SKILLSELECTAI

Group Members

Raja Ramish Tahir (01-131222-040)

Yashal Naeem (01-131222-049)

Supervisor: Ma'am Rafia Hassan

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

THESIS COMPLETION CERTIFICATE

Student Name: Raja Ramish Tahir Enrolment No: 01-131222-040

Student Name: Yashal Naeem Enrolment No: 01-131222-049

Program of Study: Bachelor of Software Engineering

Project Title: SkillSelectAI

It is to certify that the above students' project has been completed to my satisfaction, and I believe that this standard is appropriate for submission for evaluation. I have also conducted a plagiarism test of this thesis using HEC prescribed software and found a similarity index at _____ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: _____

Date: _____ Name: Engr. Rafia Hassan

CERTIFICATE OF ORIGINALITY

This is to certify that the intellectual contents of the project SkillSelectAI are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, the internet, etc. solely to support, elaborate, compare, extend, and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor shall it be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorize the University to cancel my/our degree.

Name of the Student: Raja Ramish Tahir

Signature: _____ Date: _____

Name of the Student: Yashal Naeem

Signature: _____ Date: _____

SKILLSELECTAI

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation, and Infrastructure
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	
4	Familiarity of issues	Involve infrequently encountered issues	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

SkillSelectAI is an AI-powered recruitment platform developed to automate and optimize the hiring process. Traditional recruitment procedures are often slow and inefficient because of manual CV screening, delays in interview scheduling, and difficulties in evaluating candidates accurately. Moreover, recruitment information is commonly spread across different platforms, making candidate management challenging for recruiters and HR teams.

To solve these issues, SkillSelectAI provides an intelligent recruitment solution that automates resume screening, candidate-job matching, interview scheduling, and AI-assisted candidate evaluation. The platform is designed to improve recruitment efficiency, reduce manual workload, and enable faster and more effective hiring decisions.

SkillSelectAI offers recruiters a centralized system for managing job postings, candidates, and interview workflows. The platform uses Natural Language Processing (NLP) techniques for resume parsing and intelligent candidate-role matching based on skills, qualifications, and job requirements. It also supports template-based interview questions customized for specific job roles, along with asynchronous voice and video interviews that provide flexibility for both recruiters and applicants.

The system further functions as an AI-powered evaluation platform that integrates web-based interfaces, backend data management, and intelligent processing services to support automated assessment workflows. It analyzes multimodal inputs including speech, text, tone, and visual behavioral cues to generate structured evaluations and recruitment insights. Advanced technologies such as Natural Language Processing, speech recognition, and behavioral analysis are utilized to interpret candidate responses and interactions effectively.

In addition, the platform automatically stores and organizes interview recordings, assessment reports, and generated evaluation results for future access and review. By combining multimedia analysis with centralized data management, SkillSelectAI enables scalable, consistent, and data-driven decision-making across recruitment and candidate evaluation processes.

Dedication

Firstly, this work is dedicated to God Almighty, our Lord and Creator, whose blessings have been the reason behind its successful completion. Our power, wisdom, and determination can be credited only to Him. It was because of His blessings that this work became a reality. We thank God for His mercies at every stage of our efforts to complete this work.

Secondly, this work is dedicated to our dear parents and revered teachers. Without their blessings and guidance, it would not have been possible to achieve this work. In particular, we thank our revered supervisor, Ma'am Rafia, who has been instrumental in guiding us throughout the development of this work through her professional assistance and motivation.

This success is a result of the determination and self-confidence of those people who have always believed in us without any conditions. We feel extremely grateful to them and thank them for always standing by our side.

To our parents and our supervisor for their constant support and encouragement

Acknowledgments

We start by giving a heartfelt thanks to Almighty Allah SWT, who has helped us to complete our Final Year Project titled SkillSelectAI.

We also owe an immeasurable debt to our supervisor, Ma'am Rafia, who has guided and provided positive feedback throughout this project. She reminded us not to lose track and inspired us throughout by providing us with constant supervision, constructive ideas, and encouragement. She was ever willing to assist us on any challenge that we faced and her effort was quite instrumental in enhancing our work. Additionally, we would also wish to acknowledge the Department of Software Engineering and all the revered faculty members who have made our learning experience so accommodating and have assisted us with the knowledge and resources needed in our project.

Lastly, we are so thankful to our cherished parents and relatives that prayed and loved us amidst our resilience.

Table of Contents

List of Figures

List of Tables

Chapter # 1

Introduction

Chapter 1

Introduction

In this chapter, I will be dissecting SkillSelectAI by getting into the basic components as well as the issues it seeks to address. The project is practically a cross platform tool that is put in-between recruiters and candidates to do the heavy lifting of the early picks of hiring. The system can analyze more than a text by parsing CVs and carrying out multimodal interviews, based on the tonal, facial expressions, and even the precision with which a candidate responds to the question. This is useful in establishing a hierarchical list of candidates on the basis of verbal and non-verbal communication, thus making the entire screening process a bit more objective. This was one of the primary motives behind the construction as it was necessary to reduce huge volumes of manual labor that HR teams can encounter. To address this, I paid attention to the conformity of the selection process where each candidate would be assessed according to the same criteria. I also integrated a backend Admin Module to make sure the system stays reliable. This will enable real time monitoring and auditing of technical performance to ensure that the automation is smooth and gives the recruiters an advantage of high level of monitoring.

1.1 Motivation

The fact that traditional hiring is rather broken was the key driving factor when we got to work on SkillSelectAI. It is simply unrealistic to expect a human to sift through the applications to a job posting with hundreds of applications in a single job posting. The exhaustion sets in, and even very good applicants have been missed out on the job only because their resume was at the end of a colossal stack. When determining this solution, I wished to create something that will help anyone who applies to get a fair opportunity no matter the time they applied.

The other large problem that I observed was the subjectivity in human interviews. Depending on the mood or own bias of a recruiter ratings can change, that is not fair towards the candidate. Objectivity can be given priority by shifting towards an automated system. The AI does not prioritize any individual and instead of relying on mere gut feeling to come up with a score, it uses data and particular metrics. It is a method by which companies can locate the finest talent in the most brief period whilst maintaining the overall procedure both coherent and clear.

1.2 Problem statement

Recruitment can be a pain in itself since it is a process that is normally slow and fragged. Recruiters have to deal with stacks of resumes to which they must carefully sort through each individual one by hand, not only exhausting to do but also a formula that leads to human error. Such an ineffective process of shortlisting means that the selection of candidates is ineffectual or more of a personal vibe of a recruiter than actual data. Besides that, organizations typically have five various screening, interviewing, and tracking tools, and the entire workflow appears to be disorganized and disjointed.

To resolve this, I understood that we had to have one, integrated system that incorporates everything into one roof. The concept was to develop a system that will automate the resume screening and have a structured form of interview with pre-set questions. The guesswork will be eliminated by introducing AI to determine how applicants react to the situation in reality. By doing so, recruiters make far smarter decisions as it reduces manual work, makes the evaluation criteria similar and uniform across the board, and ultimately makes the hiring process seem like an easy-breezy, logical process.

1.3 Objectives

- The goal is to create an AI-based recruitment system that will accelerate the process of hiring people, decrease manual workloads, and improve the quality of decision-making.
- The system is expected to analyze CVs and interpret them by converting unstructured resume data into organized information about skills, education, and experience of a person.
- Another goal of the system is to identify how well a particular candidate corresponds to job requirements and produce shortlists of candidates according to how suitable they are for the position.
- During interviews, the system is going to analyze verbal responses and general behavior of a person for getting a better understanding of their abilities and skills.
- To create a fair and consistent scoring system that reduces human bias and evaluates all candidates using the same standards.

1.4 Main Contributions

What sets SkillSelectAI apart is its ability to manage the entire process of recruiting from start to finish in an effective manner. Unlike solutions that have multiple separate components which do not integrate well together, I came up with one solution that covers all aspects of recruiting.

Here are the main contributions of the project:

- **Seamless Recruitment Pipeline:** We designed a solution that includes everything from creating a job and analyzing resumes to scheduling and conducting interviews and ranking candidates.
- **Smart Resume Matching:** The system uses a blended methodology to extract relevant data from resumes. Instead of using only keywords, the system considers semantic relevance and similar skill sets to measure how much the candidate fits the specific role.
- **Role-Specific Questions:** In order to make sure that the interview process is conducted fairly, I have devised a mechanism whereby the interviews can be conducted using questions which are generated automatically and tailored for the job in question.
- **Multimodal AI Analysis:** One of the biggest problems was the creation of an assessment process that would go beyond text-based evaluations. Through natural language processing of the content, audio processing of the voice, and visual processing of the behavior, the system creates a much better and broader assessment of the applicant.
- **Decision Support for Recruiters:** For the recruiters to be kept updated, the system provides explicit information about their scores and ranking. This promotes making decisions based on facts rather than using one's gut feeling.
- **Flexible Interview Modes:** The software supports the option for either video or audio only interviews. The assessment is specifically designed according to the selected format, thereby giving companies more leeway in terms of meeting their needs.
- **User-Focused Web Design:** Lastly, I implemented it as a full-stack web application with an up-to-date user interface design. The aim was to provide an easy-to-use platform for both job seekers and recruiters in charge of the recruitment process.

1.5 Report organization

The brief structure of our report is as follows:

Chapter:1 1

In Chapter 1, there is an overview of the project, which includes the objective of SkillSelectAI, motivation for the creation of the software, the problem statement, and the major contributions. This chapter creates a solid base from where one can understand the need and importance of the project.

Chapter:2

This chapter focuses on providing an overview related to artificial intelligence-based recruitment tools. It examines the previous works done in this regard, along with other similar approaches. Moreover, the chapter highlights the limitations of the current recruitment tools. This chapter also outlines the research gap addressed by SkillSelectAI.

Chapter:3

The requirements for SkillSelectAI will be discussed in Chapter 3. This chapter will include both high-level and detailed use cases, along with functional requirements. Interface requirements, database requirements, and non-functional requirements will also be discussed, along with the feasibility study of the system.

Chapter:4

The system design of SkillSelectAI is discussed in Chapter 4. This chapter discusses the general structure of the system, design approach, and design limitations. Furthermore, it includes several UML diagrams, such as class diagram, sequence diagram, activity diagram, and ER diagram.

Chapter:5

In Chapter 5, it is explained how the process of implementing the system takes place by explaining how the different modules such as resume analysis, interviews, analysis through artificial intelligence, and job matching are built and combined together.

Chapter:6

The 6 chapter explains the process of testing and evaluation of SkillSelectAI. This involves several types of tests like unit test, integration test, system test, and use case-based test to validate the functionality of the system.

Chapter:7

Chapter 7 rounds off this report by summarizing the entire project, its contribution to knowledge and its results. Chapter 7 will also talk about possible improvements that could be done on the existing system.

Chapter # 2

Background study/Literature Review

Chapter 2

Background Study/Literature Review

The overall concept of SkillSelectAI lies in re-engineering the conventional recruitment approach by streamlining it through advanced artificial intelligence technology. Given how rapidly technologies such as natural language processing and multimodal AI have evolved in recent years, it is evident that modern recruitment practices should be steered in a data-centric direction. In today's highly competitive talent market, organizations simply receive too many applicants to sort manually without compromising on accuracy and fairness.

All companies recognize that hiring is an essential activity, but the traditional way of doing things just seems a little cumbersome at times. It’s not only time-consuming but also manual and, to tell you the truth, quite biased too. Most companies end up manually going through a candidate’s resume and hiring him or her based on what they call their “gut feeling.” There is also a lack of software solutions that offer resume scanning, job matching, and interview scoring all in one package.

In order to solve the mentioned problems, I have created a comprehensive system which can take care of all the most labor-intensive aspects of this work. It is capable of parsing resumes, matching applicants with their proper job positions, and generating interview questions for them. The most interesting part is that my system utilizes multimodal evaluation, meaning that it takes into account not only what the applicant says, but also his/her behavior and communication skills. As a result, everybody gets evaluated according to the same criteria.

2.1 Analysis of the existing work- Where is it strong and where is it weak?

Several big names such as HireVue, Interviewer.AI, and Intervue already exist in the world of online recruitment. They do quite a good job at video interview recording and screening through AI technologies. Thus, HireVue will be a great solution for large corporations that are willing to automate their hiring process. It works excellently with scheduling and integrates perfectly with the company's current management system. Interviewer.AI is another platform that deals with asynchronous interviews and evaluates candidates' competencies through AI

scoring displayed on an online dashboard. Intervue, however, uses a completely different strategy.

Although these tools are undoubtedly helpful in accelerating the process and making it more structured, they do come with several disadvantages. They are mainly designed for recruiters, thus providing little to nothing in terms of feedback for candidates. Neither HireVue nor Interviewer.AI gives any suggestions for improvement, while the fact that Intervue is based on the expertise of humans makes it difficult to scale. Another drawback of these tools is that they seem to prefer taking either-or options, focusing exclusively on human evaluation or artificial intelligence.

This is what led to my creating SkillSelectAI with a slightly more balanced perspective. As opposed to these other systems that focus mainly on either voice or video interviews along with their ability to analyze emotions and confidence levels, our system takes both into consideration. Our system has been designed in such a way that the scores will be automatically calculated and allow for manual intervention if desired. This makes it easier for recruiters to leverage both perspectives.

2.2 Other Similar Work

In terms of current platforms like HireVue, Interviewer.AI, and Intervue, they each have their place when it comes to recruitment. HireVue does all the heavy lifting of enterprise automation for big companies, while Interviewer.AI excels at rapid asynchronous interviews and Intervue employs professionals in order to make complex analysis possible. The truth is, they work really well for winnowing out thousands of applicants. However, what strikes me as being the case is that they tend to be somewhat disjointed in nature.

One of the problems I identified is that all of these solutions are designed to help the recruiter reach a conclusion, with absolutely no thought put into what the candidate should do. In most cases, the only thing a candidate will get is either a simple "yes" or "no," or even an arbitrary number that doesn't give anything in return. Moreover, it appears that these platforms choose their sides and either go with pure artificial intelligence analysis or purely human-based assessment. My idea was to develop a more equal approach, allowing for customized interview flows and candidate behavior analytics.

How SkillSelect AI resolves these issues:

- Offers interviews using both voice and video-based AI.

- Conducts multimodal assessment through facial expressions, tone of voice, and quality of responses.
- Provides automated feedback to candidates about their performance.
- Includes an AI-assisted score with manual verification options.
- Integrates scheduling features and a recruiter dashboard.
- Features custom question templates and dynamic interviews.

Whereas other systems do not incorporate all aspects involved in an AI-based interview and evaluation process, SkillSelect AI aims to be a complete solution by incorporating features that enable its use as an interviewing and assessment tool.

2.3 Literature review:

- **Smart Employer – The Revolution of Recruitment with AI [1]**

  In this study, the focus will be on the development of the AI-based platform for preparing for job interviews using NLP techniques to enhance one's performance. Essentially, the tool analyzes responses of candidates and provides structured feedback for them to improve their communicative skills through simulation of the interview process. However, the issue with it is the narrow scope limited to textual and verbal input, failing to include the fascinating multimodal aspects such as behavioral analysis and facial recognition. SkillSelect AI makes a leap forward in this aspect by analyzing both verbal and visual cues to provide the most comprehensive assessment.

  From my preliminary research, one interesting point was that it cannot be concluded about a person's performance simply by looking at what he or she is saying. Thus, to address this challenge, the solution implemented will allow for integrating behavioral tracking into the process. This allows filling a rather large gap created in previous studies that limited themselves to NLP-based assessment. Thus, the solution will provide a more holistic approach and allow recruiting managers to evaluate performance more comprehensively.

- **Development of an AI-Based Interview System for Remote Hiring (IJARET, 2021) [2]**

  The research introduces an artificial intelligence solution designed for remote interviewing. As its advantage, it offers analysis not only of the text but also of the intonation and facial expressions. The idea behind it seems quite noble because it is

aimed at standardizing remote hiring processes. Still, it should be noted that the system has limitations since there is no "big picture". In particular, it does not support CV parsing, scheduling, and candidate ranking. The system is unable to provide a dedicated dashboard and feedback for candidates; hence, it is rather limited.

To compensate for all the shortcomings described above, SkillSelect AI was developed as an integrated solution. Namely, the structure of the system has been expanded by adding structured scoring as well as a centralized platform for management of the recruiters. In other words, rather than having just the part dealing with analysis of the interview, a whole process is covered by a single product.

