Faculty of Engineering

Department of Informatics Engineering

Software and Information System Engineering



AI-Powered Recruitment Platform Development

A senior project report - submitted to complete the requirements

For obtaining a bachelor's degree in informatics engineering

Prepared by

Ali Hussain Alaswad

Supervised by

Dr. Kadan Jumma

Eng.Anas Abdulaziz

2024-2025

كليـــــــــــــة الهندســـــــــــــــة
قســـــــم الهندســـــة المعلـوماتيـــة
هندسة البرمجيات ونظم المعلومات

تطوير منصة توظيف مدعومة بالذكاء الاصطناعي

تقرير مشروع تخرج - مُقدَّم لاستكمال متطلبات الحصول على درجة البكالوريوس في هندسة المعلوماتية

إعداد

علي حسين الأسود

إشراف

د. كادان جمعة

م.أنس عبدالعزيز

2024-2025

SUPERVISION CERTIFICATION

Supervisor Certification

I Certify that the preparation of this project entitled **Develop A Recruitment Platform Powered by AI Techniques**

Prepared by **Ali Hussain Alaswad** was made under my supervision at Faculty of Informatics Engineering in partial Fulfillment of the Requirements for the Degree of Bachelor of Software and Information system Engineering

Name: …….

Signature: …….

رسالة شكر

الى أبي :

يا سندي الأول وأبي العزيز..
أنت الرجل الذي علمني أن العزم يصنع المعجزات. لقد رأيتُ في عينيكَ تحديات الحياة، لكنني لم أرَ فيها يوماً استسلاماً. شكراً لأنك جعلت التعليم أولويةً رغم الصعاب، ولأنك علمتني أن الفشل مجرد محطةٍ لا نهاية. كل خطوة في جامعتي كانت ببركة دعائك وسندك.

دعائي لكَ:

"اللهم أطِل عمره في صحّةٍ ورضا، وارزقه برّي. اللهم اجعله من عبادك الشاكرين، وارزقه لذّة النظر إلى وجهك الكريم في الجنة مع الصديقين".

الى أمي:

أمي الغالية..
كيف أبدأ شكري وأنتِ من علّمتني حروف الامتنان؟ كنتِ السند الذي يحمل همومي قبل أن أُحمّلها، والنور الذي يضيء دربي في أصعب اللحظات. لا أنسى ليالي سهركِ وأنتِ تراقبين مذاكرتي، ولا دموعكِ الخفية حين رأيتِني أتعب. كل نجاحٍ حققته هو ثمرةٌ من روحكِ الطاهرة.

دعائي لكِ:

"اللهم ارحمها كما ربتني صغيرًا، واغفر لها كما ضحت من أجلي. اللهم أظلها تحت عرشك يوم لا ظل إلا ظلك، واجعلها من اللواتي يسمعن: {ادْخُلِي الْجَنَّةَ بِمَا كُنْتِ تَعْمَلِينَ}".

الى أخي :

يا أخي الحبيب، يا فلذة كبدي..
لقد كنتُ محظوظاً لأن الله منحني أخاً يجسد معنى "الصديق والأخ والصاحب". شكراً لكل محادثةٍ منتصف الليل حين كنتُ أتشكك في نفسي، ولكل ضحكةٍ مزّقتْ صعوبة الامتحانات. لن أنسى كيف كنتَ تدفع عني هموم الدراسة بمواقفك الطيبة.

دعائي لكَ:

"اللهم احفظه من كل مكروه، واجعله قرة عينٍ لوالديه. اللهم ارزقه عملاً صالحاً يقرّبه إليك، واجعلنا في الجنة رفقاء كما كنا في الدنيا إخوة".

الى اهل أبي و أمي:

اعمامي، عماتي، اخوالي، خالاتي..
لقد كان دعمكم كالمطر الهادئ الذي يروي الزروع دون ضجيج. شكراً لاستقبالكم الدافئ حين أتعب، ولأسئلتكم الحنونة عن دراستي، ولصلاتكم التي كانت خلف نجاحي. أنتم الجذور التي جعلت شجرتنا أقوى.

الى صديقي وافي أحمد:

وافي يا أخي ورفيق دربي،
أنت الذي جسّدت معنى الاسم "وفاء" بكل حرفٍ فيه! لولا صبرك على ساعات العمل الطويلة، وتفانيك في مراجعة البيانات، وتضحياتك التي قدّمتها دون تردد، لما رأى مشروعنا النور. كنتَ العين التي ترى الأخطاء التي لا أراها، واليد التي تسندني حين أتعثر. شكراً لأنك جعلت التحدّي مغامرةً جميلةً لا عبئاً ثقيلاً.

دعائي لكَ:

"اللهم اجعله من عبادك المُخلِصين، وارزقه عملاً يُقرّبه إليك. اللهم اكتب له التوفيق حيثما حلّ، وارزقه رفيقاً في الحياة كما كان رفيقاً لي في العلم. واجمعنا في جنّتك كما جمعتنا في العمل".

الى صديقي محمد البلخي :

محمد يا صديقي العزيز،
كلمات الشكر لن توفيك حقك.. أنت الركن الذي استندت إليه في أصعب لحظات المشروع، كان ذكاؤك الحادّ وإصرارك على التميز مصدر إلهامٍ لي. لا أنسى تلك الليالي التي قضيناها نناقش التفاصيل التقنية، ونبحث عن حلولٍ للإشكاليات المستعصية، وكيف كنتَ تبثُّ فيّ الروح حين كنتُ أشعر بالإحباط. لقد كنتَ أكثر من صديق؛ كنتَ الشريكَ الذي يُقسّم التعب ويُضاعف الأمل.

دعائي لكَ:

"اللهم ارفع قدره في الدنيا والآخرة، واجعل عمله هذا في ميزان حسناته. اللهم وفّقه في كل خطوة، واجعل مشاريعه جسراً لخدمة الإسلام والعلم. واحفظه كما حفظني بصدقه وإخلاصه".

الى اهل ابي واهل امي :

أعمامي، عماتي، أخوالي، خالاتي..
لقد كان دعمكم كالمطر الهادئ الذي يروي الزروع دون ضجيج. شكراً لاستقبالكم الدافئ حين أتعب، ولأسئلتكم الحنونة عن دراستي، ولصلاتكم التي كانت خلف نجاحي. أنتم الجذور التي جعلت شجرتنا أقوى.

الى الدكتور كادان جمعة مشرف المشروع :

سعادة الدكتور الفاضل كادان..
كلمات الشكر لا تكفي فضلك، فأنت من علمتني أن البحث العلمي ليس أرقاماً وحسب، بل أمانةٌ وضمير. شكراً لساعاتك التي وهبتنا إياها خارج الدوام، ولصبرك على أخطائي، ولإيمانك بقدراتي حين شككتُ في نفسي.

الى المهندس أنس عبدالعزيز مشرف المشروع :
سيدي المهندس أنس..
نصائحك العملية كانت جوهرةً في تخصصي. شكراً لأنك لم تبخل بخبراتك، ولأنك علمتني أن المهندس الناجح هو من يبني بأخلاقه قبل حساباته.

الى عميد الكلية الدكتور مهيب النقري :
سعادة العميد الدكتور مهيب النقري..
بقيادتكم الحكيمة، تحوّلت كليتنا إلى بيتٍ للعلم والأخلاق. شكراً لسيادتكم على سياسة الباب المفتوح، والحرص على سمعة طلابكم في سوق العمل.

ABSTRACT

The **AI-Powered Recruitment Platform** is designed to optimize the hiring process by leveraging **AI-driven job recommendations, automated candidate assessments, and intelligent networking suggestions**. It enhances recruitment workflows through **uploading their CV, Connect with Potential Employers, job matching, and social interaction features**, improving efficiency for both employers and job seekers.

Traditional recruitment methods often rely on **manual screening and keyword-based searches**, leading to inefficiencies and hiring delays. This platform addresses these challenges using **Natural Language Processing (NLP) and AI-based ranking** to analyze candidate profiles, job descriptions, and market trends. Additionally, **AI-powered assessments** to ensure objective candidate evaluation.

The system supports **two type of users job seekers, and companies**, offering features such as **profile management, job applications, job postings, and AI-driven recommendations**. The **backend is built with Django and Django REST Framework (DRF)** for scalability and security, while the **frontend uses ReactJS** for a dynamic user experience. AI functionalities integrate **pre-trained NLP models** rather than custom machine learning models.

This documentation provides a **detailed overview of the system's architecture, features, and implementation**. A **comparative study of existing recruitment platforms** highlights how this solution **improves hiring efficiency, enhances job matching accuracy, and provides a seamless experience** for job seekers and employers.

ملخص

تم تصميم **منصة التوظيف المدعومة بالذكاء الاصطناعي** لتحسين عملية التوظيف من خلال **التوصيات الذكية للوظائف، التقييمات الآلية للمرشحين، واقتراحات التواصل الذكية** .تعزز المنصة سير عمل التوظيف عبر **رفع السيرة الذاتية، والتواصل مع أصحاب العمل المحتملين، التوافق الوظيفي، وميزات التفاعل الاجتماعي**، مما يزيد من كفاءة التوظيف لكل من أصحاب العمل والباحثين عن عمل.

تعتمد أساليب التوظيف التقليدية على **المراجعة اليدوية والبحث بالكلمات المفتاحية**، مما يؤدي إلى عدم الكفاءة وتأخير التوظيف. تعالج المنصة هذه المشكلات باستخدام **معالجة اللغة الطبيعية (NLP) وآليات الترتيب الذكية** لتحليل ملفات المرشحين، أوصاف الوظائف، واتجاهات السوق. بالإضافة إلى ذلك، تضمن **التقييمات المستندة إلى الذكاء الاصطناعي والمقابلات الآلية** تقييمًا موضوعيًا للمرشحين.

يدعم النظام **نوعين من المستخدمين الباحثين عن العمل، والشركات**، ويوفر ميزات مثل **إدارة الملفات الشخصية، التقدم للوظائف، نشر الوظائف، والتوصيات الذكية** .تم بناء **الواجهة الخلفية باستخدام Django وDjango REST Framework (DRF)**لضمان الأمان وقابلية التوسع، بينما تعتمد **الواجهة الأمامية على ReactJS** لتوفير تجربة مستخدم ديناميكية. يتم دمج وظائف الذكاء الاصطناعي باستخدام **نماذج معالجة لغة طبيعية (NLP) مدربة مسبقًا** بدلاً من نماذج تعلم آلي مخصصة.

توفر هذه الوثيقة **نظرة تفصيلية على بنية النظام، الميزات، والتنفيذ التقني** .كما تسلط **دراسة مقارنة للمنصات الحالية** الضوء على كيفية تحسين هذا الحل **لكفاءة التوظيف، دقة التوافق بين الوظائف والمرشحين، وتجربة المستخدم بشكل عام** لكل من الباحثين عن عمل وأصحاب العمل.

TABLE OF CONTENT

SUPERVISION CERTIFICATION iii

رسالة شكر 0

ABSTRACT 4

ملخص 5

TABLE OF CONTENT 6

List of tables 8

List of Figures 9

List of abbreviations 10

Chapter1 Introduction 12

1. Introduction: 13
2. Problem Definition: 13
3. Project Structure: 14
4. *Report Structure*: 15

Chapter2 Basic Concepts and Literature Review 16

1. Introduction: 17
2. Basic concepts: 17
3. Literature review: 21

Chapter3 Project Management 25

1. Introduction: 26
2. Project charter: 26
3. The SOW document: 28
4. Project plan – Gantt chart: 32
5. Risk Management: 33
6. Summary: 34

Chapter4 System Analysis 35

1. Introduction: 36
2. Requirements Elicitation: 36
3. SRS Document: 43
    1. Introduction: 43
    2. Overall description and product features: 45
    3. Nonfunctional requirements: 53

4. Conclusion:.....55
4. Requirements modeling:.....56
5. Initial Test Cases:.....79
6. Initial RTM Requirements Tractability Matrix:.....93
Chapter5 System Design.....97
1. Introduction:.....98
2. System Architecture:.....98
System decomposition - Component diagram.....100
3. Detailed design for the system components:.....114
Used design patterns:.....120
Best Practices Implemented:.....126
4. Database Design:.....127
5. Updating the RTM:.....128
Chapter6 Practical Implementation.....136
1. Introduction:.....137
2. Used Tools:.....137
3. AI Used Technologies and Tools:.....139
4. System Interfaces:.....144
5. Test Cases Execution:.....155
Testing the AI components - Experiment on Real Data.....175
6. RTM Last Version:.....179
Chapter7 Future Prospects.....190
1 Introduction.....191
2 Proposed Future Enhancements.....191
3 Conclusion.....193
Chapter8 Conclusion.....194
1. Introduction:.....195
2. Summary of Achievements:.....195
3. Addressing the Challenges of Traditional Recruitment.....196
4. System Reliability and Performance.....196
5. Conclusion.....197
6. References:.....198

List of tables

Table 1 List of abbreviations....10
Table 2 Summary of reference study....24
Table 3 Project Charter....26
Table 4 Statement of work table....29
Table 5 Risk management....34
Table 6 Requirements Database....36
Table 7 Register Usecase Specification....57
Table 8 Login Usecase Specification....59
Table 9 Manage Social Interaction Usecase Specification....61
Table 10 Manage Recruitment Usecase Specification....64
Table 11 Apply for job Usecase Specification....66
Table 12 Manage Account Settings Usecase Specification....68
Table 13 Table 12 Manage Account Profile Usecase Specification....69
Table 14 Manage Setup Profile Usecase Specification....70
Table 15 Register Initial Test Case....79
Table 16 Login Initial Test Case....80
Table 17 Manage Social Interaction Initial Test Case....81
Table 18 Manage Recruitment Initial Test Case....83
Table 19 Apply For Job Initial Test Case....84
Table 20 Manage Account Settings Initial Test Case....86
Table 21 Manage Profile Initial Test Case....87
Table 22 Setup Profile Initial Test Case....89
Table 23 Initial RTM....93
Table 24 RTM In Design Chapter....135
Table 25 Test Cases Execution....155
Table 26 RTM Final Version....179

List of Figures

Figure 1 Gant Chart 1 32
Figure 2 Gant Chart 2 32
Figure 3 High-level Use-case Diagram 56
Figure 4 View Home Page Detailed Use-case Diagram 60
Figure 5 Manage Recruitment Detailed Use-case 64
Figure 6 Apply for Job Detailed Ucase-case 66
Figure 7 Manage Account Detailed Use-case 68
Sequence diagram: Figure 8 Sequence Diagram – Manage Recruitment 74
Figure 9 Sequence Diagram – Apply For Job 75
Figure 10 Activity Diagram – Apply For Job 76
Figure 11 Analysis Class Diagram 77
Figure 12 Entity Relationship Diagram 78
Figure 13 Component Diagram 100
Figure 14 System Architecture 108
Figure 15 User Management - Details Class Diagram 115
Figure 16 Job Management - Detailed Class Diagram 116
Figure 17 Posts Management – Detailed Class Diagram 117
Figure 18 AI Recommendation - Detailed Class Diagram 118
Figure 19 Notification - Detailed Class Diagram 119
Figure 20 Post Management – Singleton 121
Figure 21 Strategy Pattern 122
Figure 22 Observer Patterns – Notification Management 123
Figure 23 MySQL - Database Modeling 127
Figure 24 Implementation - Register Interface 144
Figure 25 Implementation - Login Interface 144
Figure 26 Implementation - Home Page Interface 145
Figure 27 Implementation - Profile Interface 145
Figure 28 Implementation - Edit Profile Dialog Interface 146
Figure 29 Implementation - Edit Privacy Settings Interface 147
Figure 30 Implementation - Job Application List Interface 147
Figure 31 Implementation - Job Application Details Interface 148
Figure 32 Implementation - Quiz Interface 148
Figure 33 Implementation - Ban Interface 149
Figure 34 Implementation - Quiz Result Interface 149
Figure 35 Implementation - View Applicants Interface 150
Figure 36 Implementation - Job Application Form Interface 151
Figure 37 Implementation - Saved Jobs Interface 151
Figure 38 Implementation - Private Profile View Interface 152
Figure 39 Implementation - Follow Request Notification For a Private Profile Interface 152
Figure 40 Implementation - Password Reset Password Interface 153
Figure 41 Implementation - Password Reset Interface 153
Figure 42 Implementation - Reset Password Link by Email Interface 154
Figure 43 Implementation - Job Offer by Email Interface 154

List of abbreviations

Abbreviation	Definition
API	Application Program Interface
NLP	Natural Language Processing
AI	Artificial Intelligence
UML	Unified Modeling Language
NoSQL	Not only Structured Query Language.
SQL	Structured Query Language.
SOW	Statement of Work
HireHub	AI-Powered Recruitment Platform
LLM	Large Language Model

Table 1 List of abbreviations

Chapter1 Introduction

1. Introduction:

In this chapter, we will introduce our project, discussing the main issues and reasons for building this system. We will also explain the objectives and goals we aim to accomplish with this system. Finally, we will provide an overview of the report structure.

2. Problem Definition:

Recruitment is a fundamental yet complex process that involves multiple stages, including **resume screening, candidate evaluation, and selection**. Traditional hiring methods often suffer from inefficiencies, making it challenging for recruiters to find the right candidates quickly and effectively. One major issue is the **time-consuming nature of manual resume screening**, which slows down the hiring process and increases operational costs. Additionally, human-driven candidate selection is prone to **bias**, leading to potential unfairness in hiring decisions and limiting diversity within organizations. Another key challenge is **ineffective job matching**, where job seekers receive irrelevant job suggestions due to poor filtering mechanisms and a lack of AI-driven personalization. Furthermore, recruiters often struggle with **limited candidate evaluation tools**, making it difficult to assess a candidate's skills beyond what is written on their resume. The **user experience** on

many existing recruitment platforms is also problematic, as job seekers frequently encounter **irrelevant job postings, lack of engagement features, and inefficient application processes**. Addressing these challenges requires an intelligent, automated system that enhances **job recommendations, candidate assessments, and recruitment efficiency**.

3. Project Structure:

HireHub aims to **enhance and streamline the hiring process** by leveraging **AI-driven recommendations, automated skill assessments, and intelligent networking features**. The primary objective is to **automate job matching** through advanced **AI recommendation algorithms**, ensuring that job seekers receive relevant job opportunities based on their skills, experience, and preferences. The system will also **improve candidate evaluation** by integrating **resume submitting and AI-based assessments**, allowing recruiters to make data-driven hiring decisions. Another critical goal is to **reduce hiring time and increase efficiency** by automating repetitive tasks such as resume screening and initial candidate filtering. Additionally, the platform will **enhance professional networking** by suggesting relevant **connections** based on industry trends, mutual interests, and professional backgrounds. To ensure security and scalability, the system will implement **role-based authentication and secure API integration**. Lastly, a key focus of the project is to provide a **user-friendly interface**, utilizing **ReactJS for the frontend and Django for**

the backend, ensuring seamless interaction for both job seekers and recruiters. Through these objectives, the platform aims to **optimize the recruitment process, improve hiring accuracy, and create a better experience for all users involved**.

4. *Report Structure*:

The report will be organized in the following chapters:

- Chapter 1: introduction.
- Chapter 2: Basic concepts and literature review.
- Chapter 3: project management.
- Chapter 4: System analysis.
- Chapter 5: design study for the system
- Chapter 6: practical implementation.
- Chapter 7: Future Prospects.
- Chapter 8: Conclusion.

Chapter2 Basic Concepts and Literature Review

1. Introduction:

In this chapter, we will provide an overview of the main concepts and tools used and required in this project, along with a reference study of similar systems and tools. This foundational knowledge is essential for understanding the technical and theoretical basis of our approach and how it has been informed by existing solutions in the field.

2. Basic concepts:

HireHub integrates several key functionalities to streamline and optimize the hiring process. Some of the fundamental concepts that underpin the system include:

- Hiring Process:

  The recruitment process generally involves **job postings, resume uploads, candidate assessments, and final selection**. AI enhances each stage by **automating job recommendations and ranking candidates** based on **job title, required skills, and job description in relation to the candidate's skills and experience.**

- AI-Based Job Recommendations:

  AI analyzes candidate profiles, job descriptions, to suggest the most relevant job opportunities. This feature is inspired by platforms like LinkedIn and indeed, which use AI-driven recommendation engines.

- Social Interaction in Recruitment:

  Modern hiring platforms, such as LinkedIn, encourage professional networking by allowing users to connect, engage in discussions, and share industry-related content. Social interaction plays a key role in talent acquisition and employer branding.

- Sentence Transformers in Job Recommendations:

  Natural Language Processing (NLP) models, such as sentence transformers, help in analyzing the job descriptions to improve job matching accuracy. These models extract relevant information from candidate profiles and rank job listings based on skill alignment.

- Authentication and Authorization:

  Secure access control is essential for recruitment platforms. The system implements JWT-based authentication and role-based access control (RBAC) to ensure data security and privacy.

- Sentence Transformers Integration:

  The use of Sentence Transformers with the 'all-MiniLM-L6-v2' pre-trained model enables intelligent job matching through text embeddings and similarity calculations. The system converts job descriptions and user profiles into high-dimensional vectors, allowing for precise matching based on semantic similarity rather than simple keyword matching. This approach significantly reduces the need for manual filtering by:

- Converting job descriptions and requirements into meaningful numerical representations
- Transforming user skills and experience into comparable vector formats
- Calculating cosine similarity scores to determine the best matches between jobs and candidates
- Providing a score from 0-100 to indicate match quality
- This implementation offers a more focused and efficient solution specifically tailored for the recruitment matching process, making the job matching process both automated and intelligent.

- Large Language Models (LLMs) :

  Large Language Models (LLMs) are advanced AI models designed to generate human-like text based on given prompts. In this project, we utilize a local LLM (inspired by llama-3.2-3b-instruct) to generate domain-specific multiple-choice quizzes that assess candidates' skills.

  The system constructs a structured prompt, detailing job-specific requirements, skill levels, and knowledge areas. The LLM then generates a JSON-formatted response with quiz questions, ensuring alignment with the role's required expertise.

Key Characteristics of LLMs in this System:

- Dynamic Quiz Generation: The LLM creates quizzes tailored to the experience level of job applicants.
- Structured JSON Output: The model responds with a well-formed structure containing 10 multiple-choice questions and 4 answer choices per question.
- Automated Evaluation: AI-generated quizzes are later used to evaluate candidates efficiently.

- Embedding-Based Semantic Analysis

  Embedding models convert textual data into numerical representations, allowing similarity computations. The SentenceTransformer (all-MiniLM-L6-v2) model is used in this project to analyze job descriptions and candidate profiles.

- This approach ensures:
  - Accurate Job Matching: By embedding both job descriptions and candidate data into vector space, we measure their similarity using cosine similarity.
  - Text Normalization: Before embedding, the text is standardized (lowercased, whitespace-trimmed, and cleaned) to maintain consistency.

- The CV parsing feature
  - automates the extraction of key information from unstructured CV documents, transforming them into a structured, machine-readable format like JSON. This streamlines the recruitment process by automating data entry and making candidate information easily searchable and manageable. HireHub's implementation leverages advanced multi-modal Large Language Models (LLMs), specifically Google Gemma via the OpenRouter API. This involves converting PDF CVs into images, which the LLM then "reads" and parses to extract details such as candidate name, contact information, skills, work experience, and education. The extracted data is then used to automatically update the user's profile on the platform, ensuring efficiency and accuracy in candidate management.

3. Literature review:

A comparative study of existing recruitment platforms was conducted to analyze their strengths, limitations, and unique features. The platforms reviewed include LinkedIn, Indeed, HackerRank, HireVue, and Vervoe.

LinkedIn

A professional networking platform that utilizes AI for job recommendations, connection suggestions, and learning content recommendations. However, its recruitment features focus more on networking rather than comprehensive AI-driven hiring solutions.

Indeed: A job search platform that uses AI to enhance job recommendations and employer insights. However, it lacks advanced AI-driven candidate assessments and skill-based evaluations.

HackerRank

Primarily focused on technical hiring, it offers skill-based coding assessments but does not provide generalized AI-based job matching.

HireVue

Specializes in AI-powered video interviews, utilizing speech analysis and facial recognition to assess candidates. While useful, its AI models raise ethical concerns related to bias in automated assessments.

Vervoe

Focuses on AI-driven skill assessments but lacks networking features and AI-based job recommendations, making it less effective for holistic talent acquisition.

Numerous studies and research papers support the methodologies used in this project:

Advancements in NLP for Content Generation

Research demonstrates that LLMs can generate domain-specific content accurately, including quizzes, structured assessments, and interview

questions. Studies show that **LLMs outperform rule-based question generators** in terms of contextual relevance and adaptability to different industries.

Embedding-Based Semantic Analysis

State-of-the-art embedding models, such as **SentenceTransformer**, have been widely adopted in recommendation systems, chatbots, and intelligent search engines. In recruitment, these models help **match candidates based on skills and experience**, overcoming the limitations of traditional keyword-based approaches.

Applications in Recruitment

AI-powered recruitment tools have seen growing adoption in industry-leading platforms like **LinkedIn and Indeed**, but existing systems lack:

- **Real-time skill assessment** (our project addresses this with AI-generated quizzes).
- **Effective embedding-based candidate-job matching** (our system enhances this using weighted text embeddings).

The methodologies applied in this project align with modern AI-driven hiring solutions, making it a **robust, research-backed approach** to recruitment automation.

The Summary of the reference study:

Your proposed recruitment platform aims to integrate the best aspects of these platforms while addressing their limitations. Below is a comparison table that highlights where your system stands out:

Table 2 Summary of reference study

System / Feature	Job Recommendations	Connection Suggestions	AI for Skill Assessment	Customizable Hiring Process	Content & Learning Recommendations	Video Interviews
LinkedIn	✔	✔	✗	✗	✔	✗
Indeed	✔	✗	✗	✗	✗	✗
HackerRank	✗	✗	✔	✔	✗	✔
HireVue	✗	✗	✗	✔	✗	✔
Vervoe	✗	✗	✔	✔	✗	✔
Our System	✔	✔	✔	✔	✔	✔

Chapter3 Project Management

1. Introduction:

In this chapter, we will dive into the management phase of the project, which is a critical aspect of ensuring the project's success. We will examine the project charter, project plan, Statement of Work (SOW) document, stakeholder analysis, and risk management strategies to effectively guide and control the project from initiation to completion.

2. Project charter:

A project charter is a formal document that serves as an official authorization for the start of a project. It acts as a reference point throughout the project, providing a clear understanding of the project's purpose and establishing a foundation for decision-making and project governance.

Table 3 Project Charter

Project title	*AI-Powered Recruitment Platform*
Project start date	*September – 21 – 2024*
Project finish date	*July – 12 – 2025*
Project manager	*Dr.Kadan Jumma Eng.Anas Abdulaziz*

Project objectives	*The main objectives of this project are to develop an AI-powered recruitment system that improves job matching accuracy, resume parsing, and enhances candidate assessments. The platform will also provide secure authentication and a user-friendly interface to improve the experience for both job seekers and recruiters.*
Approach	1. *The project follows a **structured development methodology**, ensuring a systematic progression from **conceptualization to deployment**. The key phases are:* 2. ***Determining the scope of the project**, defining essential features, user roles, and system capabilities.* 3. ***Conducting a literature review** to analyze similar recruitment platforms, identifying their strengths, limitations, and best practices.* 4. ***Performing requirements analysis**, gathering **functional and non-functional requirements** based on industry needs and user expectations.* 5. ***Designing the system architecture**, including defining the **database structure, API endpoints, and AI-driven recommendation mechanisms**.* 6. ***Developing the backend** using **Django REST Framework**, ensuring secure authentication, API endpoints, and business logic implementation.* 7. ***Developing the frontend** using **ReactJS**, focusing on creating an intuitive and responsive UI for job seekers and recruiters.* 8. ***Integrating AI components** for job recommendations and resume uploading, utilizing **pre-trained NLP models and AI-driven ranking algorithms**.*

	9. ***Testing the system***, *including* ***functional testing, performance evaluation, and security assessments*** *to ensure reliability.* 10. ***Deploying the platform*** *to a cloud-based or server-hosted environment, making it publicly accessible.* 11. ***Documenting all phases of the project***, *including architecture, API documentation, system workflows, and security protocols.*

Roles and responsibilities:

Name	*Role*	*Responsibility*
Dr.Kadan Jumma Eng. Anas Abdulaziz	*Project manager – supervisor*	*Project management and work monitoring*
Ali Hussain Alaswad	*Engineer*	*Back-end development – Front-end Development - documentation – AI components - analysis and design phase team leader.*

3. The SOW document:

Statement of Work is a comprehensive document that defines the scope of work for a project. It outlines the specific tasks, deliverables, timeline, and responsibilities. The SOW document provides a clear understanding of what needs to be accomplished, the project's objectives, and the criteria for success.

Table 4 Statement of work table

Project Title:	*AI-Powered Recruitment Platform*
Project Description and Objectives:	*This project aims to develop an **AI-powered recruitment platform** that enhances the hiring process by integrating **automated job recommendations, AI-driven resume parsing, and intelligent candidate assessments**. The platform will offer a **scalable, secure, and user-friendly** experience for both job seekers and recruiters. The main objectives include:* *- Automating **job matching** and improving **candidate-job fit** using AI-based ranking.* *- Enhancing **candidate assessments** with AI-driven **resume parsing**.* *- Providing **secure authentication and role-based access control** for different user types.* *- Offering an **interactive and dynamic UI** for a seamless user experience.*
Project Scope:	***In Scope:*** *- Developing a **web-based** recruitment platform with **backend and frontend integration**.* *- Implementing **AI-powered job recommendations** using **pre-trained NLP models**.* *- Ensuring **secure authentication (JWT-based) and role-based access control**.* *- Developing **job posting, candidate profile management, and application tracking features**.*

	Out of Scope: *- Developing a custom **machine learning model** from scratch.* *- Integration with **third-party recruitment platforms**.*
Project goals:	▪ *The project is designed to meet the following goals:* ☑ ***Improve Hiring Efficiency**: Reduce the time and effort required to **screen and shortlist** candidates by automating the matching process.* ☑ ***Enhance Job Recommendations**: Provide **personalized job suggestions** for candidates based on AI-driven ranking and NLP-based analysis.* ☑ ***Increase Recruiter Productivity**: Enable recruiters to **post jobs, track applicants, and assess resumes** with minimal manual effort.* ☑ ***Secure and Scalable System**: Implement a **role-based access control system** with **secure authentication** to protect user data and ensure privacy.* ☑ ***User-Friendly Experience**: Develop a **modern and intuitive UI** that allows job seekers and employers to interact effortlessly with the platform.*
Project Deliverables:	*The expected deliverables include:* *- A **fully functional AI-powered recruitment platform** with job matching and resume parsing.* *- A **secure authentication system** supporting **role-based access control**.* *- A **frontend UI** that ensures an intuitive and engaging user*

	experience. *- **API documentation** for system integration and scalability.* *- A **final project report** detailing design, development, and implementation.*
Technology and Tools:	*Programming Languages: Python, JavaScript – html - CSS.* *Frameworks: Django for backend, React for frontend* *Database: MYSQL.*
Assumptions:	*- The platform will rely on **pre-trained AI models** for job matching.* *- Users will have **internet access** to interact with the system.* *- The authentication system will support **multi-role access (job seekers, recruiters, and admins)**.*
Project Resources	*Human Resources:* *Dr.Kadan Jumma* *Eng.Anas Abdulaziz*
Schedules:	*Project Start Date: September 21, 2024* *First second project Seminar: April 26, 2025* *Techniqal Review : June 22, 2025* *Project Finish Date: March 08, 2025.*

4. Project plan – Gantt chart:

A document outlining tasks, deadlines, and resources needed to achieve project objectives, serving as a roadmap for successful project execution.

