Plos one*, vol. 16, no. 1, p. e0245485, 2021.
- [10] Y. Ouyang, H. Long, R. Gao, and J. Liu, “Course recommendation model based on layer dropout graph differential contrastive learning,” *IEEE Access*, 2024.
- [11] Aditya Babbar, Manu J, “Resumonk.” <https://www.resumonk.com>, 2012. Accessed: 2024-10-09.
- [12] Reid Hoffman, Konstantin Guericke, Allen Blue, Eric Ly, Jean-Luc Vaillant, “Linkedin.” <https://www.linkedin.com>, 2003. Accessed: 2024-10-09.
- [13] Ian Siegel, Ward Poulos, Joe Edmonds, Will Redd, “Ziprecruiter.” <https://www.ziprecruiter.com>, 2010. Accessed: 2024-10-09.