- **Interview AI-ssistant: Real-Time Human-AI Collaboration (Zhe Liu, 2025) [3]**
  The article describes an AI that assists the interviewer in conducting the job interview process through real-time suggestions on questions to ask and other insights. It is meant to assist the human interviewer in decision-making rather than conducting the interview entirely independently. The drawback here is the lack of ability to assess candidates' responses, which is an important function for an AI in assisting in recruiting processes.
- SkillSelect AI that I have developed, on the other hand, completely takes responsibility of conducting the interview process off the shoulders of the recruiter. One of the challenges that had to be overcome in designing such an AI was its robotic character, so the tool was designed to conduct the whole interview process on behalf of the recruiter. However, the element of human review remained an option, so a recruiter can participate at any point in the decision-making process.
- **AI-Powered Mock Interview System with Real-Time Voice & Emotion Analysis (IJNRD, 2025) [4]**
  It's an interesting approach because it emulates the environment of an actual interview using real-time data about the voice, emotions, etc. The algorithm evaluates not only speech patterns and intonations but modifies the set of questions depending on the interviewee's response. Although it is an effective way to practice, it remains a simulation tool only since it doesn't assist organizations in hiring people because there is no such important boring part as CV management, candidate ranking, or scheduling.

  The solution was to take the core aspect of behavioral analysis, implement it into the whole system with many features that can be useful for recruitment managers. With the

introduction of SkillSelect AI, one can not only simulate a real-life interview but also schedule interviews, analyze CVs, conduct rankings of candidates, etc. There was one major difficulty in implementing this feature: the inability to integrate deep analysis of emotions with the ranking engine. However, this issue was successfully resolved.

- **Smart Hire: AI Interview Platform (IJRASET, 2025) [5]**
  This solution is quite neat in that it employs artificial intelligence to conduct the interview, assessing candidates not only for their communication skills but also based on confidence and content of responses. Some cool elements it features include adaptive questioning and speech recognition technology to make the process more engaging. However, the issue with this product is its rather closed nature. This particular solution isn't fully integrated into recruitment processes; it is thus devoid of a central dashboard and does not provide any kind of hiring workflow.

In this instance, the goal was to enhance the features through a development process whereby they will be used efficiently in addressing challenges facing recruitment. Recruiting teams face inefficiencies because they utilize several systems to conduct interviews, track candidates, and evaluate their capabilities. SkillSelectAI is intended to solve these challenges by offering an integrated platform that brings together recruitment activities in one location, hence streamlining recruiting efforts.

Recruiters are now capable of using a single platform to collect information about potential employees, conduct interviews, and evaluate their suitability to work in the firm. Apart from performing basic recruitment functions, SkillSelectAI provides insights on hiring decisions based on artificial intelligence (AI) technologies. An important consideration made during development was ensuring that there was effective interaction between various aspects of the system. It was imperative to develop a system whereby information obtained from the interview module would easily be channeled to the recruiter dashboard. In this manner, recruiters have access to information concerning candidate performances at one point.

Furthermore, the whole process is designed to enhance collaboration and usability of the recruitment team through less manual input and lower risks of missing or misplacing information on candidates. The combination of the three key functions, interview management, artificial intelligence evaluation and reporting into one tool has enabled SkillSelectAI to offer an efficient recruiting solution.

Chapter # 3

System Requirements

Chapter 3

System Requirements

System Requirements are the computer's specifications required to ensure efficient operation of SkillSelectAI. System Requirements guarantee proper functioning of various aspects including candidate evaluation using AI technology, management of interviews, and proper processing of data.

3.1 System Level Use Case Diagram

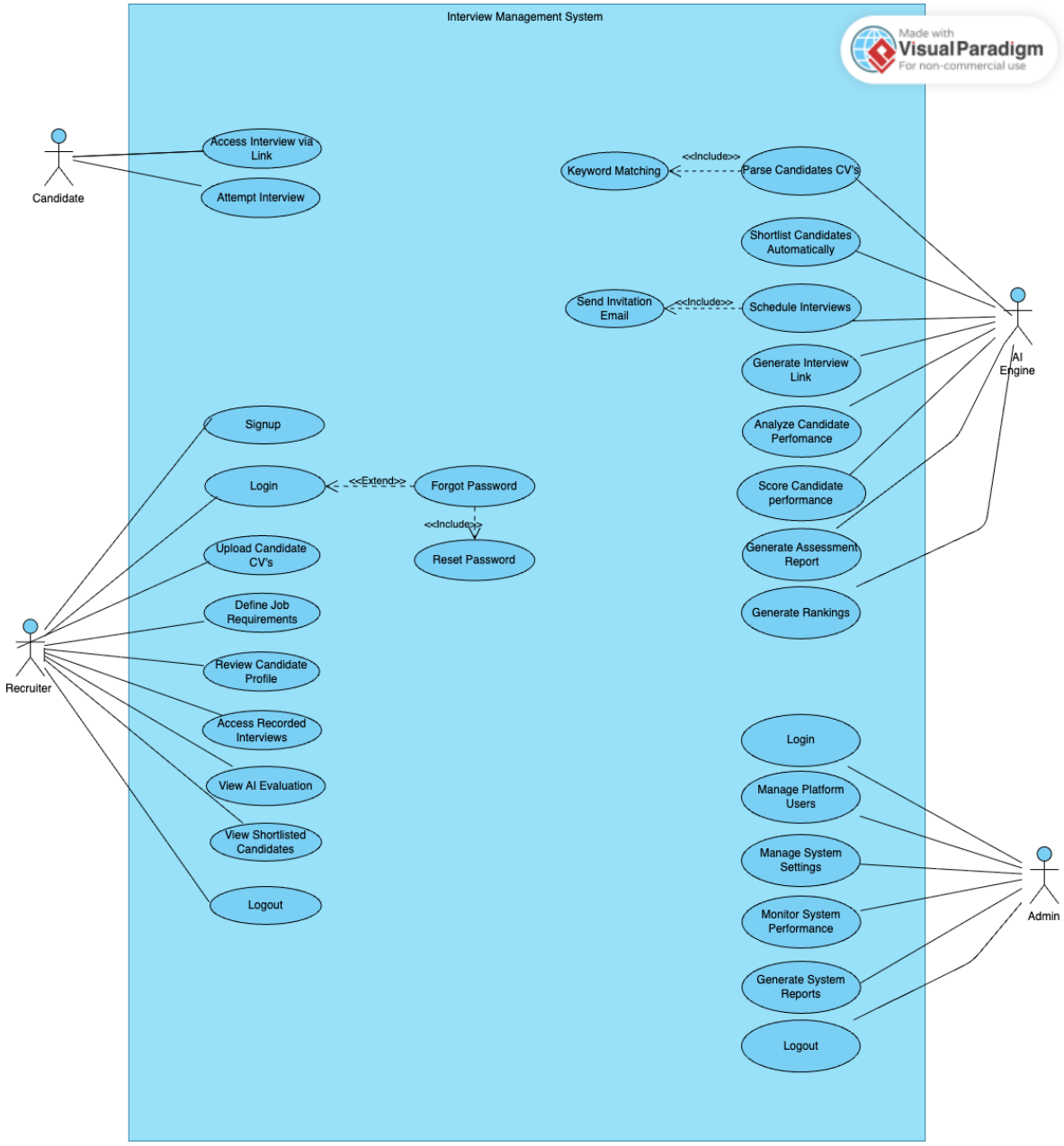


Figure 1 System Level Use Case Diagram

Explanation:

The use-case diagram outlined above demonstrates the interactions between Candidates, Recruiters, and Admins, which altogether make the process of hiring less cumbersome for everyone involved. In terms of Candidates, the goal was to make their experience as easy as possible. The idea is that they do not need to go through an elaborate platform but just click the interview link provided by the recruiter and begin the interview.

In the case of Recruiters, the process is slightly more complicated. First, recruiters must register and log into the website, including the "Forgot Password" option because let's face it, people always forget their credentials. Next, recruiters can upload CVs, create job specifications, and view candidate profiles. Furthermore, they are granted access to the recorded interviews and automatic assessments performed by the AI. After evaluating the shortlisted candidates, recruiters are able to safely log out from the application.

The real magic happens in the AI Engine where all the boring tasks will be automated. The AI Engine will automatically read the resume, find keywords, and schedule appointments. At the same time, the process of interviewing includes analyzing the response, creating a report, and sorting out candidates based on their performance score. These functions are performed automatically to spare the time of the recruiter since there is no need for him/her to spend time and effort figuring out who the better candidate is.

Moreover, the AI Engine ensures that the entire recruiting process becomes consistent and fair through evaluating all the candidates based on the same assessment criteria. This software also saves interviews' information, candidate assessments, and reports centrally for easy viewing by recruitment teams at any time. This software not only makes recruiting processes efficient but also promotes quick and well-informed decisions during recruitments. The Admin tool, on the other hand, manages system users and performance.

3.2 Detailed Use cases:

3.2.1 User Management

3.2.1.1 Recruiter Signup:

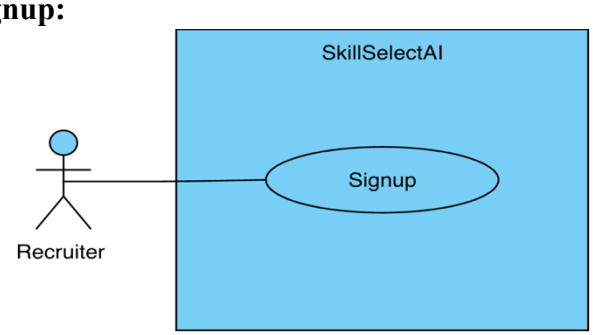


Table 1: Use Case Description of Signup

Use Case ID:	**UC1**
Use Case Name:	Recruiter Signup
Actor(s):	Recruiter
Pre-Conditions:	Recruiter must have a valid business email
Priority:	High
Basic Flow:	1. Open app 2. Click on Signup 3. Enter company name, email, and password 4. Click Register

Actor Actions	**System Response**
The Recruiter enters company name, email, and password. The Recruiter clicks on the "Register" button.	The system checks the user's credentials. If the credentials are valid, the recruiter account is created and a success message is displayed.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. The Recruiter forgot to fill a required field.	An error message will be displayed to fill all fields.
2. The Recruiter enters an invalid email format.	An error message will be displayed: "Please enter a valid email."
3. The Recruiter enters an email that already exists in the system.	An error message will be displayed: "An account with this email already exists."

4. The Recruiter enters a weak password.	An error message will be displayed: "Password must be at least 8 characters and include letters and numbers."

3.2.1.2 Employer Login

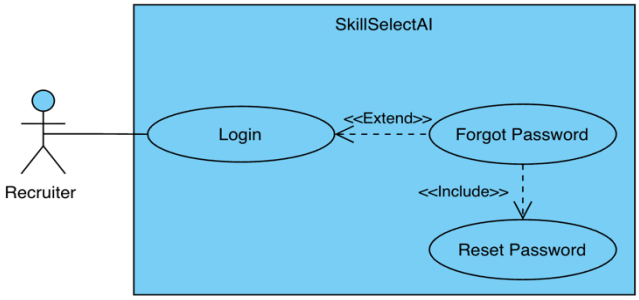


Table 2: Use Case Description of Employer Login

Use Case ID:	**UC2**
Use Case Name:	Employer Login
Actor(s):	Employer
Pre-Conditions:	Employer must have a registered account with a valid email and password.
Priority:	High
Basic Flow:	1. Open login screen 2. Enter email and password. 3. Click Login System redirects to dashboard.

Actor Actions	**System Response**
1. The User enters their registered email and password.	The system verifies the user's credentials and displays the dashboard if valid.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. The user forgot to fill a field.	An error message will be displayed.
2. The user enters an incorrect password.	An error message will be displayed: "Invalid Credentials."
3. The user enters an unregistered email or wrong credentials.	An error message will be displayed: "Invalid Credentials."

3.2.1.3 Forgot Password

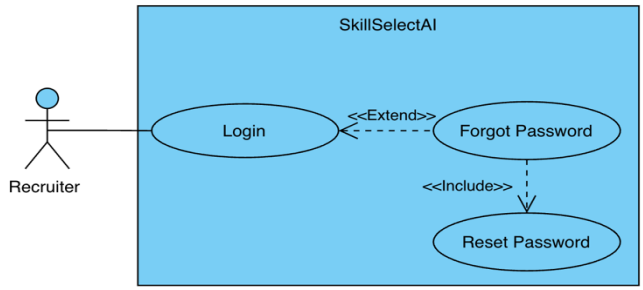


Table 3: Use Case Description of Forgot Password

Use Case ID:	**UC3**
Use Case Name:	Forgot Password
Actor(s):	Recruiter
Pre-Conditions:	Email must be registered
Priority:	Medium
Basic Flow:	1. Click on "Forgot Password" 2. Enter registered email 3. Submit 4. Open email and click reset link 5. Enter new password 6. Submit Password is reset and redirect to login page

Actor Actions	**System Response**
1. The user enters a valid registered email on the Forgot Password page.	The system verifies the email and sends a password reset link to the email address.
2. The user clicks the password reset link and enters a new password.	The system validates the new password, updates it and redirects to the login page.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. The user enters an unregistered email.	An error message will be displayed: "This email does not exist."
2. The user enters a weak password during reset.	An error message will be displayed: "Password is weak. Please use at least 8 characters including letters and numbers."

3.2.1.4 Upload Candidate's CV

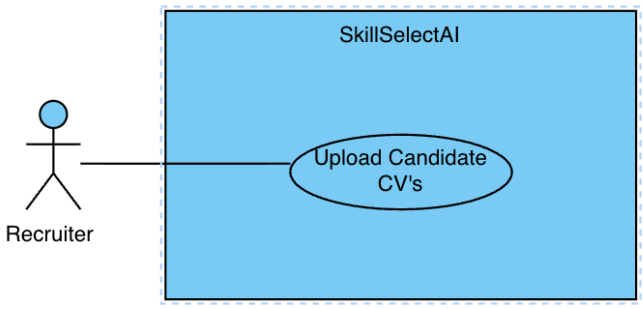


Table 4: Use Case Description of Upload Candidate's CV

Use Case ID:	**UC4**	
Use Case Name:	Upload Candidate's CV	
Actor(s):	Recruiter, System	
Pre-Conditions:	Recruiter must be logged in	
Priority:	High	
Basic Flow:	1. Select one or multiple CV files 2. Confirm upload 3. System processes CVs 4. CVs are linked to Job ID for parsing	
Actor Actions		**System Response**
1. The Recruiter selects one or multiple CV files.		System validates file type, file size, and checks for corruption.
2. The Recruiter confirms the upload.		System stores CVs securely and generates Candidate IDs.
3. The system processes uploaded CVs.		System parses CVs and links them to the relevant Job ID for further analysis.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1. No file is selected and upload is attempted.		System prompts to select at least one CV file before proceeding.

3.2.1.5 Define Job Requirements

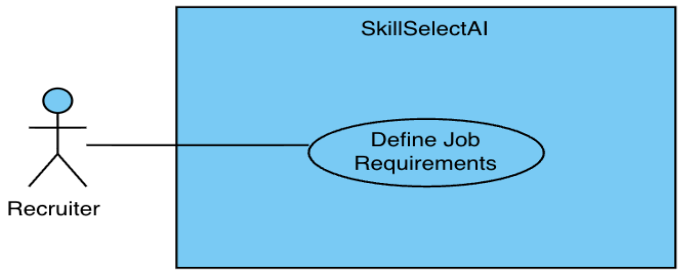


Table 5: Use Case Description of Define Job Requirements

Use Case ID:	**UC5**
Use Case Name:	Create Jobs
Actor(s):	Recruiter
Pre-Conditions:	Recruiter is logged into the system and parsed data must be available.
Priority:	High
Basic Flow:	1. Enter job details (job name and job description) 2. Upload CV files 3. Click "Create Job" System processes and stores job

Actor Actions	**System Response**
1. Recruiter enters job name, job description, and uploads CV files.	System temporarily stores input data and uploaded CV files.
2. Recruiter clicks on "Create Job" button.	System validates required fields and CV's.
3. System processes the request.	System saves job details, stores CV and generates a Job ID.
4. Recruiter observes the result.	System displays a confirmation message.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Job name is empty.	System displays: "Job name is required."
2. Job description is left empty and "Create Job" is clicked.	System displays: "Job description is required."
3. No CV files are uploaded and "Create Job" is clicked.	System displays: "At least one CV file must be uploaded."