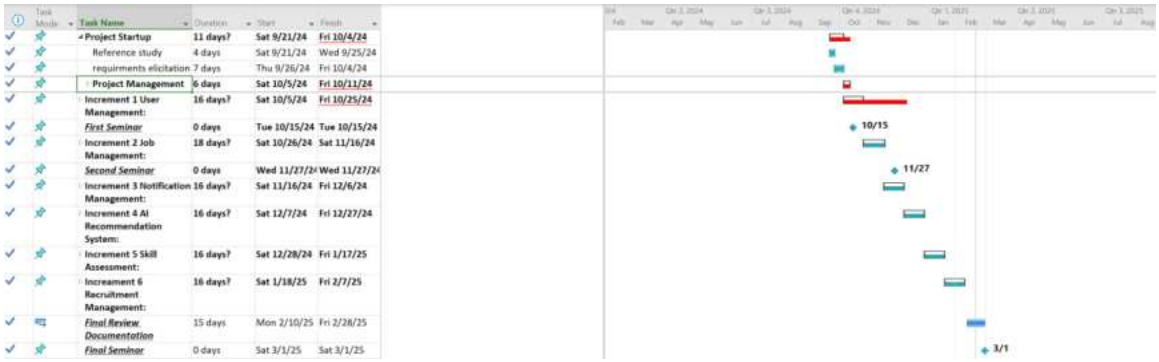


Figure 1 Gant Chart 1



Figure 2 Gant Chart 2

5. Risk Management:

Risk ID	RK-01	RK-02	RK-03	RK-04
Risk title	Single-Team Member Workload	fuzzy project scope	must learn and use new technologies	big Separation of work in the implementation stage
Risk Description	The project is being developed by a **single student**, leading to an **overwhelming workload** and potential delays in development.	The project scope is **not well-defined**, leading to **shifting requirements** and possible rework.	The project requires learning and integrating **new technologies** such as **ReactJS, Django, AI-based recommendation systems**, which may slow down progress.	A **large gap in development stages** may lead to **integration issues** and inefficiencies in implementation.
Raised Date	11/10/2024	11/10/2024	15/12/2024	25/12/2024
Tracking Frequency	weekly	daily	weekly	daily
State	active	closed	Under Mitigation	active
Impact	high	medium	high	medium
Mitigation Plan	Implement **time management strategies**, use **Waterfall methodologies**,	Establish clear project goals, maintain a well-	Allocate **dedicated learning time**, use **online tutorials/documentation**, and break tasks into	Maintain a **continuous integration approach**, ensure **regular testing**,

	and seek external guidance when needed.	documented scope with detailed functional and non-functional requirements.	**manageable learning phases**.	and align backend and frontend development through **consistent documentation**.

Table 5 Risk management

The process of identifying, evaluating, and mitigating potential risks that could impact the success of a project or a team.

6. Summary:

In conclusion, good project management is vital for successful software system development. It provides structure, ensuring projects are completed within scope and schedule.

Overall, strong project management enhances the delivery of high-quality software systems that meet the goals of the project.

Chapter4 System Analysis

1. Introduction:

In this chapter we will introduce the analytical study of the system using the needed UML diagrams for system requirements modeling.

2. Requirements Elicitation:

During the requirements elicitation process, we have taken several steps to ensure a comprehensive understanding of the project. We conducted a study and analysis of similar systems, also analyzed, and anticipated the needs of the application's users.

The resulted Requirement database:

Table 6 Requirements Database

Index	REQ-ID	Title	Category	Priority	Actor(s)
1	FR-UM-001	Register User	User Management	High	Job Seeker, Company
2	FR-UM-002	Login User	User Management	High	Job Seeker, Company
3	FR-UM-003	Logout	User Management	High	Job Seeker, Company
4	FR-UM-004	Reset Password	User Management	High	Job Seeker, Company
5	FR-UM-005	View Own Profile	User Management	High	Job Seeker, Company

6	FR-UM-006	Collect User Profile Interests	User Management	High	Job Seeker, Company
7	FR-UM-007	Add Profile and Cover Picture	User Management	High	Job Seeker, Company
8	FR-UM-008	Add User Details	User Management	High	Job Seeker, Company
9	FR-UM-009	Set Up Basic User Information	User Management	High	Job Seeker, Company
10	FR-UM-010	Update Personal and Work Information	User Management	High	Job Seeker, Company
11	FR-UM-011	Update Profile and Cover Pictures	User Management	High	Job Seeker, Company
12	FR-UM-012	Update Basic User Information	User Management	High	Job Seeker, Company
13	FR-UM-013	Manage Profile Privacy Settings	User Management	Medium	Job Seeker, Company
14	FR-UM-014	Switch to Private Profile	User Management	High	Job Seeker, Company
15	FR-UM-015	Display Locked Profile for Non-Followers	User Management	Medium	System
16	FR-UM-016	Delete User Account	User Management	Medium	Job Seeker, Company
17	FR-UM-017	Display Home Page	User Management	High	Job Seeker, Company
18	FR-UM-018	Display List of followers	User Management	Medium	Job Seeker, Company
19	FR-UM-019	Display List of following	User Management	Medium	Job Seeker, Company

20	FR-UM-020	Display Profile Summary	User Management	Medium	Job Seeker, Company
21	FR-UM-021	Display Connection Suggestions	User Management	High	Job Seeker, Company
22	FR-UM-022	Display Recommended Job Postings	User Management	High	Job Seeker, Company
23	FR-UM-023	View Other User Profiles	User Management	High	Job Seeker, Company
24	FR-UM-024	Search for Profiles	User Management	High	Job Seeker, Company
25	FR-UM-025	Create Post	User Management	High	Job Seeker, Company
26	FR-UM-026	Restrict Media Upload Sizes	User Management	Medium	System
27	FR-UM-027	Display Posts List	User Management	High	Job Seeker, Company
28	FR-UM-028	Like Posts	User Management	High	Job Seeker, Company
29	FR-UM-029	Comment on Posts	User Management	High	Job Seeker, Company
30	FR-UM-030	Like Comments	User Management	High	Job Seeker, Company
31	FR-UM-031	Delete Comments	User Management	High	Job Seeker, Company
32	FR-UM-032	Edit Comments	User Management	High	Job Seeker, Company
33	FR-UM-033	Reply on Comments	User Management	High	Job Seeker, Company

34	FR-UM-034	Like Reply	User Management	High	Job Seeker, Company
35	FR-UM-035	Delete Reply	User Management	High	Job Seeker, Company
36	FR-UM-036	Edit Reply	User Management	High	Job Seeker, Company
37	FR-UM-037	Send Follow Request	User Management	High	Job Seeker, Company
38	FR-UM-038	Accept Follow Request	User Management	High	Job Seeker, Company
39	FR-UM-039	Reject Follow Request	User Management	High	Job Seeker, Company
40	FR-UM-040	Ignore Follow Request	User Management	High	Job Seeker, Company
41	FR-UM-041	Follow a User	User Management	High	Job Seeker, Company
42	FR-UM-042	Unfollow User	User Management	High	Job Seeker, Company
43	FR-JM-001	Post Job Application	Job Management	High	Company
44	FR-JM-002	Search for Jobs Using Filters	Job Management	High	Job Seeker
45	FR-JM-003	Save Job	Job Management	Medium	Job Seeker
46	FR-JM-004	View Saved Job List	Job Management	Medium	Job Seeker
47	FR-JM-005	Unsave Job	Job Management	Medium	Job Seeker
48	FR-JM-006	Automatically Expire Job Postings	Job Management	Medium	System
49	FR-JM-007	Delete Job Application	Job Management	Medium	Company
50	FR-NTF-001	Notify Suitable Users of Job Opportunities	Notifications	High	Notification System

51	FR-NTF-002	Notify New Connection Requests	Notifications	High	Notification System
52	FR-NTF-003	Notify Private Profile Owners of Follow Requests	Notifications	High	Notification System
53	FR-NTF-004	Notify Accepted Request	Notifications	High	Notification System
54	FR-NTF-005	Post Like Notification	Notifications	High	Notification System
55	FR-NTF-006	Comment Notification	Notifications	High	Notification System
56	FR-NTF-007	Comment Reply Notification	Notifications	High	Notification System
57	FR-NTF-008	Comment Like Notification	Notifications	High	Notification System
58	FR-NTF-009	Job Offer Notification	Notifications	High	Notification System
59	FR-NTF-010	Job Offer Reminder Notification	Notifications	Medium	Notification System
60	***FR-NTF-011***	***Message Notification***	***Notifications***	***High***	***Notification System***
61	***FR-MS-01***	***Send Direct Messages***	***In-app messaging***	***High***	***In-app messaging***
62	***FR-MS-02***	***Receive Direct Messages***	***In-app messaging***	***High***	***In-app messaging***
63	***FR-MS-03***	***Edit Messages***	***In-app messaging***	***High***	***In-app messaging***
	FR-MS-04	***Delete Messages***	***In-app messaging***	***High***	***In-app messaging***
64	***FR-MS-05***	***Search for User to start Converstion***	***In-app messaging***	***High***	***In-app messaging***
65	FR-AI-001	Collect User Job Preferences	AI-Powered Recommendations	High	Recommendation System
66	FR-AI-002	Analyze User Job Preferences	AI-Powered Recommendations	High	Recommendation System
67	FR-AI-003	Recommend Jobs Based on Preferences	AI-Powered Recommendations	High	Recommendation System
68	FR-AI-004	Suggest Connections Based on Job Interests	AI-Powered Recommendations	High	Recommendation System

69	FR-AI-005	Recommend Posts Based on Preferences	AI-Powered Recommendations	High	Recommendation System
70	FR-AI-006	Suggest Connections Based on Mutual Connections	AI-Powered Recommendations	High	Recommendation System
71	FR-AI-007	Analyze User and Job Data for Connection Suggestions	AI-Powered Recommendations	Medium	Recommendation System
72	FR-AI-008	Dynamic Connection Suggestions	AI-Powered Recommendations	High	Recommendation System
73	FR-AI-009	Recommend Posts Based on Skills	AI-Powered Recommendations	High	Recommendation System
74	FR-AI-010	Recommend Users	AI-Powered Recommendations	High	Recommendation System
75	FR-AI-011	Updates in AI Recommendations	AI-Powered Recommendations	High	Recommendation System
76	FR-AI-012	Generate Personalized Recommendations	AI-Powered Recommendations	High	Recommendation System
77	FR-AI-013	Smart Resume Parsing	AI-Powered Recommendations	High	CV Pasing – User Management
78	FR-SK-001	Generate Dynamic Assessments Based on Job Skills	Skill Assessment	High	Recommendation System
79	FR-SK-002	Display Timers for Each Question and Automatically Submit	Skill Assessment	High	System
80	FR-SK-003	View Quiz Result	Skill Assessment	High	Job Seeker
81	FR-SK-004	View already attempted to quiz	Skill Assessment	High	Job Seeker
82	FR-SK-005	Upload CV	Skill Assessment	High	Job Seeker

83	FR-RM-001	Compare Job Application Skills and experience with Profile Skills and experience	Recruitment Management	High	System
84	FR-RM-002	Display Completion Message After CV Submission	Recruitment Management	Medium	System
85	FR-RM-003	View applicants for each job application	Recruitment Management	High	Company
86	FR-RM-004	Display Candidate Information and Comparison Results in Employer Dashboard	Recruitment Management	High	Company
87	FR-RM-005	View CV	Recruitment Management	High	Company
88	FR-RM-006	Send Job Offer	Recruitment Management	High	Company
89	FR-Admin-DASH-001	Manage User Accounts	Recruitment Management	High	System
90	FR-Admin-DASH-002	Manage Permissions	Admin Dashboard	High	System
91	***FR-RM-007***	***Schedule Personal Interviews and Notify Applicants***	***Recruitment Management***	***High***	***Company***
92	***FR-RM-007***	***Generate Calendar Event for Scheduled Interviews***	***Recruitment Management***	***High***	***Company***
93	NF-REQ-01	Authentication and Authorization Security	Admin Dashboard	High	System
94	NF-REQ-02	Password Security and Management	Security	High	System

95	NF-REQ-03	Session Management	Security & User Management	High	System
96	NF-REQ-04	Profile Data Management	Security & User Experience	High	System
97	NF-REQ-05	Performance Optimization	Data Management & Privacy	High	System
98	NF-REQ-06	Real-time Updates and Polling	System Responsiveness	Medium	System
99	NF-REQ-07	Privacy Controls	Privacy & User Controls	High	System

3. SRS Document:

1. Introduction:

1. *Purpose:*

The purpose of this **Software Requirements Specification (SRS)** document is to define the **functional and non-functional requirements** of the **AI-Powered Recruitment Platform**. This document serves as a foundational reference for stakeholders, including **developers, project managers, recruiters, and job seekers**, ensuring a **clear and structured understanding** of the system's capabilities, features, and limitations.

2. *Project scope:*

The **AI-Powered Recruitment Platform (HireHub)** is designed to enhance the hiring process by leveraging **AI-driven job recommendations, dynamic skill assessments, and professional**

networking features. The system ensures seamless interaction between **job seekers and company users** by integrating:

User management, allowing users to register, log in, update profiles, and manage privacy settings.

AI-powered job matching, utilizing pre-trained NLP models to suggest relevant job postings based on candidate preferences.

AI-generated quizzes, dynamically creating skill assessments for candidates based on job descriptions and requirements.

Social interaction & networking, enabling users to post updates, engage with posts (like, comment, share), and receive AI-recommended content.

Job application tracking, allowing recruiters to post job openings, manage applications, and interact with potential candidates.

The system ensures **data security and scalability** through **JWT-based authentication, role-based access control, and secure API integration**.

Interview Scheduling: Enables company users to schedule, reschedule, cancel, and manage interviews for job applicants, integrating with platforms like Zoom and Google Meet to generate meeting links and send automated notifications.

CV Parsing: Automatically extracts structured information (e.g., contact details, skills, experience) from PDF CVs using a multi-modal LLM, then uses this data to populate and update candidate profiles on the platform.

Adaptive Quiz: Dynamically generates job-specific quizzes with varying difficulty levels using an LLM, adjusting question difficulty in real-time based on the candidate's performance to provide a personalized assessment.

Messaging System: Provides real-time, direct communication between users with features for sending, editing, and deleting messages, utilizing WebSockets for live updates and managing message read status.

3. Document overview:

The software requirements specification SRS document will be structured as follows:

1. Introduction.
2. Overall description and product features.
3. Nonfunctional requirements.
4. Conclusion.

2. Overall description and product features:

1. Product features:

The product features derived from the High-Level Use Case Diagram are as follows:

3.2.1.1 Manage Accounts (UC-01): This feature allows users to manage their accounts by configuring personal information, privacy settings, and account

preferences. Users can update their email, phone number, profile picture, and privacy settings to control who can view their profile. They also have the option to delete their accounts permanently.

3.2.1.2 Manage Profile (UC-02): This feature allows users to update and personalize their profiles by adding key information such as profile pictures, phone numbers, emails, and privacy settings. Users can also modify their work experience, skills, and certifications.

3.2.1.3Setup Profile (UC-03): This feature ensures that users complete their profiles by providing essential details such as job preferences, skills, location, and profile picture. Users can also configure privacy settings related to their profile visibility.

3.2.1.4 Manage Account Settings (UC-04): This feature enables users to configure their account preferences, including updating personal details, managing privacy settings, and setting AI-based job recommendations.

3.2.1.5 Register (UC-05): This feature allows users to create a new account by selecting their role as either a Job Seeker or a Company User. The system requires basic information such as email, password, and user details for registration.

3.2.1.6 Login (UC-06): This feature provides a secure authentication mechanism, allowing users to log in to access the platform.

3.2.1.7 Reset Password (UC-07): This feature enables users to recover their accounts in case they forget their password by sending a reset link via email.

3.2.1.8 Manage Recommendations (UC-08): This feature enables the AI system to analyze user profiles and job preferences to provide personalized job recommendations. Based on user skills, experience, and preferences, the system suggests the most relevant job postings and Normal Posts.

3.2.1.9 Suggest Connections (UC-09): This feature allows the AI system to suggest professional connections based on job interests, industry preferences, and mutual connections.

3.2.1.10 View Home Page (UC-10): This feature provides users with a centralized Page displaying job recommendations, networking suggestions, posts, and hiring insights. Users can access their profile, job listings, and suggested connections from the home page.

3.2.1.11 Search for Users (UC-11): This feature allows users to search for professionals within the platform, helping them expand their professional networks.

3.2.1.12 Manage Social Interaction (UC-12): This feature allows users to interact with posts and updates within the platform. Users can create posts, like, comment.

3.2.1.13 Send Notifications (UC-13): This feature ensures that users receive real-time notifications regarding job postings, shared posts by users, networking requests, and social interactions.

3.2.1.14 Apply for Jobs (UC-14): This feature allows job seekers to browse available job listings and apply for relevant positions through the platform.

3.2.1.15 View Jobs List (UC-15): This feature provides job seekers with a comprehensive list of job postings, including job details, employer information, and application requirements.

3.2.1.16 Search Using Filter (UC-16): This feature enhances job search functionality by allowing users to filter job postings based on criteria such as job title, required skills, experience level, employment type, and location.

3.2.1.17 Manage Recruitment (UC-17): This feature allows company users to manage the recruitment process efficiently. Recruiters can track job applications, review candidate profiles, and shortlist potential hires.

3.2.1.18 Post Job Application (UC-18): This feature allows employers to create and post job openings by specifying details such as job title, description, required skills, salary range, and employment type.

3.2.1.19 View Applicants (UC-19): This feature enables recruiters to review job applications and assess candidate qualifications before making hiring decisions.

3.2.1.20 Delete Job Application (UC-20): This feature allows recruiters to remove job listings when a position is filled or no longer available.

3.2.1.21 View Profile (UC-21): This feature allows users to view both their own profile and other user profiles. The visibility of profile details depends on privacy settings.

3.2.1.22 Log out: The Logout feature allows users to securely exit their accounts, ensuring the protection of their session and personal data. When a job seeker or company user chooses to log out, the system terminates their active session, revokes authentication tokens, and redirects them to the login page.

3.2.1.23 The Save/Unsaved Job Listing: feature allows job seekers to bookmark job postings for later review. When a job seeker saves a job, the system stores the listing under their profile, making it easily accessible. If the user decides they are no longer interested, they can unsaved the job, removing it from their saved listings.

3.2.1.24 The View Saved Job Listings feature enables job seekers to access a dedicated section where they can review all previously saved jobs. This functionality helps users manage their job search efficiently by keeping track of opportunities they want to apply for later.

3.2.1.25 Manage User Accounts: This feature allows the Admin to log in to the system using the Django Admin Panel and oversee all user accounts.

Through this interface, the Admin can view a list of registered Job Seekers and Company Users, edit user details, deactivate accounts, or permanently delete users when necessary. This feature ensures efficient user management, maintaining system integrity and security.

3.2.1.26 Manage Permissions: This feature enables the Admin to control user access levels and functionalities within the system. Using the Django Admin Panel, the Admin can assign or revoke permissions, restricting specific actions such as job posting, profile editing, or social interactions. This feature ensures that different user roles (e.g., Job Seekers, Company Users) have appropriate privileges, maintaining a secure and well-regulated platform.

3.2.1.27 Manage Messaging : his feature enables real-time, direct communication between users on the platform, including both job seekers and company representatives. The messaging system supports one-to-one conversations, allowing users to initiate, send, receive, edit, and delete messages securely. Built on Django REST Framework and Django Channels, the system uses WebSockets for instant message delivery and read receipts, ensuring a seamless chat experience. All messages and conversations are stored in the database, with user-friendly APIs for fetching conversation history and searching for users to start new chats. The system enforces access control so that only conversation participants can view or

interact with messages. Notifications are triggered for new messages, and message status (read/unread, edited, deleted) is synchronized in real time across all user devices. This feature ensures efficient, secure, and interactive communication within the recruitment platform.

3.2.1.28 Interview Scheduling: This feature enables company users to efficiently schedule interviews with job applicants directly through the platform. Using a dedicated interface, company representatives can select an applicant, specify the interview date, time, platform (such as Zoom or Google Meet), and assign an interviewer. The system manages all interview details, stores them securely in the database, and ensures that only authorized users can create, view, or modify interview records. Automated notifications are sent to both applicants and interviewers upon scheduling, rescheduling, or cancellation of interviews. The feature supports viewing upcoming and completed interviews, updating interview status, and integrating with third-party meeting platforms for seamless virtual interview experiences. This ensures a streamlined, organized, and transparent interview process for both recruiters and candidates.

3.2.1.29 CV Parsing:This feature enables users to upload their CVs in PDF format and have key information automatically extracted and structured by the system. Leveraging advanced AI and large language models, the platform parses uploaded CVs to identify and extract details such as candidate name, contact information, skills, work experience, education, and certifications. The extracted data is

then used to populate the user's profile, reducing manual data entry and ensuring consistency. The system validates file type and size, supports secure file storage, and provides clear feedback on parsing results. This feature streamlines the onboarding process for job seekers and enhances the accuracy and completeness of candidate profiles for recruiters.

2. User classes and characteristics:

Our system has two main types of users, each with distinct roles and functionalities:

Job Seeker: The primary user of the platform who is searching for job opportunities. Job seekers can create and update their profiles, specify their job preferences, and apply for job listings. They can also engage in professional networking by following other users, interacting with posts, and receiving AI-powered job recommendations. Additionally, job seekers have access to job search filters and receive real-time notifications about job openings, connection requests, and application status updates.

Company User: User how post job opportunities and managing job applications. Company users can create job application, view and evaluate applicants, and communicate with potential candidates. They can also manage recruitment processes, and track job applications through the system. Additionally, company users can engage in networking activities, follow professionals, and interact with job-related posts to enhance their hiring reach.

3. Assumptions:

Network Connection: It is assumed that users will have a stable and reliable internet connection. Since the platform is web-based, its performance, accessibility, and real-time features depend on a consistent internet connection.

Browser Requirements: Users are expected to access the system through a modern web browser that supports HTML5, CSS3, and JavaScript. The platform is optimized for browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge to ensure smooth user experience and functionality.

User Roles and Interaction: It is assumed that Job Seekers and Company Users will actively engage with the system. Job seekers will regularly update their profiles, apply for jobs, and interact with networking features, while company users will consistently post job openings and manage applicants.

AI Recommendations Dependency: The accuracy of job and connection recommendations depends on the information provided by users. It is assumed that users will accurately complete their profiles, including job preferences, skills, and experience, to receive relevant suggestions.

Scalability Considerations: The platform is designed to handle a growing user base, assuming that multiple job seekers and company users will simultaneously use the system for job searching and recruitment.

3. Nonfunctional requirements:

The AI-Powered Recruitment Platform must adhere to the following non-functional requirements to ensure security, efficiency, scalability, and user experience:

- **NFR-01 (Authentication and Authorization Security):** The system must implement a secure JWT-based authentication system with a token refresh mechanism, protected routes, and role-based access control to ensure that only authorized users can access specific functionalities.

- **NFR-02 (Password Security and Management):** The platform must enforce secure password policies, requiring at least 8 characters, including numbers and special characters. Additionally, the system should provide secure password reset functionality with time-limited tokens, email verification, and encrypted password storage.

- **NFR-03 (Session Management):** The system must include secure session handling mechanisms, such as token expiration, automatic refresh, and token blacklisting for secure logout and session control.

- **NFR-04 (Profile Data Management):** The system must provide flexible profile management, supporting different user types (Job Seeker and Company User), configurable privacy settings, and secure storage of user information.

- NFR-05 (Performance Optimization): The platform should be optimized for high performance using lazy loading, pagination for large lists (such as job postings, comments, and user interactions), and efficient database queries to improve response times.

- **NFR-06 (Real-time Updates and Polling):** The system should provide real-time updates for profile changes, new job recommendations, connection requests, and notifications using an optimized polling mechanism with controlled intervals to reduce server load while maintaining responsiveness.

- **NFR-07 (Privacy Controls):** The platform must provide granular privacy settings, allowing users to control the visibility of their personal information, such as profile details, contact information, and communication preferences.

4. Conclusion:

This Software Requirements Specification (SRS) document outlines the critical functionalities, user roles, and assumptions essential for developing a robust and user-friendly AI-powered recruitment platform. By adhering to these requirements, the system will effectively support job seekers in finding relevant job opportunities and company users in managing job postings and recruitment processes efficiently. Additionally, the platform ensures secure authentication, AI-driven recommendations, real-time notifications, and interactive social networking features, streamlining the hiring and job search experience for all users.

4. Requirements modeling:

Use case diagram: use-case diagrams model the behavior of a system and help to capture the requirements of the system.

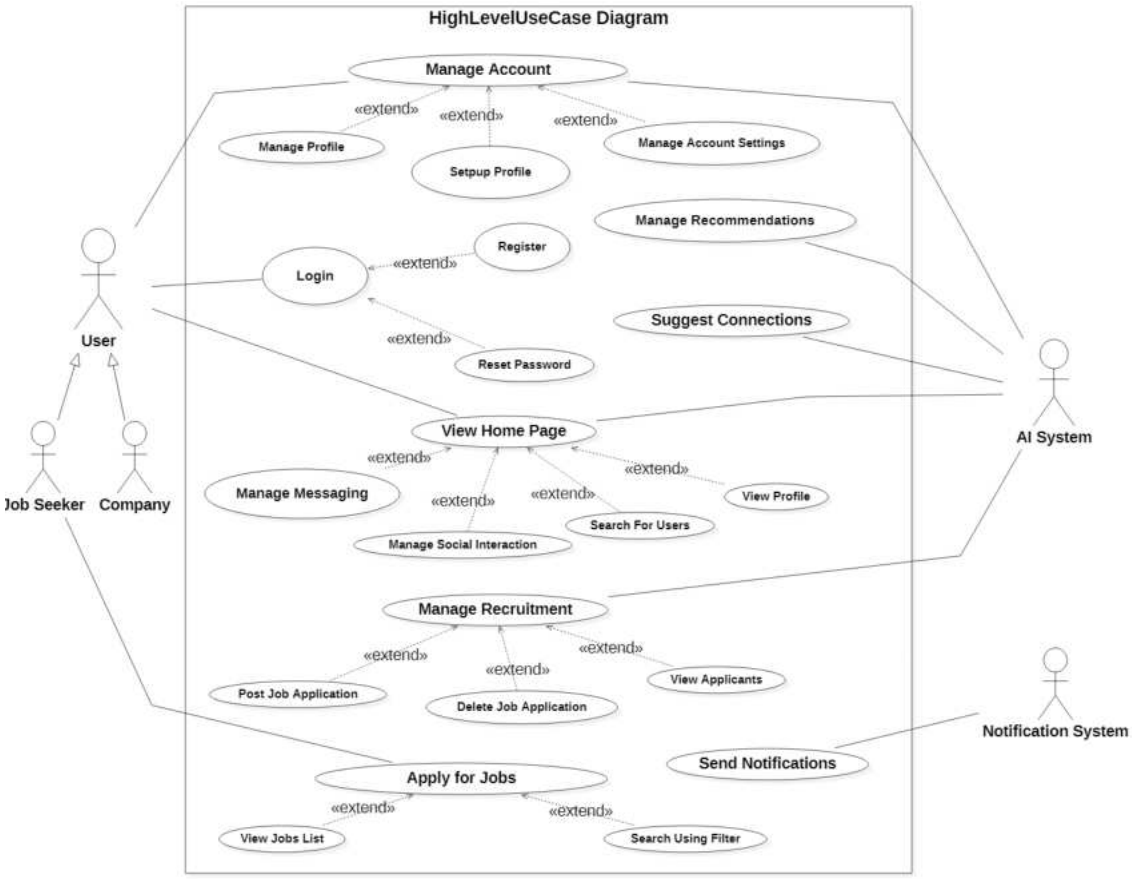


Figure 3 High-level Use-case Diagram

System features – use cases:

- Register (UC-05):

Table 7 Register Usecase Specification

Use case name:	Register
Participating Actors:	Job Seeker, Company
Preconditions	1. The user must not have an existing account. 2. The system must be online and accessible. 3. The user must be on the registration page.
Postconditions	1. A new user account is created in the system. 2. The system redirects the user to the login page.
Flow Of Event	A.1 Basic Flow **(Main Success Scenario)** 1. The user navigates to the registration page. 2. The system displays the registration form, requiring the user to enter: - User Type (Job Seeker or Company User). 3. Based on the selected user type, the system prompts for different required fields: - If Job Seeker: • Email • Password

	• First Name • Last Name • Date of Birth - If Company User: • Email • Password • Company Name 4. The user fills in the form and submits it. 5. The system validates the input data: - Ensures the email is in the correct format. - Checks the password meets security criteria. - Confirms that the email is not already registered. 6. If the validation succeeds, the system creates a new account. 7. The system redirects the user to the login page.
Alternative Flows	A.2.1 Invalid Input Data - If the user enters an invalid email format or a weak password, the system displays an error message and prompts them to re-enter valid information.

	A.2.2 Email Already Registered - If the email is already in use, the system notifies the user and suggests logging in instead.
Related Use Cases	Login (After registration, users log in to access the system).

- Log In (UC-06):

Table 8 Login Usecase Specification

Use case name:	Log in
Participating Actors:	Job Seeker, Company.
Preconditions	1. The user must have an existing registered account. 2. The system must be available. 3. The user is on the login page.
Postconditions	1. The user is authenticated successfully. 2. The system redirects the user to the Home Page.
The flow of events:	A.1 Basic Flow **(Main Success Scenario)** 1. The user navigates to the login page. 2. The system displays the login form, prompting the user to enter: - Email - Password 3. The user fills in the form and submits it. 4. The system validates the credentials by checking:

	- If the email exists in the database. - If the password matches the stored credentials. 5. If the validation succeeds, the system authenticates the user. 6. The system redirects the user to the Home Page, where they can view job recommendations, connection suggestions, and posts.
Alternative flows:	A.2.1 Invalid Credentials - If the user enters an incorrect email or password, the system displays an error message and prompts them to re-enter valid credentials.
Related Use Cases	Register User (A prerequisite before logging in). View Home Page (Next step after successful login).

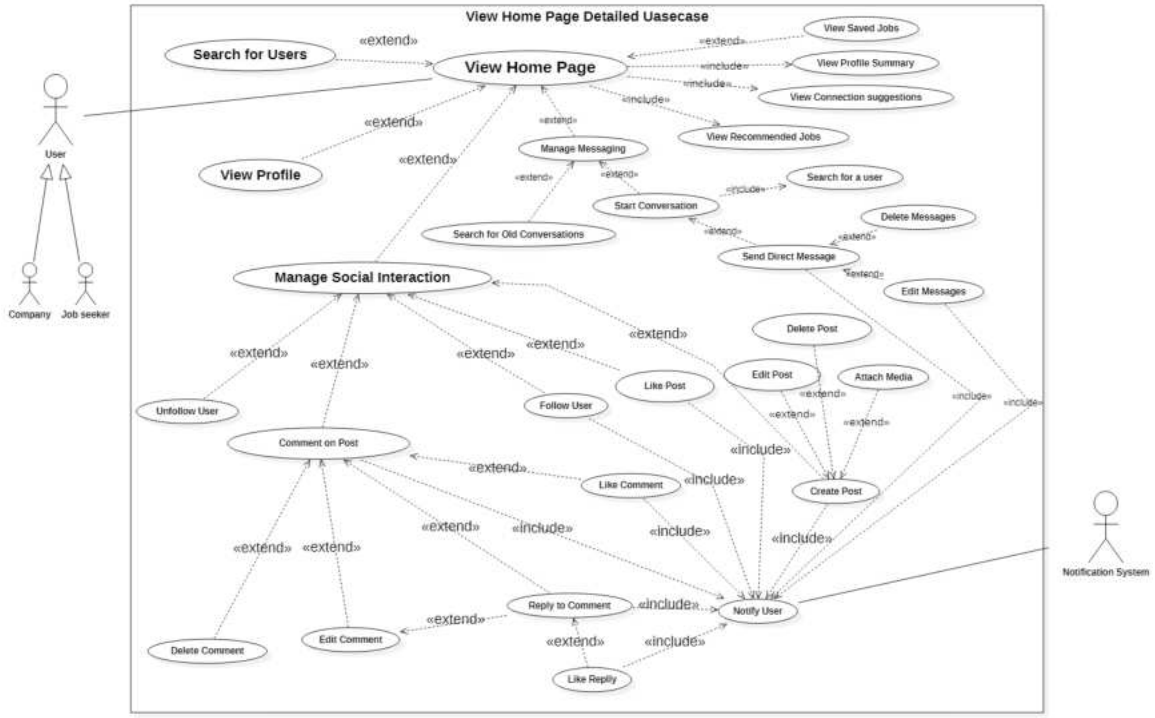


Figure 4 View Home Page Detailed Use-case Diagram

Manage Social Interaction (UC-12):

Table 9 Manage Social Interaction Usecase Specification

Use case name:	Manage Social Interaction
Participating Actors:	Job Seeker, Company.
Preconditions	1. The user must be logged into the system. 2. The system must be online and accessible. 3. The user must have an active account. 4. The home page must be successfully loaded.
Postconditions	1. The user successfully engages in social interactions (liking, commenting, replying, following, or posting). 2. The system updates interactions in real-time. 3. Notifications are sent to relevant users when an interaction occurs.
The flow of events:	**Main Success Scenario:** 1. The user navigates to the home page. 2. The system displays posts from followed users and trending posts. 3. The user can like a post. 4. The user can comment on a post. 5. The user can reply to an existing comment. 6. The user can like a comment or reply. 7. The user can edit or delete their comments. 8. The user can follow or unfollow another user. 9. If the user creates a new post, they can attach media files (images/videos).