3.2.1.6 Review Candidate Profile

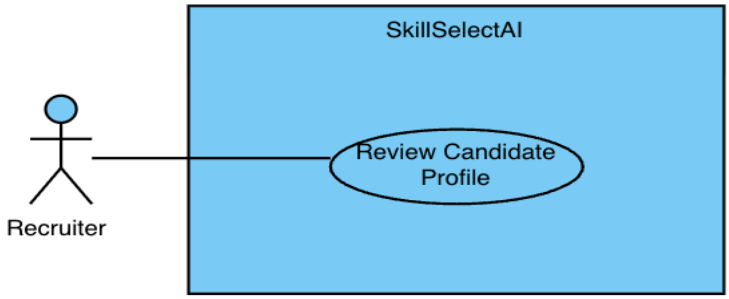


Table 6: Use Case Description of Review Candidate Profile

Use Case ID:	**UC6**
Use Case Name:	Review Candidate Profile
Actor(s):	Recruiter
Pre-Conditions:	CVs must be parsed and candidate profiles must be available in the system.
Priority:	High
Basic Flow:	1. Open candidate profile list 2. Select a candidate profile 3. View parsed candidate details (skills, experience, education) 4. Review and analyze candidate information

Actor Actions	**System Response**
1. Recruiter opens the list of parsed candidate profiles.	System displays all available candidate profiles linked to the job.
2. Recruiter selects a candidate profile.	System retrieves and loads the full candidate details.
3. Recruiter views candidate information (skills, education, experience).	System displays structured parsed data in a readable format.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. No candidate profiles are available.	System displays: "No candidate profiles found."
2. Selected profile data is incomplete.	System displays available data and represents missing fields as empty values.

3.2.1.7 Access Recorded Interviews

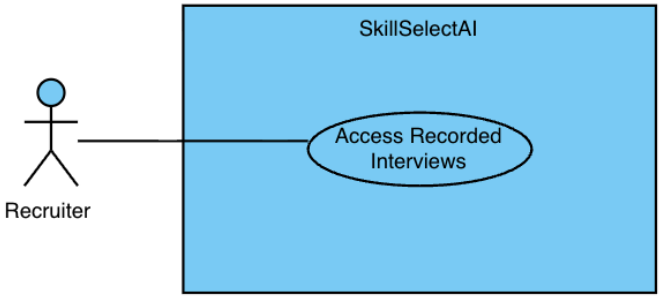


Table 7: Use Case Description of Access Recorded Interviews

Use Case ID:	**UC7**
Use Case Name:	Access Recorded Interviews
Actor(s):	Recruiter
Pre-Conditions:	Candidate profile must exist and interview must have been conducted and stored in the system.
Priority:	High
Basic Flow:	1. Open candidate profile 2. Check interview section 3. Access recorded interview (if available) 4. View or play interview recording

Actor Actions	**System Response**
1. Recruiter opens a candidate profile.	System loads the candidate's stored profile data.
2. Recruiter navigates to the interview section.	System checks for available recorded interviews linked to the profile.
3. Recruiter views or plays the interview.	System streams the recorded interview for review.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. No interview has been conducted for the candidate.	Interview section is not displayed in the candidate profile.
2. Interview recording is unavailable or corrupted.	System displays: "Unable to load interview recording."

3.2.1.8 View AI Evaluation

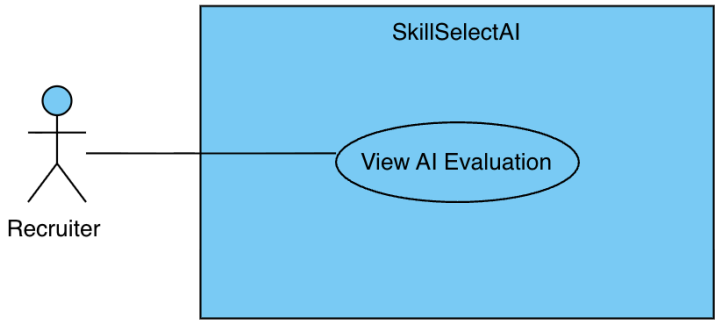


Table 8: Use Case Description of View AI Evaluation

Use Case ID:	**UC8**
Use Case Name:	View AI Evaluation
Actor(s):	Recruiter, System
Pre-Conditions:	AI evaluation must be completed
Priority:	High
Basic Flow:	1. Open candidate list 2. Select candidate 3. Navigate to AI Evaluation section 4. View AI evaluation report

Actor Actions	**System Response**
1. Recruiter opens a candidate profile.	System loads candidate details.
2. Recruiter reviews evaluation results.	System displays full AI evaluation report.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. AI evaluation has not been completed.	AI evaluation section is not displayed.

3.2.1.9 View Shortlisted Candidates

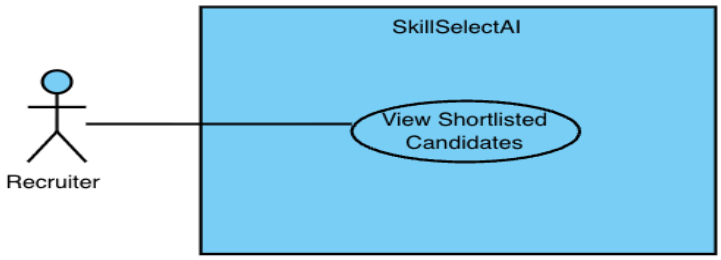


Table 9: Use Case Description of View Shortlisted Candidates

Use Case ID:	**UC9**
Use Case Name:	View Shortlisted Candidates
Actor(s):	Recruiter
Pre-Conditions:	Candidates must be evaluated and ranked for the selected job.
Priority:	High
Basic Flow:	1. Open candidate ranking page 2. Select job 3. View ranked candidates 4. View candidate contact details

Actor Actions	**System Response**
1. Recruiter opens the candidate ranking page.	System displays job selection options.
2. Recruiter selects a job.	System retrieves and displays ranked candidates from highest to lowest rank.
3. Recruiter clicks on a ranked candidate.	System displays candidate's contact information.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. No job is selected.	System does not display ranked candidates.
2. No ranked candidates available for selected job.	System displays empty ranking list.
3. Candidate contact information is missing.	System displays available details and leaves missing fields blank.

3.2.1.10 Employer Logout

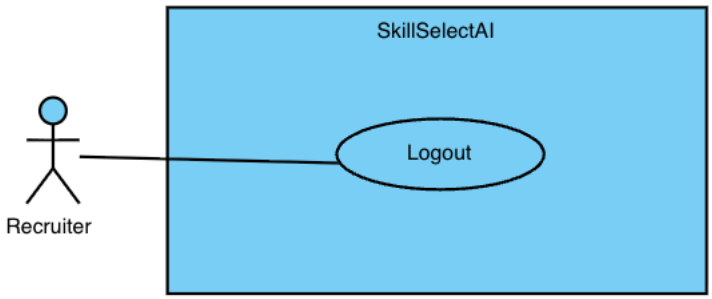


Table 10: Use Case Description of Employer Logout

Use Case ID:	**UC10**
Use Case Name:	Employer Logout
Actor(s):	Employer
Pre-Conditions:	1. The user is logged in 2. The user no longer wants to be logged in.
Priority:	Medium
Basic Flow:	1. The user is done using the application 2. The user clicks on the logout button 3. The system logs the user out.

Actor Actions	**System Response**
1. Users click on the logout button	The landing page is displayed.
Alternative Course of Action (if any)	
Actor Action	**System Response**
NA	NA

3.2.2 Candidate Management

3.2.2.1 Access Interview via Link

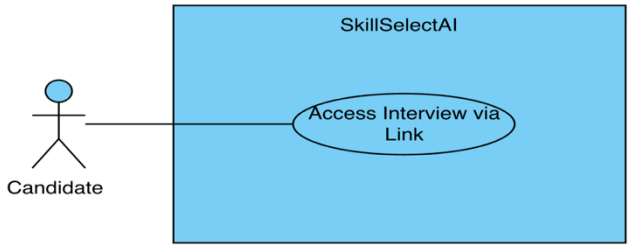


Table 11: Use Case Description of Access Interview via Link

Use Case ID:	**UC12**
Use Case Name:	Access Interview via Link
Actor(s):	Candidate
Pre-Conditions:	Candidate must be shortlisted and interview link must be generated and sent via email
Priority:	High
Basic Flow:	1. Receive interview email 2. Click interview link 3. Redirect to interview page 4. Start interview

Actor Actions	**System Response**
1. Candidate receives interview invitation email.	System sends interview link to shortlisted candidates who meet job requirement threshold.
2. Candidate clicks the interview link.	System verifies the link and redirects to interview page.
3. Candidate begins the interview.	System starts the interview recording and session.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Candidate is not shortlisted.	Interview link is not sent to the candidate.
2. Candidate clicks invalid or expired link.	System displays: “Invalid or expired interview link.”

3.2.2.2 Attempt Interview

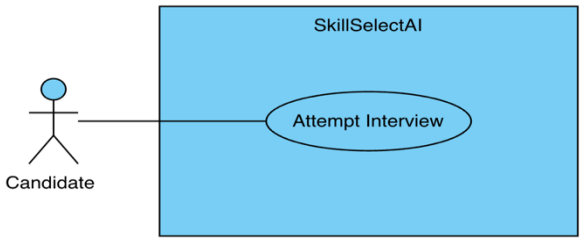


Table 12: Use Case Description of Attempt Interview

Use Case ID:	**UC12**
Use Case Name:	Conduct Interview
Actor(s):	Candidate, System
Pre-Conditions:	Candidate has interview link
Priority:	High
Basic Flow:	1. Click interview link received via email 2. System validates link and loads interview questions 3. Record interview responses (video/audio) 4. Submit interview 5. System saves recordings and sends for analysis

Actor Actions	**System Response**
1. Candidate clicks the interview link received via email.	System validates the link and loads the appropriate interview questions.
2. Candidate begins recording their responses (webcam for video / microphone for audio)	System securely captures and saves the recordings.
3. Candidate submits the interview after completing all responses.	System notifies the module to analyze the submitted responses.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Candidate clicks an expired or invalid interview link.	System displays: “Invalid or expired link. Please request a new interview link.”
2. Candidate’s webcam fails.	System displays: “Unable to detect video device.”
3. Candidate’s microphone fails.	System displays: “Unable to detect audio input.”

3.2.3 AI Engine Management

3.2.3.1 Parse Candidates' CVs

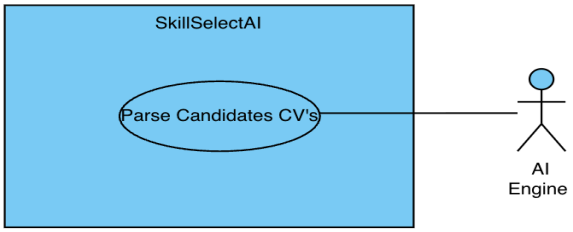


Table 13: Use Case Description of Parse Candidates CV's

Use Case ID:	**UC13**
Use Case Name:	Parse CV's
Actor(s):	System
Pre-Conditions:	CVs have been successfully uploaded to the system.
Priority:	High
Basic Flow:	1. CV files are uploaded to the system 2. System reads and processes the CV files 3. System extracts candidate information (personal details, education, experience, skills) 4. System generates structured candidate profiles and displays parsed results

Actor Actions	**System Response**
1. Recruiter uploads one or multiple CV files.	System reads the file and prepares it for extraction.
2. Recruiter waits while the parsing is being processed.	System extracts personal details, experience, education, and skills using rule-based parsing and regex.
3. Recruiter proceeds to view the parsed result.	System generates a structured candidate profile and displays the parsed data on screen.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Recruiter initiates parsing on a corrupted or incomplete CV.	System extracts partial data and displays: "Some information could not be extracted."

3.2.3.2 Shortlist Candidates Automatically

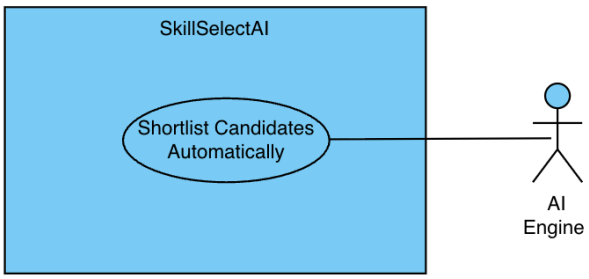


Table 14: Use Case Description of Shortlist Candidates

Use Case ID:	**UC14**
Use Case Name:	Shortlist Candidates
Actor(s):	System
Pre-Conditions:	Job requirements are defined.
Priority:	High
Basic Flow:	1. Define job requirements 2. System compares candidate profiles with job requirements 3. System calculates match scores 4. Candidates are ranked based on scores 5. View shortlisted candidates

Actor Actions	**System Response**
1. Recruiter defines a job requirement.	System receives job requirements and initiates candidate comparison.
2. Recruiter waits for candidate evaluation.	System compares candidate CV profiles with job requirements and calculates match scores.
3. Recruiter views the list.	System displays shortlisted candidates
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Recruiter requests shortlist but no candidate meets minimum match threshold.	System displays: “No candidates meet the job requirements.”

3.2.3.3 Schedule Interviews

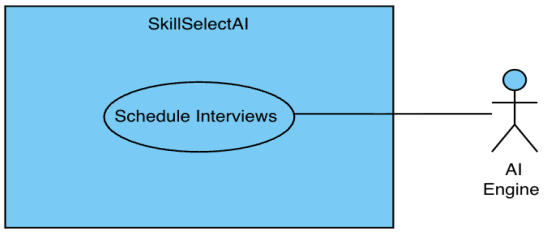


Table 15: Use Case Description of Schedule Interviews

Use Case ID:	**UC15**
Use Case Name:	Schedule Interviews
Actor(s):	System, Candidate
Pre-Conditions:	Candidate profile must exist
Priority:	High
Basic Flow:	1. Shortlist candidates 2. Send interview link 3. Receive interview email 4. Click interview link 5. Start interview session

Actor Actions	**System Response**
1. Profiles are created.	System sends email with interview instructions and secure link.
2. Candidate opens email and clicks the link.	System creates a secure interview session.
3. Candidate reviews instructions and starts interview.	System loads interview session and allows recording.
4. Candidate answers interview questions.	System tracks progress and saves responses.
5. Candidate submits the completed interview.	System attaches interview results to candidate profile.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Candidate clicks interview link after expiration.	System validates link and displays "Invalid or expired link."
2. Camera or microphone permission not enabled.	System detects missing permissions and blocks interview start.

3.2.3.4 Generate Interview Link

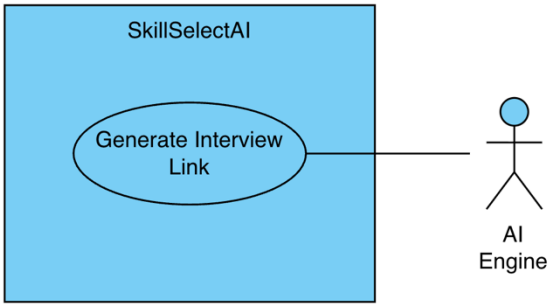


Table 16 : Use Case Description of Generate Interview Link

Use Case ID:	**UC16**
Use Case Name:	Generate Interview Link
Actor(s):	Recruiter, System
Pre-Conditions:	Candidates must be shortlisted for the selected job
Priority:	High
Basic Flow:	1. Select job 2. Click send interview invites 3. System generates interview links 4. Emails sent to candidates

Actor Actions	**System Response**
1. Recruiter selects job and clicks "Send Interview Invites".	System validates request and retrieves shortlisted candidates.
2. Recruiter waits for processing.	System generates unique interview links for each candidate.
3. Invite generation completes.	System sends interview emails to candidates.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Invites already sent.	System prevents duplicate invites.

3.2.3.5 Generate Assessment Report

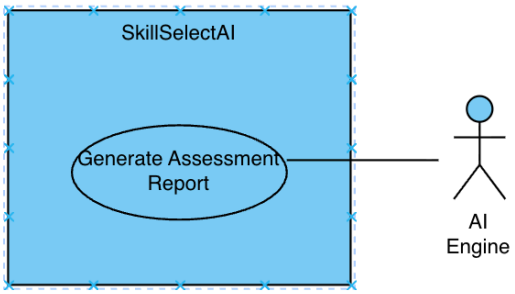


Table 17: Use Case Description of Generate Assessment Report

Use Case ID:	**UC17**
Use Case Name:	Generate Assessment Report
Actor(s):	System
Pre-Conditions:	Candidate recording must exist
Priority:	High
Basic Flow:	1. Candidate submits interview recording 2. System sends recording to AI module 3. AI analyzes responses and generates scores 4. System attaches evaluation to candidate

Actor Actions	**System Response**
1. Candidate recording is available.	System sends recording to AI module.
2. AI module processes recording.	AI analyzes communication, technical, and behavioral skills.
3. AI completes analysis.	AI generates performance summary and score.
4. Results are finalized.	System attaches AI evaluation to candidate profile.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Recording is corrupted or incomplete.	System requests re-attempt of recording.

3.2.3.6 Generate Rankings

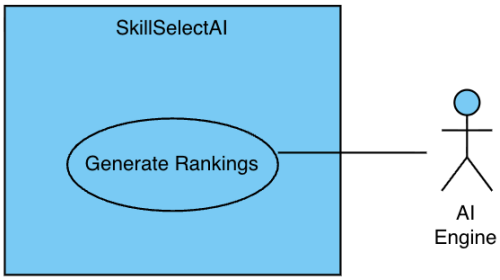


Table 18: Use Case Description of Generate Rankings

Use Case ID:	**UC18**	
Use Case Name:	Generate Rankings	
Actor(s):	System	
Pre-Conditions:	All evaluations must be available	
Priority:	High	
Basic Flow:	1. Retrieve candidate evaluation scores 2. Process scores for ranking 3. Calculate candidate rankings 4. Generate ordered candidate list 5. Display ranked candidates	
Actor Actions		**System Response**
1. System gathers all final evaluation scores from database.		System sends candidate scores to the ranking engine.
2. Ranking engine receives scores and begins processing.		Ranking engine calculates ranking and returns the ordered list.
3. System receives ranked list from ranking engine.		System displays final ranked list to recruiter.
Alternative Course of Action (if any)		
Actor Action		**System Response**
1. Missing score		System excludes candidate.

3.2.4 Admin Management

3.2.4.1 Manage Platform Users

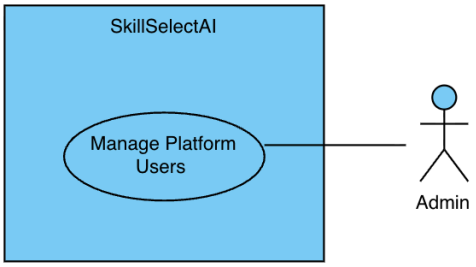


Table 19: Use Case Description of Manage Platform Users

Use Case ID:	**UC19**
Use Case Name:	Manage Platform Users
Actor(s):	Admin
Pre-Conditions:	Admin must be logged into the admin portal.
Priority:	High
Basic Flow:	1. Open admin panel 2. View platform users 3. Search or select user 4. Update permissions or delete user 5. System saves changes

Actor Actions	**System Response**
1. Admin opens recruiter management panel.	System loads platform users.
2. Admin selects a user.	System displays selected user details.
3. Admin updates permissions.	System saves updated user permissions.
4. Admin deletes a user.	System removes user and related data.
5. Admin completes action.	System records admin activity in logs.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. No users available.	System displays empty user list.

3.2.4.1 Monitor System Performance

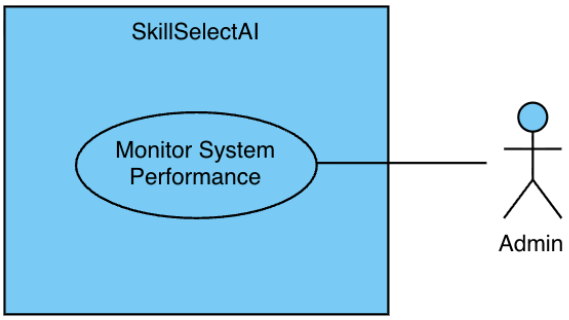


Table 20: Use Case Description of Monitor System Performance

Use Case ID:	**UC20**
Use Case Name:	Manage Platform Users
Actor(s):	Admin
Pre-Conditions:	Admin must be logged into the admin portal.
Priority:	Medium
Basic Flow:	1. Open admin dashboard 2. View system performance 3. Monitor service status

Actor Actions	**System Response**
1. Admin opens system health view.	System displays performance metrics.
2. Admin checks service status.	System shows server, database, and API status.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Service failure occurs.	System marks service as unavailable.

3.2.4.1 Generate System Reports

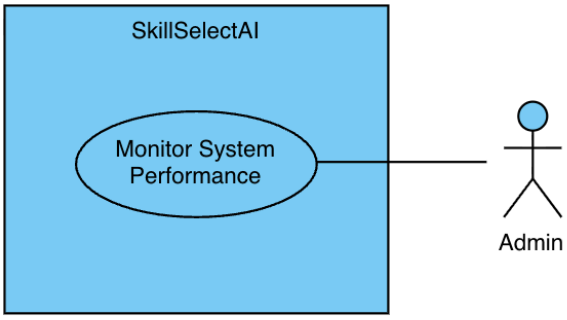


Table 21: Use Case Description of Generate System Reports

Use Case ID:	**UC21**
Use Case Name:	Generate System Reports
Actor(s):	Admin
Pre-Conditions:	Admin must be logged into the admin portal.
Priority:	Medium
Basic Flow:	1. Open admin panel 2. Click generate report 3. Download report

Actor Actions	**System Response**
1. Admin clicks “Generate System Report”.	System compiles system data.
2. Admin downloads report.	System exports report file.
Alternative Course of Action (if any)	
Actor Action	**System Response**
1. Report generation fails.	System displays error message.

3.3 Functional Requirements:

3.3.1 User Management

3.3.1.1 Recruiter Registration

Table 22: Functional Requirement of Recruiter Registration

Identifier	**FR1**
Title	Recruiter Registration
Requirement	Recruiter can register by providing company name, email and password.
Source	Web Application
Rationale	Enables recruiters to access hiring features.
Restrictions and Risk	Duplicate emails must be rejected.
Dependencies	N/A
Priority	High

3.3.1.2 Recruiter Login

Table 23: Functional Requirement of Recruiter Login

Identifier	**FR2**
Title	Recruiter Login
Requirement	Recruiter can login using registered email and password.
Source	Web Application
Rationale	Provides secure access to recruiter dashboard.
Restrictions and Risk	Invalid credentials should be rejected.
Dependencies	FR1
Priority	High

3.3.1.3 Forgot Password

Table 24: Functional Requirement of Forgot Password

Identifier	**FR3**
Title	Reset Password
Requirement	Recruiter can reset password using registered email and reset link.
Source	Web Application
Rationale	Allows account recovery.
Restrictions and Risk	Link may expire.
Dependencies	FR2
Priority	Medium

3.3.1.4 Create Job

Table 25: Functional Requirement of Create Job

Identifier	**FR4**
Title	Create Job
Requirement	Recruiter can create job by entering job details and uploading CVs.
Source	Recruiter Module
Rationale	Enables job-based candidate screening.
Restrictions and Risk	CV upload required.
Dependencies	FR2
Priority	High

3.3.1.5 Upload CV’s

Table 26: Functional Requirement of Upload CV's

Identifier	**FR5**
Title	Upload CV’s
Requirement	Recruiter can upload one or multiple CV files for screening.
Source	CV Processing Module
Rationale	Allows candidate data collection.
Restrictions and Risk	Unsupported formats must be rejected.
Dependencies	FR4
Priority	High

3.3.1.6 View Shortlisted Candidates

Table 27: Functional Requirement of View Shortlisted Candidates

Identifier	**FR6**
Title	View Shortlisted Candidates
Requirement	Recruiter can view candidates selected for job.
Source	Ranking Module
Rationale	Helps recruiter select candidates.
Restrictions and Risk	No candidates if threshold not met.
Dependencies	FR12
Priority	High

3.3.1.7 Review Candidate Profile

Table 28: Functional Requirement of Review Candidate Profile

Identifier	**FR7**
Title	Review Candidate Profile
Requirement	Recruiter can view candidate details and parsed information.
Source	Candidate Module
Rationale	Allows detailed candidate review.
Restrictions and Risk	Missing fields may appear blank.
Dependencies	FR11
Priority	High

3.3.1.8 View AI Evaluation

Table 29: Functional Requirement of View AI Evaluation

Identifier	**FR8**
Title	View AI Evaluation
Requirement	Recruiter can view AI-generated candidate evaluation.
Source	Evaluation Module
Rationale	Helps recruiter decision making.
Restrictions and Risk	Evaluation must be completed.
Dependencies	FR14
Priority	High

3.3.2 Candidate Management

3.3.2.1 Access Interview via Link

Table 30: Functional Requirement of Access Interview via Link

Identifier	**FR9**
Title	Access Interview
Requirement	Candidate accesses interview using secure email link.
Source	Interview Module
Rationale	Allows candidates to attend interview.
Restrictions and Risk	Expired links must be blocked.
Dependencies	FR13
Priority	High

3.3.2.2 Record Interview

Table 31: Functional Requirement of Record Interview

Identifier	**FR10**
Title	Record Interview
Requirement	System records candidate video/audio responses.
Source	Interview Engine
Rationale	Enables AI evaluation.
Restrictions and Risk	Device permission required.
Dependencies	FR9
Priority	High

3.3.3 AI Service Management

3.3.3.1 Parse CV

Table 32: Functional Requirement of Parse CV

Identifier	**FR11**
Title	Parse CV
Requirement	System extracts candidate details such as skills, education and experience.
Source	CV Parsing Engine
Rationale	Converts CV into structured data.
Restrictions and Risk	Incomplete CV may result in missing data.
Dependencies	FR5
Priority	High

3.3.3.2 Rank Candidates

Table 33: Functional Requirement of Rank Candidates

Identifier	**FR12**
Title	Rank Candidates
Requirement	System compares candidate profiles with job requirements and ranks them.
Source	Ranking Engine
Rationale	Identifies best candidates automatically.
Restrictions and Risk	Requires parsed candidate data.
Dependencies	FR11
Priority	High

3.3.3.2 Generate Interview Link

Table 34: Functional Requirement of Generating Interview Link

Identifier	**FR13**
Title	Generate Interview Link
Requirement	System generates unique interview links for shortlisted candidates.
Source	Interview Scheduler
Rationale	Enables remote interview access.
Restrictions and Risk	Duplicate invites must be prevented.
Dependencies	FR12
Priority	High

3.3.3.3 Generate Evaluation Report

Table 35: Functional Requirement of Generating Evaluation Report

Identifier	**FR14**
Title	Generate Evaluation Report
Requirement	AI analyzes recording and generates performance report.
Source	AI Engine
Rationale	Provides automated candidate assessment.
Restrictions and Risk	Recording must exist.
Dependencies	FR10
Priority	High

3.3.3.3 View AI Evaluation

Table 36: Functional Requirement of View AI Evaluation

Identifier	**FR15**
Title	View AI Evaluation
Requirement	Recruiter can view AI-generated candidate evaluation.
Source	Evaluation Module
Rationale	Helps recruiter decision making.
Restrictions and Risk	Helps recruiter decision making.
Dependencies	FR14
Priority	High

3.3.4 Admin Management

3.3.4.1 Manage Platform Users

Table 37: Functional Requirement of Managing Platform Users

Identifier	**FR16**
Title	Manage Platform Users
Requirement	Admin can view, update permissions, and delete recruiter accounts.
Source	Admin Dashboard
Rationale	Controls platform access.
Restrictions and Risk	Deleting user removes related data.
Dependencies	Admin Login
Priority	High

3.3.4.2 Monitor System Performance

Table 38: Functional Requirement of Monitoring System Performance

Identifier	**FR17**
Title	Monitor System Performance
Requirement	dmin can view system health, server status and database connection.
Source	Admin Dashboard
Rationale	Ensures system reliability.
Restrictions and Risk	Requires system monitoring services.
Dependencies	Admin Login
Priority	Medium

3.3.4.3 Generate System Report

Table 39: Functional Requirement of Generating System Report

Identifier	**FR18**
Title	Generate System Report
Requirement	Admin can generate downloadable system report.
Source	Admin Dashboard
Rationale	Enables auditing and analysis.
Restrictions and Risk	Report generation may fail if data unavailable.
Dependencies	Admin Login
Priority	Medium

3.4 Interface Requirements

3.4.1 User Interfaces

React.js was chosen for the development of the front-end because it enables the creation of a contemporary and responsive web interface. As we were striving for a seamless user experience, much emphasis was placed on such classic principles of human-computer interaction (HCI) as consistency and clarity. Thus, one of the key objectives was to avoid any obstacles regardless of whether you are a recruiter or a job-seeker.

To structure our SkillSelectAI implementation, I used a role-based approach. Both recruiters and administrators have access to their respective dashboards. Candidates, however, are not required to register since the only action needed is to complete an online interview without the need for logging in to an account via a safe URL link.

UI-1: User-Centric Design

In constructing the screens, ease of learning and use were important in HCI considerations. This is because my aim was to ensure that the first time user would be able to navigate the platform without consulting a manual. Using the familiar layout and labeling eased the mental burden of both the job recruiter and candidate.

UI-2: Recruiter Dashboard

The recruiter's portion of the platform is the most complex section of the user interface. I have divided it into several modules that will manage the entire process of hiring a candidate. These include:

Job Management: An area for creating and editing job postings.

Candidate Pool & Ranking: The location for seeing all candidates and the ranking done by the AI algorithm.

Interview Management: A module for scheduling and tracking interviews.

UI-3: Candidate Interview Page

In order to simplify things for the candidate, it became imperative to make the process simple. I created a special page for them through which they can log on using a unique link. Without

any login page or dashboard, they can perform their activities here without being disturbed by anything.

UI-4: Admin Dashboard

Admin portal is considered to be the spine for the management of the whole platform. Some functionalities that were added in order to manage the whole system by the admin include the management of users, monitoring of system performance, and generation of reports. This provides a bird's-eye view of the functioning of SkillSelectAI.

UI-5: Single Page Application (SPA) Behavior

In order to create an impression that it is a regular desktop application and not an awkward website, I made sure that the application follows SPA patterns. With the help of React routing, there is no need for a browser to reload all the contents when moving between different pages of the application.

3.4.2 Hardware Interfaces

SkillSelectAI has been developed using web technology, hence the software will be accessible from virtually any type of computer such as laptop, tablet, and desktop provided there is access to the internet. One issue that I needed to consider while developing this software was the requirement for multimedia hardware in order to record the interviews.

- **Camera:** The camera is a must-have since the interview involves videos and thus needs to collect data.
- **Microphone:** A microphone is needed as well, as the video interview involves recording the audio part.
- **Display:** Any regular display will work just fine for both the recruiter and administration dashboards.
- **General Hardware:** Apart from the listed above browser-dependent devices, no other sophisticated or costly hardware was needed.

3.4.3 Software Interfaces

Table 40: Software Interfaces

Software tools and technology	Version	Rationale
Visual Studio Code	1.83	IDE for development
React.js	19.2.0	Front-end UI Framework
React Router	7.13.1	Client-side routing
MUI	7.3.7	UI component library
Node.js	20.x	Backend server environment
Express.js	4.18.2	Web framework for REST APIs
MongoDB	6.x	NoSQL database
Mongoose	9.2.1	MongoDB object modeling
Flask	3.1.2	Python-based interview service
Celery	5.6.2	Background task processing
DeepFace	0.0.98	Emotion detection
MediaPipe	0.10.32	Face & gesture processing
Whisper	20250625	Speech-to-text transcription
TensorFlow	2.15.0	Machine learning models
Groq SDK	0.37.0	AI-based evaluation & generation

Note:

The AI analysis and evaluation during interviews may be relatively time-consuming because of complexity issues. This is addressed through the use of loading bars while the requests are being processed. The response time will depend on audio-video duration and processing load for the AI.

3.4.4 Communication Interfaces

- **Frontend and Backend Link:** The connection between the frontend, built in React, and backend, based on Node.js, takes place via HTTP RESTful API calls.

- **Backend Endpoints:** I configured specific backend routes for the jobs, candidates, interviews, authentication as well as the admin and AI components.
- **The Python Connection:** In order to make use of my AI-related logic which worked better in Python, I had to create a new Flask module which communicated with my Node backend.
- **Handling Communication:** I enabled CORS to make sure that communication between the frontend, Node server, and Flask backend could take place seamlessly.
- **Security and Config:** In terms of security, I decided to implement JWT token-based authentication to secure my API access, with the API keys being in the form of environment variables.
- **Wait Times:** Considering that AI analysis takes time, I dealt with it using asynchronous messages and loading indicators.

3.5 Database Requirements

For the backend implementation, MongoDB has been chosen as the NoSQL database. The reason behind this is the flexible nature of storage options that can store different kinds of data, including the evaluation score of the candidates using artificial intelligence and information regarding the profiles that are expected to undergo changes with time. In order to manage schemas, I have used Mongoose.

DB-1: Database and Schema Management

For this assignment, MongoDB was employed as the main repository for storing data. The use of Mongoose helped me define appropriate schemas while at the same time retaining the scalability of NoSQL database. In this way, it became easy to integrate our node code with the actual data.