	10. The system sends notifications when another user interacts with a post (likes, comments, replies).
Alternative flows:	User Tries to Follow a Private Profile: Instead of an automatic follow, a follow request is sent, awaiting approval
Related Use Cases	Create Post, Like Post, Comment on Post, Reply to Comment, Follow User, Unfollow User, Send Notifications

Manage Messaging (UC-10.5):

Use case name:	Manage Messaging
Participating Actors:	Job Seeker, Company.
Preconditions	1. User is authenticated and logged into the platform 2. User has a valid account (Job Seeker or Company).
Postconditions	1. Messages are sent, received, edited, or deleted as requested. 2. Conversations are updated and stored in the system. 3. Users can view their conversation history.
The flow of events:	**Main Success Scenario:** 1. The user navigates to the home page. 2. User navigates to the messaging section. 3. User searches for another user (Job Seeker or Company) to start a conversation.

	4. User selects a user from the search results and initiates a conversation. 5. User sends a direct message.. 6. The recipient receives the message in real time. 7. User can view all ongoing and past conversations 8. User can edit or delete their own messages within the allowed timeframe 9. All changes are reflected in the conversation history for both participants.
Alternative flows:	**A1: No Search Results** : if no users match the search query, the system displays a "No users found" message **A2: Message Edit/Delete Timeout :** If the user tries to edit or delete a message after the allowed timeframe, the system displays an error and prevents the action. **A3: Empty Conversation List:** If the user has no previous conversations, the system displays an empty state message.
Related Use Cases	Send Direct Message, Recives Direct Message, Notify user, Start Conversation , Search for users , Search for Old Conversation

Table 27 Manage Messaging Use-case Specification

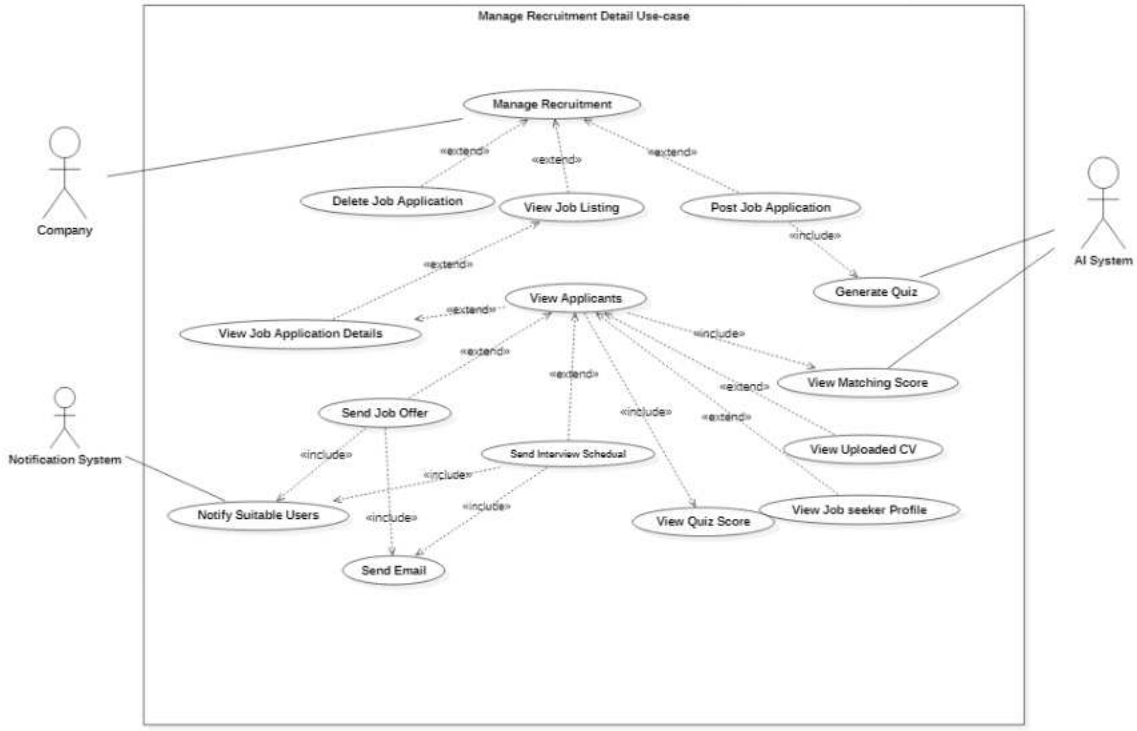


Figure 5 Manage Recruitment Detailed Use-case

- Manage Recruitment (UC-17):

Table 10 Manage Recruitment Usecase Specification

Use case name:	Manage Recruitment
Participating Actors:	Primary Actor: Company. Secondary Actors: AI System, Notification System
Preconditions	1. The Company User must be logged in. 2. The system must be online and accessible. 3. There must be active job listings or applicants in the system.
Postconditions	1. The company successfully manages job postings and applicants. 2. The system notifies job seekers when a new job is posted. 3. The AI system generates and evaluates quizzes for applicants. 4. Job offers are sent to selected candidates.

The flow of events:	Main Success Scenario: 1. The user can perform the following actions: - Post a new job application. - View existing job listings. - Delete a job application if necessary. 2. The system notifies suitable job seekers about the new job posting. 3. Job seekers apply for jobs by submitting applications. 4. The company can view applicants for a job. 5. The company user selects an applicant and can: - View the job seeker's profile. - View the uploaded CV. - Check the quiz score generated by the AI system. - View the matching score between the job and the applicant’s skills. 6. If an applicant meets the requirements, the company schedual interview by add the date and time of the interview, the platform that the interview will be (Google Meets or ZOOM), and The interviewer name. After fill the information about the Interview Schedual , The company user send email notifcation 7.After the meet the interview requirements , the company user send a job offer via email notification 8. The system sends an email notification to the selected job seeker.
Related Use Cases	Post Job Application, View Applicants, View Uploaded CV, View Matching Score, Generate Quiz, View Quiz Score, Send Job Offer, Notify Suitable Users.

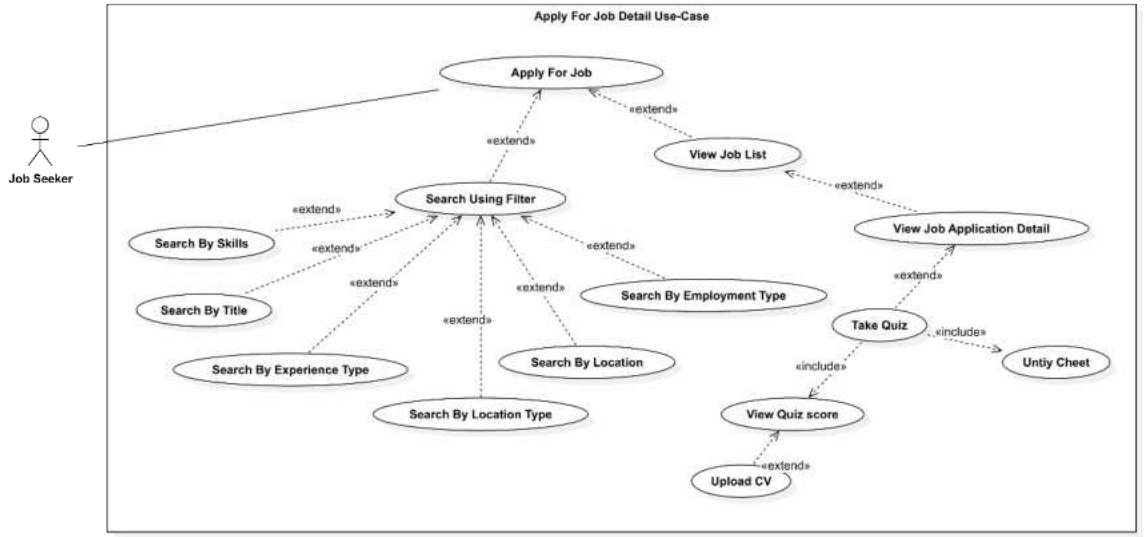


Figure 6 Apply for Job Detailed Ucase-case

- Apply for Job (UC-14):

Table 11 Apply for job Usecase Specification

Use case name:	Apply for Job
Participating Actors:	Primary Actor: Job Seeker Secondary Actors: AI System
Preconditions	1. The user successfully applies for a job. 2. The AI system evaluates the job seeker's qualifications. 3. The system stores the application and uploaded CV.
Postconditions	1. The user successfully applies for a job. 2. The AI system evaluates the job seeker's qualifications. 3. The system stores the application and uploaded CV.
The flow of events:	Main Success Scenario: 1. The Job Seeker navigates to the Apply for Job section. 2. The user searches for jobs using filters, including: - Search by skills

	- Search by title - Search by experience type - Search by location type - Search by employment type 3. The system displays a list of job postings matching the selected criteria. 4. The Job Seeker views job application details. 5. The user proceeds to take a skill-based quiz. 6. The AI system evaluates the quiz and generates a score. 7. The user uploads their CV. 8. The system stores the application.
Related Use Cases	Search Using Filter, View Job List, View Job Application Detail, Take Quiz, View Quiz Score, Upload CV.

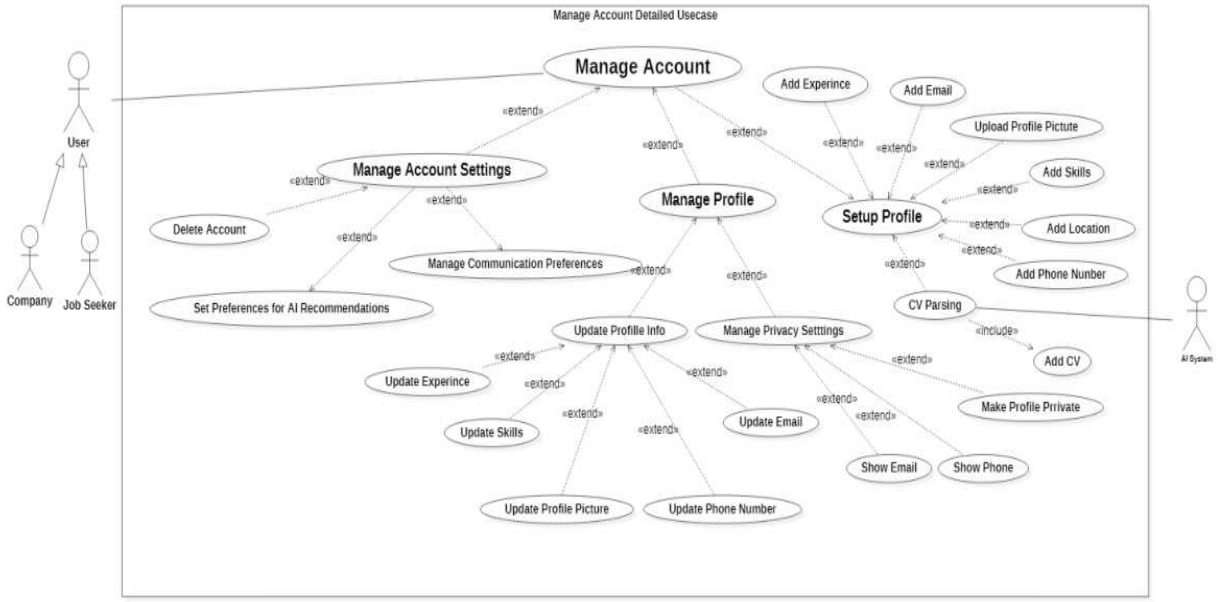


Figure 7 Manage Account Detailed Use-case

- Manage Account Settings (UC-04):

Table 12 Manage Account Settings Usecase Specification

Use case name:	Manage Account Settings
Participating Actors:	Job Seeker, Company.
Preconditions	The user must be logged in and have a new or incomplete profile.
Postconditions	1. The system saves the updated profile information. 2. The user can now interact with job postings and networking features.

The flow of events:	Main Success Scenario: 1. The user navigates to the account settings section. 2. The system displays options for managing settings, including: - Manage Communication Preferences. - Set Preferences for AI Recommendations. - Delete account 3. The user selects an option and updates the necessary details. 4. The system validates and saves the changes. 5. The updated settings take effect immediately.
Alternative Flows	1. Invalid Input: - If the user enters incorrect or incomplete data (e.g., invalid skills), the system prompts for correction.

- Manage Profile (UC-02):

Table 13 Table 12 Manage Account Profile Usecase Specification

Use case name:	Manage Profile
Participating Actors:	Job Seeker, Company.
Preconditions	1. The user must be logged into their account. 2. The profile must exist in the system.
Postconditions	1. The user successfully updates profile details. 2. The system saves changes and reflects them in real-time.

The flow of events:	Main Success Scenario: 1. The user navigates to the Manage Profile section. 2. The system displays profile details, allowing the user to: - Update Profile Info (Email, Phone Number, and Profile Picture). - Update Privacy Settings (Show/Hide Email and Phone). 3. The user selects an option and makes modifications. 4. The system validates and applies the changes.
Alternative Flows	1. Profile Picture Change: - If the user uploads an invalid format or large file, the system prompts for correction. 2. Privacy Settings: - The user can switch their profile to private mode, restricting access.
Related Use Cases	Update Experience, Update Skills, Update Profile Picture, Update Phone number, Update Email

- Setup Profile (UC-03):

Table 14 Manage Setup Profile Usecase Specification

Use case name:	Setup Profile
Participating Actors:	Job Seeker, Company.
Preconditions	The user must be logged in and have a new or incomplete profile.
Postconditions	1. The system saves the updated profile information. 2. The user can now interact with job postings and networking features.

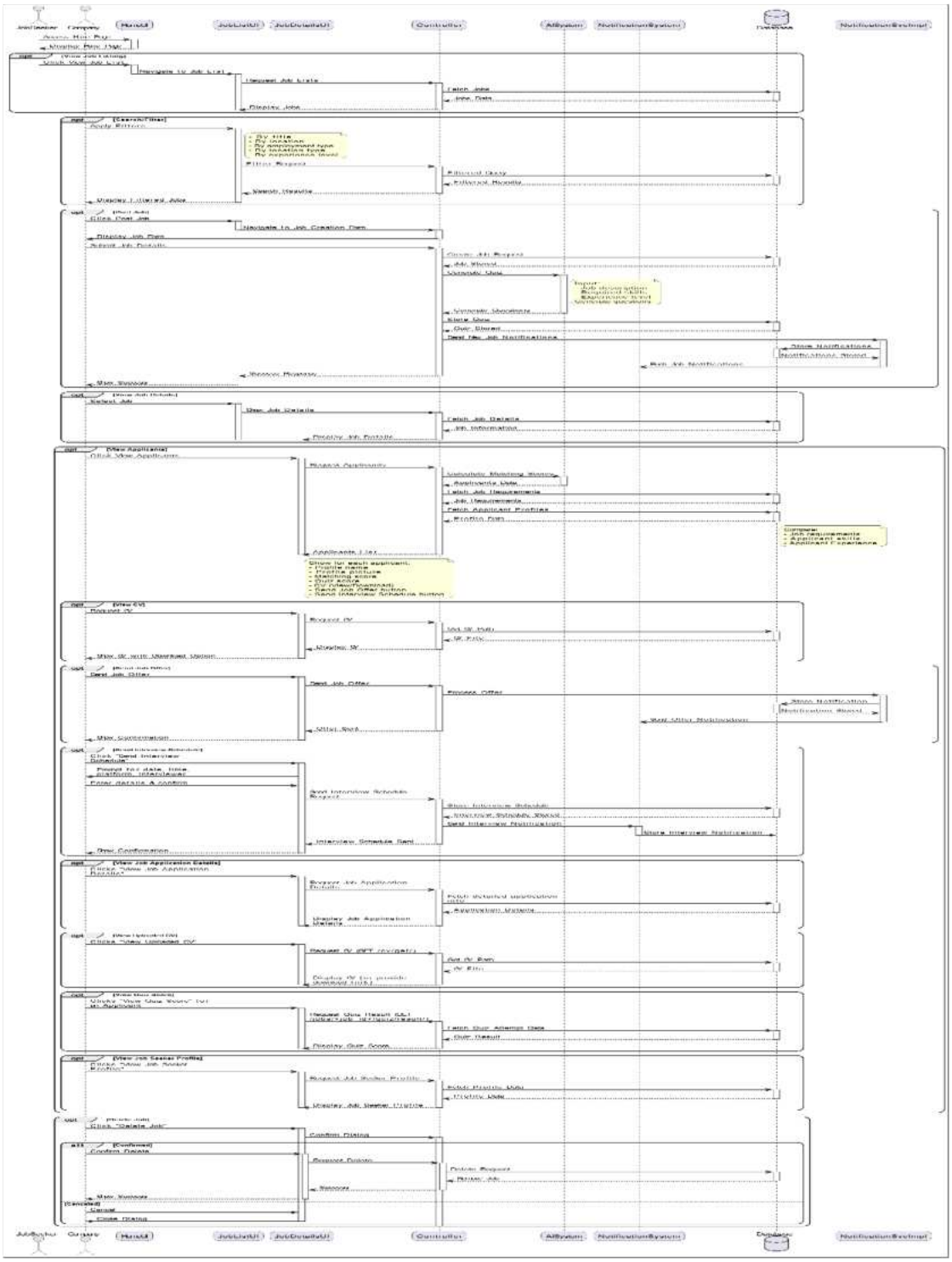
The flow of events:	Main Success Scenario: 1. The user navigates to the Setup Profile section. 2. The system prompts the user to complete missing details, including: - Upload Profile Picture - Add Email & Phone Number - Add Skills & Experience - Add Profile Picture & Location 3. The user enters the required information and submits it. 4. The system validates and stores the data. 5. The profile setup is completed successfully.
Alternative Flows	1. Missing Required Information: - If mandatory fields are left empty, the system prompts the user to complete them. 2. Invalid Email/Phone Format: - The system rejects incorrect formats and requests valid input.
Related Use Cases	Upload Profile Picture, Add Skills Add Experience, Add Location, Add Email, Add Phone Number.

CV parsing (UC-1.3.7):

Use case name:	CV Parsing
Participating Actors:	Job Seeker
Preconditions	- The user must be logged in and have a new or incomplete profile. - User has a CV in PDF format.
Postconditions	1. The user's CV is uploaded and securely stored. 2. The system parses the CV and extracts structured information 3. The user's profile fields are auto-filled with the parsed data. 4. The user can review and further edit the auto-filled information
The flow of events:	Main Success Scenario: 1. User navigates to their profile page. 2. User clicks "Edit Profile" 3. User selects and uploads their CV (PDF format) 4. System validates the file type and size. 5. System uploads and stores the CV. 6. User clicks "Parse CV" 7. System parses the CV using an AI-powered parser (LLM). 8. System extracts key information (name, contact, skills, experience, education, etc.). 9. System auto-fills the corresponding profile fields with the parsed data 10. User reviews and can further edit the auto-filled information 11. User saves the updated profile.

Alternative Flows	1. Invalid File Type/Size : - If the uploaded file is not a PDF or exceeds the size limit, the system displays an error and rejects the upload. 2. Parsing Failure: - If the system fails to parse the CV, an error message is shown and the user can try again or enter data manually. 3. Partial Data Extraction: - If some fields cannot be extracted, the system fills only the available fields and notifies the user to complete the rest manually.
Related Use Cases	----------

Sequence diagram: *Figure 8 Sequence Diagram – Manage Recruitment*



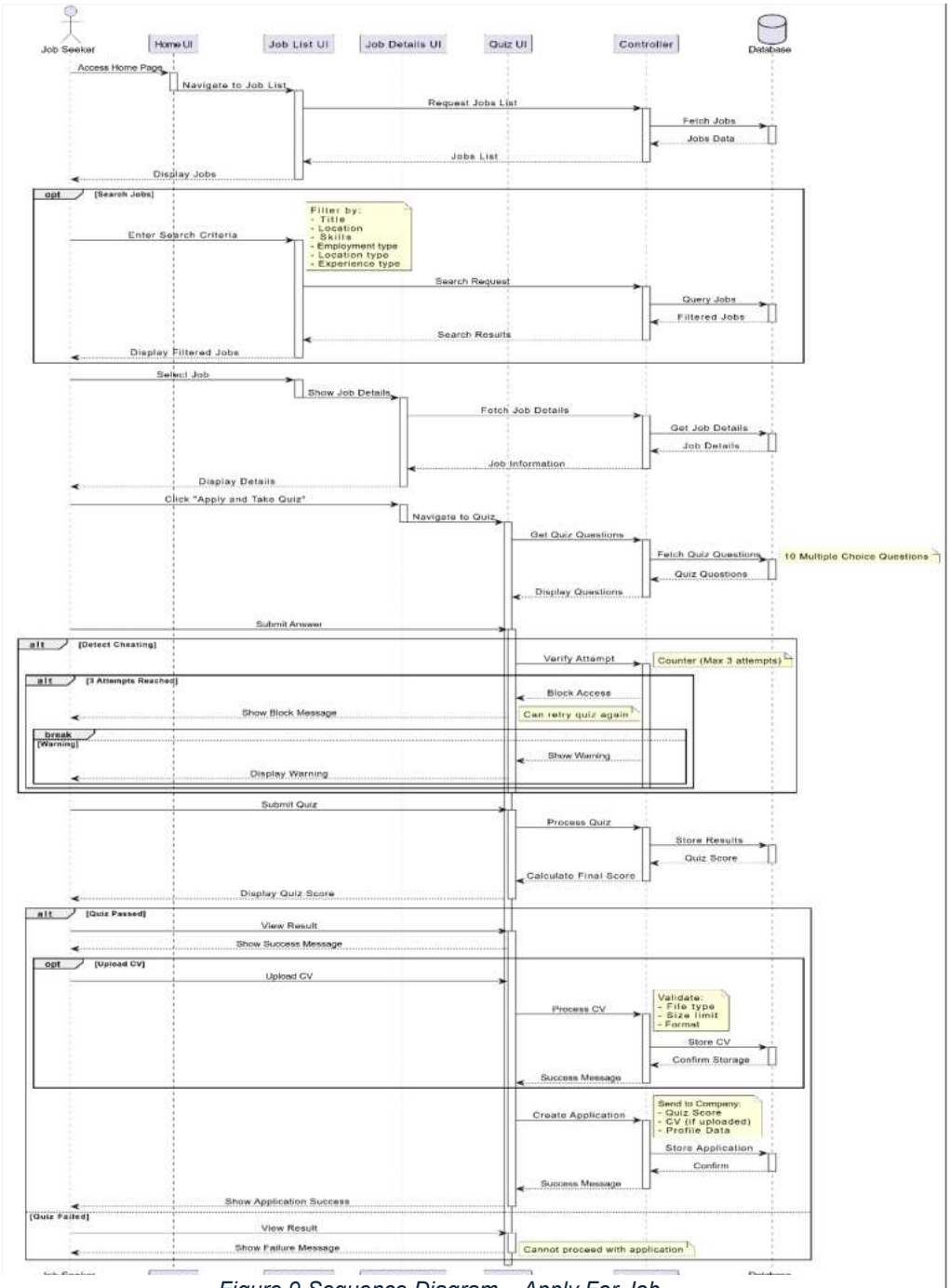


Figure 9 Sequence Diagram – Apply For Job

Activity diagram:

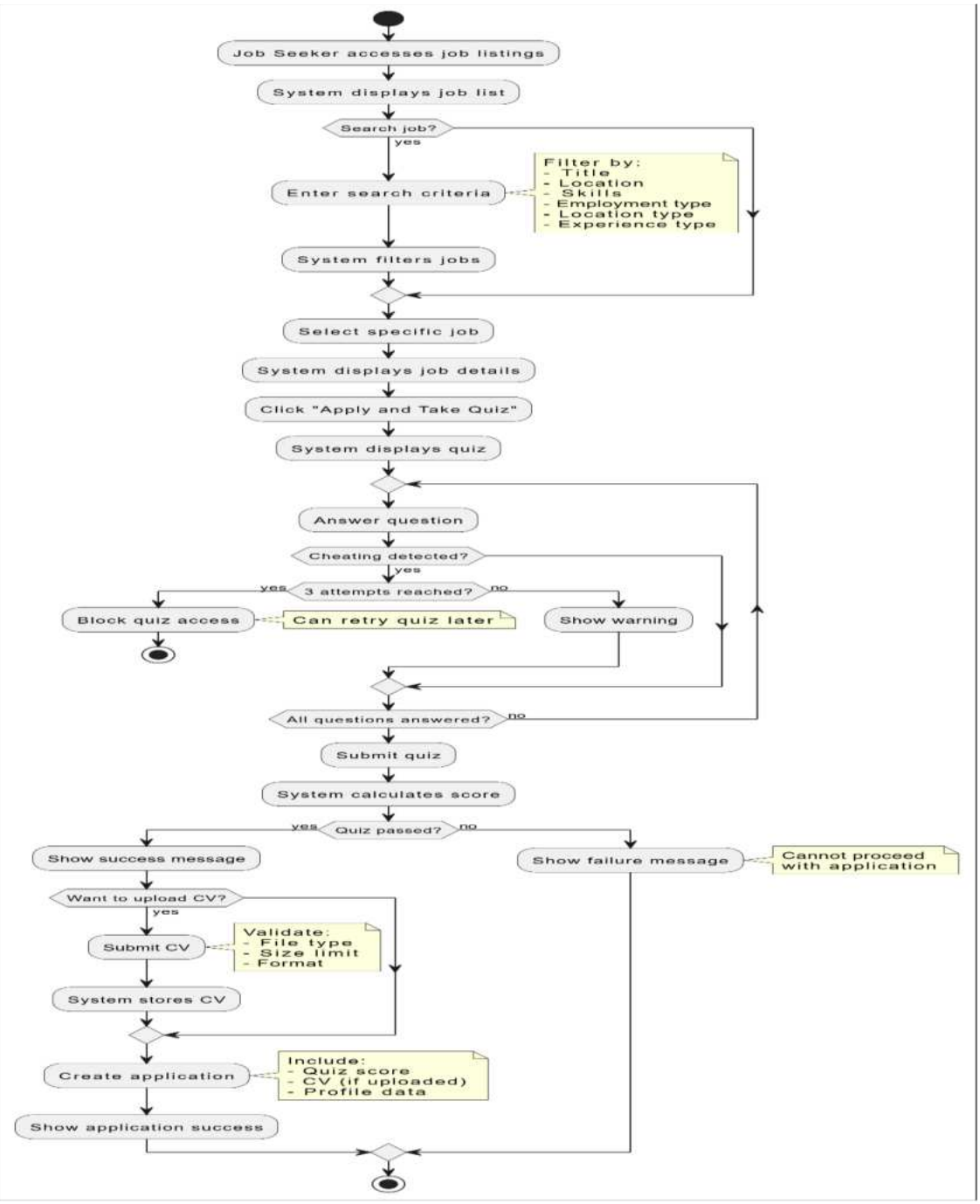


Figure 10 Activity Diagram – Apply For Job

Analysis class diagram:

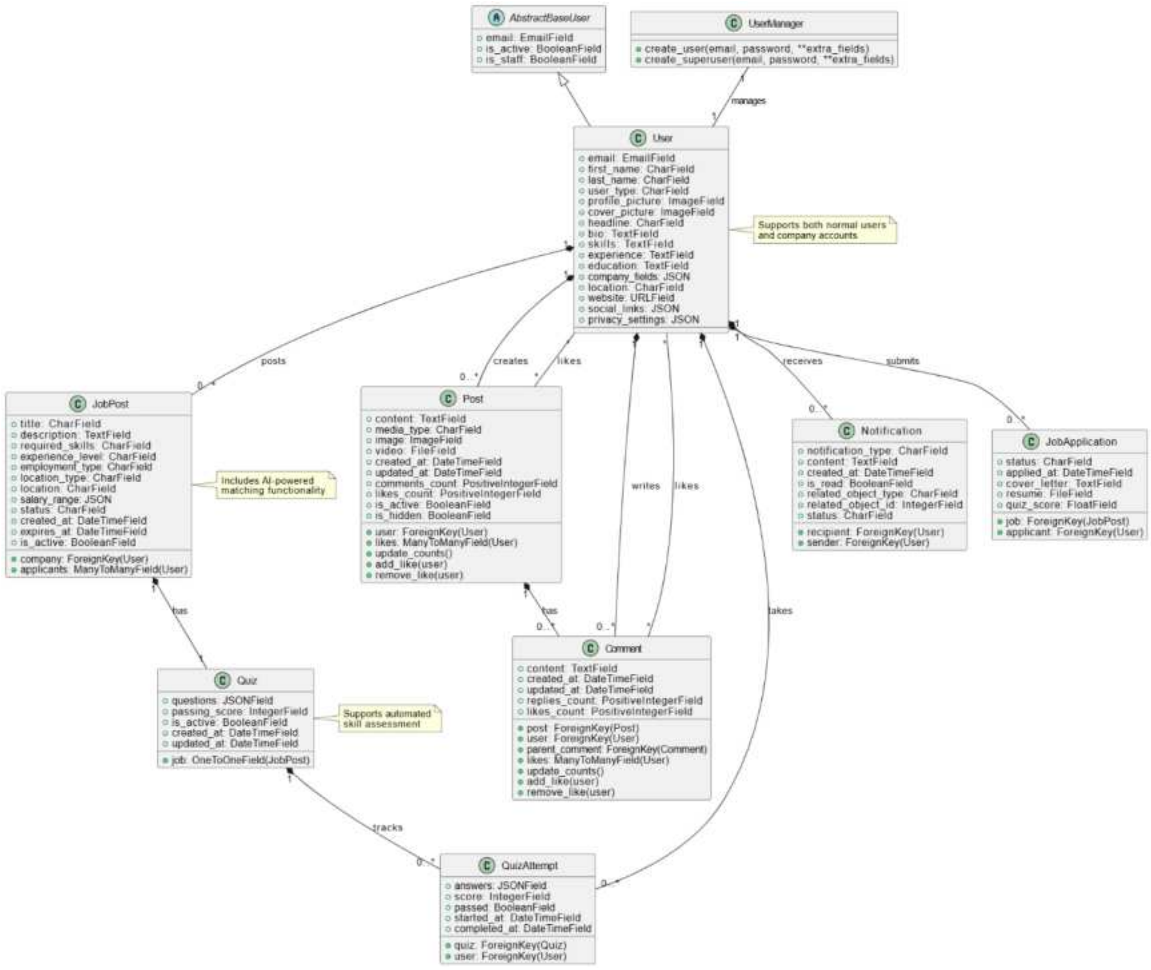


Figure 11 Analysis Class Diagram

Entity Relationship Diagram ERD:

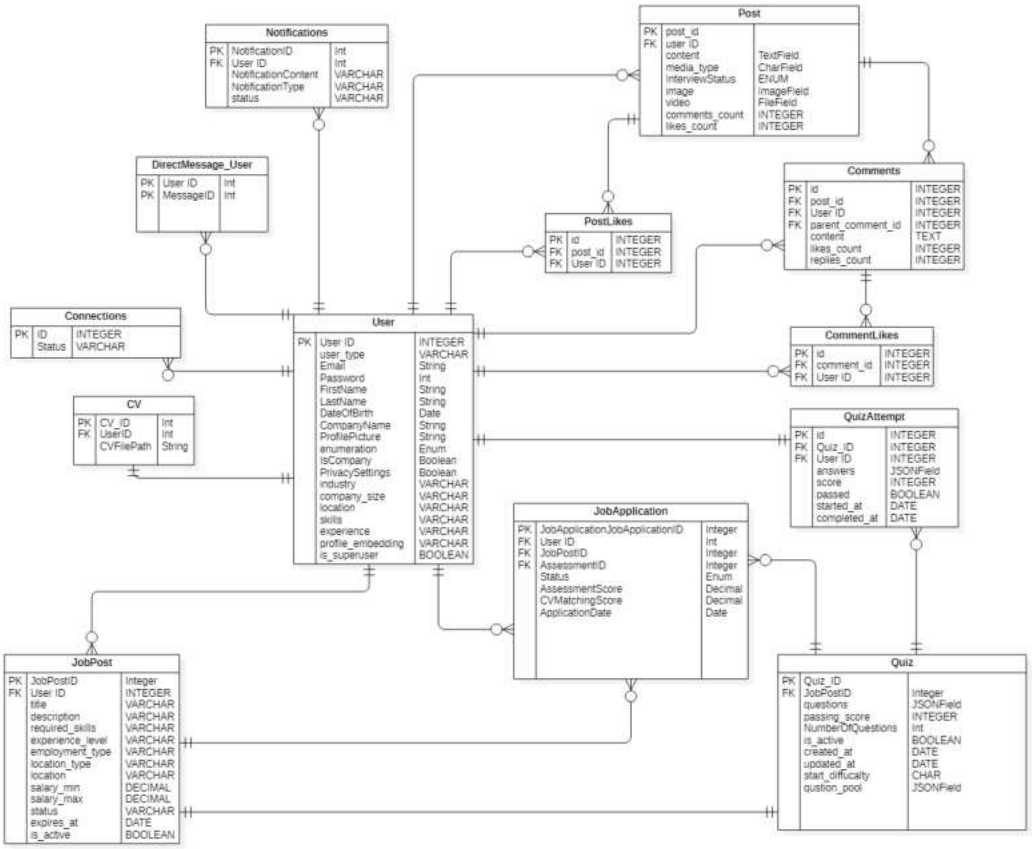


Figure 12 Entity Relationship Diagram

5. Initial Test Cases:

Table 15 Register Initial Test Case

Test ID	TC-001
Associated Use Case	Register
Title	User Registration
Objective	Verify that a new user (Job Seeker or Company) can register successfully.
Prerequisites	1. The system must be online and accessible. 2. The user must not have an existing account. 3. The user must be on the registration page.
Test Data	Job Seeker: - Email: testuser@example.com - Password: Test@1234 - First Name: John - Last Name: Doe - Date of Birth: 01-01-1995 Company User: - Email: company@example.com - Password: Secure@4567 - Company Name: Tech Solutions
Steps	1. The user navigates to the registration page. 2. The system displays the registration form with options for Job Seeker and Company User. 3. The user selects their type and enters the required details.