DB-2: Data Collections

The process is divided into various major groups that help in managing the process flow. I have created special groups for user, job, candidate, and interview information. Another group contains information on evaluation, which stores the output produced by the AI after analyzing the interview.

DB-3: Record Structure and Validation

In order to ensure that our information will remain clean, each entry in our database will have its own unique identifier. In addition, there is field validation for required fields in order to prevent ourselves from getting ghost entries.

DB-4: Performance Optimization

The more applicants there are, the more likely I am to experience issues with performance. To address this, I implemented indexes on searchable fields, such as email, jobId, and candidateId. With this measure, recruiters are able to locate their candidates or jobs instantly.

DB-5: Security and Access Control

Security was one of the main things that mattered because we were dealing with personal data. I ensured that none of the passwords are stored in a plaintext form; instead, they would be stored using hashing algorithms. In addition to that, I restricted the database's access.

3.6 Non-Functional Requirements

3.6.1 Performance Requirements

For the application to become functional, it was crucial not only that it be easy to use, but also responsive and robust. For example, I determined that the loading time of the dashboard should not exceed three seconds on a working connection. This is because people find it frustrating to wait. We also designed it to serve multiple recruiters and candidates at one time, due to the nature of the job market process. While some of the AI algorithms such as CV processing and evaluation – require more time because of their complexity, they were optimized where possible..

NFR1: System Responsiveness

A problem that occurred was ensuring that the entire application does not get frozen when the AI is processing data behind the scenes. I ensured that the system would remain responsive during these processes to allow a recruiter to continue working while waiting for the CV parsing process to complete.

NFR2: Visual Feedback

As some of the AI operations take some time to complete, I did not want the user to doubt whether the application was working correctly or whether it had shut down. I provided feedback through icons and indicators while processing or evaluating an interview.

NFR3: Optimized Operations

For the core "heavy" functionality, I paid attention to the optimization of response times. Regardless of whether it is about ranking the candidates or creating a complete report, we made sure that there were no delays affecting the productivity.

NFR4: Interface Consistency

In order to ensure consistent interface, I aimed at keeping the interface minimalistic and intuitive enough. From the administrator's interface to the candidate one, the same visual design will be used. Thus, the transition between various parts of the application becomes more intuitive for the user.

3.6.2 Safety Requirements

Given that SkillSelectAI works with confidential data such as individual CV's and interviews, security played a key role in our considerations. We ensured that user data, interview data, and AI outcomes would be safely stored and accessible only by authorized individuals. Another important issue was reliability, as we needed to make sure that no data would get lost and no corruptions would happen.

3.6.3 Security Requirements

NFR5: User authentication is required via secure login using the user's email and password.
NFR6: The application uses JSON Web Tokens for secure authentication of the user session and API communications.

NFR7: Sensitive information like passwords and interview details must be securely stored and kept private.

3.6.4 Software Quality Attributes

3.6.4.1 Usability

I have paid attention to simplicity because it would be very difficult for recruiters, candidates, and admins to use the platform. This was achieved by utilizing a clean layout and logical workflows that make it easy for any user to access everything he needs. The navigation process has been made simple and therefore makes the entire process very easy.

3.6.4.2 Availability

The system was built with the goal of ensuring its availability 24/7, which means that recruiters will not have to worry about working outside regular hours, nor will the time difference become a problem. The user can schedule an interview at night, and the candidate can take the AI evaluation on weekends. In this way, both the recruiter and the candidate can be assured that the system is always up and running.

3.6.4.3 Correctness

The emphasis was made on making SkillSelectAI dependable, meaning that every feature implemented had to function correctly. It means that all operations performed by the system – including generating a job ad without mistakes, processing a CV, and making an assessment and rankings would have to be executed correctly. The need for this level of precision was one of the main concerns for me when working on this project.

3.6.4.4 Flexibility

The architecture was designed such that it will be scalable and flexible in nature, thereby allowing for future development needs. This ensures that any additional AI modules or recruitment features required can be plugged into the current structure without any hassle. Another benefit of having an architecture in place is that it would not need a complete rework when adding external employment websites into the picture.

3.6.4.5 Maintainability

The platform has been created using a modular structure. Therefore, maintaining its operation and fixing the bugs becomes significantly easier because different elements of the software can be updated independently from each other, which prevents any malfunctioning of the system. It was important to create such a design that would not limit further improvements and modifications as the platform grows, but will allow performing all necessary changes behind the scenes without affecting users' experience.

3.6.4.6 Reliability

A special emphasis has been placed on increasing reliability of the platform as well. The goal is to prevent any failures that could happen at a critical stage of work when the platform is analyzing the CV, recording the interviews or performing some of the other workflows. In order to achieve the highest level of reliability, it has been important to focus on avoiding crashes or errors at these particular stages.

3.6.4.7 Reusability

The modularity in the design of this project, where the key modules such as CV parser, candidate ranking and AI assessment are reusable, is another strong point of this effort. Instead of tying everything up in this version of SkillSelectAI, the flexibility of such modules will enable their use in future improvements of the system or even in different systems altogether. Modular design of the application has been a crucial factor to ensure that our current efforts can form the base for future applications.

3.7 Project Feasibility

- Technical Feasibility: In terms of technical feasibility, I decided to make SkillSelectAI web based modular because scalability and manageability are crucial for the success of the project. The modern tech stack that includes React on the frontend, Node.js on the backend and MongoDB on the database layer became the base for creating a powerful solution. In addition, I implemented AI services in Python to cope with complicated data processing related to recruitment.
- Legal and Ethical Feasibility: Speaking about legal and ethical aspects, I focused on designing SkillSelectAI from the perspective of protecting privacy considering the fact that the project involves personal data of candidates. One of the key advantages of automated evaluation is its ability to avoid human bias and thus create an environment where recruitment is much more ethical.

3.8 Conclusion

In this chapter, the requirements for SkillSelectAI have been explained. These include functional and non-functional requirements. In order to optimize the recruitment process using AI, the process includes resume analysis, candidate selection, and evaluation of interviews.

Chapter # 4

System Design

Chapter no. 4

System Design

4.1 Design Approach

For the architecture of the SkillSelect AI application, the chosen architectural model is modular architecture and layered architecture. The choice was mainly motivated by the need for separation among components such that the UI components will not be mixed with back end and AI-related functionalities. This allows much easier debugging and modification.

- **Presentation Layer (Frontend):** The frontend was implemented in React.js. That's essentially everything a user is interacting with. I have done my best to stick to HCI standards when designing the front end to ensure its usability and good presentation on phone and laptop screens. Candidates will use it to upload their resumes and conduct the interviews.
- **Application Layer (Backend):** As for the core of my app, I decided to use Node.js and Express framework to implement it. This layer contains everything related to logic such as login verification and management of API requests. In other words, it works as a bridge between frontend and either AI or database layers.
- **AI Processing Layer:** This is where the bulk of the work is done. For implementing this layer, I chose Python because it simply fits much better into the field of AI. Its responsibilities are:
    1. CV recognition and parsing.
    2. Comparison of candidate skills to position requirements.
    3. Generation of interview questions and evaluation of answers.
    4. Providing feedback based on pre-trained models.
- **Data Layer (Database):** I have chosen MongoDB as the database here. This was because the data here is varied in terms of resume and interviews among other things, hence a NoSQL approach would be best since it can easily retrieve and store data without adhering to any strict format.

4.2 Design Constraints

Table 41: Design Constraints

Constraint	Description
Reliability	The system must maintain stable communication between frontend, backend, database, and AI services so recruiters can complete hiring workflows without interruption.
Criticality of the Application	SkillSelect AI performs functions such as scheduling interviews and evaluation of candidates, and therefore, critical operations that include invitation sending, accessing interviews, and obtaining results have to be timely and reliable.
Safety and Security Considerations	The platform handles sensitive recruiter and candidate data, including CVs and interview recordings, so secure authentication, protected APIs, and safe data handling are essential.
Internet Connection	SkillSelect AI is an online platform and depends on API communication, storage, and AI processing; therefore, stable internet connectivity is required for smooth operation.
Hardware Limitations	Video interview features require access to a camera, microphone, and sufficient processing capability. Low-end devices may experience lag, reduced quality, or slower analysis performance.
Scalability Constraints	The application needs to deal with several recruiters and candidate interview sessions at once; however, when traffic and AI processing increase, so does the load on the server.
Integration Constraints	In SkillSelect AI, there are React components used for the front end, Node.js/Express components used for back end development, and Python/API-based AI components.

4.3 System Architecture

In choosing the architecture for SkillSelect AI, I opted for one that is scalable and easily manageable by integrating a layering methodology with the Model View Controller design paradigm. This essentially involves segregating the user interface, the business logic, and data manipulation functionalities from one another to prevent interference.

I then proceeded to divide my architecture into four main components:

- **The Frontend (Presentation Layer):** The presentation layer was implemented using React.js, creating a single-page application (SPA) which provides users with an extremely smooth and interactive experience with no page reloads involved. This is where all of the interaction occurs, such as users uploading their CVs, attending interviews, and reviewing their scorecards.
- **The Backend (Application Layer):** This is the controller layer of the system implemented using Node.js and Express.js. It acts as the middleware of the project, handling user authentication and controlling the flow of data from the frontend and AI modules to the database. Security and speed were both significant concerns when implementing this layer.
- **The AI Processing Layer:** This is the layer responsible for carrying out the "intelligence" tasks within the project. As such, it was implemented using Python which is superior to JavaScript at performing data science tasks. This module performs the following tasks:
  i) **Parsing CVs:** Extraction of relevant information from CVs.
  ii) **Skill Matching:** Evaluating whether a candidate matches the requirements of the job description.
  iii) **Interview Logic:** Creating the interview questions and analyzing the answers using NLP.
  iv) **Feedback:** Creating the feedback reports generated after the interviews.

The Database (Data Layer): I decided to use MongoDB since it provides flexibility in terms of schemas that are required to store our data types, which consist of user details and uploaded files like PDFs, interview transcripts, and feedback reports from AI models. The NoSQL feature helped me not to worry about having a fixed schema for different data types.

4.4 Logical Design

4.4.1 Class Diagram:

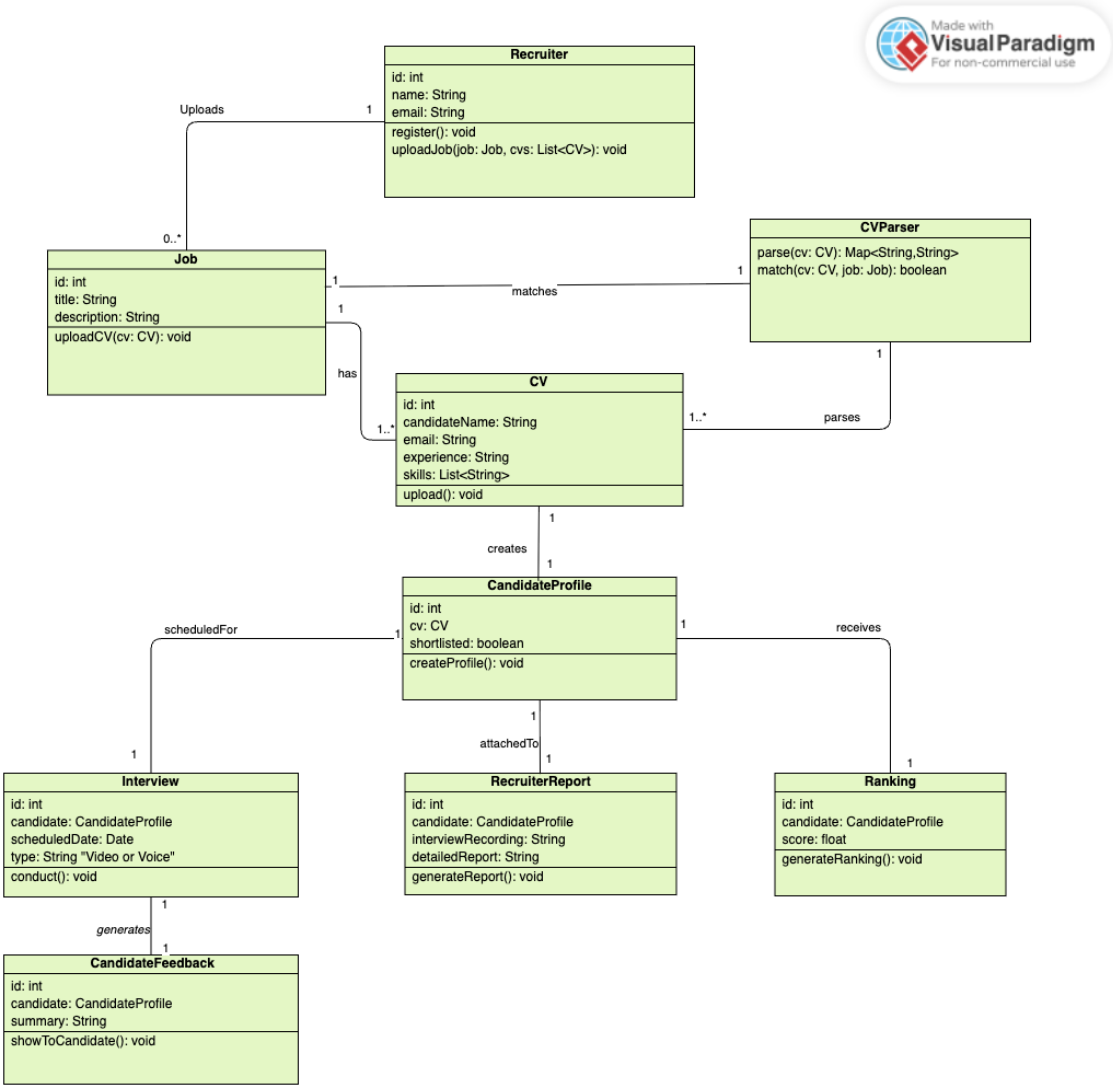


Figure 2: Class Diagram

Description:

1. Recruiter Class

This is the profile of the recruiting organization. This class contains information such as their identification number, name, and email address. The idea here is to design a class which will enable the recruiter to post various job opportunities and recruit people through these jobs.

2. Job Class

Each time the recruiter recruits employees, he/she uses this class. It contains information such as the designation and responsibilities attached to the position. It also acts as an anchor for the resumes which are posted for this position.

3. CV Class

It acts as an anchor for the resumes which contain raw data about the candidates. This raw data includes personal information of the individual, work experience, and skills.

4. CVParser Class

This class contains the AI algorithm which converts unstructured information contained in a CV into structured information. Furthermore, it compares the skills of candidates to the required skills as per the job.

5. CandidateProfile Class

After parsing a CV, the next step is creating a CandidateProfile which is a more “refined” format of information which includes details such as whether the candidate has been shortlisted or not.

6. Interview Class

This is an operational part of the algorithm. It includes information about the interview ID, type of interview (voice/video), date and time. Moreover, it is related to the candidate profile so that information about the candidate can be obtained.

7. CandidateFeedback Class

Since I wanted to ensure that the candidate is not left in suspense regarding his performance, the class contains details of his performance, indicating his strengths and weakness areas.

8. RecruiterReport Class

This class will be used by the recruitment team. The class contains a compilation of interview recordings along with the assessment results given by the AI.

9. Ranking Class

Lastly, the class is designed to rank the candidates on the basis of scores generated for a certain position. It greatly benefits recruiters as it sorts out the most suitable candidates at the top.

4.5 Dynamic View

4.5.1 Sequence Diagram:

4.5.1.1 Candidate Module

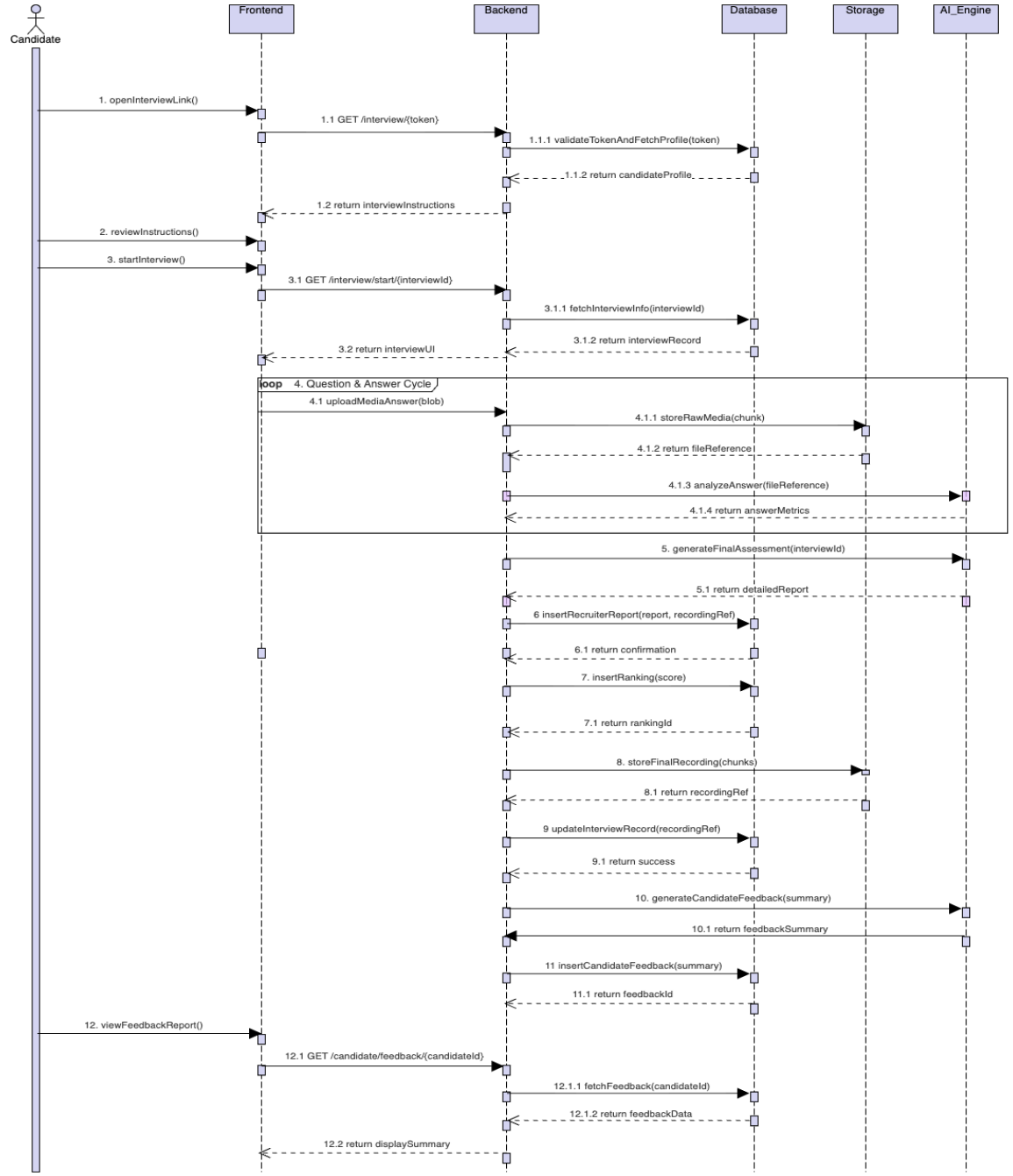


Figure 3: Sequence Diagram of Candidate Module

Description: The above sequence diagram shows the AI-interview workflow. First, the candidate opens the interview URL. This action prompts the frontend to ask for information about the interview from the backend server. The backend will validate the authentication token against the database and fetch the candidate's profile and return the interview instructions to the frontend.

The candidate studies the interview instructions and begins the interview. This prompts the frontend to ask for interview data from the backend. The backend then queries the database and returns the interview interface to the frontend.

A process of looping through Q&A where media responses are provided by the candidates takes place during the interview stage. Such media responses are saved to the storage and the references of media responses are passed to the AI engine, which analyzes these responses and generates the scores of the candidate.

At last, the backend produces a report using the results produced by the AI engine, which are saved in the database together with a reference to the recording of an interview session. The rank of the candidate is determined according to his/her score and the final version of recording of the interview session is saved in storage.

Lastly, feedback is generated for the candidate and saved in storage. The candidate may obtain a copy of the feedback report using references to such document saved in the database.

4.5.1.2 Recruiter Module

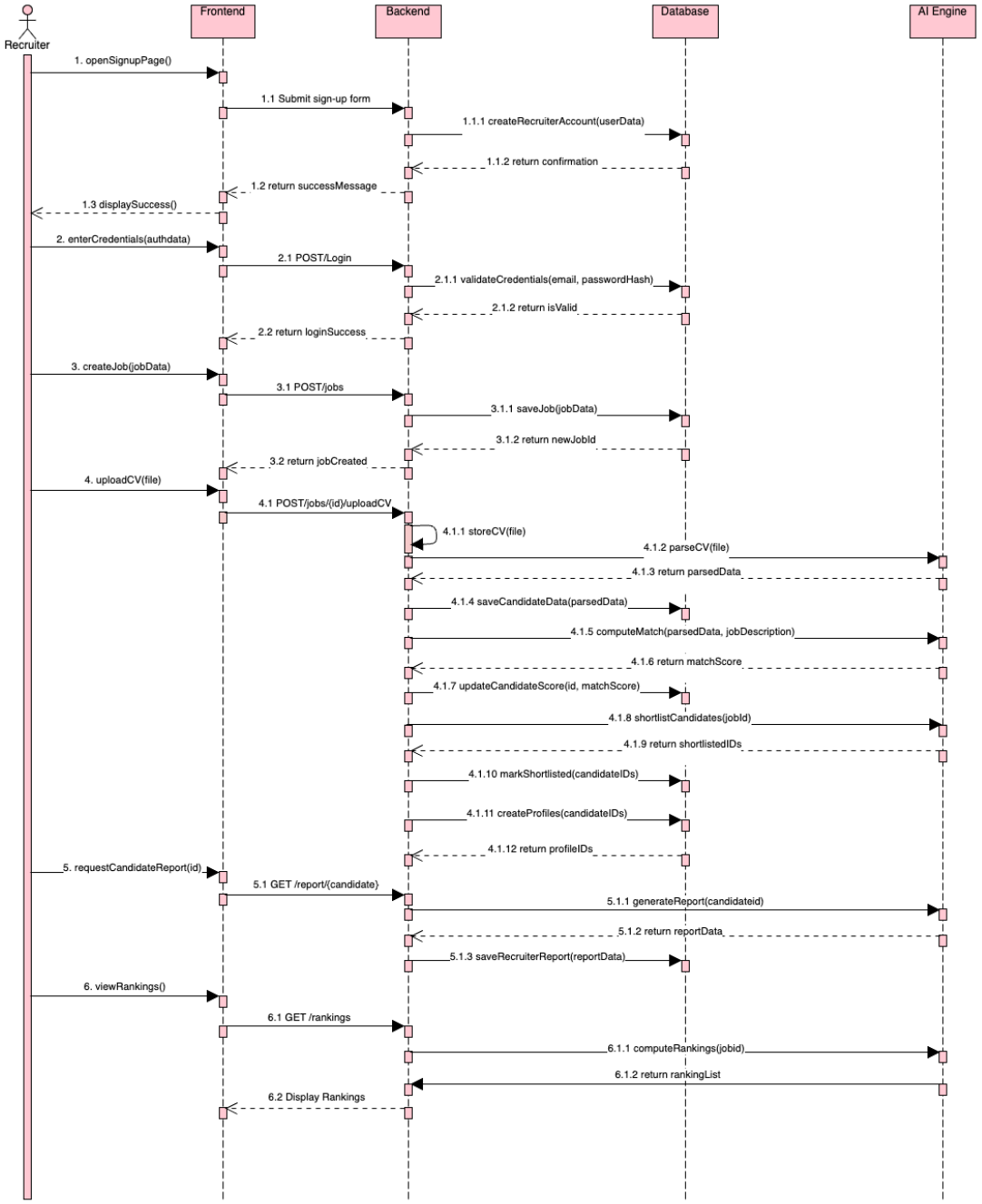


Figure 4: Sequence Diagram of Recruiter Module

Description: This is a sequence diagram illustrating the flow process of the communication between the recruiter and the AI-powered recruitment candidate assessment system. In the first step, the recruiter navigates to the sign up page, enters his/her registration information, and prompts the backend to save these details into the database. After successful creation of an

account, the recruiter signs into the application system and undergoes authentication process before he/she gains access.

Upon authentication, the recruiter creates a job opening by submitting relevant job details into the database. Next, the recruiter uploads CVs of the recruited candidates that will be evaluated by the backend system, which submits the uploaded CVs to the AI system for parsing. After parsing, the collected information is saved, and the similarity score for each candidate and the job posting is generated. In the next step, the candidates are selected and their profiles generated and saved into the database.

The recruiter requests for a report for the selected candidates, which is generated by the AI system and saved in the database. In the final step, the recruiter gets the ranking of the recruited candidates.

4.5.1.3 Admin Module

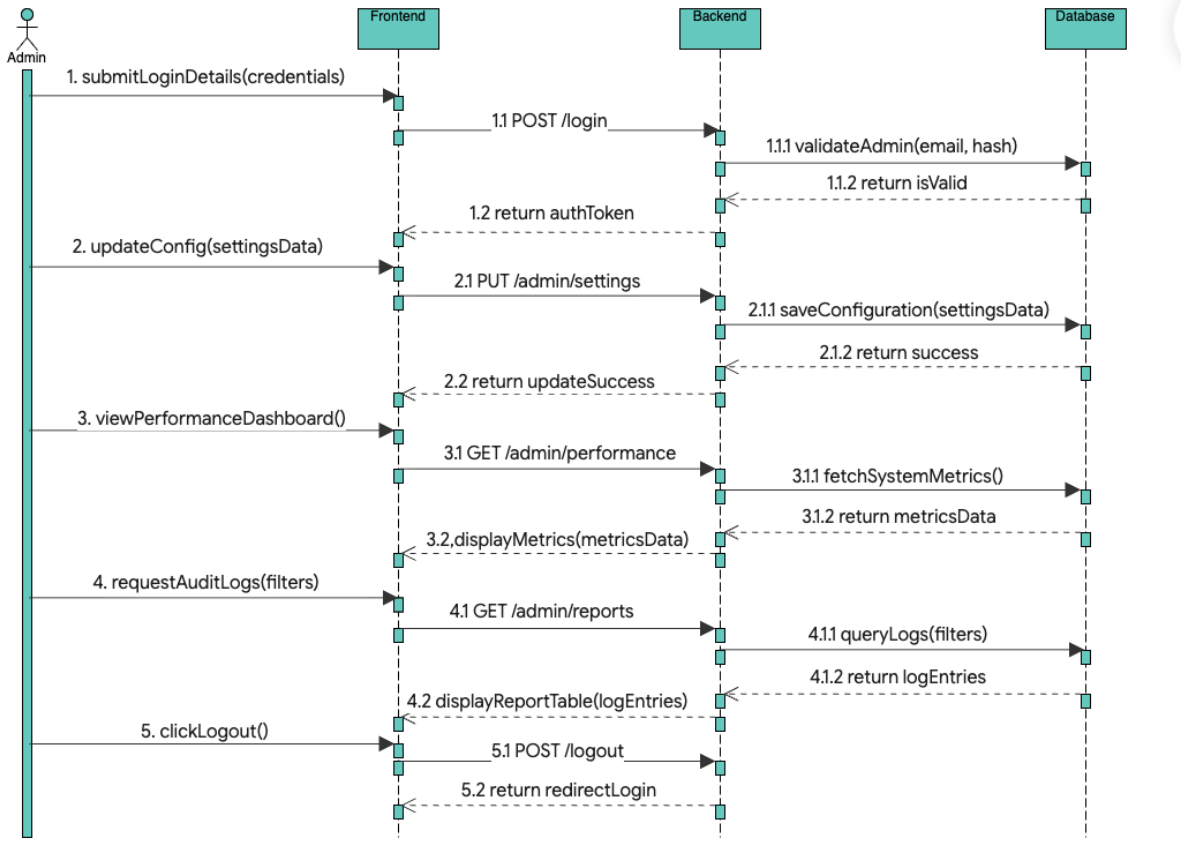


Figure 5: Sequence Diagram of Admin Module

Description: This diagram showcases the flow of operations involved in using the system for configuration management and performance monitoring. The process starts when the administrator sends login information to the backend server via the frontend interface. The backend server authenticates the user by comparing the information against the information stored in the database, and once this is accomplished, the user receives an authorization token.

Having received the authorization token, the administrator now manages the system configurations by providing necessary configuration details to the backend server. Afterward, a confirmation message is sent back to the administrator, confirming successful configuration.

The user is now able to view the system performance details from the performance dashboard. The process starts when the frontend requests information about the system performance from the backend. The backend server then gathers performance information from the database.

Moreover, the admin can also generate audit logs using filters. The backend will fetch the logs from the database according to the specified criteria and then send them to the frontend in the form of a report.

Lastly, the admin signs out of the system and sends a sign-out request to the backend.

4.5.1.3 Candidate CV Upload

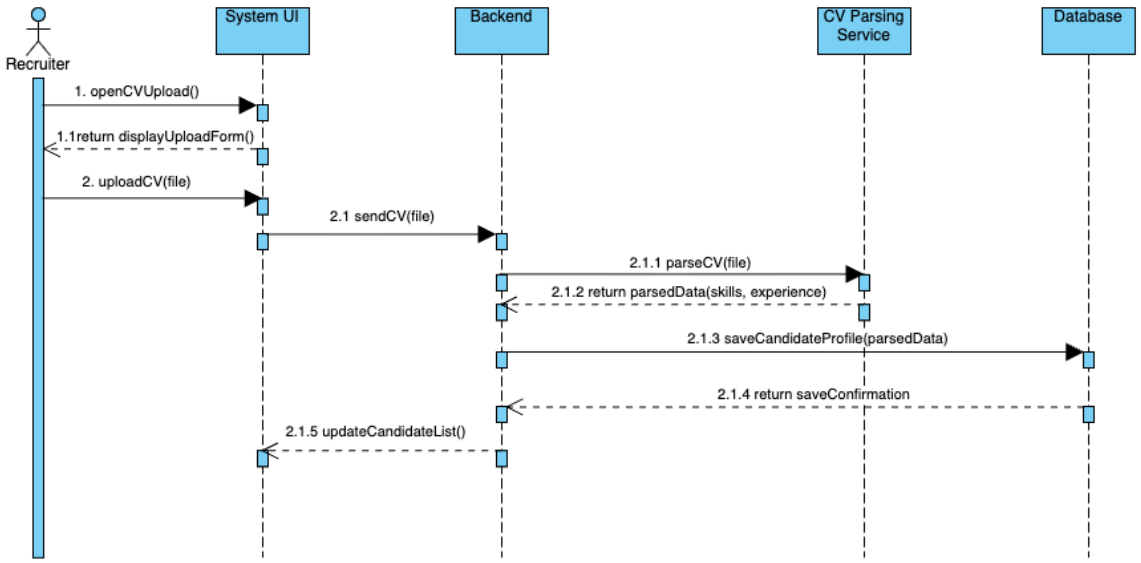


Figure 6: Sequence Diagram of Candidate CV Upload

Description: This diagram explains the sequence of operations of a recruiter using the system to upload the candidate's CV and parsing of it within the system. Firstly, the recruiter uploads the CV file using the upload operation via the System UI. Upon receiving the request for uploading, the system shows the uploading page to the recruiter, who can select the required CV file and send it to the system.

Upon receiving the uploaded CV, the System UI transfers it to the backend server, which forwards it to the CV Parsing service. This service performs analysis of the CV file to extract the necessary information such as skills and experience of the applicant.

Subsequently, after extracting information from the CV file, the backend stores all of it in the database.

This is followed by a return response from the backend to the System UI. This will prompt the user interface to update its candidate list to include the newly-added candidate.

The above steps illustrate the efficiency of a streamlined process that the system adopts when dealing with CVs. It involves uploading, parsing, and storing data while simultaneously updating the user interface in real-time.