	4. The user submits the form. 5. The system validates the data (email format, password strength, uniqueness). 6. If all validations pass, the system creates an account. 7. The system redirects the user to the login page.
Pass Criteria	The system successfully creates the account and navigates the user to the login page.
Pass Criteria	The system displays an error message due to: - Invalid email format. - Weak password. -Email already registered.

Table 16 Login Initial Test Case

Test ID	TC-002
Associated Use Case	Login
Title	User Login
Objective	Verify that an existing user (Job Seeker or Company) can log in successfully.
Prerequisites	1. The system must be available. 2. The user must have an existing registered account. 3. The user is on the login page.
Test Data	Valid Credentials: - Email: user@example.com - Password: Valid@1234 Invalid Credentials:

	- Email: invalid@example.com - Password: WrongPass123
Steps	1. The user navigates to the login page. 2. The system displays the login form. 3. The user enters their email and password. 4. The user submits the form. 5. The system validates the credentials by checking: - If the email exists in the database. - If the password matches the stored credentials. 6. If validation is successful, the system authenticates the user. 7. The system redirects the user to the Home Page.
Pass Criteria	The system successfully logs in the user and navigates them to the Home Page.
Pass Criteria	The system displays an error message due to: - Incorrect email. - Incorrect password. - Unregistered email.

Table 17 Manage Social Interaction Initial Test Case

Test ID	TC-003
Associated Use Case	Manage Social Interaction
Title	User Interactions on Posts

Objective	Verify that a user (Job Seeker or Company) can interact with posts (like, comment, reply, and follow/unfollow) successfully.
Prerequisites	1. The user is logged into the system. 2. The system is online and accessible. 3. The home page is successfully loaded.
Test Data	Valid Interaction Actions: - Like a post - Comment on a post - Reply to a comment - Like a comment/reply - Follow/unfollow a user - Create a post with media
Steps	1. The user navigates to the Home Page. 2. The system displays posts from followed users and trending posts. 3. The user likes a post. 4. The user comments on a post. 5. The user replies to an existing comment. 6. The user likes a comment or reply. 7. The user edits or deletes their comments. 8. The user follows or unfollows another user. 9. If the user creates a new post, they can attach images or videos. 10. The system sends notifications when another user interacts with a post (likes, comments, replies).

Pass Criteria	- The system successfully records the interactions. - Notifications are sent to relevant users. - The user can see updates in real-time.
Pass Criteria	- The system fails to record likes, comments, or replies. - Notifications are not sent. - User interactions are not updated in real-time.

Table 18 Manage Recruitment Initial Test Case

Test ID	TC-004
Associated Use Case	Manage Recruitment
Title	Verify that a Company User can manage job postings, review applicants, and send job offers successfully.
Objective	Verify that a Company User can manage job postings, review applicants, and send job offers successfully.
Prerequisites	1. The Company User is logged into the system. 2. The system is online and accessible. 3. Active job listings or applicants exist in the system.
Test Data	Valid Job Actions: - Create a job posting - Delete a job posting - View job listings - View applicants - Review applicant profiles & CVs - View quiz scores - View AI-generated matching scores

	- Send a job offer
Steps	1. The Company User navigates to the Home Page. 2. The user creates a new job posting and submits it. 3. The system notifies suitable job seekers about the new job. 4. Job seekers apply for jobs by submitting applications. 5. The Company User views the list of applicants for a specific job. 6. The Company User selects an applicant and: - Views the applicant's profile - Views the uploaded CV - Checks the quiz score generated by the AI system - Reviews the matching score between job and applicant skills 7. If the applicant meets requirements, the Company User sends a job offer. 8. The system sends a job offer email notification to the selected job seeker.
Pass Criteria	- The system successfully creates a job post. - Job seekers receive notifications about new jobs. - Company Users can view applicants, CVs, and quiz/matching scores. - The job offer is sent successfully, and the applicant receives a notification.
Pass Criteria	- The system fails to save or display job postings. - Job seekers do not receive job notifications. - The Company User cannot view applicants or uploaded CVs. - The AI system does not generate a quiz or matching score. - The job offer is not sent.

Table 19 Apply For Job Initial Test Case

Test ID	TC-005
Associated Use Case	Apply for Job
Title	Job Application Process
Objective	Verify that a Job Seeker can search for jobs, take a quiz, and submit an application successfully.
Prerequisites	1. The Job Seeker is logged into the system. 2. The system is online and accessible. 3. Active job listings exist in the system.
Test Data	Valid Job Application Data: - Skills: Python, Django - Job Title: Backend Developer - Experience: 2+ years - Location Type: Remote - Employment Type: Full-time Valid Quiz Submission: - Score: 85% Valid CV Upload: - File: resume.pdf (PDF , DOC , DOCS)
Steps	1. The Job Seeker navigates to the Apply for Job section. 2. The user searches for jobs using filters (skills, title, experience, location type, employment type). 3. The system displays job postings matching the selected criteria. 4. The Job Seeker views job application details.

	5. The user takes a skill-based quiz. 6. The AI system evaluates the quiz and generates a score. 7. The user uploads their CV. 8. The system stores the application and marks it as submitted.
Pass Criteria	- The system successfully filters and displays relevant jobs. - The quiz is generated and evaluated by the AI system. - The application is successfully submitted and stored.
Pass Criteria	- The system fails to display job listings. - The quiz is not generated or evaluated. - The CV upload fails. - The application is not stored in the system.

Table 20 Manage Account Settings Initial Test Case

Test ID	TC-006
Associated Use Case	Manage Account Settings
Title	Update and Manage Account Settings
Objective	Verify that a Job Seeker or Company User can update their account settings successfully.
Prerequisites	1. The user is logged into the system. 2. The profile is new or incomplete. 3. The system is online and accessible.

Test Data	Valid Account Settings Data: - Communication Preferences: Email notifications enabled - AI Preferences: Enable job recommendations - Deleting User Profile
Steps	1. The user navigates to the Account Settings section. 2. The system displays options for managing settings. 3. The user selects an option To delete his account. - Communication preferences - AI recommendation preferences 4. The system validates the input and saves the changes. 5. The updated settings take effect immediately.
Pass Criteria	- The system successfully saves and applies the updated settings. - The changes are reflected immediately.
Pass Criteria	- The system fails to save the settings. - The user receives an error message for invalid data. - The profile remains empty despite valid updates.

Table 21 Manage Profile Initial Test Case

Test ID	TC-007
Associated Use Case	Manage Profile
Title	Update and Manage Profile Information
Objective	Verify that a Job Seeker or Company User can update their profile details successfully.

Prerequisites	1. The user is logged into their account. 2. The profile exists in the system. 3. The system is online and accessible.
Test Data	Valid Profile Data: - Email: newemail@example.com - Phone: +9876543210 - Profile Picture: profile.jpg (JPEG, PNG) - Experience: 5 years of back-end development - Skills: React, Django, Documentation - Privacy Setting: Show/Hide Contact Info
Steps	1. The user navigates to the Manage Profile section. 2. The system displays existing profile details. 3. The user selects an option and updates: - Email, phone number, profile picture, skills, Experience - Privacy settings (show/hide contact info) 4. The system validates the input and saves the changes. 5. The updated profile details are reflected immediately.
Pass Criteria	- The system successfully updates and displays the modified profile. - Privacy settings take effect immediately.

Pass Criteria	- The system fails to save the updated details. - Invalid inputs (e.g., incorrect email format, unsupported image format) trigger an error message. - Changes do not appear after submission.

Table 22 Setup Profile Initial Test Case

Test ID	TC-008
Associated Use Case	Setup Profile
Title	Complete Profile Setup
Objective	Verify that a Job Seeker or Company User can successfully complete their profile setup.
Prerequisites	1. The user is logged into their account. 2. The profile is new or incomplete. 3. The system is online and accessible.
Test Data	Valid Profile Setup Data: - Profile Picture: profile.jpg (JPEG, PNG) - Email: user@example.com - Phone: +1234567890 - Experience: Mid-level .Net - Skills: Python, Django, React - Location: New York, USA
Steps	1. The user navigates to the Setup Profile section. 2. The system prompts the user to complete missing details: - Upload Profile Picture & Location

	- Add Email & Phone Number - Add Skills & Experience 3. The user enters the required information and submits it. 4. The system validates and stores the data. 5. The profile setup is completed successfully.
Pass Criteria	- The system successfully saves and updates the profile. - The user can now interact with job postings and networking features.
Pass Criteria	- The system fails to store the profile data. - The user is prompted for missing mandatory fields. - Invalid inputs (incorrect email format, unsupported image type) trigger an error message.

Test ID	TC-010
Associated Use Case	Manage Account (CV Parsing)
Title	Upload Valid CV
Objective	Ensure user can upload a valid PDF CV
Prerequisites	User is authenticated and on profile edit page
Test Data	Valid PDF file under 5MB (AliCV.pdf)
Steps	1.Go to profile edit 2. Click upload CV 3.Select valid PDF 4.Submit
Pass Criteria	CV is uploaded, stored, and user receives success message

Test ID	TC-011
Associated Use Case	Manage Messaging
Title	Parse Uploaded CV
Objective	Ensure system parses uploaded CV and fills profile
Prerequisites	User has uploaded a valid CV
Test Data	----------
Steps	1.Go to profile edit 2. Click "Parse CV" 3.Waiting For Parse
Pass Criteria	Profile fields are auto-filled with extracted data; user sees success message

Test ID	TC-012
Associated Use Case	Manage Interview Schedualing
Title	Schedule Interview (Valid)
Objective	Ensure company can schedule interview for applicant
Prerequisites	Company user is authenticated, job and applicant exist
Test Data	Valid date/time, platform, interviewer (12/7/2025 , 9:00 AM , ZOOM, HR.Ali Alaswad)
Steps	1.Go to job details 2. Click "View Applicants" 3.Click "Schedule Interview" 4.Enter valid details 5.Submit
Pass Criteria	Interview is created, notifications sent via Email, interview appears in lists.

Test ID	TC-013
Associated Use Case	Manage Messaging
Title	Search for User
Objective	Ensure user can search for other users to start conversation
Prerequisites	User is authenticated
Test Data	Search query (e.g., "Ali Alasewad" or “G42”)
Steps	1.Go to messaging 2.Enter search query 3.View results
Pass Criteria	Users matching query are listed

Test ID	TC-013
Associated Use Case	Manage Messaging
Title	Start Conversation and Send Message
Objective	Ensure user can start conversation and send message
Prerequisites	User is authenticated, recipient exists
Test Data	Message text (For any User “Hi welcome”)
Steps	1.Search for user 2.Start conversation 3.Type message 4.Send
Pass Criteria	Message appears in conversation, recipient receives it in real time

6. Initial RTM Requirements Tractability Matrix:

Table 23 Initial RTM

REQ-ID	Use-case	Analysis	System Design	Detail Design	Implementation	Test Cases
FR-UM-001	UC-05	3.2.1.5				
FR-UM-002	UC-06	3.2.1.6				
FR-UM-003	UC-HIGH	3.2.1.22				
FR-UM-004	UC-07	3.2.1.7				
FR-UM-005	UC-21	3.2.1.21				
FR-UM-006	UC-HIGH	3.2.1.8				
FR-UM-007	UC-01.3.3	3.2.1.3				
FR-UM-008	UC-03	3.2.1.3				
FR-UM-009	UC-03	3.2.1.3				
FR-UM-010	UC-01.2.1	3.2.1.4				
FR-UM-011	UC-01.2.1.3	3.2.1.2				
FR-UM-012	UC-01.2.1.3	3.2.1.2				
FR-UM-013	UC-01.2.2	3.2.1.2				
FR-UM-014	UC-01.2.2.3	3.2.1.2				
FR-UM-015	UC-21	3.2.1.1				
FR-UM-016	UC-01.4.1	3.2.1.4				
FR-UM-017	UC-10	3.2.1.10				
FR-UM-018	UC-10	3.2.1.10				
FR-UM-019	UC-10	3.2.1.10				
FR-UM-020	UC-10.3	3.2.1.10				
FR-UM-021	UC-10.2	3.2.1.9				
FR-UM-022	UC-10.1	3.2.1.10				
FR-UM-023	UC-21	3.2.1.21				

FR-UM-024	UC-11	3.2.1.11				
FR-UM-025	UC-12.5	3.2.1.10				
FR-UM-026	UC-12.5.3	3.2.1.10				
FR-UM-027	UC-12	3.2.1.10				
FR-UM-028	UC-12.4	3.2.1.10				
FR-UM-029	UC-12.2	3.2.1.10				
FR-UM-030	UC-12.2.4	3.2.1.10				
FR-UM-031	UC-12.2.1	3.2.1.10				
FR-UM-032	UC-12.2.2	3.2.1.10				
FR-UM-033	UC-12.2.3	3.2.1.10				
FR-UM-034	UC-12.2.3.1	3.2.1.10				
FR-UM-035	UC-12.2.1	3.2.1.10				
FR-UM-036	UC-12.2.2	3.2.1.10				
FR-UM-037	UC-12.3	3.2.1.12				
FR-UM-038	UC-12	3.2.1.12				
FR-UM-039	UC-12	3.2.1.12				
FR-UM-040	UC-12	3.2.1.12				
FR-UM-041	UC-12.3	3.2.1.12				
FR-UM-042	UC-12.1	3.2.1.12				
FR-JM-001	UC-18	3.2.1.18				
FR-JM-002	UC-15	3.2.1.16				
FR-JM-003	UC-16	3.2.1.23				
FR-JM-004	UC-10.4	3.2.1.24				
FR-JM-005	UC-10.4	3.2.1.23				
FR-JM-006	UC-18	3.2.1.17				
FR-JM-007	UC-20	3.2.1.20				
FR-NTF-001	UC-13	3.2.1.13				

FR-NTF-002	UC-13	3.2.1.13				
FR-NTF-003	UC-13	3.2.1.13				
FR-NTF-004	UC-13	3.2.1.13				
FR-NTF-005	UC-13	3.2.1.13				
FR-NTF-006	UC-13	3.2.1.13				
FR-NTF-007	UC-13	3.2.1.13				
FR-NTF-008	UC-13	3.2.1.13				
FR-NTF-009	UC-13	3.2.1.17				
FR-NTF-010	UC-13	3.2.1.13				
FR-NTF-011	UC-NT	3.2.1.27				
FR-MS-01	UC-10.5	3.2.1.27				
FR-MS-02	UC-10.5.1	3.2.1.27				
FR-MS-03	UC-10.5.1.2	3.2.1.27				
FR-MS-04	UC-10.5.1.3	3.2.1.27				
FR-MS-05	UC-10.5.2	3.2.1.27				
FR-AI-001	UC-01.4.2	3.2.1.8				
FR-AI-002	UC-HIGH	3.2.1.8				
FR-AI-003	UC-HIGH	3.2.1.8				
FR-AI-004	UC-09	3.2.1.9				
FR-AI-005	UC-HIGH	3.2.1.8				
FR-AI-006	UC-09	3.2.1.9				
FR-AI-007	UC-09	3.2.1.9				
FR-AI-008	UC-09	3.2.1.9				
FR-AI-009	UC-HIGH	3.2.1.8				
FR-AI-010	UC-HIGH	3.2.1.8				
FR-AI-011	UC-HIGH	3.2.1.8				

FR-AI-012	UC-HIGH	3.2.1.8				
FR-AI-013	UC-AI	3.2.1.29				
FR-SK-001	UC-18.1	3.2.1.17				
FR-SK-002	UC-16.1.1.2	3.2.1.17				
FR-SK-003	UC-16.1.1.1	3.2.1.14				
FR-SK-004	UC-16.1.1.1	3.2.1.14				
FR-SK-005	UC-16.1.1.1.1	3.2.1.14				
FR-RM-001	UC-19.1.1.5	3.2.1.17				
FR-RM-002	UC-16.1.1.1.1	3.2.1.14				
FR-RM-003	UC-19.1.1	3.2.1.19				
FR-RM-004	UC-19.1.1.3	3.2.1.19				
FR-RM-005	UC-19.1.1.5	3.2.1.19				
FR-RM-006	UC-19.1.1.1	3.2.1.17				
FR-RM-007	UC-19.1.1.6	3.2.1.28				
FR-RM-008	UC-19.1.1.7	3.2.1.28				
FR-Admin-DASH-001		3.2.1.25	-	-		
FR-Admin-DASH-002		3.2.1.26	-	-		

Chapter5 System Design

1. Introduction:

The system design phase is a critical component of the software development lifecycle (SDLC), serving as the blueprint that guides the development and implementation of the system. This chapter provides an in-depth analysis of the design process, including System Architecture, Detailed Class Diagram, Design Patterns, and Solid Principles. It aims to ensure the efficient collection, processing, and utilization of data within the system while maintaining scalability, security, and user experience.

This chapter elaborates on system architecture by defining key system components and their interactions. Additionally, it covers the detailed design, which focuses on breaking down system components into smaller modules, ensuring a modular and structured approach to development.

By the end of this chapter, readers will gain a comprehensive understanding of how the system is structured, the rationale behind design decisions, and how different components interconnect to deliver the intended functionality. This ensures that the system is built with efficiency, maintainability, and flexibility, allowing for future enhancements and adaptations as needed.

2. System Architecture:

In this section we will discuss system design where system design involves defining the overall structure and architecture of the software, ensuring that all components work seamlessly together.

This phase establishes the blueprint for efficient data flow, interaction, and integration within the system.

The first constraints to take into consideration when building an architecture for a software system are **the non-functional requirements (quality attributes),** so first we will start with the non-functional requirements of our system

1. **Extensibility:** The system should demonstrate high extensibility to upgrade the AI components with minimal impact on the existing codebase or system integrity. (Average of 10% changes when adding or upgrading features).

2. **Security:** The system should be able to protect its all-sensitives users' data successfully, this includes using authentication, authorization, strong password, and good architecture.

3. **Scalability**: The system analyzes a total of 10,000 reviews and their associated requirements per hour across all projects, with each project limited to 100 requirements.

4. **Performance**: The system should return the result of a requested operation in an average of 1 minute. Most existing operations respond within a few seconds. However, specific operations require more processing time:

AI-generated quizzes (for job applications) may take up to 2 minutes to generate questions.

User profile updates trigger an embedding process that takes up to 1 minute to generate personalized recommendations.

Analysis of project reviews and requirements scales to 10,000 reviews per hour.

System decomposition - Component diagram:

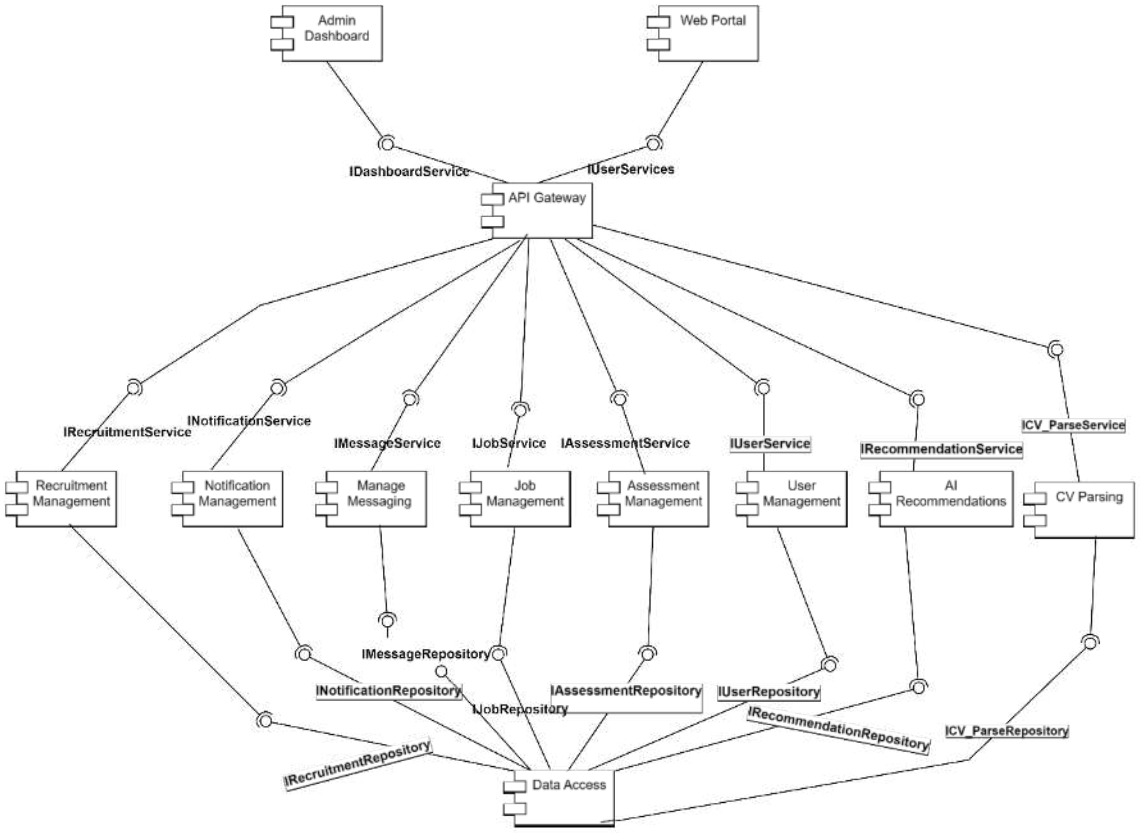


Figure 13 Component Diagram

Component's functionalities:

1. Web Portal:

The Web Portal serves as the primary user interface for both Job Seekers and Company Users. It allows users to interact with the system by providing functionalities such as:

- User Authentication & Account Management: Users can log in, register, and manage their profiles.
- Job Search & Application: Job Seekers can search for job listings using filters, view job details, and apply for jobs.
- Social Interaction Features: Users can view posts, create posts, comment, like, and reply to existing interactions.
- AI-Skill Based Quiz Taking: Job Seekers can take skill-based quizzes related to job postings, with AI-driven scoring.
- Recommendation System Integration: The portal provides personalized job and connection recommendations.
- Dashboard & Analytics: Users can view job application statuses, quiz results, and other key metrics.

2. API Gateway

The API Gateway acts as the centralized communication layer between the Web Portal and Backend Services. It is responsible for:

- Routing API Requests: It forwards requests to the appropriate service (e.g., Job Management, Recruitment Management, and User Management).

- Authentication & Authorization: Ensures that API calls are made by authenticated users and that access control policies are enforced.
- Load Balancing & Rate Limiting: Helps distribute traffic efficiently to avoid overloading backend services.
- Error Handling & Logging: Provides consistent error responses and logs API activities for debugging and monitoring.
- Security Layer: Protects the backend services from unauthorized access.

3. Recruitment Management

The Recruitment Management Service is responsible for handling job-related operations performed by Company Users. The key functionalities include:

- Posting New Job Listings: Allows companies to create and manage job postings, specifying required skills, experience levels, and employment types.
- Managing Job Applications: Provides an interface for viewing applicants, filtering applications, and reviewing submitted CVs.
- Deleting Job Listings: Enables employers to remove outdated or filled job postings.
- AI-Based Job Matching: Facilitates AI-driven matching scores between job seekers and job postings.
- Sending Interview Schedualing : The Company user can send for the selected Job seeker interview via email notidfication includes Date and Time , Platform (Google Meet or ZOOM), and the interviewer.

- Sending Job Offers: Once an applicant is selected, the system sends job offer notifications.
- Integration with Notification System: Ensures job seekers are notified about new job postings, application updates, and Send job offers

4. Notification Management

The Notification Management Service is responsible for sending and managing notifications within the system. The main functions include:

- Real-time Alerts: Users receive instant notifications when relevant actions occur, such as:
    - A job seeker receives a job offer.
    - A job seeker gets a new job application posting.
    - A job seeker gets a new connection request.
    - Interactions on social posts (likes, comments, replies).
- Email Notifications: Critical updates (such as job offer and password resets) are sent via email.
- Integration with Other Services: This component interacts with Recruitment Management, Social Interaction, and AI Recommendation Services to trigger notifications.

5. Job Management

The Job Management Service facilitates job-related processes in the system, providing functionalities such as:

- Job Listings & Filtering: Job Seekers can search for jobs based on:

- Skills
- Job Title
- Experience Level
- Location Type
- Employment Type

- Viewing Job Details: Users can view job descriptions, requirements, salaries, and other relevant details.
- Managing Job Status: Jobs can be marked as active, expired, or removed based on recruitment progress.
- Integration with AI-Based Matching: Provides a recommended jobs to be sorted first of job listing with a chip for recognizing the recommended job application.
- Application Process Handling: Ensures that users can apply for jobs, submit CV, and receive responses from Company.

6. Assessment Management

The Assessment Management Service is responsible for generating and evaluating skill-based quizzes for job seekers. The functionalities include:

- Quiz Generation: When a company user fills out the job application form and creates a job, the system generates a quiz based on the job's required skills using the LLaMA-3.2-3B-Instruct model."
- Scoring & Evaluation: The AI system calculates the quiz score and provides it to the job seeker.

- Fraud Prevention & Attempt Limits: Users have a limited number of attempts per quiz, and cheating behaviors (e.g., switching to another browser tab multiple times) are detected. If a user is caught attempting to cheat 3 times, the system will ban them from applying for jobs for 24 hours.
- Storing Quiz Results: Quiz scores are saved and associated with the job application for employer review.
- Integration with Job Management: Company can view quiz scores alongside job applications to assess candidates effectively.

7. User Management

The User Management Service is responsible for handling user-related functionalities, including:

- User Registration & Authentication: Enables Job Seekers and Company Users to create accounts and log in securely.
- Profile Management: Users can update personal details, profile pictures, skills, and contact information.
- Privacy & Security Settings: Users can manage:
    - Profile visibility (public/private).
    - Contact details privacy (email, phone number visibility).
- Social Interaction Handling: Facilitates:
    - Following/unfollowing users.
    - Managing connections between job seekers and Company.

- Account Recovery & Security Measures: Includes:
    - Password reset functionality.
    - Secure logouts.

8. AI Recommendation

The AI Recommendation Service provides personalized recommendations for job seekers and companies. The main features include:

- Job Recommendation for Job Seekers:
    - Uses AI algorithms to recommend relevant job postings based on a user's skills, experience, and past job applications.
- Social Feed Recommendations:
    - Suggests connections (people to follow), relevant posts, and skill-based content.
- Integration with Other Services:
    - Works with Job Management, User Management, and Social Interaction features to refine suggestions.

9. Manage Messaging:

The manage messaging service provide for the users to make a conversations with each others by sending/reciving and edit/delete messages at the real-time

10. CV Parsing :

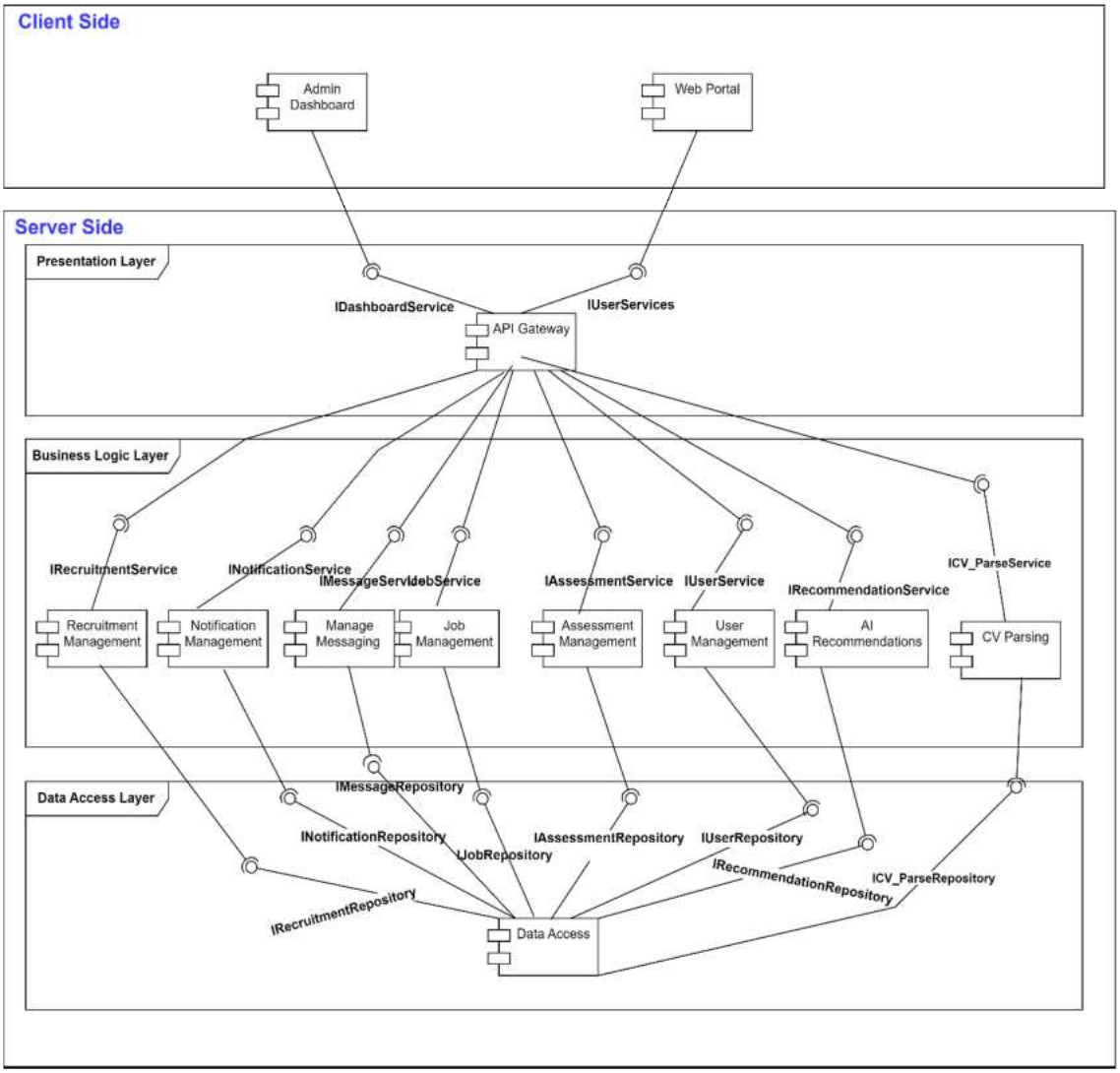
The cv parsing service provide for the user to upload his cv at his profile then The system parser his cv and Fill the information with the extracted information From this CV and fill it automatically with the corrected Fields

9. Data Access

The Data Access Layer (DAL) acts as the centralized storage and retrieval system for all backend services. It is responsible for:

- Handling Database Interactions:
    - Manages read/write operations for users, jobs, applications, posts, notifications, and assessments.
- Ensuring Data Consistency:
    - Prevents data duplication and maintains transaction integrity.
- Optimizing Query Performance:
    - Uses efficient indexing, caching, and pagination to optimize response times.
- Security & Compliance:
    - Ensures secure data handling, including encryption for sensitive user information (passwords, personal details).
- Providing Repositories for Services:
    - Each service (e.g., UserService, JobService, RecruitmentService) interacts with dedicated repositories to fetch/update data efficiently.

System Architecture:



3. Architectural Style: Layered Architecture with Client-Server Model

The system is structured using two key architectural patterns:

A. Client-Server Architecture

- The system follows a client-server model, where the frontend (client) and backend (server) communicate through RESTful APIs.

- The client-side application is built using the React framework, while the server-side application is developed using the Django framework.
- This separation allows for independent development and scaling of the frontend and backend, ensuring a responsive user experience.

B. Layered Architecture (Applied on the Server-Side)

The backend is divided into multiple layers, where each layer serves a distinct purpose. This ensures separation of concerns, enhanced security, and easier maintenance.

The key layers of the server-side include:

1. Presentation Layer (API Layer)
2. Business Logic Layer (Service Layer)
3. Data Access Layer (Repository Layer & ORM)

4. Client-Side Architecture (Frontend)

The client-side (frontend) application is built using React, a component-based JavaScript framework. The key aspects of the frontend architecture include:

- User Interface (UI): Provides an interactive and dynamic UI for Job Seekers and Company Users.
- State Management: Uses React Context API to manage global states across the application, including: User authentication, profile data, Job listings, filters, recommendations, Social posts and interactions.

  Real-time notifications, updates, Profile following system and application-wide loading and error states.
- API Communication: Uses Axios for RESTful API communication with the Django backend, including:

  - Centralized API services

- Interceptors for authentication
- Error handling and response processing

- Component-Based Design: Implements a modular architecture with:
  - Reusable UI components (JobCard, ProfileSummary, etc.)
  - Feature-specific components (Jobs, Profile, Social)
  - Common components for shared functionality
  - Context-based state management integration
- Security Features: Implements JWT authentication and protected routes to ensure secure access to user data.

The client-side application is responsible for handling all user interactions, sending requests to the backend via HTTP APIs, and rendering dynamic content based on the responses.

3. Server-Side Architecture (Backend)

The backend application is developed using Django, a high-level Python web framework. It is structured using a layered architecture to promote scalability, security, and maintainability.

A. Presentation Layer (API Layer)

The Presentation Layer is responsible for handling all incoming and outgoing API requests. It acts as a bridge between the frontend and backend services.

Main Responsibilities:

- Exposes RESTful APIs:
  - Implements Django REST Framework (DRF) to define API endpoints.
  - Handles CRUD (Create, Read, Update, and Delete) operations for various system entities (users, jobs, applications, assessments, etc.).

- Handles Authentication & Authorization:
    - Uses JWT-based authentication for securing API requests.
    - Implements role-based access control (RBAC) to restrict API access.
- Request Validation & Serialization:
    - Uses serializers to validate incoming data before passing it to the business logic layer.
    - Formats API responses to ensure consistency across the system.
- Error Handling & Logging:
    - Implements error-handling middleware to return structured error messages.
    - Logs API requests for monitoring and debugging.

B. Business Logic Layer (Service Layer)

The Business Logic Layer is the core of the backend application, containing all the main logic and operations performed by the system.

Main Responsibilities:

- Implements Business Rules:
    - Defines core functionalities such as job application workflows, AI-based recommendations, and assessment processing.
- Manages AI Processing:
    - Handles quiz generation, matching score calculations, and job recommendations based on AI-driven algorithms.
- Processes Job Applications:

- Ensures that job seekers can apply for jobs and companies can review applications.

- Manages Social Features:
  - Controls post creation, comments, likes, follows/unfollows, and notifications.
- Security & Data Integrity:
  - Implements password hashing, access control rules, and validation checks before updating data.

Example Business Logic Implementations:

1. Job Matching Algorithm:
   - Uses Natural Language Processing (NLP) models to compare job descriptions with user skills.
   - Assigns matching scores to rank job seekers for specific job postings.
2. Quiz Scoring System:
   - Evaluates job seeker performance based on multiple-choice questions.
   - Calculates a quiz score that is attached to job applications.
3. Recommendation System:
   - Suggests jobs for job seekers and potential candidates for recruiters.
   - Suggests potential knows people based on mutual connections or sharing the same interests
   - Uses AI models trained on user interactions and previous applications.

4. Data Access Layer (Repository Layer & ORM)

The Data Access Layer (DAL) is responsible for retrieving, storing, and managing data. It ensures a structured interaction with databases using Django's Object-Relational Mapping (ORM).

Main Responsibilities:

- Manages Database Queries Efficiently:
  - Uses optimized SQL queries and indexing to retrieve data quickly.
  - Implements query caching for frequently accessed data.
- Ensures Data Consistency:
  - Uses database transactions to prevent data corruption.
  - Maintains referential integrity between different entities (e.g., users, jobs, applications).
- Implements Repository Pattern:
  - Separates database queries from business logic, making the system more modular and maintainable.
- Handles File Storage:
  - Stores user-uploaded CVs, profile pictures, and media files in a secure storage system.

5. Summary & Benefits of the Architecture

By following a layered client-server architecture, the system ensures:

1. Modularity & Maintainability

- Each layer handles a specific responsibility, making the system easier to update and scale.

2. Security & Access Control
   - JWT authentication, role-based permissions, and data encryption secure user information.
3. Scalability & Performance Optimization
   - The API Gateway and Load Balancing ensure the system can handle high traffic efficiently.
4. AI-Powered Decision Making
   - The AI-driven matching system and quiz evaluation enhance recruitment accuracy and efficiency.

3. Detailed design for the system components:

In this section we will dive into the detailed design for each component of the system, also mentions the design principles and patterns used to build a strong software system that fulfills its requirements.

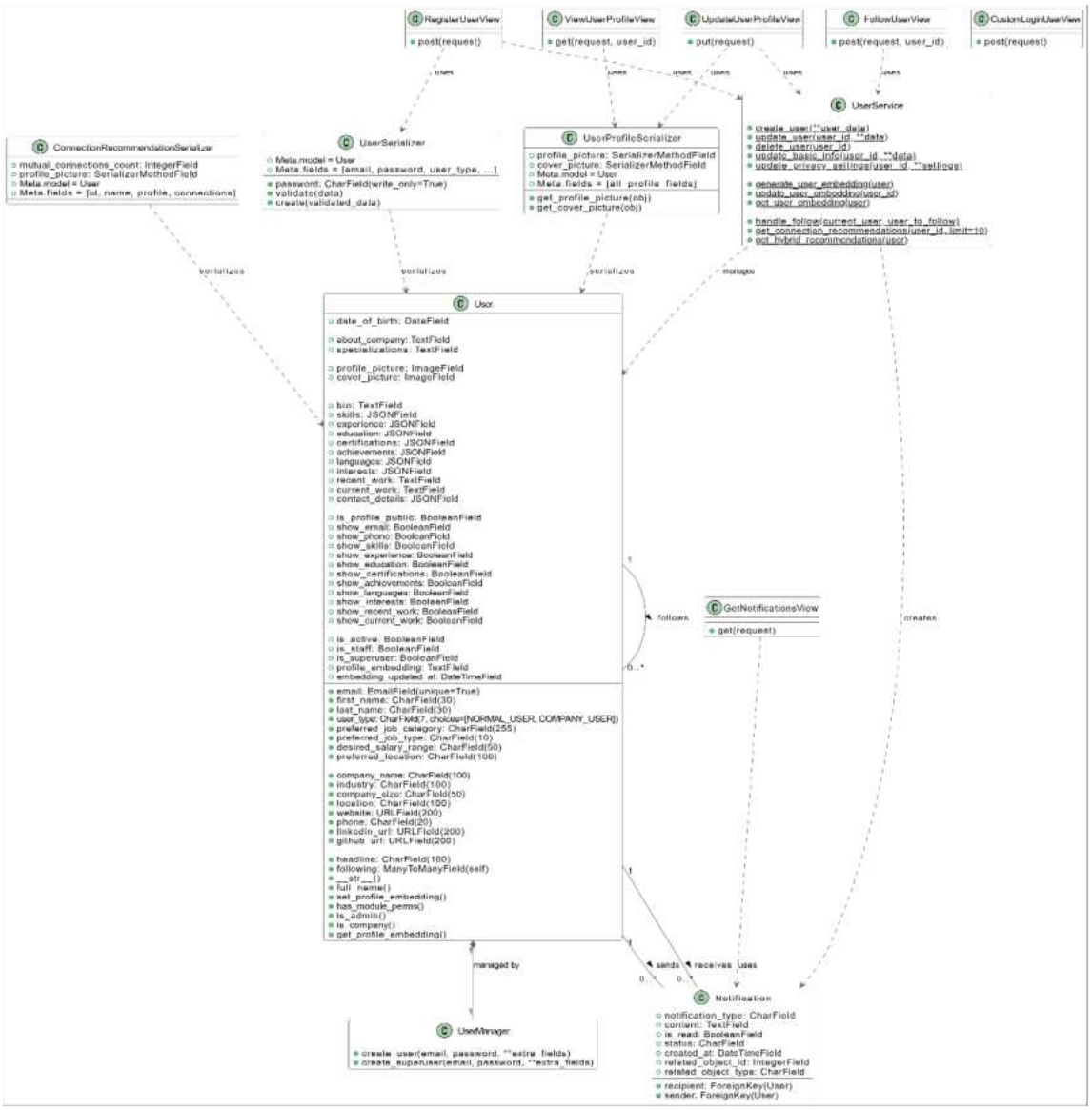


Figure 15 User Management - Details Class Diagram

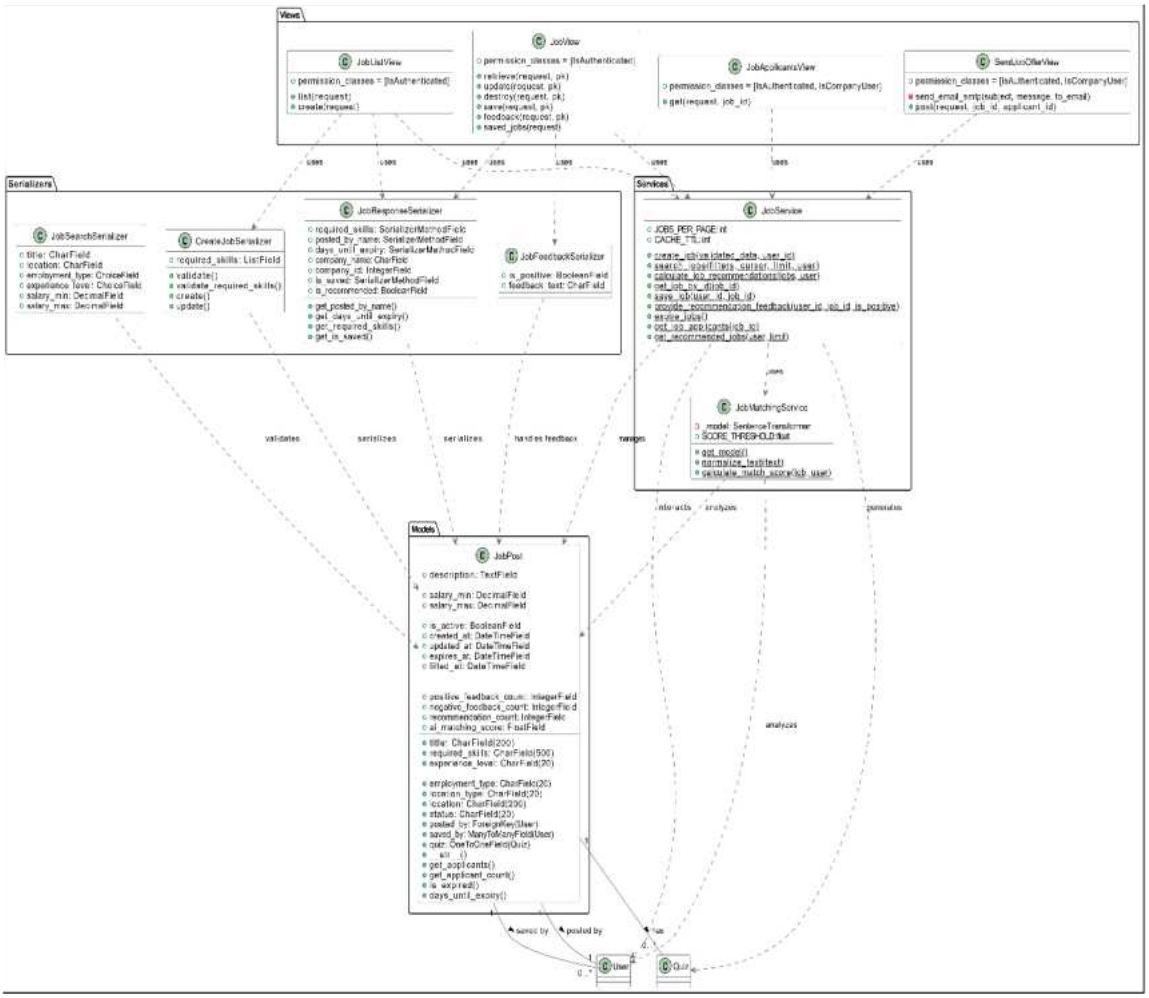


Figure 16 Job Management - Detailed Class Diagram

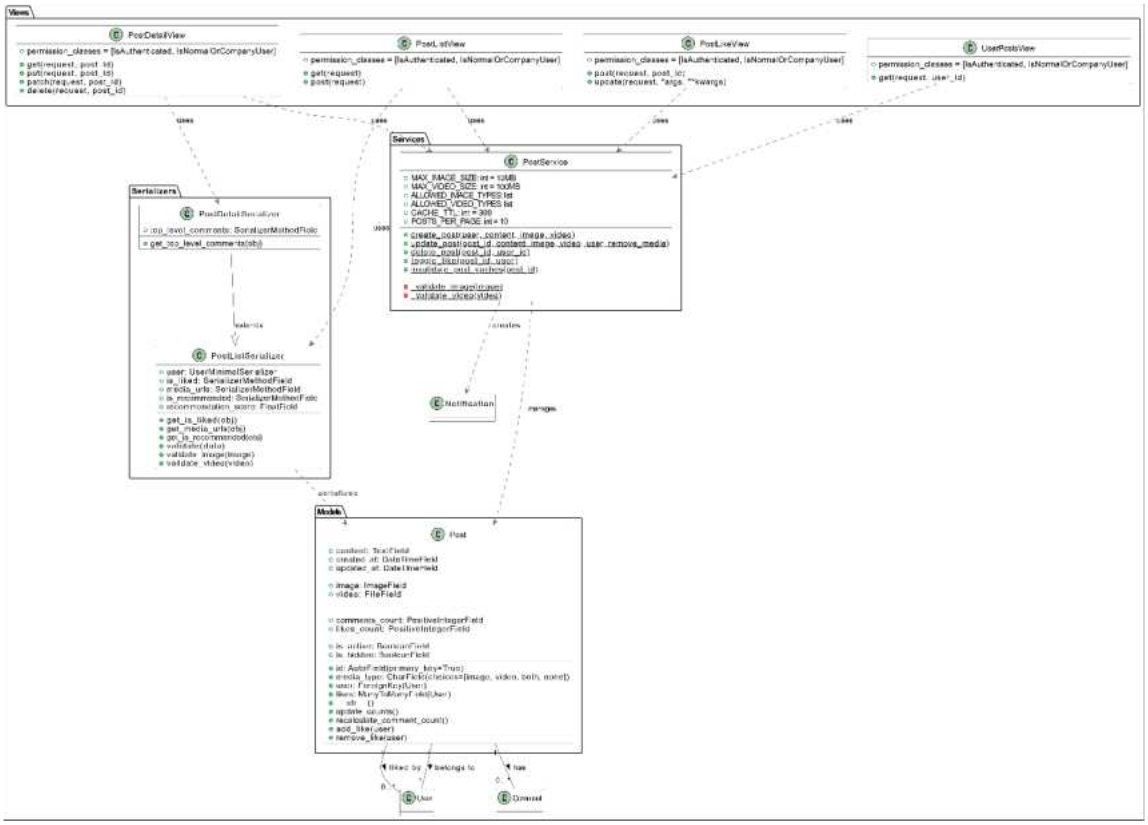


Figure 17 Posts Management – Detailed Class Diagram

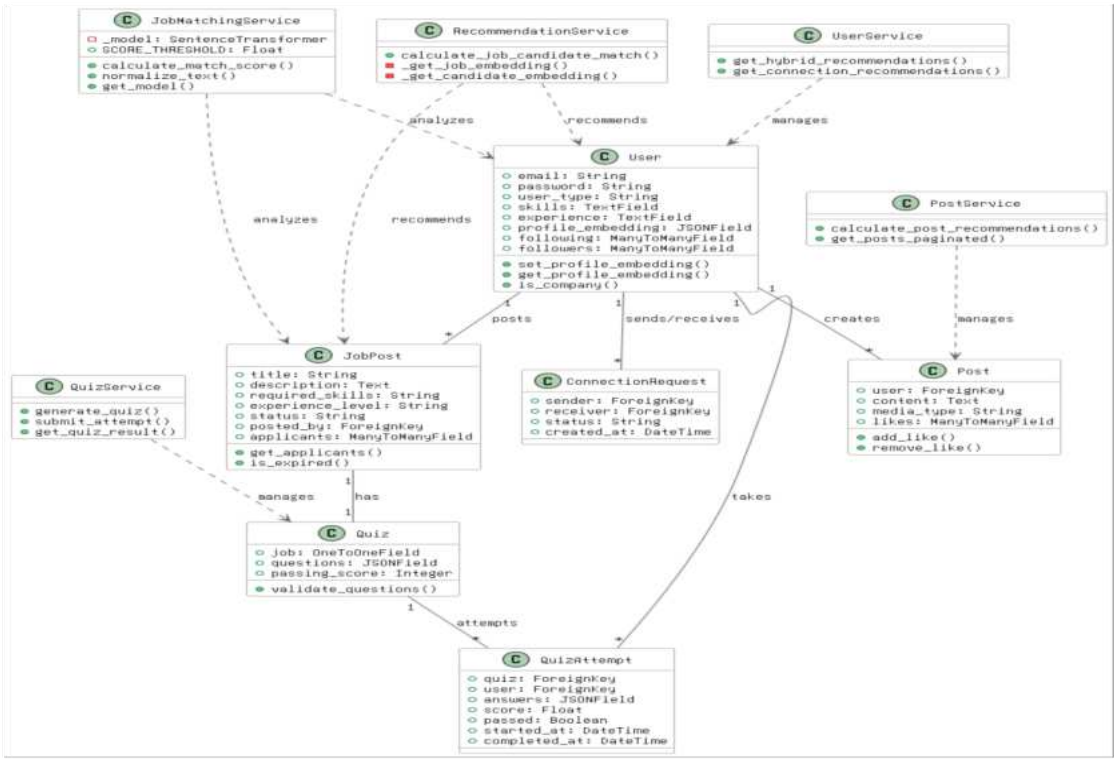


Figure 18 AI Recommendation - Detailed Class Diagram

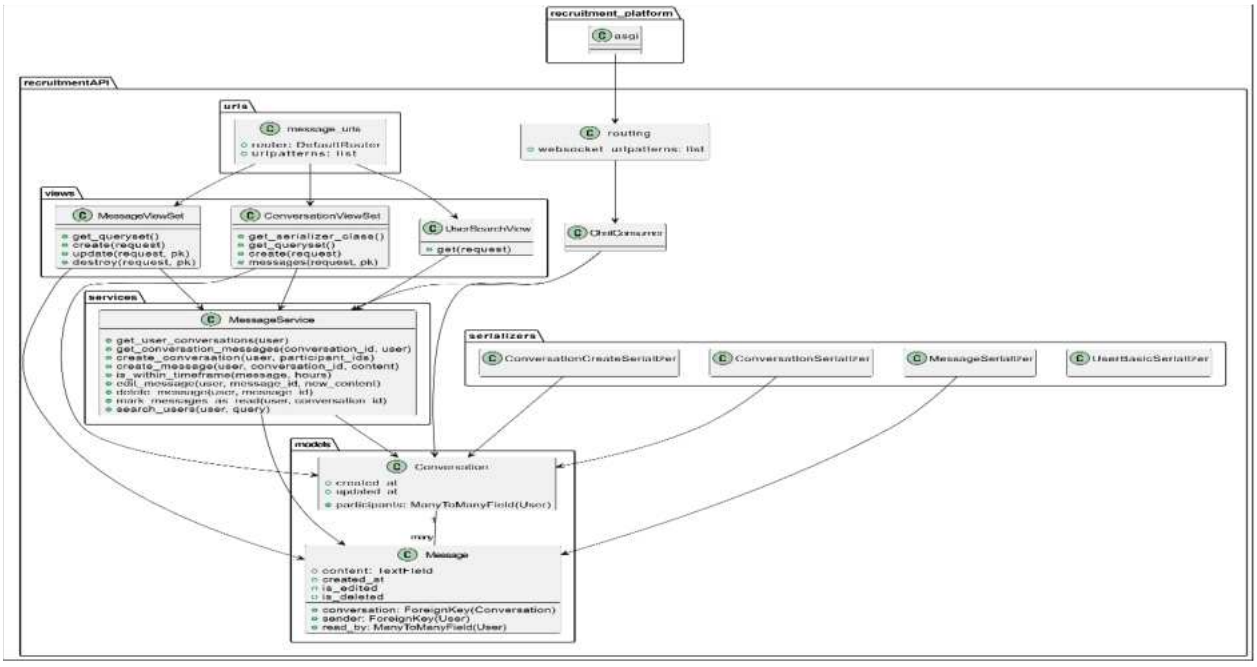


Figure 45 Manage Messaging Detailed Class Diagram

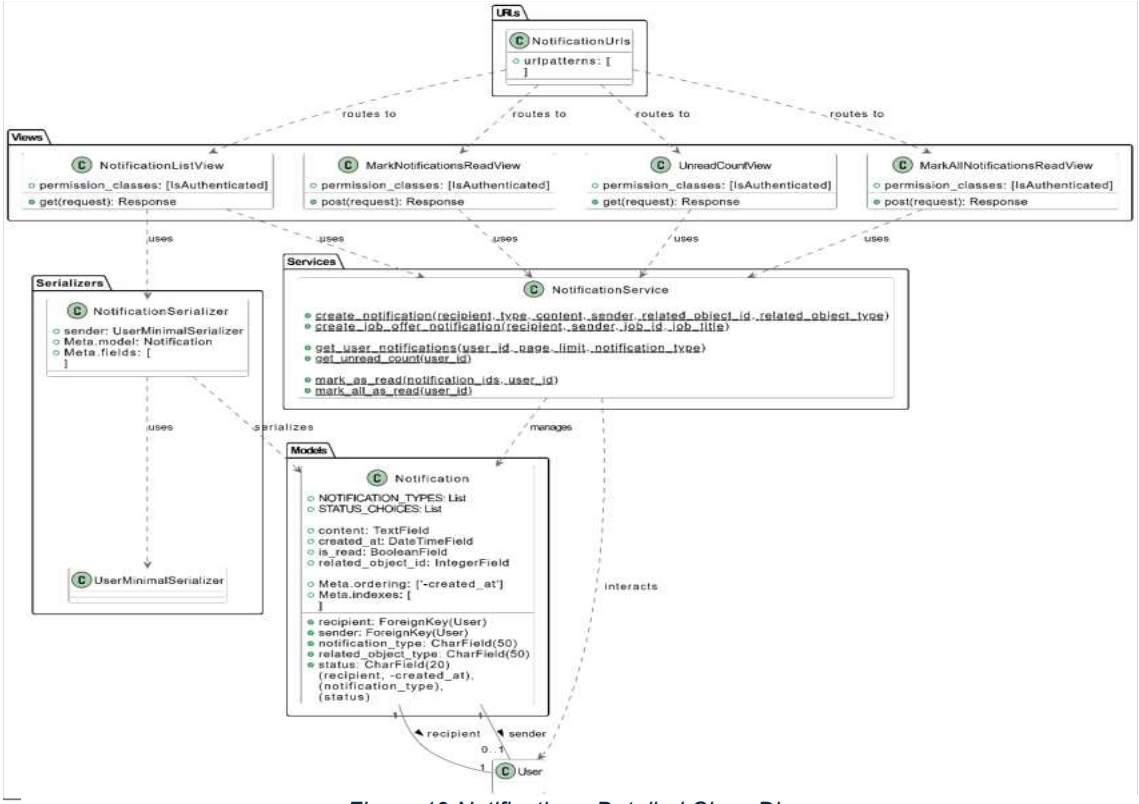


Figure 19 Notification - Detailed Class Diagram

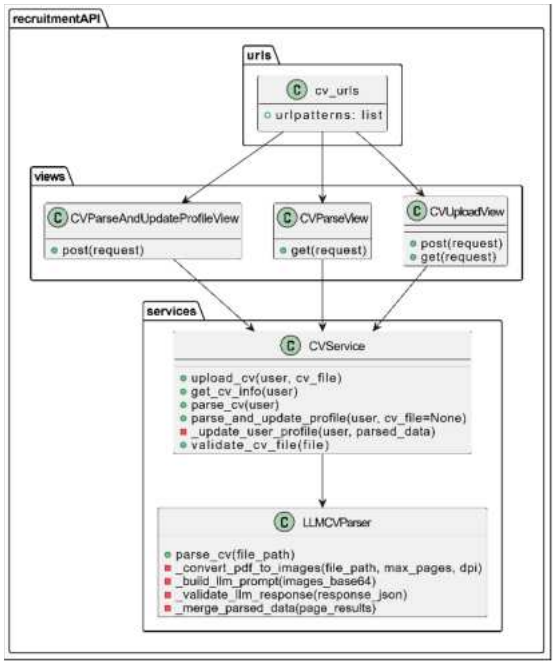


Figure 44 CV Parsing Detailed Class Diagram

Used design patterns:

1. **Posts Management:**

In this component, particularly in handling posts and their content processing, we implement a pattern similar to the Singleton Pattern (though not a true Singleton) for the Sentence Transformer model, which is a resource-intensive machine learning model used for text embedding and similarity calculations. The pattern is crucial here because initializing multiple instances of Sentence Transformer would be computationally expensive and memory-intensive, as each instance loads large model weights into memory. While our implementation uses a shared class variable rather than a private constructor, it achieves similar benefits to a Singleton as initializing multiple instances of Sentence Transformer would be computationally expensive and memory-intensive, as each instance loads large model weights into memory. By using a class-level variable with lazy initialization through get_model(), we optimize resource usage while maintaining consistent access to the model's functionality. By ensuring only one instance exists throughout the application's lifecycle through lazy initialization, we optimize resource usage while maintaining consistent access to the model's functionality. This is especially important when handling multiple post operations simultaneously, such as content analysis, recommendation generation, or similarity comparisons.

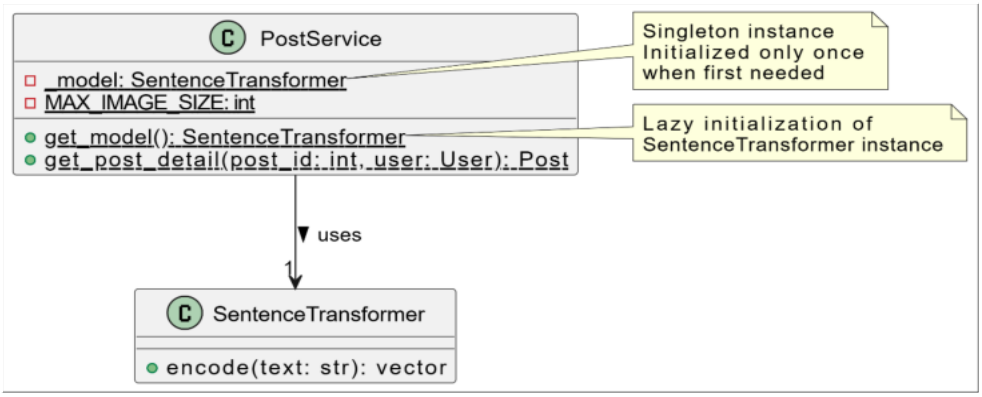


Figure 20 Post Management – Singleton

2. User Management component:

In this component, particularly in handling user access control and authorization, we implement the Strategy Pattern through Django REST Framework's permission classes. This pattern is essential as it allows different API endpoints to use different permission checking strategies based on user types (Normal User, Company User, or both). The implementation follows the classic Strategy Pattern structure where BasePermission serves as the strategy interface, defining the has_permission() method that all concrete strategies must implement. We have three concrete strategy classes (IsNormalUser, IsCompanyUser, IsNormalOrCompanyUser) that encapsulate different permission checking algorithms while adhering to the same interface. The APIView context class can then utilize any of these strategies through its permission_classes list, making our permission system both flexible and maintainable. This allows us to easily add new permission strategies without modifying existing code, as demonstrated by the additional IsJobOwner permission class that implements both has_permission() and has_object_permission() methods..