4.5.1.4 Reviewing Candidate Profile and Evaluation

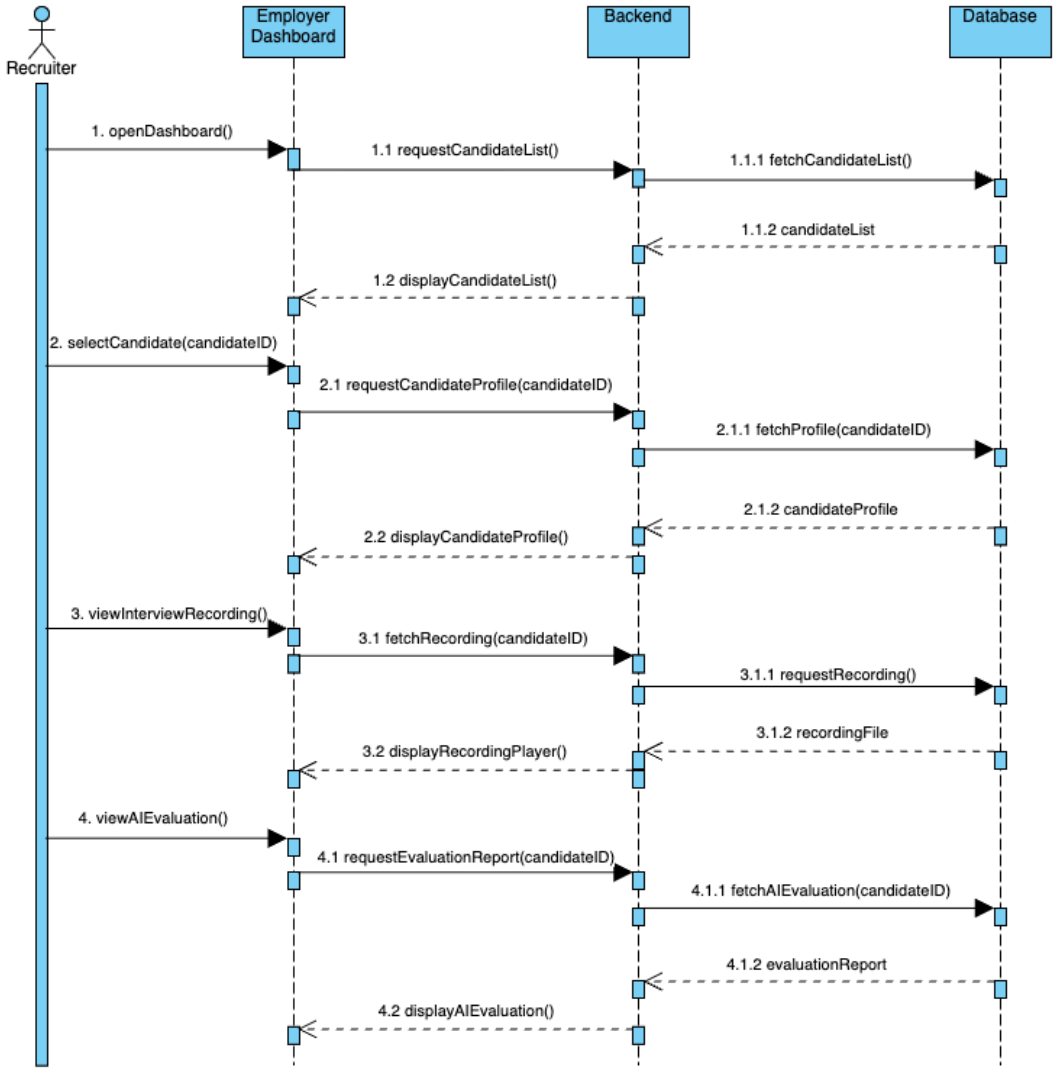


Figure 7: Sequence Diagram of Reviewing Candidate Profile and Evaluation

Description: This sequence diagram shows how the recruiter, employer dashboard, backend, and database interact in the context of evaluating the candidate's details and scores. Firstly, the recruitment cycle starts with the opening of the employer's dashboard. As a result of that, an initial request for candidates list is sent by the Employer Dashboard to the backend.

Upon fetching that list from the database and receiving it, the Employer Dashboard will display the list on the interface of the dashboard for the recruiter to select. After selecting the required candidate, an initial request for profile of the selected candidate is sent by the employer dashboard to the backend.

In response, the backend fetches the profile data of the selected candidate from the database and sends the retrieved data to the dashboard for displaying it. Thereafter, the user gets access

to viewing the video file of the interview. For this purpose, the Employer Dashboard makes a request for the file from the backend and the latter retrieves it from the database and forwards the file back.

In addition, the recruiter can gain access to the assessment done by the AI about the candidate. An invitation is made to the backend so that the evaluation report which was stored in the database is retrieved. The data is then provided by the backend to the dashboard.

In summary, the whole process demonstrates how easily recruiters can access candidate information, watch videos of interviews done, and assess the evaluation produced by the AI.

4.5.2 Activity Diagram

4.5.2.1 Automated Candidate Profile Generation Workflow

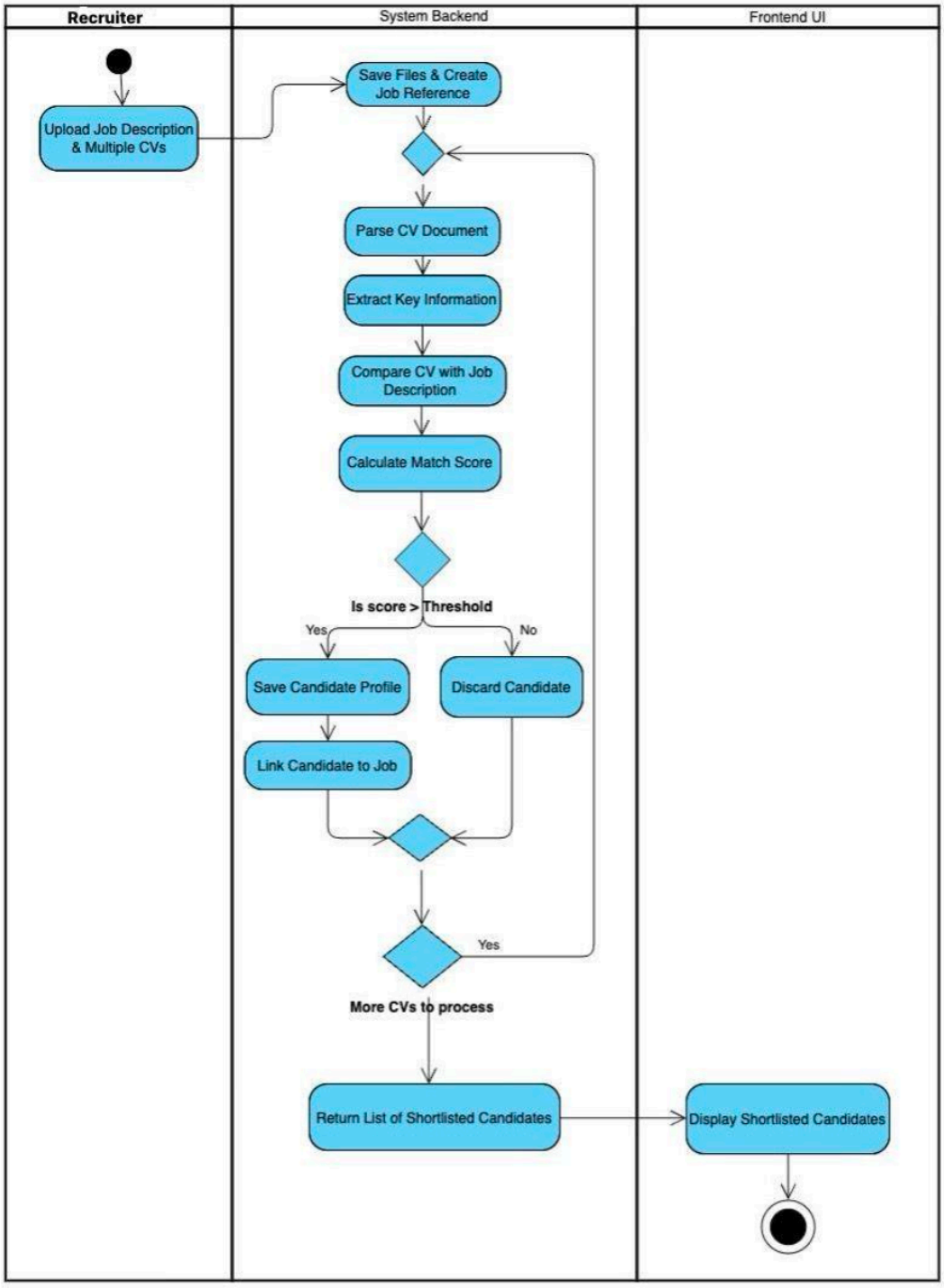


Figure 8: Activity Diagram of Automated Candidate Profile Generation Workflow

Description: This diagram depicts the automation of shortlisting for the recruitment of suitable candidates. The process begins when the Recruiter offers job descriptions and various CVs. Subsequently, the System Backend takes control of the entire process, which includes storage of data and creation of job references. Following the creation, there is looping in the process where the System Backend carries out several operations on each CV.

At first, the System Backend analyses each CV using some algorithm to check whether the data contained in the CV corresponds to the job description. If yes, then the CV qualifies; the candidate gets accepted, and the system links the job description to him/her. Otherwise, the candidate gets rejected. When all CVs are considered, the process looping ends. Finally, after completing the looping operation, the System Backend sends a list of selected candidates to the Frontend UI.

4.5.2.2 Recruiter Registration

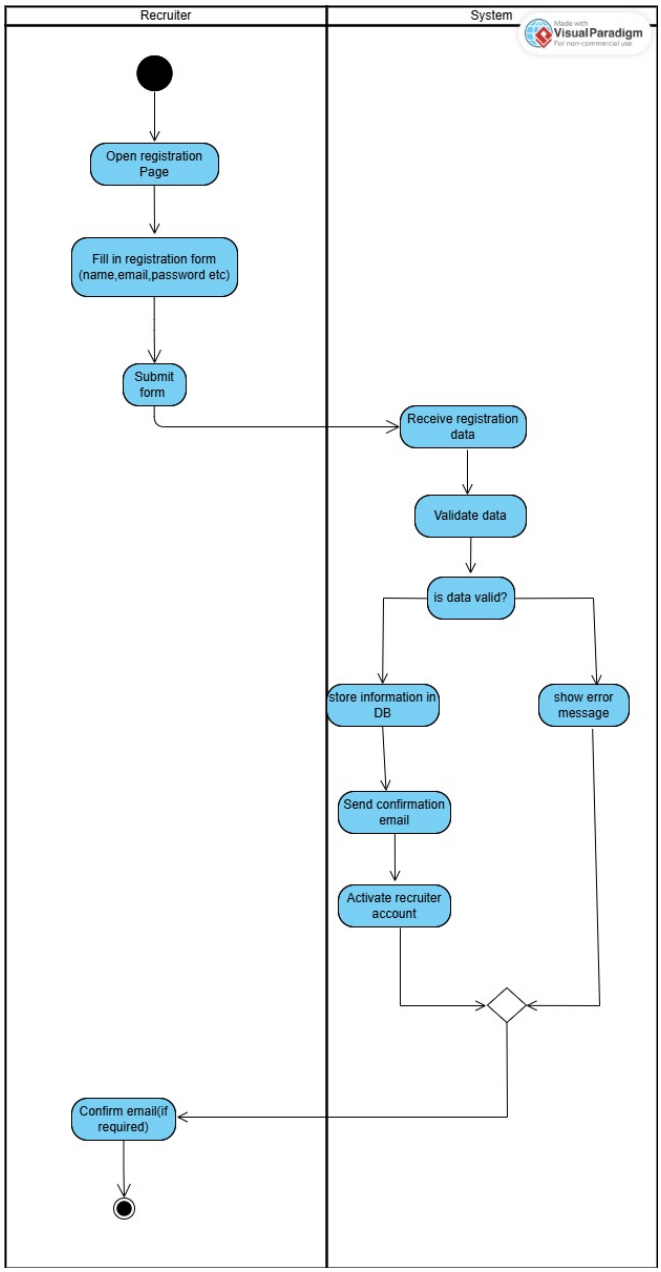


Figure 9: Activity Diagram of Recruiter Registration

Description: The registration process for the Recruiter role is described using this activity diagram. The process starts with the user accessing the registration page and filling out the necessary fields like the name, email address, and password. Once submitted, the System checks and validates the input data.

In case the input data is not valid, the system returns an error message. In case the input data passes the validation process, the recruiter's data is saved in the database, an email notification

is sent to the user, and the account gets activated. The final step would be to confirm the email, if needed.

4.5.2.3 Asynchronous Interview Process

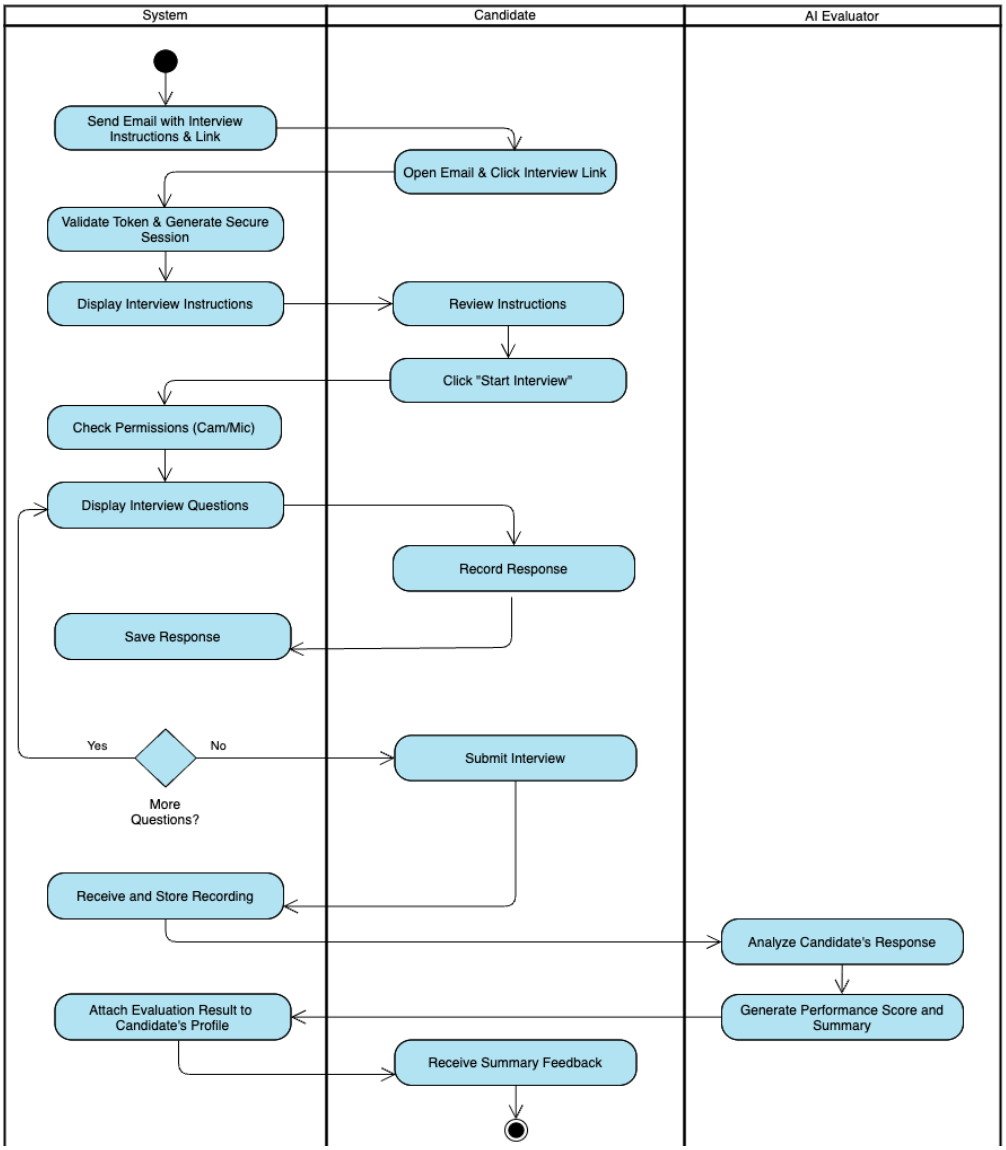


Figure 10: Activity Diagram of Asynchronous Interview Process

Description: The activity diagram above represents the process of an automated interview that consists of three actors - System, Candidate and AI Evaluator. The System at the beginning sends an e-mail to the Candidate containing instructions on the interview and a link. Later on, once the Candidate uses the provided link from the System, the latter verifies the token and establishes a secure connection. As soon as the Candidate opens the interview instructions and clicks the "Start Interview" button, the system does a validation check for the Camera and Microphone and presents the questions.

Then, the actual interview takes place in a loop, where the system presents a question to the Candidate, who provides his/her answer and then it is stored by the system. When there are no more questions, the interview is over and the answer from the Candidate including the recording is sent to the system. In turn, the AI Evaluator assesses the candidate's answers and produces the result and grade. Finally, the outcome is added to the candidate's profile.