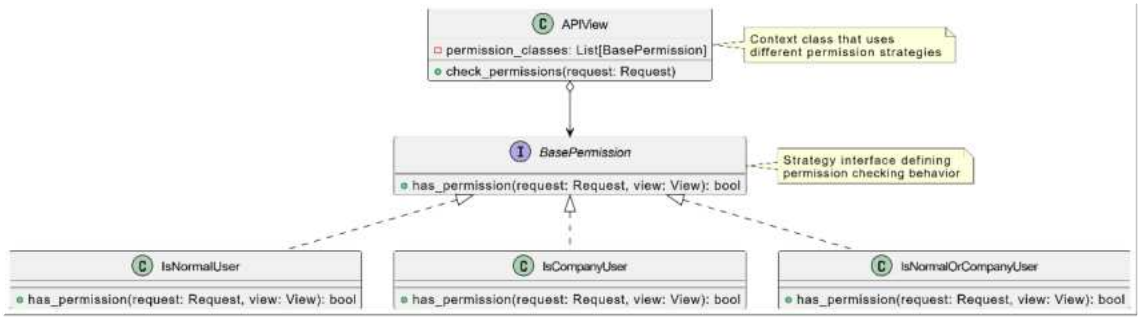


Figure 21 Strategy Pattern

3. Notification Management component:

In our recruitment platform, we implement the Observer Pattern to handle the notification system, where various services (PostService, JobService, ConnectionService) act as subjects that trigger notifications for different events. The NotificationService acts as a centralized mediator that creates and manages these notifications through its create_notification method. When significant events occur (like creating a post, sending a job offer, or making a connection request), the respective services notify users by calling appropriate NotificationService methods (e.g., create_notification, create_job_offer_notification). The User model serves as the observer, receiving notifications that are stored in their notifications_received list. This pattern effectively decouples the notification logic from the core business logic, as demonstrated by specialized notification creation methods like create_job_offer_notification that handle specific notification types while maintaining a consistent interface. The implementation includes features for notification management such as marking notifications as

read (mark_as_read, mark_all_as_read) and tracking unread notifications (get_unread_count).

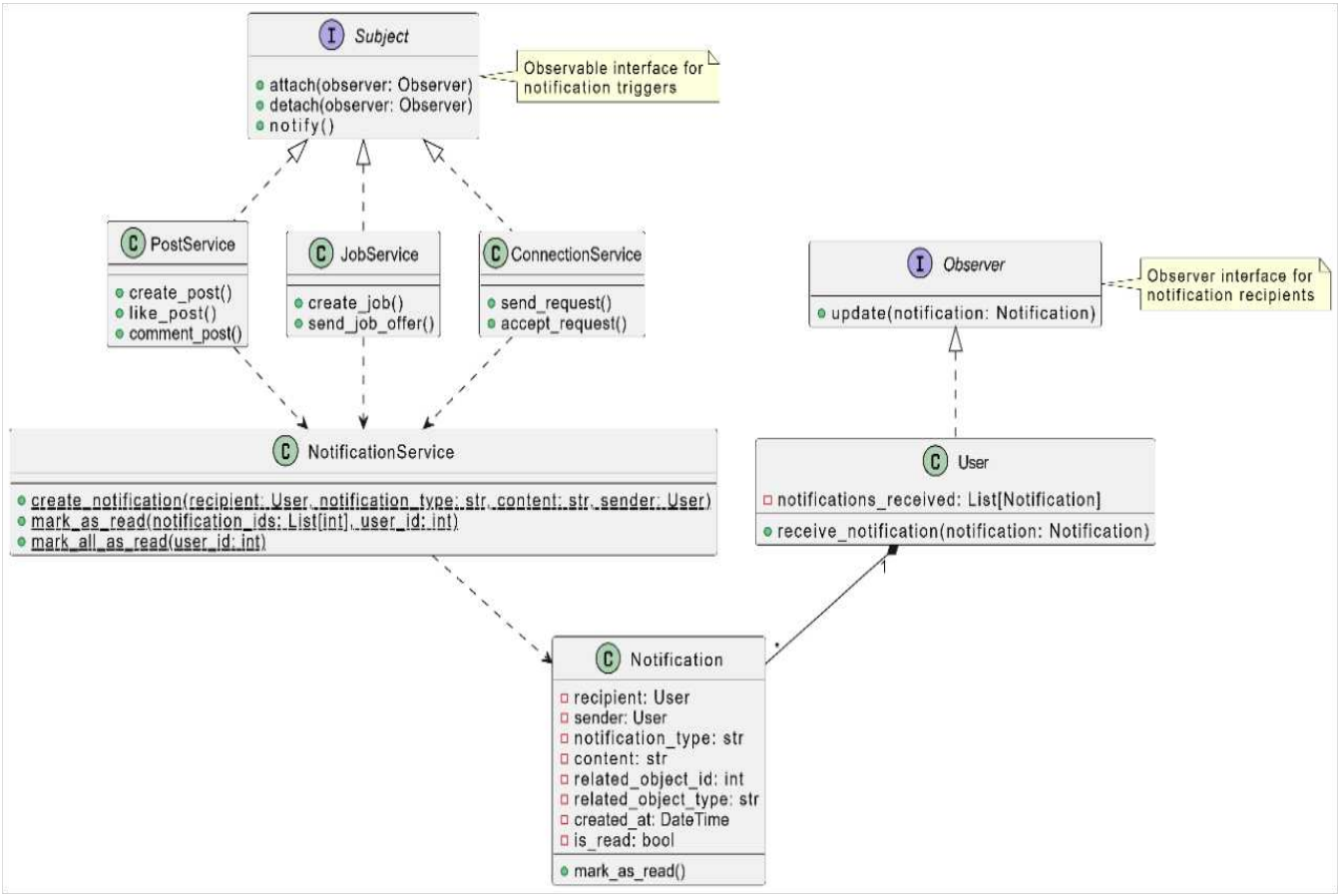


Figure 22 Observer Patterns – Notification Management

Quiz Generation Workflow

The **quiz generation service** follows these steps:

1. **Retrieves job details** (title, description, skills, experience level).
2. **Constructs a structured prompt** instructing the LLM to generate **10 multiple-choice questions**.
3. **Sends an HTTP request** to the LLM endpoint at http://localhost:1234/v1/chat/completions.
4. **Validates the response**:
    - Confirms **JSON structure** is intact.
    - Ensures **10 questions** are generated.
    - Checks that each question contains **4 complete answer choices**.
5. **Assigns metadata** to questions:
    - **Correct answer always at index 0**.
    - **Unique ID for frontend tracking**.
6. **Shuffles the questions** before saving to the database.

Job Matching Workflow

The **job-matching process** involves:

1. **Text Normalization**:
    - Converts all text to lowercase.

- Removes extra whitespace.
- Handles null values.

2. **Weighted Embedding Processing**:
   - Job titles and skills are **repeated** in the embedding input to **increase their influence**.
   - Candidate profiles are **combined from skills and experience** for more accurate matching.
3. **Computing Cosine Similarity**:
   - Uses the formula:

$$Cosine\ Similarity = \frac{A\ .B}{||A||\ \times\ ||B||}$$

where:

- A = Job embedding
- B = Candidate embedding

4. **Thresholding:**

   Candidates with a similarity score above 30.0 are considered potential matches.

Best Practices Implemented:

1. Separation of Concerns
    - Clear separation between models, views, and services
    - Dedicated serializers for data transformation
    - Separate permission classes
2. DRY (Don't Repeat Yourself)
    - Reusable services
    - Base serializers
    - Common permission classes
3. SOLID Principles
    - Single Responsibility (separate services for different functionalities)
    - Open/Closed (extensible permission classes)
    - Interface Segregation (specific serializers for different use cases)
    - Dependency Inversion (service layer abstraction)
4. Security Practices
    - JWT Authentication
    - Permission-based access control
    - Password hashing
    - Email verification

4. Database Design:

SQL Database – MySQL: In our system, MySQL is utilized for storing operational data related to accounts, projects, and requirements. This data is highly structured, making MySQL an ideal choice due to its efficiency in handling defined schemas and transactional integrity.

MySQL- database modeling:

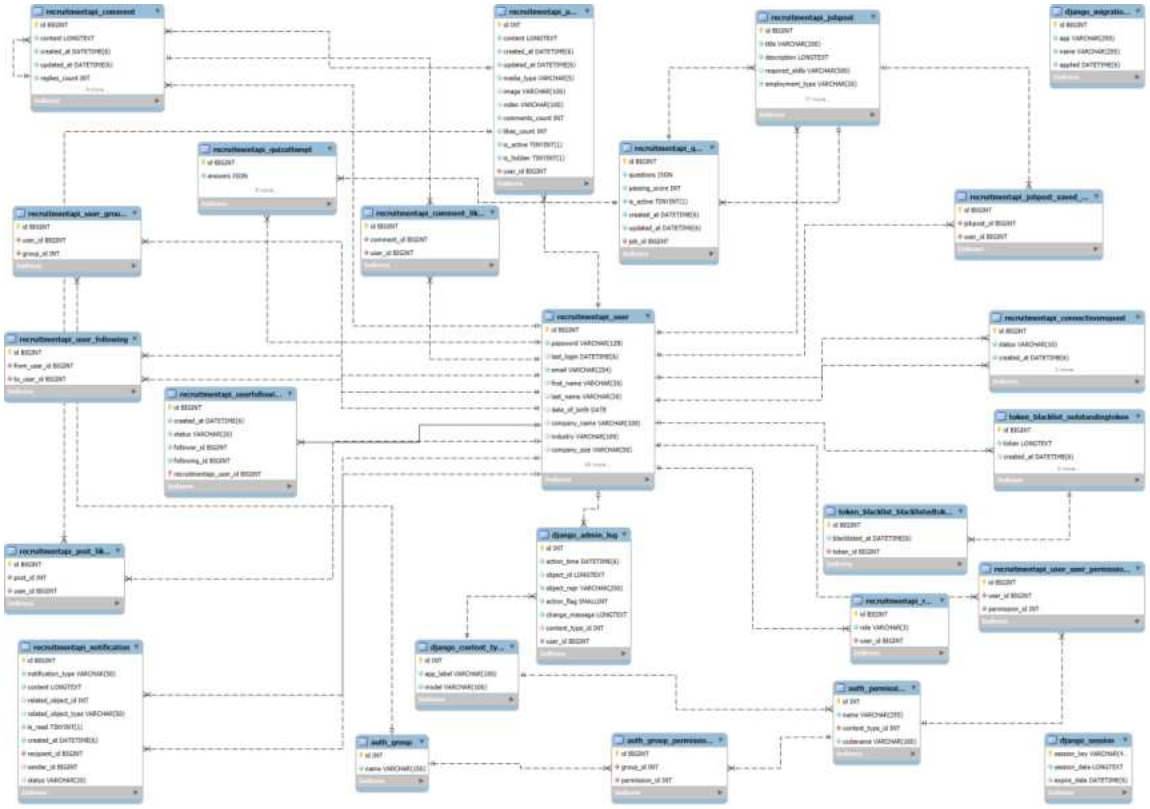


Figure 23 MySQL - Database Modeling

5. Updating the RTM:

REQ-ID	Use-case	Analysis	System Design	Detail Design	Implementation	Test Cases
FR-UM-001	UC-05	3.2.1.5	UserMgt	Detail User		
FR-UM-002	UC-06	3.2.1.6	UserMgt	Detail User		
FR-UM-003	UC-HIGH	3.2.1.22	UserMgt	Detail User		
FR-UM-004	UC-07	3.2.1.7	UserMgt	Detail User		
FR-UM-005	UC-21	3.2.1.21	UserMgt	Detail User		
FR-UM-006	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-UM-007	UC-01.3.3	3.2.1.3	UserMgt	Detail User		
FR-UM-008	UC-03	3.2.1.3	UserMgt	Detail User		
FR-UM-009	UC-03	3.2.1.3	UserMgt	Detail User		
FR-UM-010	UC-01.2.1	3.2.1.4	UserMgt	Detail User		
FR-UM-011	UC-01.2.1.3	3.2.1.2	UserMgt	Detail User		

FR-UM-012	UC-01.2.1.3	3.2.1.2	UserMgt	Detail User		
FR-UM-013	UC-01.2.2	3.2.1.2	UserMgt	Detail User		
FR-UM-014	UC-01.2.2.3	3.2.1.2	UserMgt	Detail User		
FR-UM-015	UC-21	3.2.1.1	UserMgt	Detail User		
FR-UM-016	UC-01.4.1	3.2.1.4	UserMgt	Detail User		
FR-UM-017	UC-10	3.2.1.10	UserMgt	Detail User		
FR-UM-018	UC-10	3.2.1.10	UserMgt	Detail User		
FR-UM-019	UC-10	3.2.1.10	UserMgt	Detail User		
FR-UM-020	UC-10.3	3.2.1.10	UserMgt	Detail User		
FR-UM-021	UC-10.2	3.2.1.9	UserMgt	Detail User		
FR-UM-022	UC-10.1	3.2.1.10	UserMgt	Detail User		
FR-UM-023	UC-21	3.2.1.21	UserMgt	Detail User		

FR-UM-024	UC-11	3.2.1.11	UserMgt	Detail User		
FR-UM-025	UC-12.5	3.2.1.10	UserMgt	Detail Post		
FR-UM-026	UC-12.5.3	3.2.1.10	UserMgt	Detail Post		
FR-UM-027	UC-12	3.2.1.10	UserMgt	Detail Post		
FR-UM-028	UC-12.4	3.2.1.10	UserMgt	Detail Post		
FR-UM-029	UC-12.2	3.2.1.10	UserMgt	Detail Post		
FR-UM-030	UC-12.2.4	3.2.1.10	UserMgt	Detail Post		
FR-UM-031	UC-12.2.1	3.2.1.10	UserMgt	Detail Post		
FR-UM-032	UC-12.2.2	3.2.1.10	UserMgt	Detail Post		
FR-UM-033	UC-12.2.3	3.2.1.10	UserMgt	Detail Post		
FR-UM-034	UC-12.2.3.1	3.2.1.10	UserMgt	DetailPost		
FR-UM-035	UC-12.2.1	3.2.1.10	UserMgt	Detail Post		
FR-UM-036	UC-12.2.2	3.2.1.10	UserMgt	Detail Post		

FR-UM-037	UC-12.3	3.2.1.12	UserMgt	Detail User		
FR-UM-038	UC-12	3.2.1.12	UserMgt	Detail User		
FR-UM-039	UC-12	3.2.1.12	UserMgt	Detail User		
FR-UM-040	UC-12	3.2.1.12	UserMgt	Detail User		
FR-UM-041	UC-12.3	3.2.1.12	UserMgt	Detail User		
FR-UM-042	UC-12.1	3.2.1.12	UserMgt	Detail User		
FR-JM-001	UC-18	3.2.1.18	JobMgt	Detail Job		
FR-JM-002	UC-15	3.2.1.16	JobMgt	Detail Job		
FR-JM-003	UC-16	3.2.1.23	JobMgt	Detail Job		
FR-JM-004	UC-10.4	3.2.1.24	JobMgt	D Job		
FR-JM-005	UC-10.4	3.2.1.23	JobMgt	Detail Job		
FR-JM-006	UC-18	3.2.1.17	JobMgt	Detail Job		

FR-JM-007	UC-20	3.2.1.20	JobMgt	Detail Job		
FR-NTF-001	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-002	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-003	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-004	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-005	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-006	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-007	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-008	UC-13	3.2.1.13	NotifyMgt	Detail Notify		
FR-NTF-009	UC-13	3.2.1.17	NotifyMgt	Detail Notify		
FR-NTF-010	UC-13	3.2.1.13	NotifyMgt	Detail Notify		

FR-NTF-011	UC-NT	3.2.1.27	NotifyMgt	Detail Notify		
FR-MS-01	UC-10.5	3.2.1.27	MgtMessage	Detail Messaging		
FR-MS-02	UC-10.5.1	3.2.1.27	MgtMessage	Detail Messaging		
FR-MS-03	UC-10.5.1.2	3.2.1.27	MgtMessage	Detail Messaging		
FR-MS-04	UC-10.5.1.3	3.2.1.27	MgtMessage	Detail Messaging		
FR-MS-05	UC-10.5.2	3.2.1.27	MgtMessage	Detail Messaging		
FR-AI-001	UC-01.4.2	3.2.1.8	AI	DetailAI		
FR-AI-002	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-003	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-004	UC-09	3.2.1.9	AI	DetailAI		
FR-AI-005	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-006	UC-09	3.2.1.9	AI	DetailAI		
FR-AI-007	UC-09	3.2.1.9	AI	DetailAI		
FR-AI-008	UC-09	3.2.1.9	AI	DetailAI		
FR-AI-009	UC-HIGH	3.2.1.8	AI	DetailAI		

FR-AI-010	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-011	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-012	UC-HIGH	3.2.1.8	AI	DetailAI		
FR-AI-013	UC-HIGH	3.2.1.8	CV	DetailCV		
FR-SK-001	UC-18.1	3.2.1.17	AI Assessment	DetailAI		
FR-SK-002	UC-16.1.1.2	3.2.1.17	Manage Recruitment	Detail Job		
FR-SK-003	UC-16.1.1.1	3.2.1.14	JobMgt	Detail Job		
FR-SK-004	UC-16.1.1.1	3.2.1.14	JobMgt	Detail Job		
FR-SK-005	UC-16.1.1.1.1	3.2.1.14	JobMgt	Detail Job		
FR-RM-001	UC-19.1.1.5	3.2.1.17	- Manage Recruitment -AI	DetailAI		
FR-RM-002	UC-16.1.1.1.1	3.2.1.14	JobMgt	Detail Job		
FR-RM-003	UC-19.1.1	3.2.1.19	Manage Recruitment	Detail Job		
FR-RM-004	UC-19.1.1.3	3.2.1.19	Manage Recruitment	Detail		

				Job		
FR-RM-005	UC-19.1.1.5	3.2.1.19	Manage Recruitment	Detail Job		
FR-RM-006	UC-19.1.1.1	3.2.1.17	Manage Recruitment	Detail Job		
FR-RM-007	UC-19.1.1.6	3.2.1.28	Manage Recruitment	Detail Job		
FR-RM-008	UC-19.1.1.7	3.2.1.28	Manage Recruitment	Detail Job		
FR-Admin-DASH-001		3.2.1.25	-	-		
FR-Admin-DASH-002		3.2.1.26	-	-		
NF-REQ-01		–	–	–	–	–
NF-REQ-02		–	–	–	–	–
NF-REQ-03		–	–	–	–	–
NF-REQ-04		–	–	–	–	–
NF-REQ-05		–	–	–	–	–
NF-REQ-06		–	–	–	–	–
NF-REQ-07		–	–	–	–	–

Table 24 RTM In Design Chapter

Chapter6 Practical Implementation

1. Introduction:

In this chapter, we will present the practical implementation of the system, detailing the technologies and tools employed. Additionally, we will showcase the system interfaces and conclude by executing test cases to verify the system's functionalities across different scenarios.

2. Used Tools:

- Django

  Is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

  Especially we use DRF: Django REST Framework is a widely used, full-featured API framework designed for building RESTful APIs with Django. At its core, DRF integrates with Django's core features "models, views, and URLs" making it simple and seamless to create a RESTful API

- React

  React is a popular JavaScript library for building user interfaces. It was created by Facebook and is widely used in web development. React allows developers to build reusable UI components that can efficiently update and render changes to the user interface when the underlying data changes.

Reacts primary focus is on building user interfaces, and it excels in creating interactive and dynamic web applications.

- Postman

  Postman is a powerful API platform that simplifies the process of designing, debugging, and testing APIs. It provides an intuitive interface for sending HTTP requests, inspecting responses, and automating API workflows. As a widely used tool, Postman helps developers efficiently test and document their APIs, making the development process smoother and more effective

- My SQL Database

  MySQL is an open-source relational database management system (RDBMS) that is widely used for storing, managing, and retrieving data. It is one of the most popular and widely adopted databases in the world, known for its reliability, scalability, and ease of use.

- Visual studio code (VS code):

  Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, code completion and debugging.

  We use it to develop the whole project (frontend, backend).

- GitHub:

GitHub is a Git-based source code repository hosting service that enables version control and collaborative development. It provides tools for managing code changes, continuous integration, and deployment, making software development more efficient. As our project repository, GitHub helps streamline teamwork, track progress, and ensure code quality.

3. AI Used Technologies and Tools:

HireHub the recruitment platform leverages Machine Learning (ML), Natural Language Processing (NLP), and scientific computing to provide intelligent job recommendations, accurate candidate matching, and automated assessment generation. The following key technologies form the foundation of the system:

❖Sentence Transformers

Sentence Transformers is an advanced NLP framework designed for generating text embeddings and computing semantic similarities. Built on PyTorch and Transformers, it utilizes the all-MiniLM-L6-v2 model, which is optimized for production use. This model generates 384-dimensional embeddings, providing a dense representation of text that captures semantic relationships and contextual meaning.

Usage in HireHub:

- Text Embedding Generation: Converts job descriptions, candidate profiles, and required skills into dense vector representations.
- Semantic Similarity Computation: Computes meaningful similarity scores between job postings and candidate qualifications.
- Context-Aware Matching: Identifies semantic relationships between skills and job requirements for accurate recommendations.

❖Numpy

NumPy is a core Python library for scientific computing, offering powerful tools for handling multi-dimensional arrays and performing complex mathematical operations.

Usage in HireHub:

- Vector Operations: Processes and manipulates text embedding vectors for job postings and candidate profiles.
- Similarity Score Computation: Implements cosine similarity calculations to assess the relevance of candidate profiles.
- Score Normalization: Scales and normalizes recommendation scores for consistent ranking and comparison.

❖Cosine Similarity:

Cosine Similarity is a widely used mathematical measure that calculates the similarity between two non-zero vectors by measuring the cosine of the angle between them. This metric is particularly effective for high-dimensional text embeddings, ensuring accurate comparisons.

Usage in HireHub:

- Profile Matching: Computes similarity scores between job descriptions and candidate embeddings.
- Skill Relevance Assessment: Measures how closely candidate skills align with job requirements.
- Recommendation Ranking: Orders job recommendations based on similarity scores for optimized matching.

❖ Local LLM (Large Language Model)

The project integrates llama-3.2-3b-instruct, a powerful local Large Language Model (LLM) that specializes in generating contextually relevant content. This model excels at understanding domain-specific requirements and generating intelligent responses.

Usage in HireHub:

- Automated Quiz Generation: Creates job-specific assessment questions tailored to the required skills.
- Context-Aware Questions: Adapts assessment questions based on job descriptions and experience levels.

- Natural Language Understanding: Extracts key assessment areas from job descriptions for accurate candidate evaluation.



Adaptive Quiz (AI-Driven Assessment)

The Adaptive Quiz feature uses a local LLM to dynamically generate and deliver skill assessments tailored to each job posting. The system adapts the difficulty of questions in real time based on the candidate's performance, ensuring a personalized and challenging evaluation.

Usage in HireHub:

Dynamic Question Generation: The LLM generates multiple-choice questions of varying difficulty (easy, medium, hard) based on the job description and required skills.

Adaptive Assessment Flow: The quiz engine adjusts the difficulty of subsequent questions according to the candidate's correct or incorrect answers, providing a more accurate measure of ability.

Automated Scoring and Feedback: The system automatically scores the quiz and provides instant feedback to both candidates and recruiters.

CV Parsing (AI-Powered Document Understanding)

The CV Parsing feature leverages multi-modal LLMs (such as Google Gemma via OpenRouter) and scientific libraries to extract structured information from unstructured CV documents.

Usage in HireHub:

PDF-to-Image Conversion: Uses libraries like **pdf2image** and **Pillow** to convert uploaded PDF CVs into images for processing by **vision-capable LLMs.**

Multi-Modal LLM Parsing: Sends CV images to a **vision-enabled LLM**, which extracts key candidate information (name, contact, skills, experience, education, certifications) in a structured JSON format.

Profile Auto-Fill: Automatically populates the user's profile fields with the parsed data, reducing manual entry and improving data accuracy.

NLP Libraries: Dependencies such as **spaCy and NLTK** are available for further text processing and entity recognition, supporting robust and extensible parsing workflows.

4. System Interfaces:

Sign-up interface:



Figure 24 Implementation - Register Interface

Log in interface:

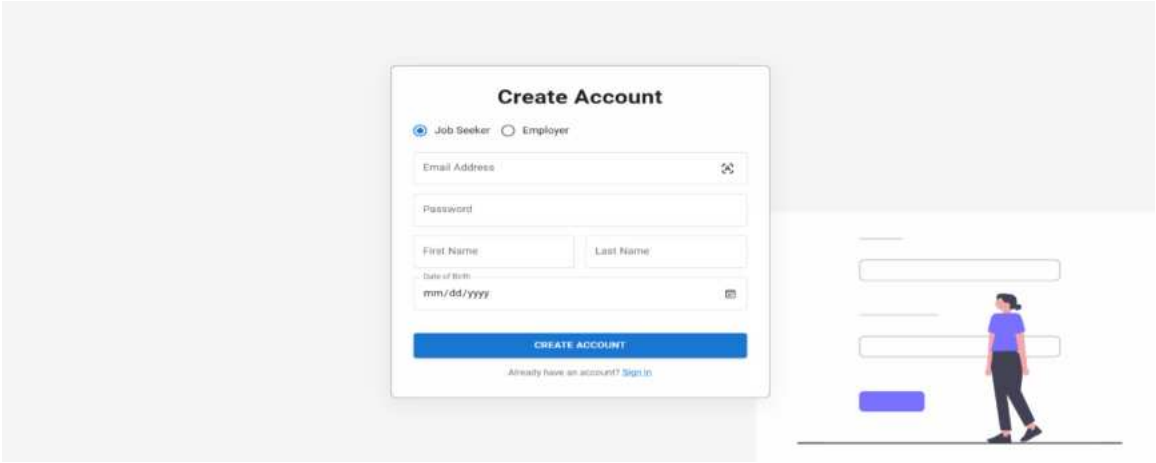


Figure 25 Implementation - Login Interface

Home Page interface:

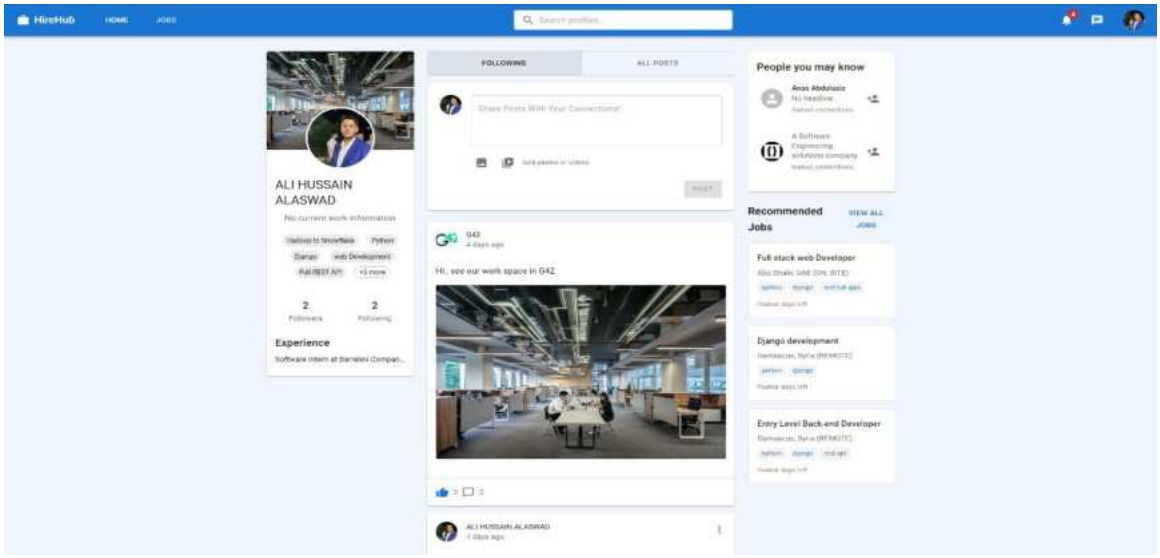


Figure 26 Implementation - Home Page Interface

Profile interface:

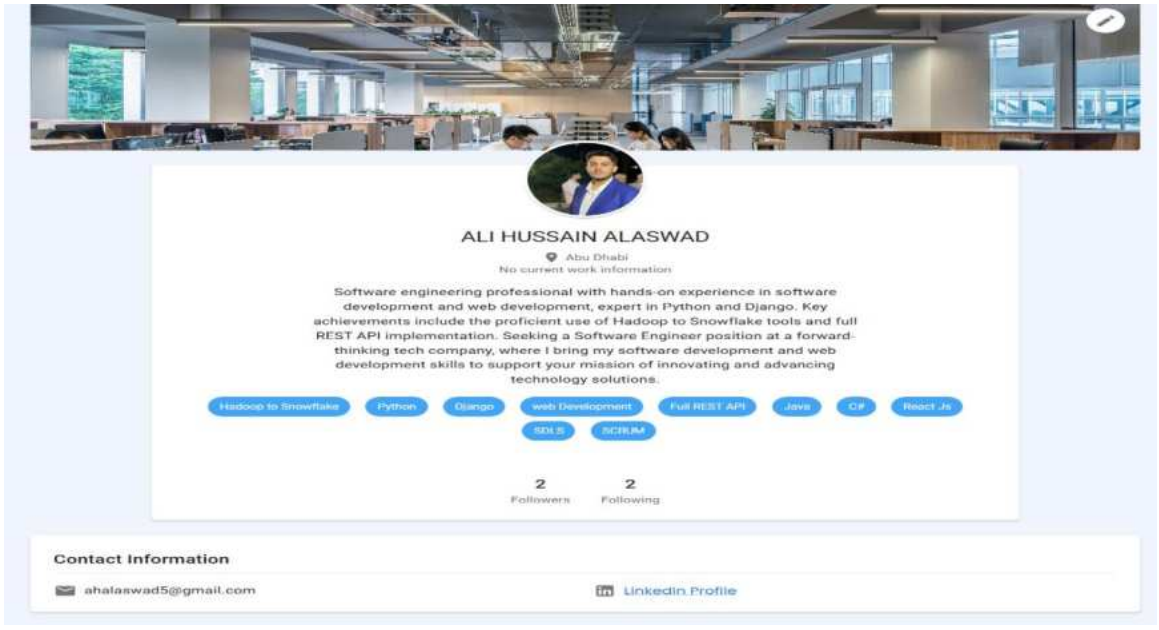


Figure 27 Implementation - Profile Interface

Edit Profile Dialog interface:

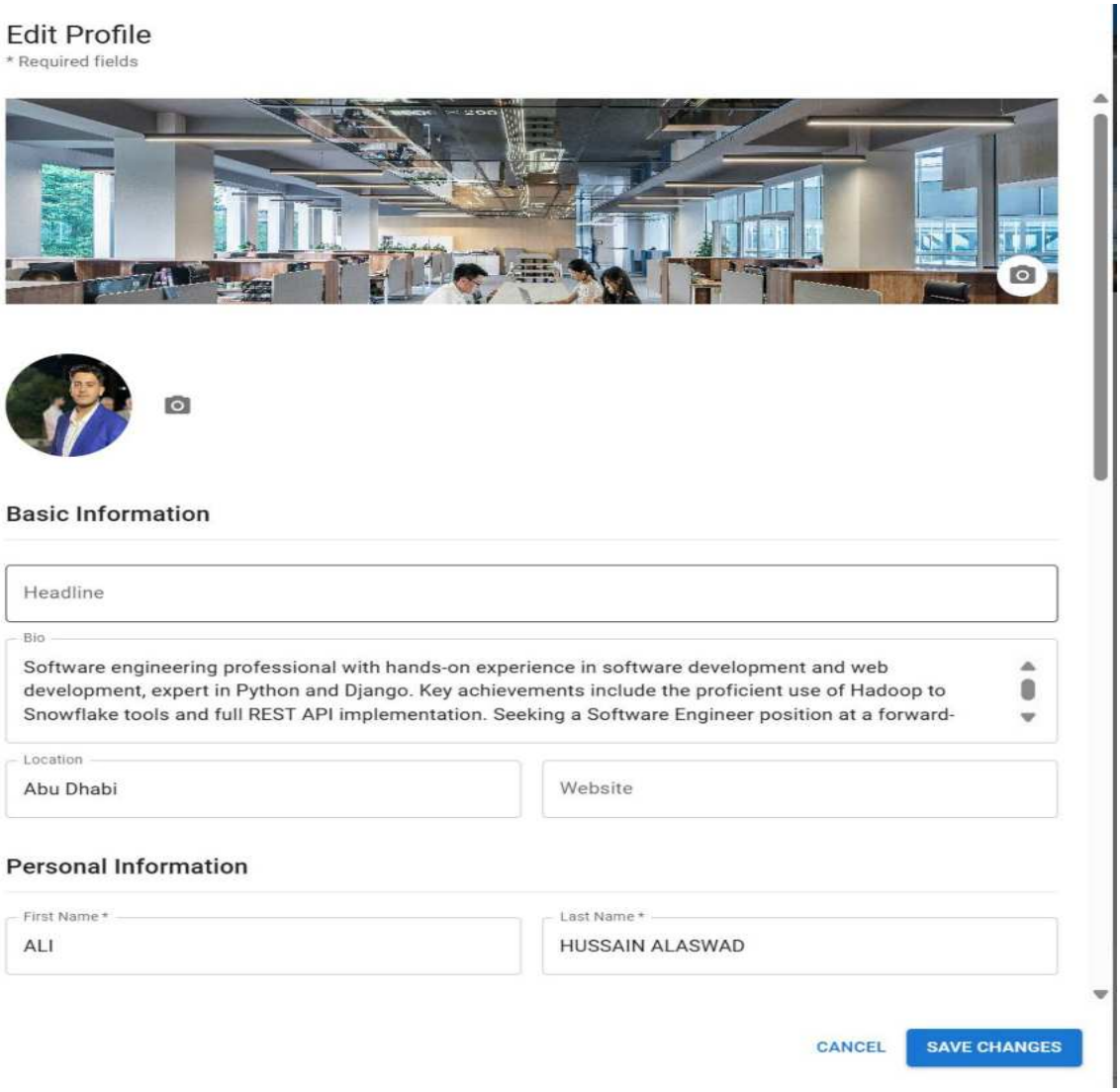


Figure 28 Implementation - Edit Profile Dialog Interface

Edit Privacy Settings interface:

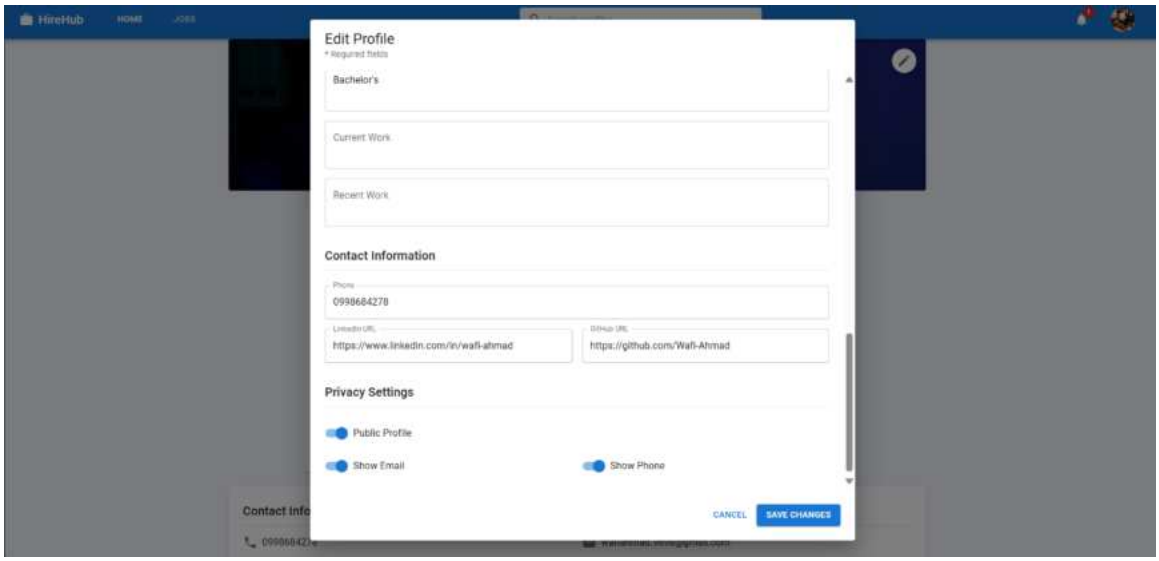


Figure 29 Implementation - Edit Privacy Settings Interface

Job Applications List interface:

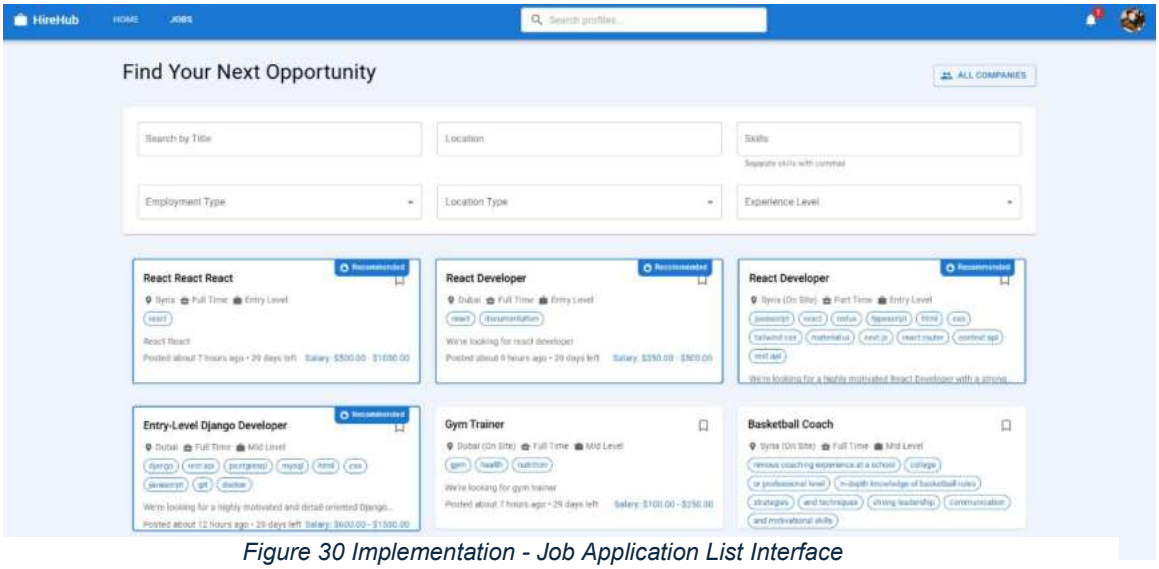


Figure 30 Implementation - Job Application List Interface

Job Application Details interface:

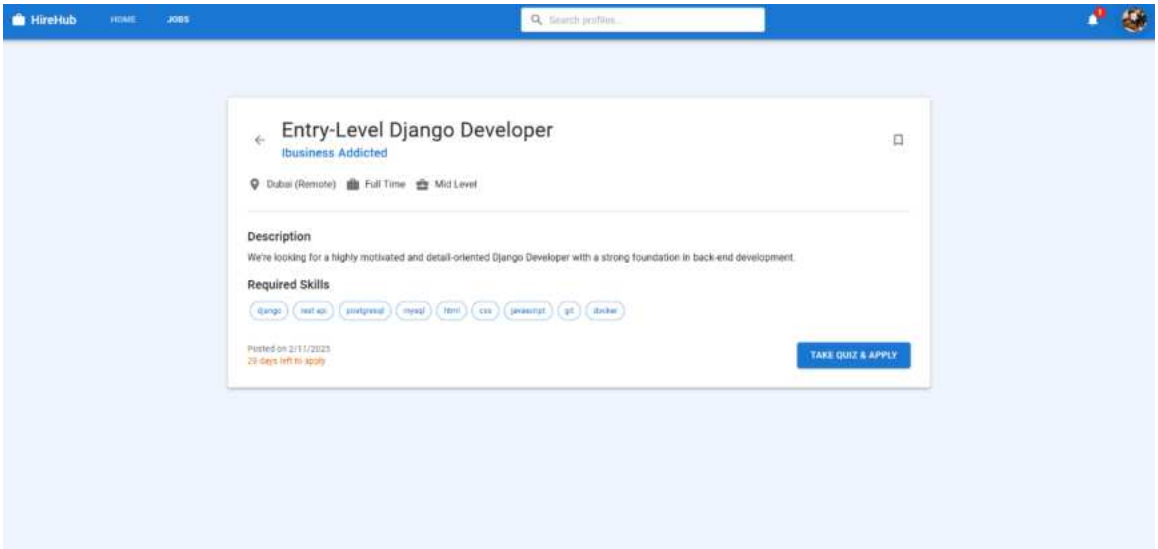


Figure 31 Implementation - Job Application Details Interface

Quiz interface:

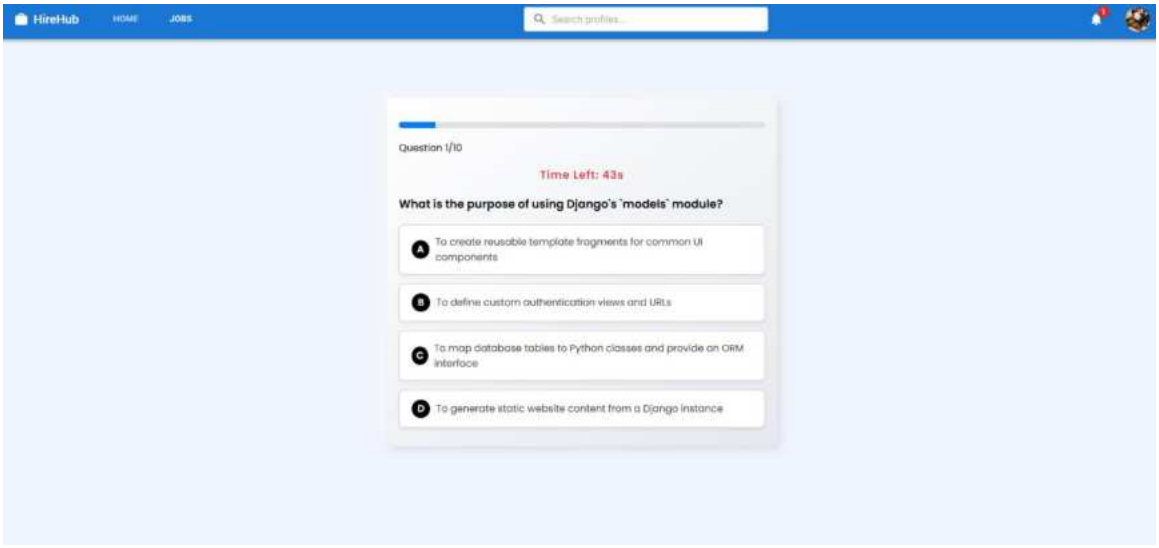


Figure 32 Implementation - Quiz Interface

Quiz Ban interface:

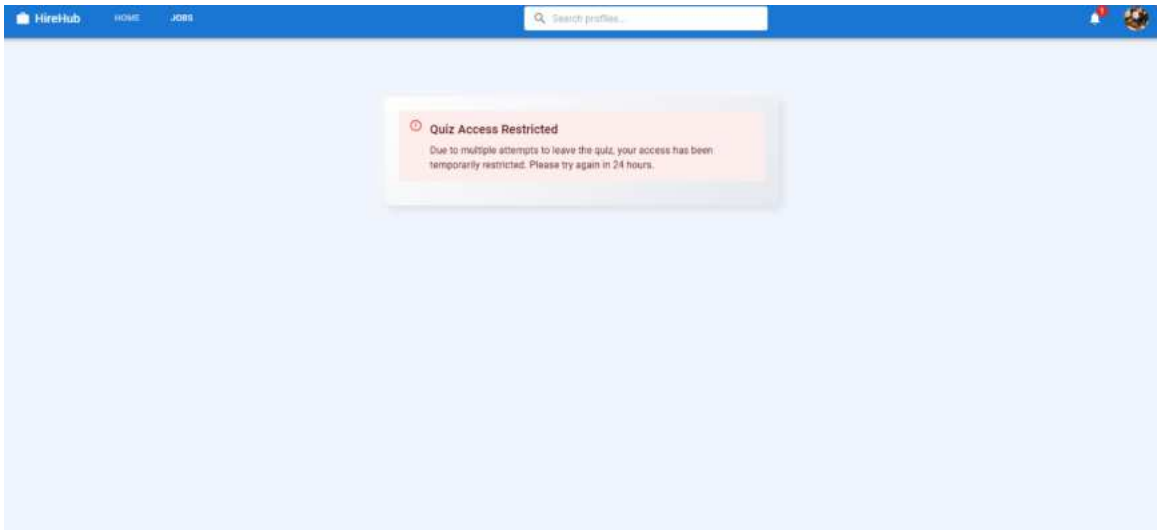


Figure 33 Implementation - Ban Interface

Quiz Result interface:

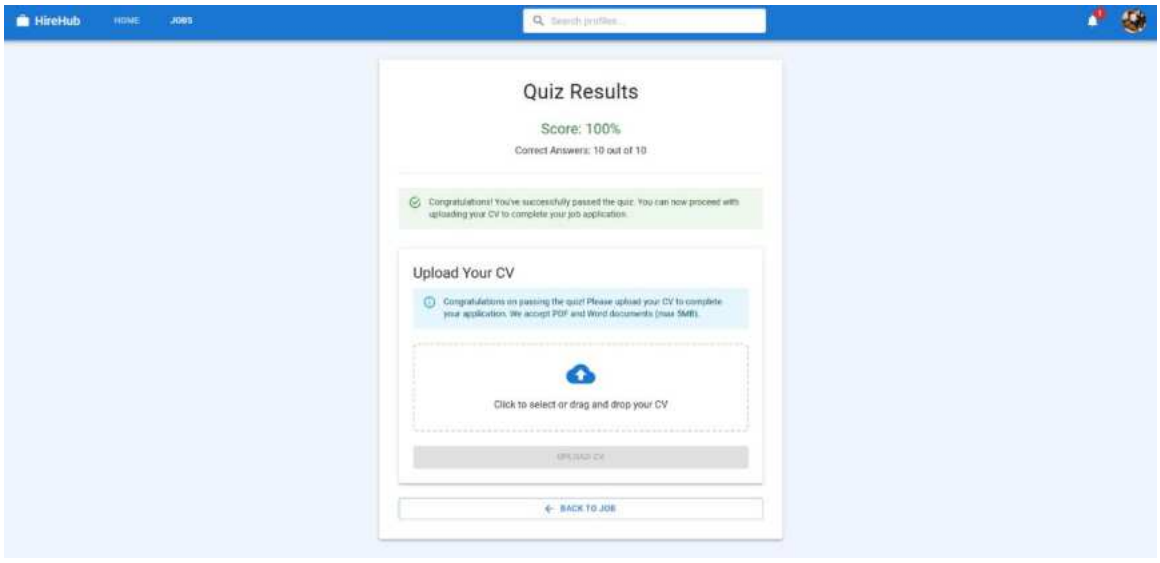


Figure 34 Implementation - Quiz Result Interface

View Applicants interface:

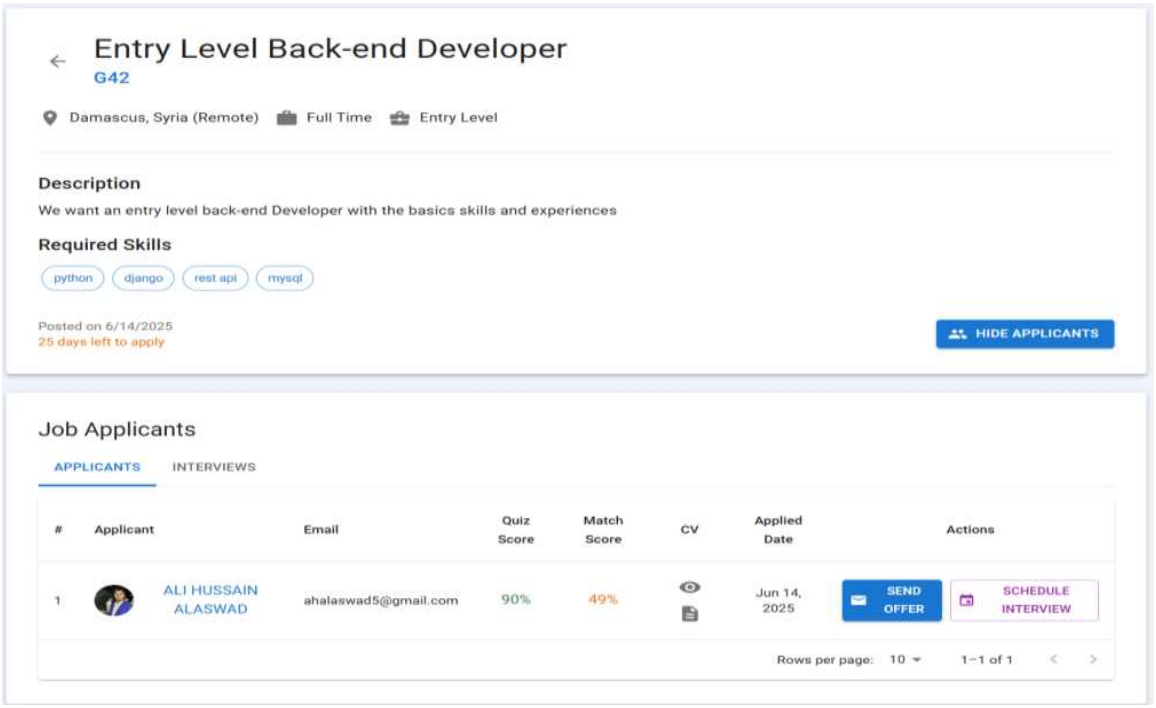


Figure 35 Implementation - View Applicants Interface

Interview Schedualing interface :

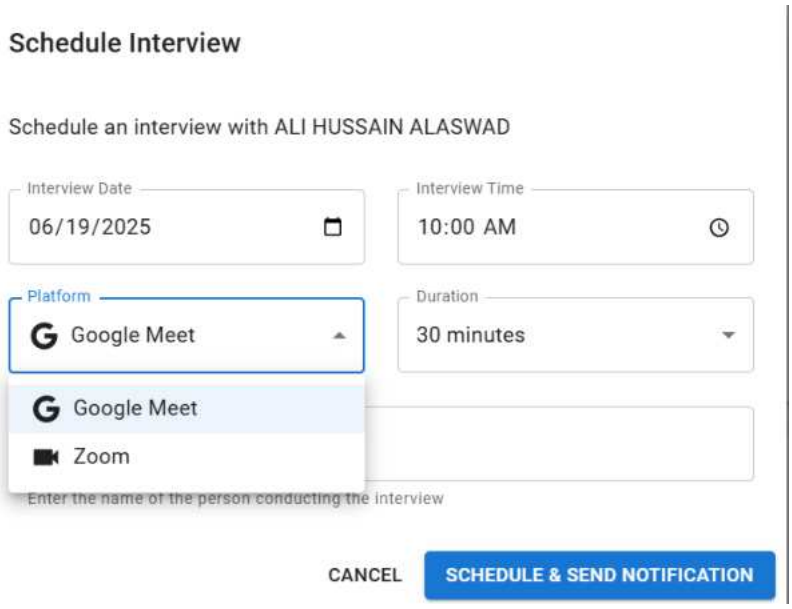


Figure 47 implementation - Interview Schedualing

Job Application Form interface:

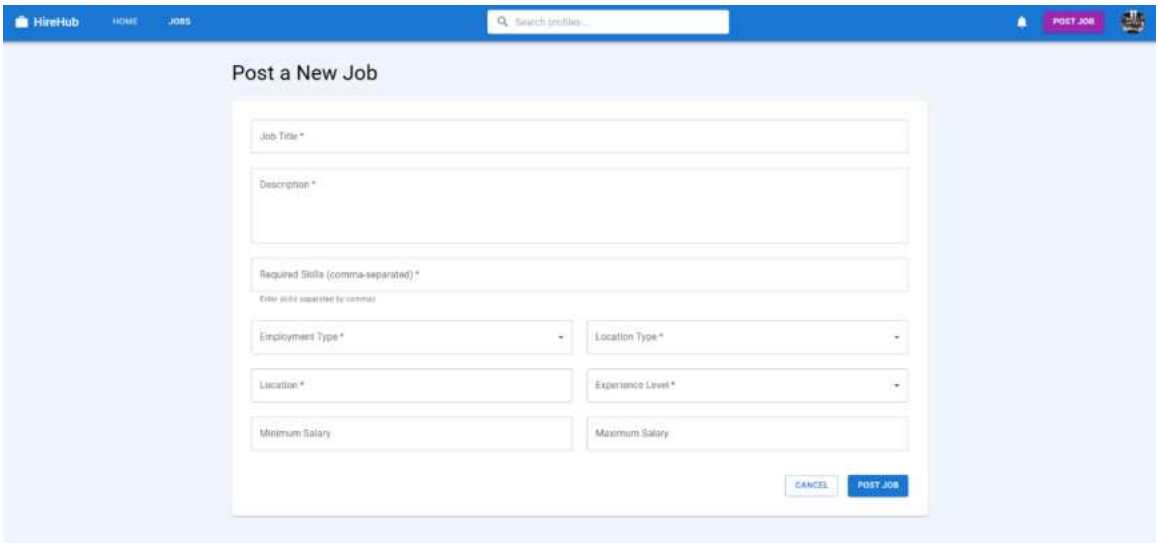


Figure 36 Implementation - Job Application Form Interface

Saved Jobs interface:

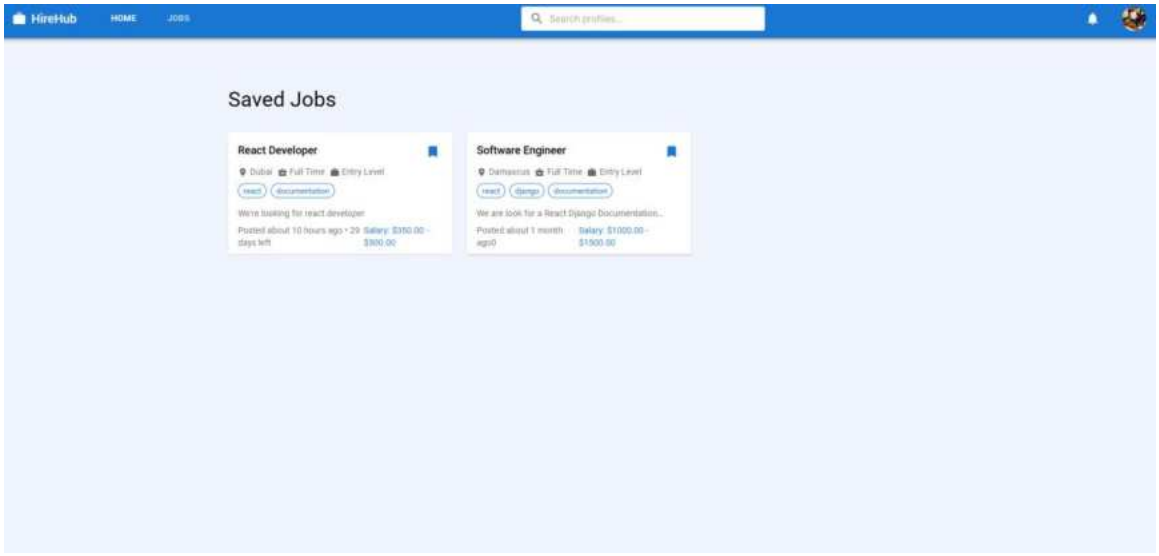


Figure 37 Implementation - Saved Jobs Interface

Private Profile View Interface:

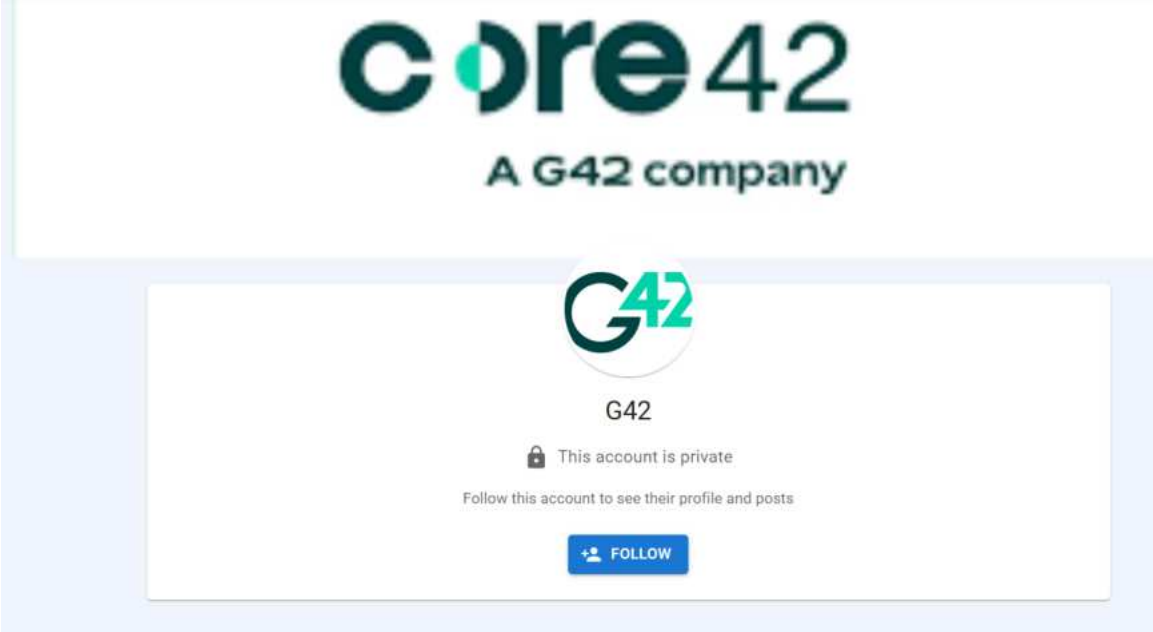


Figure 38 Implementation - Private Profile View Interface

Confirm Follow Request Interface:

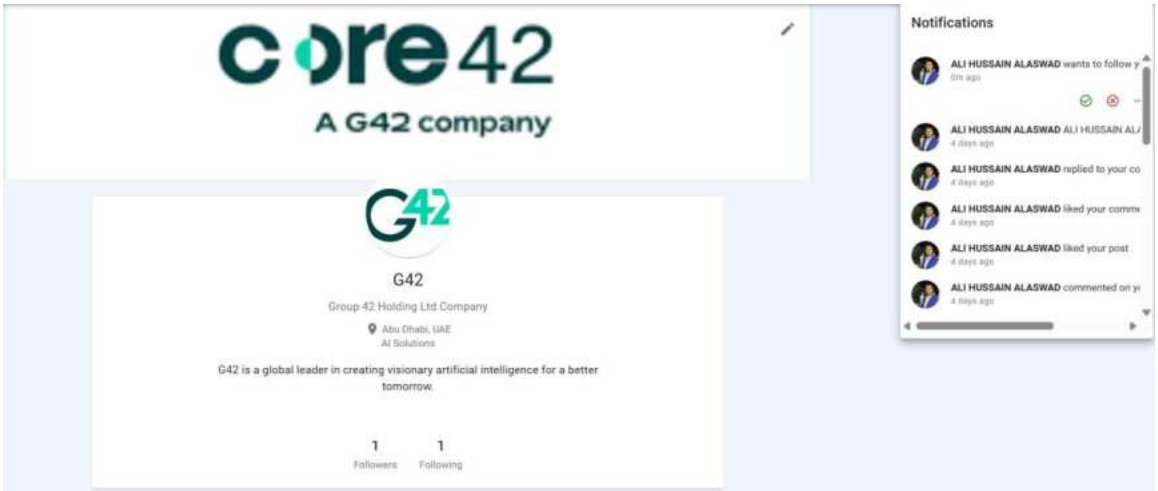


Figure 39 Implementation - Follow Request Notification For a Private Profile Interface

Reset Password Request Interface:

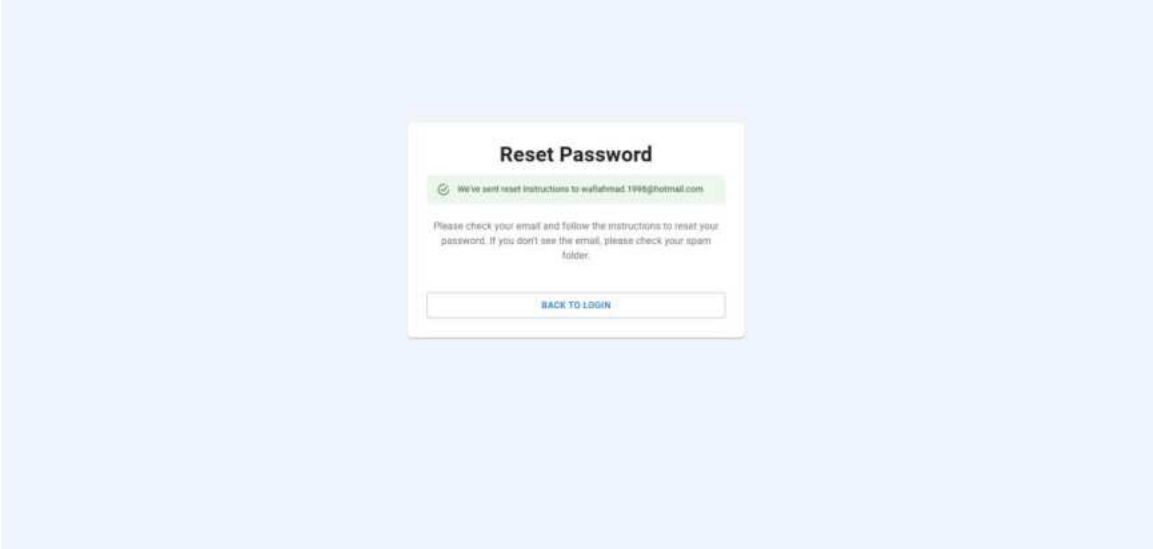


Figure 40 Implementation - Password Reset Password Interface

Reset Password Interface:



Figure 41 Implementation - Password Reset Interface

Reset Password Email Interface:



Figure 42 Implementation - Reset Password Link by Email Interface

Job Offer Email Interface:

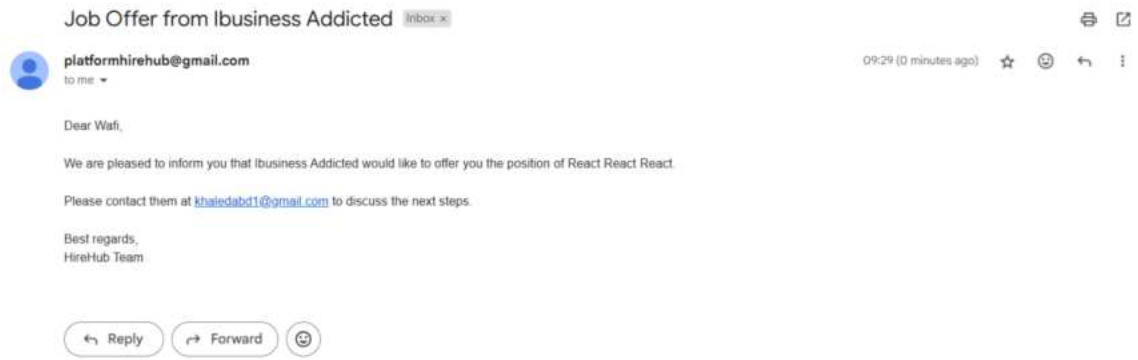


Figure 43 Implementation - Job Offer by Email Interface

Cv Parsing Interface:

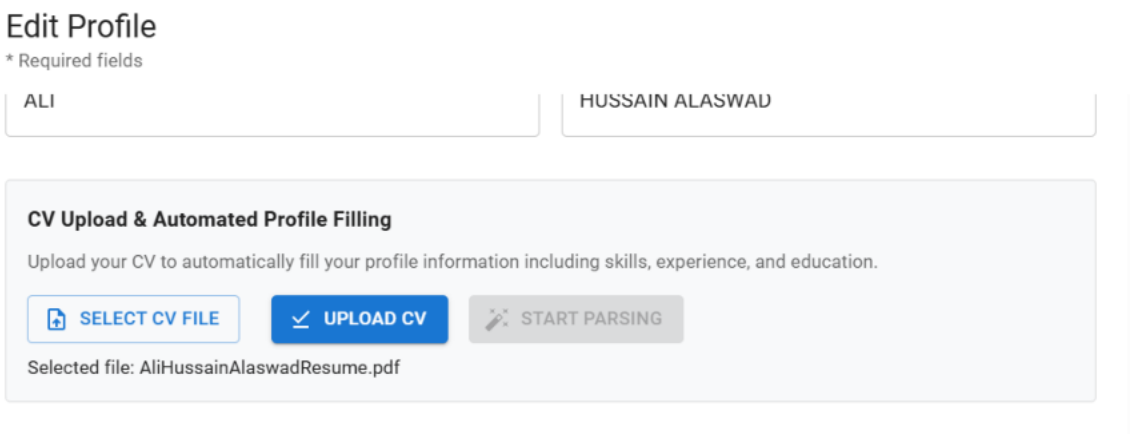


Figure 48 implementation - CV parsing

5. Test Cases Execution:

Table 25 Test Cases Execution

TC id	Test case title	Tested data	Expected result	Actual result	Pass/ fail
Tc-01	Check results on entering a valid email and password.	email=" wafiahmad.9898@gmail.com" Password=" wafi.1998"	Login successfully.	Login successfully.	Pass
Tc-02	Check results on entering an invalid email, or password.	Email= "wafi@hotmail.com" Password="P123s31s31"	Invalid credentials. Please check your email and password.	Invalid credentials. Please check your email and password.	Pass

			Email Address	Email Address	
Tc-03	Check results when a user email is empty, and the "login" button is pressed.	Email= Password="Testtest1"	Error message "Email is required"	Error message "Email is required"	Pass
Tc-04	Check results on completing the sign-up form correctly, for job seeker and selected sign-up	Email=" wafiahmad.9898@gmail.com " First name=" Wafi" Last name="Ahmad" Date = 10/6/1998 Password=raghad1243 Password=wafi.1998	"Account created successfully "	"Account created successfully"	Pass
Tc-05	Check results on entering an existing email.	Email=" wafiahmad.1998@hotmail.com" First name="wafi" Last name="ahmad" Date = 10/6/1998 Password=raghad1243 Password=raghad1243	Error message "user with this email already exists"	Error message "user with this email already exists"	Pass
Tc-06	Check results when a user chose a weak	Email= "wafiahmad.1998@hotmail.com"	Error message "Password	Error message "Password	Pass

	password (less than 8 characters).	First name=" wafi" Last name="ahmad" Date = 10/6/1998 Password=123 Password=123	must be at least 8 characters"	must be at least 8 characters"	
Tc-07	Check results when a user email is empty, and the "login" button is pressed.	Email= Password=Testest	Error message "Please correct the errors below Email is required"	Error message "Please correct the errors below Email is required"	Pass
Tc-08	Check results on entering valid Email & Password	Email="wafiahmad.1998@hotmail.com" Password=wafi.1998	The system should navigate user to home page.	The system should display the user info.	Pass
Tc-09	Check results on deleting user account		The account should be deleted.	The account should be deleted.	Pass
Tc-10	Check results when reset password is required.	Email="wafiahmad.1998@hotmail.com"	We've sent reset instructions to wafiahmad.	We've sent reset instructions to wafiahmad.9	Pass

			9898@gmail.com.	898@gmail.com.	
Tc-11	Check results on entering new password.	New password: "Wafi.1998" Confirm New password: "Wafi.1998"	The System will navigate user to login page	The System will navigate user to login page	Pass
Tc-12	Check results on viewing saved jobs UI		The System will display list of saved jobs card	The System will display list of saved jobs card	Pass
Tc-13	Check results on uploading the correct format for changing profile pictures.	Image = "user.png"	The photo must be changed successfully.	The photo must be changed successfully.	Pass
Tc-14	Check results on uploading incorrect format.	Image = "user.svg"	Error message " profile must be JPEG, PNG, or JPG".	Error message " profile must be JPEG, PNG, or JPG".	Pass
Tc-15	Check results on updating the username with strings	First name = "Wafi" Last name = " Ahmad"	The username updated.	The username updated.	Pass

Tc-16	Check results on updating the profile while leaving empty First name and Last name.	First name = “” Last name = “”	Error message Please fix the errors before submitting	Error message Please fix the errors before submitting	Pass
Tc-17	Check results on valid attempt of updating user profile.		Profile updated successfully	Profile updated successfully	Pass
Tc-18	Check results on viewing home page when user not following any users.		No posts to show. Follow some users to see their posts here, or create your own post!	No posts to show. Follow some users to see their posts here, or create your own post!	Pass
Tc-19	Check results on viewing home page when user is following users.		The System shall display list of posts from users	The System shall display list of posts from users	Pass
Tc-20	Check results of creating a new post with image or video	Content=’”We're looking for employees”	The System shall display	The System shall display	Pass

		Image= “We're Hiring.png”	a new post form	a new post form	
Tc-21	Check results of editing post	Content=”We’re Hiring” Image= “We’re Hiring.png”	The system shall display updated post	The system shall display updated post	Pass
Tc-22	Check results of deleting a post		Post deleted.	Post deleted.	Pass
Tc-23	Display User’s Own Profile		The system displays the user's own profile with relevant personal details.	The system displays the user's own profile with relevant personal details.	Pass
Tc-24	Check results when trying to like or comment posts	Like button= “Pressed” Comment= “Interesting”	Like is registered, and the comment "Interesting " is posted.	Like is registered, and the comment "Interesting" is posted.	Pass
Tc-25	Check results when user unlike post	Like button= “Pressed”	Like is removed, and the like count	Like is removed, and the like count	Pass

			updates accordingly.	updates accordingly.	
Tc-26	Check results when deleting comment		The comment is removed, no longer appears under the post, and the comment count updates accordingly.	The comment is removed, no longer appears under the post, and the comment count updates accordingly.	Pass
Tc-27	Check results when edit comment in a post		The comment is updated with the new content and displayed correctly.	The comment is updated with the new content and displayed correctly.	Pass
Tc-28	Check results of switching profile to private		The profile becomes private, and	The profile becomes private, and	Pass

			other users see a restricted profile view with limited information.	other users see a restricted profile view with limited information.	
Tc-29	Check results of following public user.		The follow action is successful, the sender's following count increases, and the receiver's followers count updates accordingly.	The follow action is successful, the sender's following count increases, and the receiver's followers count updates accordingly.	Pass
Tc-30	Check results of following a private user		A follow request is sent to the private account,	A follow request is sent to the private account, and	Pass

			and the receiver sees options to accept, reject, or ignore the request.	the receiver sees options to accept, reject, or ignore the request.	
Tc-31	Check results of posting a job application	Job Title: Software Engineer Description: Develop and maintain scalable applications. Skills: Python, Django, React, REST API Employment Type: Full-Time Location Type: Remote Location: New York, USA Experience Level: Mid-Level Salary: 350 - 900	The job is successfully posted with the provided details, and it appears in the job listings.	The job is successfully posted with the provided details, and it appears in the job listings.	Pass
Tc-32	Check results of posting empty job application	Job Title: Description: Skills: Employment Type: Location Type:	Error message “Please fill	Error message “Please fill	Pass

		Location: Experience Level: Salary:	out this form".	out this form".	
Tc-33	Check results deleting job application		The job application is removed successfully and no longer appears in the application list.	The job application is removed successfully and no longer appears in the application list.	Pass
Tc-34	Check results for deleting a requirement.		The requirement should be deleted successfully.	The requirement should be deleted successfully.	Pass
Tc-35	Check results for job search with Title	Title: Software USA	Relevant job listings matching the search criteria are displayed correctly.	Relevant job listings matching the search criteria are displayed correctly.	Pass

Tc-36	Check results for job search with Location	Engineer Location: Syria	Relevant job listings matching the search criteria are displayed correctly.	Relevant job listings matching the search criteria are displayed correctly.	Pass
Tc-37	Check results for job search with Skills	Skills: Python, Django, React	Relevant job listings matching the search criteria are displayed correctly.	Relevant job listings matching the search criteria are displayed correctly.	Pass
Tc-38	Check results for job search with Employment Type	Employment Type: Full-Time	Relevant job listings matching the search criteria are displayed correctly.	Relevant job listings matching the search criteria are displayed correctly.	Pass
Tc-39	Check results for job search with Location Type	Location Type: Remote	Relevant job listings matching the search	Relevant job listings matching the search	Pass

			criteria are displayed correctly.	criteria are displayed correctly.	
Tc-40	Check results for job search with Experience Level	Experience Level: Mid-Level	Relevant job listings matching the search criteria are displayed correctly.	Relevant job listings matching the search criteria are displayed correctly.	Pass
Tc-41	Check results of taking a quiz		The system navigates the user to the quiz UI, where they can answer multiple-choice questions, submit responses, and receive feedback or results.	The system navigates the user to the quiz UI, where they can answer multiple-choice questions, submit responses, and receive feedback or results.	Pass

Tc-42	Check results of submitting CV	- File: WafiCV.pdf (PDF , DOC , DOCS)	The CV is successfully uploaded, and the system confirms submission.	The CV is successfully uploaded, and the system confirms submission.	Pass
Tc-43	Check results for View Job Applicants		The system displays a list of applicants for the job, showing details such as name, email, quiz score, match score, CV, application date, and available actions (e.g.,	The system displays a list of applicants for the job, showing details such as name, email, quiz score, match score, CV, application date, and available actions (e.g., sending a job offer).	Pass

			sending a job offer).		
Tc-44	Check results searching for users	First Name: Wafi Last Name: Ahmad Company Name: IBusiness Addicted	The system returns a list of users matching the search criteria based on first name, last name, or company name.	The system returns a list of users matching the search criteria based on first name, last name, or company name.	Pass
Tc-45	Check results for notifications when user likes a post.		A new notification appears in the list, indicating the like action with the correct user and timestamp.	A new notification appears in the list, indicating the like action with the correct user and timestamp.	Pass
Tc-46	Check results for notifications when		A new notification	A new notification	Pass

	user comments a post.		appears in the list, indicating the comment action with the correct user and timestamp.	appears in the list, indicating the comment action with the correct user and timestamp.	
Tc-47	Check results for notifications when new followers	A user follows another user.	A notification appears, stating that a specific user has started following.	A notification appears, stating that a specific user has started following.	Pass
Tc-48	Check results for notifications when followed companies post new job applications		A notification appears in the user's notification list, stating that the company has posted a	A notification appears in the user's notification list, stating that the company	Pass

			new job opportunity .	has posted a new job opportunity.	
Tc-49	Check results for notifications when companies send job offers to users.		The user receives an email notification about the job offer. The email contains job details, company name, and contact information . The email is formatted correctly and is readable. The recipient can reply or contact the company	The user receives an email notification about the job offer. The email contains job details, company name, and contact information. The email is formatted correctly and is readable. The recipient can reply or contact the company via	Pass

			via the provided email.	the provided email.	
Tc-50	Check results for displaying recommended jobs in home page		Expected Result: The system displays up to 3 job recommend ations tailored to the user. Each job listing includes title, location, skills, and posting date. A “View All Jobs” option is available for additional listings.	Expected Result: The system displays up to 3 job recommenda tions tailored to the user. Each job listing includes title, location, skills, and posting date. A “View All Jobs” option is available for additional listings. The UI correctly	Pass

			The UI correctly handles fewer than 3 jobs without issues.	handles fewer than 3 jobs without issues.	
TC-51	Upload Valid CV	Valid PDF file under 5MB	CV is uploaded, stored, and user receives success message	CV is uploaded, stored, and user receives success message	Pass
TC-52	Upload Invalid File Type	.docx or .jpg file	System displays error: "Only PDF files are allowed"	System displays error: "Only PDF files are allowed"	Pass
TC-53	Upload Oversized CV	PDF file >5MB	System displays error: "File size cannot exceed 5MB"	System displays error: "File size cannot exceed 5MB"	Pass
TC-54	Parse Uploaded CV	Uploaded valid CV	Profile fields are	Profile fields are auto-	Pass

			auto-filled with extracted data; user sees success message	filled with extracted data; user sees success message	
TC-55	Parse Corrupted CV	Corrupted PDF	System displays error: "Failed to parse CV" or similar	System displays error: "Failed to parse CV"	Pass
TC-56	Schedule Interview (Valid)	Valid date/time, platform, interviewer	Interview is created, notifications sent, interview appears in lists	Interview is created, notifications sent, interview appears in lists	Pass
TC-57	Schedule Interview (Past Date)	Date/time in the past	System displays error: "Interview date must be in the future"	System displays error: "Interview date must be in the future"	Pass

TC-58	Search for User	Search query (e.g., "Ali")	Users matching query are listed	Users matching query are listed	Pass
TC-59	Start Conversation and Send Message	Message text	Message appears in conversation, recipient receives it in real time	Message appears in conversation, recipient receives it in real time	Pass
TC-60	Edit Message (Within Timeframe)	New message text	Message content is updated, marked as edited	Message content is updated, marked as edited	Pass
TC-61	Delete Message (Within Timeframe)	N/A	Message is marked as deleted for both users	Message is marked as deleted for both users	Pass
TC-62	View Conversation History	Existing conversations	All previous messages are displayed in correct order	All previous messages are displayed in correct order	Pass

Testing the AI components - Experiment on Real Data

Introduction

In this analysis, we evaluated the performance of our AI-powered recommendation and matching systems within HireHub. Our objective was to assess how effectively our system matches jobs with candidates, recommends relevant posts, and suggests meaningful connections based on user profiles and interactions

System Components Tested:

1. Job Matching System:
    - Implementation: Using Sentence Transformer ('all-MiniLM-L6-v2') for semantic matching.
    - Features tested: Job-candidate matching based on skills, experience, and job requirements.
    - Data: Real user profiles and job postings entered into the system.
    -
2. Post Recommendations:
    - Implementation: Content-based filtering using embedding similarities.
    - Features tested: Post relevancy based on user profiles and interests.
    - Data: User-generated posts and professional content.

3. Connection Recommendations:
    - Implementation: Profile similarity matching.
    - Features tested: Professional network suggestions based on skills and industry alignment.

Testing Methodology:

We tested the system with real user data, including:

- Job postings with detailed requirements.
- User profiles with skills and experience.
- Professional posts and interactions.
- Connection networks.

Result:

1. Job Matching Performance:
    - Successfully identified relevant job matches based on user skills.
    - Accurate calculation of match scores between candidates and job requirements.
    - Effective filtering of job recommendations based on the 30% threshold.
2. Post Recommendations:
    - Demonstrated strong correlation between user interests and recommended posts.
    - Effectively prioritized content from followed connections.
    - Successfully identified professionally relevant content.

3. Connection Suggestions:
    - Accurately suggested professional connections based on industry and skill alignment.
    - Successfully identified mutual interests and potential professional relationships.

Quiz Services Module (quiz_services.py)

- **generate_quiz()**:
    - Constructs a prompt based on **job details**.
    - Sends an API request to **local LLM endpoint**.
    - Validates the response format.
    - **Assigns metadata**, including the correct answer.
    - **Shuffles questions** before saving them to the database.
- **submit_attempt()**:
    - Retrieves **quiz questions** by ID.
    - **Compares user answers** with the stored correct answer.
    - Computes a **score percentage**.
    - Saves the **quiz attempt record**.

Job Matching Module (job_matching_service.py)

- **calculate_match_score()**:
    - **Prepares text data** (normalization, embedding input construction).
    - Computes **embeddings** using SentenceTransformer.
    - **Applies cosine similarity** to determine candidate-job compatibility.

- **Returns a match score** (scaled from **0 to 1**).

Recommendation Module (recommendation_services.py)

- **calculate_job_candidate_match()**:
  - Fetches job & candidate details.
  - Generates **embeddings for both**.
  - Computes **cosine similarity**.
  - Normalizes the score **between 0 and 1**.
  - **Handles errors gracefully** (e.g., missing data).

Conclusion

The testing demonstrated that our implementation of AI components, particularly the Sentence Transformer model for semantic matching, effectively serves our platform's needs for job matching, content recommendations, and professional networking. The system showed strong performance in providing relevant recommendations across all tested components, validating our approach to using embedding-based similarity calculations for matching and recommendations.

6. RTM Last Version:

Table 26 RTM Final Version

REQ-ID	Use-case	Analysis	System Design	Detail Design	Implementation	Test Cases
FR-UM-001	UC-05	3.2.1.5	UserMgt	Detail User	UserService	TC-04 TC-05 TC-06
FR-UM-002	UC-06	3.2.1.6	UserMgt	Detail User	UserService	TC-01 TC-02 TC-03 TC-08
FR-UM-003	UC-HIGH	3.2.1.22	UserMgt	Detail User	UserService	TC-08
FR-UM-004	UC-07	3.2.1.7	UserMgt	Detail User	UserService	TC-10 TC-11
FR-UM-005	UC-21	3.2.1.21	UserMgt	Detail User	UserService	TC-23
FR-UM-006	UC-HIGH	3.2.1.8	AI	DetailAI	UserService	TC-17

FR-UM-007	UC-01.3.3	3.2.1.3	UserMgt	Detail User	UserService	TC-13 TC-14
FR-UM-008	UC-03	3.2.1.3	UserMgt	Detail User	UserService	TC-15 TC-16
FR-UM-009	UC-03	3.2.1.3	UserMgt	Detail User	UserService	TC-15 TC-16 TC-17
FR-UM-010	UC-01.2.1	3.2.1.4	UserMgt	Detail User	UserService	TC-17
FR-UM-011	UC-01.2.1.3	3.2.1.2	UserMgt	Detail User	UserService	TC-13 TC-14
FR-UM-012	UC-01.2.1.3	3.2.1.2	UserMgt	Detail User	UserService	TC-15 TC-16
FR-UM-013	UC-01.2.2	3.2.1.2	UserMgt	Detail User	UserService	TC-28
FR-UM-014	UC-01.2.2.3	3.2.1.2	UserMgt	Detail User	UserService	TC-28

FR-UM-015	UC-21	3.2.1.1	UserMgt	Detail User	UserService	TC-28
FR-UM-016	UC-01.4.1	3.2.1.4	UserMgt	Detail User	UserService	TC-09
FR-UM-017	UC-10	3.2.1.10	UserMgt	Detail User	UserService	TC-18 TC-19
FR-UM-018	UC-10	3.2.1.10	UserMgt	Detail User	UserService PostService JobsService	TC-08
FR-UM-019	UC-10	3.2.1.10	UserMgt	Detail User	UserService	TC-08
FR-UM-020	UC-10.3	3.2.1.10	UserMgt	Detail User	UserService	TC-23
FR-UM-021	UC-10.2	3.2.1.9	UserMgt	Detail User	Connection Service	TC-08
FR-UM-022	UC-10.1	3.2.1.10	UserMgt	Detail User	JobsService	TC-50

FR-UM-023	UC-21	3.2.1.21	UserMgt	Detail User	UserService	TC-23
FR-UM-024	UC-11	3.2.1.11	UserMgt	Detail User	UserService	TC-44
FR-UM-025	UC-12.5	3.2.1.10	UserMgt	Detail Post	PostService	TC-20
FR-UM-026	UC-12.5.3	3.2.1.10	UserMgt	Detail Post	PostService	TC-13 TC-14
FR-UM-027	UC-12	3.2.1.10	UserMgt	Detail Post	PostService	TC-18 TC-19
FR-UM-028	UC-12.4	3.2.1.10	UserMgt	Detail Post	PostService	TC-24 TC-25
FR-UM-029	UC-12.2	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-24
FR-UM-030	UC-12.2.4	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-24 TC-25

FR-UM-031	UC-12.2.1	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-26
FR-UM-032	UC-12.2.2	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-27
FR-UM-033	UC-12.2.3	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-24
FR-UM-034	UC-12.2.3.1	3.2.1.10	UserMgt	DetailPost	Comment Service	TC-24 TC-25
FR-UM-035	UC-12.2.1	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-26
FR-UM-036	UC-12.2.2	3.2.1.10	UserMgt	Detail Post	Comment Service	TC-27
FR-UM-037	UC-12.3	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30
FR-UM-038	UC-12	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30

FR-UM-039	UC-12	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30
FR-UM-040	UC-12	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30
FR-UM-041	UC-12.3	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30
FR-UM-042	UC-12.1	3.2.1.12	UserMgt	Detail User	Connection Service	TC-30
FR-JM-001	UC-18	3.2.1.18	JobMgt	Detail Job	JobsService	TC-31 TC-32
FR-JM-002	UC-15	3.2.1.16	JobMgt	Detail Job	JobsService	TC-35 TC-36 TC-37 TC-38 TC-39 TC-40
FR-JM-003	UC-16	3.2.1.23	JobMgt	Detail Job	JobsService	TC-12
FR-JM-004	UC-10.4	3.2.1.24	JobMgt	D Job	JobsService	TC-12

FR-JM-005	UC-10.4	3.2.1.23	JobMgt	Detail Job	JobsService	TC-12
FR-JM-006	UC-18	3.2.1.17	JobMgt	Detail Job	JobsService	TC-12
FR-JM-007	UC-20	3.2.1.20	JobMgt	Detail Job	JobsService	TC-33
FR-NTF-001	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-48
FR-NTF-002	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-47
FR-NTF-003	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-30
FR-NTF-004	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-30
FR-NTF-005	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-45
FR-NTF-006	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-45

FR-NTF-007	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-45
FR-NTF-008	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-45
FR-NTF-009	UC-13	3.2.1.17	NotifyMgt	Detail Notify	Notification Service	TC-49
FR-NTF-010	UC-13	3.2.1.13	NotifyMgt	Detail Notify	Notification Service	TC-49
FR-NTF-011	UC-NT	3.2.1.27	NotifyMgt	Detail Notify	MessagingServices	TC-59
FR-MS-01	UC-10.5	3.2.1.27	MgtMessage	Detail Messaging	MessagingServices	TC-59
FR-MS-02	UC-10.5.1	3.2.1.27	MgtMessage	Detail Messaging	MessagingServices	TC-60
FR-MS-03	UC-10.5.1.2	3.2.1.27	MgtMessage	Detail Messaging	MessagingServices	TC-61
FR-MS-04	UC-10.5.1.3	3.2.1.27	MgtMessage	Detail Messaging	MessagingServices	TC-61

FR-MS-05	UC-10.5.2	3.2.1.27	MgtMessage	Detail Messaging	MessagingServices	TC-61
FR-AI-001	UC-01.4.2	3.2.1.8	AI	DetailAI	RecommendationService	TC-17
FR-AI-002	UC-HIGH	3.2.1.8	AI	DetailAI	RecommendationService	TC-23
FR-AI-003	UC-HIGH	3.2.1.8	AI	DetailAI	RecommendationService	TC-50
FR-AI-004	UC-09	3.2.1.9	AI	DetailAI	Connection Service	TC-50
FR-AI-005	UC-HIGH	3.2.1.8	AI	DetailAI	RecommendationService	TC-50
FR-AI-006	UC-09	3.2.1.9	AI	DetailAI	Connection Service	TC-50
FR-AI-007	UC-09	3.2.1.9	AI	DetailAI	RecommendationService	TC-50
FR-AI-008	UC-09	3.2.1.9	AI	DetailAI	Connection Service	TC-50
FR-AI-009	UC-HIGH	3.2.1.8	AI	DetailAI	UserService	TC-50
FR-AI-010	UC-HIGH	3.2.1.8	AI	DetailAI	Connection Service	TC-50
FR-AI-011	UC-HIGH	3.2.1.8	AI	DetailAI	RecommendationService	TC-50

FR-AI-012	UC-HIGH	3.2.1.8	AI	DetailAI	RecommendationService	TC-50
FR-AI-013	UC-HIGH	3.2.1.8	CV	DetailCV	CVService	TC-54
FR-SK-001	UC-18.1	3.2.1.17	AI Assessment	DetailAI	QuizService	TC-41
FR-SK-002	UC-16.1.1.2	3.2.1.17	Manage Recruitment	Detail Job	QuizService	TC-41
FR-SK-003	UC-16.1.1.1	3.2.1.14	JobMgt	Detail Job	QuizService	TC-41
FR-SK-004	UC-16.1.1.1	3.2.1.14	JobMgt	Detail Job	QuizService	TC-41
FR-SK-005	UC-16.1.1.1.1	3.2.1.14	JobMgt	Detail Job	CvService	TC-42
FR-RM-001	UC-19.1.1.5	3.2.1.17	- Manage Recruitment -AI	DetailAI	CvService	TC-43
FR-RM-002	UC-16.1.1.1.1	3.2.1.14	JobMgt	Detail Job	CvService	TC-42
FR-RM-003	UC-19.1.1	3.2.1.19	Manage Recruitment	Detail Job	JobMatching Service	TC-43

FR-RM-004	UC-19.1.1.3	3.2.1.19	Manage Recruitment	Detail Job	JobMatching Service	TC-43
FR-RM-005	UC-19.1.1.5	3.2.1.19	Manage Recruitment	Detail Job	CvService	TC-42
FR-RM-006	UC-19.1.1.1	3.2.1.17	Manage Recruitment	Detail Job	JobsService	TC-49
FR-RM-007	UC-19.1.1.6	3.2.1.28	Manage Recruitment	Detail Job	InterviewServices	TC-56
FR-RM-008	UC-19.1.1.7	3.2.1.28	Manage Recruitment	Detail Job	InterviewServices	TC-56
FR-Admin-DASH-001		3.2.1.25	-	-	RoleService	PASSED
FR-Admin-DASH-002		3.2.1.26	-	-	Permissions	PASSED

Chapter7 Future Prospects

1 Introduction

The AI-powered recruitment platform has successfully streamlined job matching, candidate assessment, and professional networking. However, as recruitment technologies evolve, there is potential to enhance the system further. This chapter outlines key future improvements that can be integrated into the platform to optimize the hiring process, improve user engagement, and enhance decision-making through AI-driven automation. The following enhancements aim to provide a more **comprehensive, intelligent, and user-friendly recruitment experience**.

2 Proposed Future Enhancements

2.1 AI-Powered Interview Question Generation

Building upon the existing interview scheduling, this enhancement will leverage AI to dynamically generate interview questions tailored to specific job roles and candidate profiles. This aims to standardize the interviewing process for recruiters while ensuring relevance and depth in assessments.**Key Functionalities:**

- **Contextual Question Generation:** The system will analyze the job description, required skills, and the candidate's CV and quiz results to suggest relevant interview questions.
- **Behavioral & Technical Questions:** Generate a mix of behavioral (e.g., "Tell me about a time...") and technical questions specific to the role.
- **Interviewer Guidance:** Provide interviewers with a structured set of questions and potential follow-ups, ensuring comprehensive evaluation.
- **Performance Metrics Integration:** Link interview questions to specific competencies or skills, allowing for structured scoring and objective feedback collection.

This enhancement will streamline interview preparation for recruiters, ensure more consistent and effective candidate evaluation, and ultimately lead to better hiring decisions.

2.2 Advanced Job Recommendation System

While basic job matching exists, this enhancement will evolve the recommendation engine to provide highly personalized and accurate job suggestions for job seekers by integrating a hybrid approach, combining content-based and collaborative filtering methods.**Key Functionalities:**

- **Hybrid Recommendation Engine:** Combine current content-based matching (skills, experience, job description) with collaborative filtering (suggesting jobs liked by similar users or jobs saved/applied to by the user's connections).
- **Behavioral Analysis:** Incorporate job seeker interactions (e.g., jobs viewed, saved, applied to, search queries) to refine recommendations over time.
- **Skill Gap Identification:** Based on desired roles, the system will identify skills lacking in a candidate's profile and suggest relevant learning resources.
- **Explainable Recommendations:** Provide job seekers with clear reasons why a job was recommended, fostering trust and transparency.

By implementing these features, job seekers will discover more relevant opportunities faster, reducing search fatigue and improving overall platform engagement and satisfaction.

2.3 Comprehensive Applicant Tracking System (ATS) Features

To provide companies with more robust tools for managing their recruitment pipeline, this enhancement will integrate a full suite of Applicant Tracking System functionalities, extending beyond simple applicant listing.**Key Functionalities:**

- **Visualized Recruitment Pipeline:** Implement a customizable drag-and-drop interface (e.g., Kanban board) to visualize applicants moving through different hiring stages (e.g., "Applied," "CV Review," "Interviewing," "Offer Sent," "Hired").
- **Automated Stage Transitions:** Allow companies to set up rules for automatically moving applicants between stages based on quiz scores, interview feedback, or manual actions.
- **Centralized Applicant Communication:** Manage all correspondence (emails, messages) with applicants from a single dashboard, including templates for common communications (e.g., rejection, interview invite).
- **Collaboration Tools:** Enable team members within a company to add notes, assign tasks, and discuss applicant profiles internally.

This enhancement will significantly improve the efficiency and organization of the hiring process for companies, providing a holistic view and control over their applicant pool.

3 Conclusion

The proposed enhancements will significantly improve the platform's **efficiency, user experience, and AI-driven decision-making capabilities**. By implementing **Visualized Recruitment Pipeline, Automated Stage Transitions, video interviews, Collaboration Tools**, the system will become a **fully comprehensive recruitment platform** that streamlines every stage of the hiring process. These future developments will ensure that **both job seekers and recruiters benefit from an intelligent, automated, and highly interactive recruitment system**.

Chapter8 Conclusion

1. Introduction:

This chapter presents a summary of the findings, achievements, and key insights derived from the development and implementation of the AI-powered recruitment platform. It highlights how the system successfully addresses the challenges in traditional hiring processes and how the integration of AI technologies enhances efficiency, accuracy, and user experience.

2. Summary of Achievements:

The AI-powered recruitment platform was designed to modernize and streamline the hiring process through intelligent job matching, automated candidate assessments, and enhanced social interaction features. The system successfully implemented several core functionalities, including:

AI-Driven Job Matching: Utilizing Natural Language Processing (NLP) and AI algorithms, the platform efficiently matches job seekers with relevant job opportunities based on their skills, experience, and preferences.

Automated Skill-Based Assessments: The system generates dynamic quizzes tailored to job descriptions, allowing employers to evaluate candidates objectively.

Comprehensive Recruitment Management: Employers can post jobs, review applicants, analyze candidate suitability using AI-based scoring, and send job offers through an intuitive interface.

Enhanced User Interaction: The platform includes social features such as networking, following professionals, creating posts, and engaging with job-related content, fostering a dynamic professional environment.

Scalable and Secure Architecture: Built with Django REST Framework for the backend and ReactJS for the frontend, the system ensures a secure, efficient, and scalable user experience.

3. Addressing the Challenges of Traditional Recruitment

The project successfully tackled key inefficiencies in conventional hiring processes:

- **Reducing Hiring Time:** By automating resume evaluation and job matching, the platform minimizes manual effort for recruiters.
- **Improving Candidate Evaluation:** AI-powered assessments ensure an unbiased and objective approach to evaluating job seekers.
- **Enhancing User Experience:** The integration of AI recommendations, professional networking, and personalized job suggestions provides a seamless experience for job seekers and employers.

4. System Reliability and Performance

Through rigorous testing and validation, the platform demonstrated high reliability, security, and performance:

- **Authentication & Security:** Implemented JWT-based authentication and role-based access control for secure user management.

- **Optimized Performance:** AI algorithms were optimized for quick processing of job recommendations and skill assessments.
- **Real-Time Updates:** 3Notifications ensure real-time interaction and engagement.

5. Conclusion

The AI-powered recruitment platform represents a significant advancement in modernizing the hiring process. By integrating AI-driven job recommendations, automated assessments, and intelligent networking, the system provides an efficient and user-friendly experience for job seekers and employers. With a scalable and secure architecture, the platform is well-positioned to adapt and expand in the evolving recruitment landscape.

This project has successfully met its objectives, laying a solid foundation for further enhancements and innovation in AI-driven recruitment solutions.

6. References:

1. (Holovaty, The Definitive Guide to Django: Web Development Done Right, 2009)

2. (Banks, Learning React: Modern Patterns for Developing React Apps., 2017)

3. (Lutz, Learning Python., 2013)

4. (DuBois, MySQL (Developer's Library)., 2013)

5. (Ricci, Recommender Systems: Introduction and Challenges., 2015)

6. (Mikolov, Efficient Estimation of Word Representations in Vector Space., 2013)