4.5.2.4 Score candidate performance

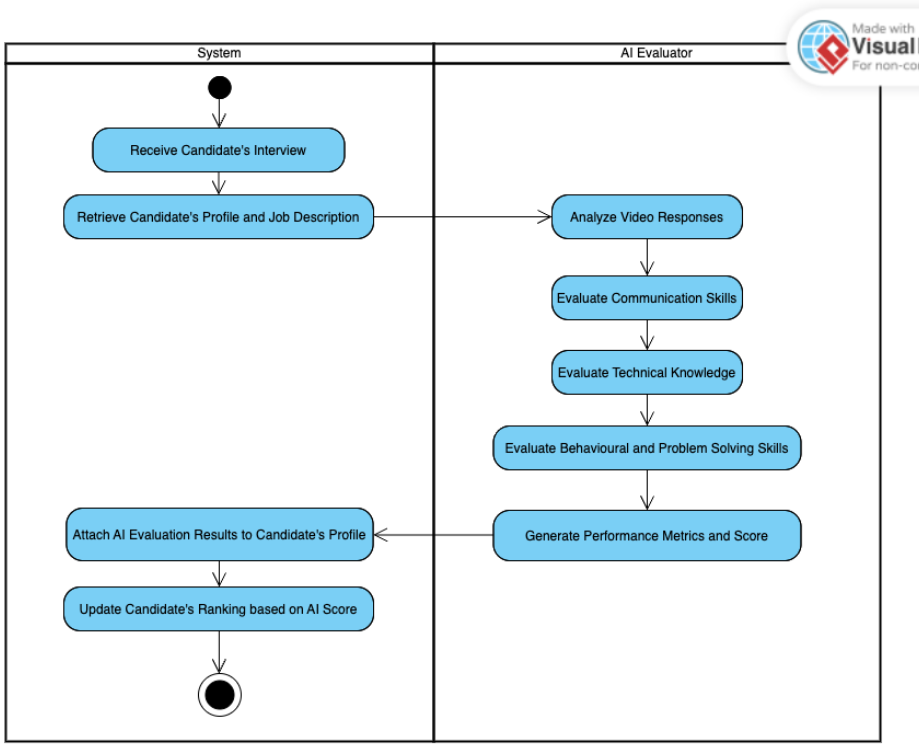


Figure 11: Activity Diagram of Score candidate performance

Description: The activity diagram depicts the full automated process of scoring the candidates' performances. First, there is the process of having the candidate conduct the video interview recorded by the system. Secondly, there is the process of the system collecting the candidate's profile and the job description. The third step entails transferring control to the AI Evaluator, where AI performs a number of levels of analysis on the uploaded videos. Some of the analyses include evaluating the candidate's communication, technical skills, and problem-solving ability.

Finally, the AI comes up with a scorecard on the performance of the candidate. Subsequently, the system collects the scores generated by AI and updates the rankings of the candidates in the recruiting process.

4.6 Component Diagram:

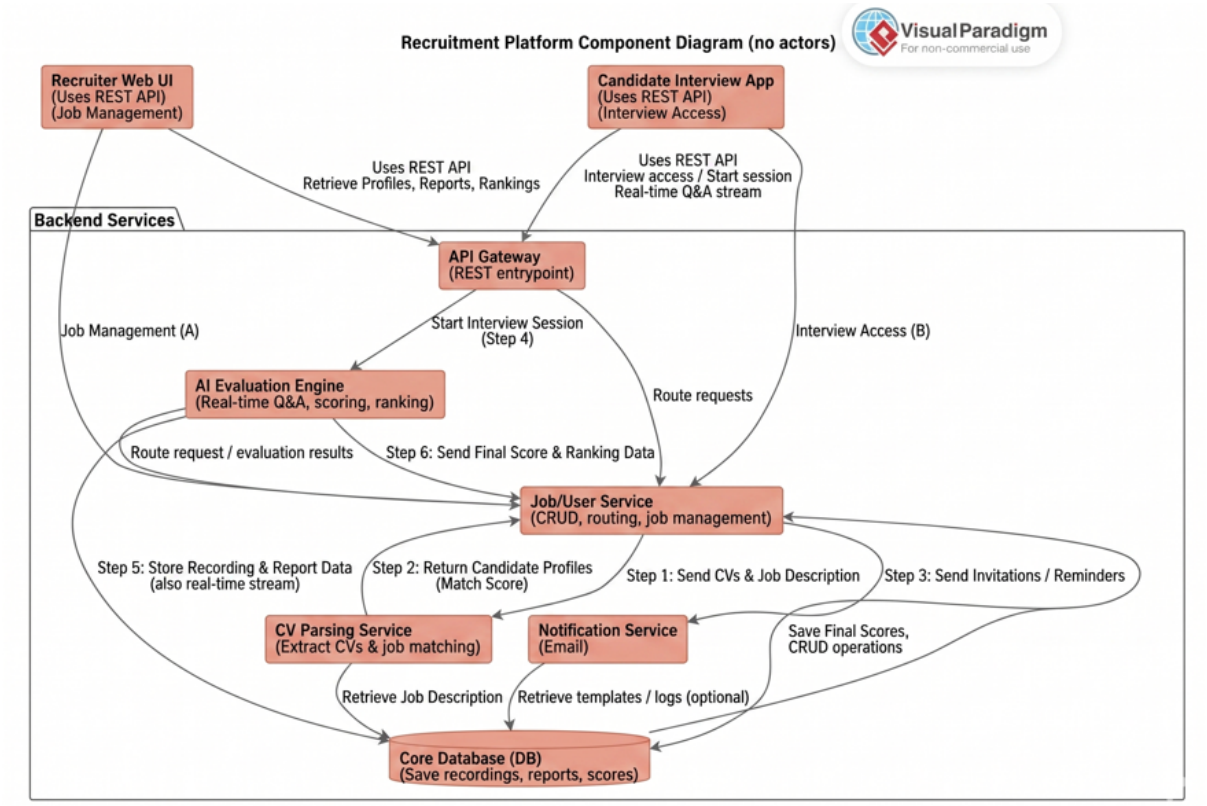


Figure 12: Component Diagram

Description: SkillSelectAI component diagram indicates a modular backend architecture aimed at automating the recruitment process. The recruiters and candidates can use the system via the Recruiter Web UI and Candidate Interview App respectively, where the latter two components interact using the API Gateway.

These components include such elements as the service for managing jobs and users, the CV parsing module, the candidate matching module, and the automated notifications service. Besides, the AI engine evaluates the performance of the candidates during interviews, whereas the centralized database stores information on jobs, recorded interviews, candidate ratings, and more.

4.7 Data Models:

4.7.1 ER Diagram:

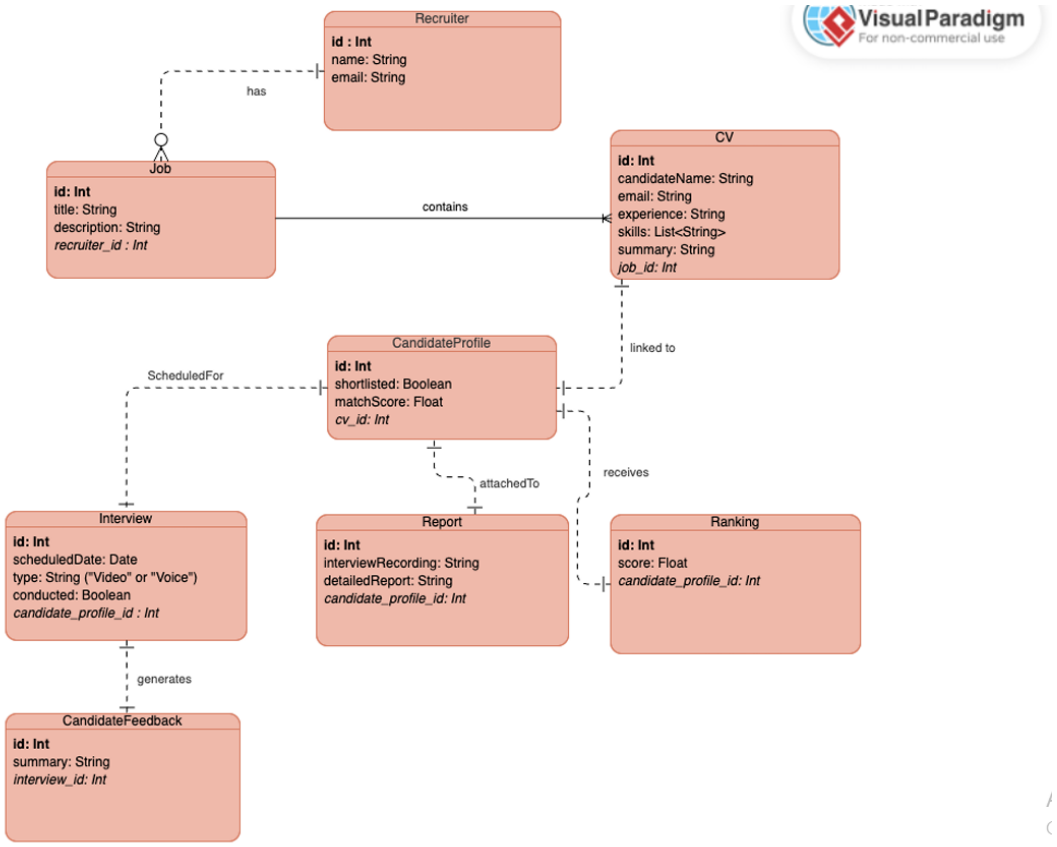


Figure 13: ER Diagram

Description: The progression from a resume to evaluation, leading to the end product of ranking of candidates on the basis of their eligibility for a specific job, is presented in the ERD for SkillSelect AI. In this process, Recruiter initiates the process of application through the Job post. Each job post will contain multiple CV entities that represent the resumes of the candidates who apply for the job.

In order to maintain clarity in this complex ERD, I decided to link the CV with the Candidate Profile, which forms the core of the entire database. This entity contains information about the candidate such as the status, which can be either inactive or live. The automated match score and shortlisting decisions are also stored in this entity. If the candidate profile is live, it will result in scheduling of a voice/video interview with the candidate. As a consequence of the interview, two entities are generated: Candidate Feedback and Recruiter Report. The first one

represents the feedback about the candidate by the interviewer while the second one contains all the information about the interview, including the recording and the ranking score.

4.8 User Interface Design

4.8.1 Screenshot of Landing Page

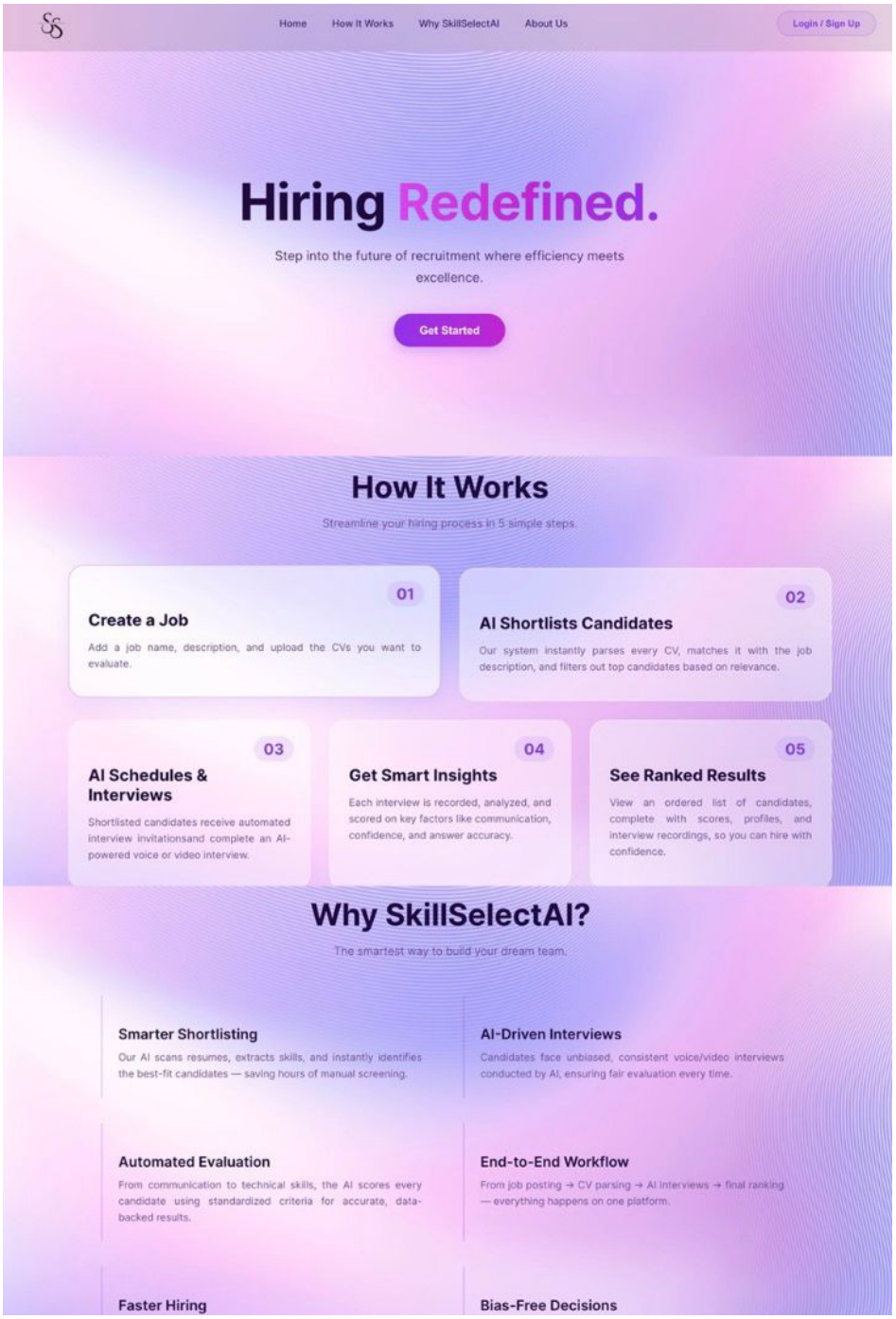


Figure 14: Landing Page

4.8.2 Screenshot of Signup Page

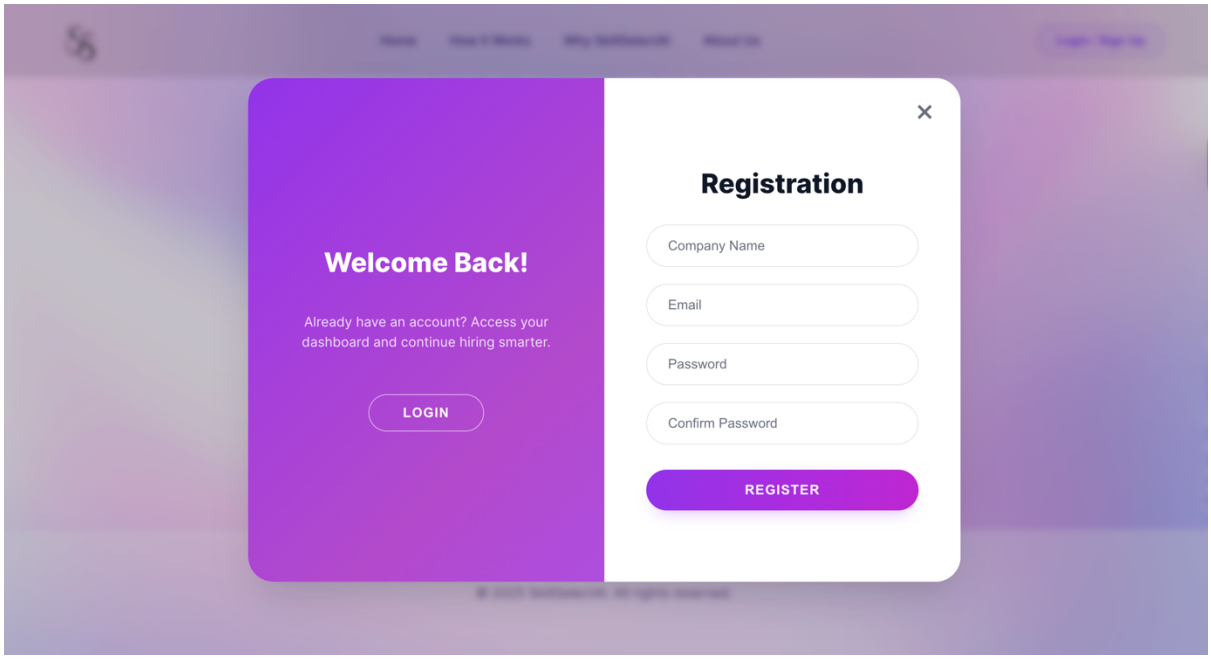


Figure 15: Signup Page

4.8.3 Screenshot of Login Page

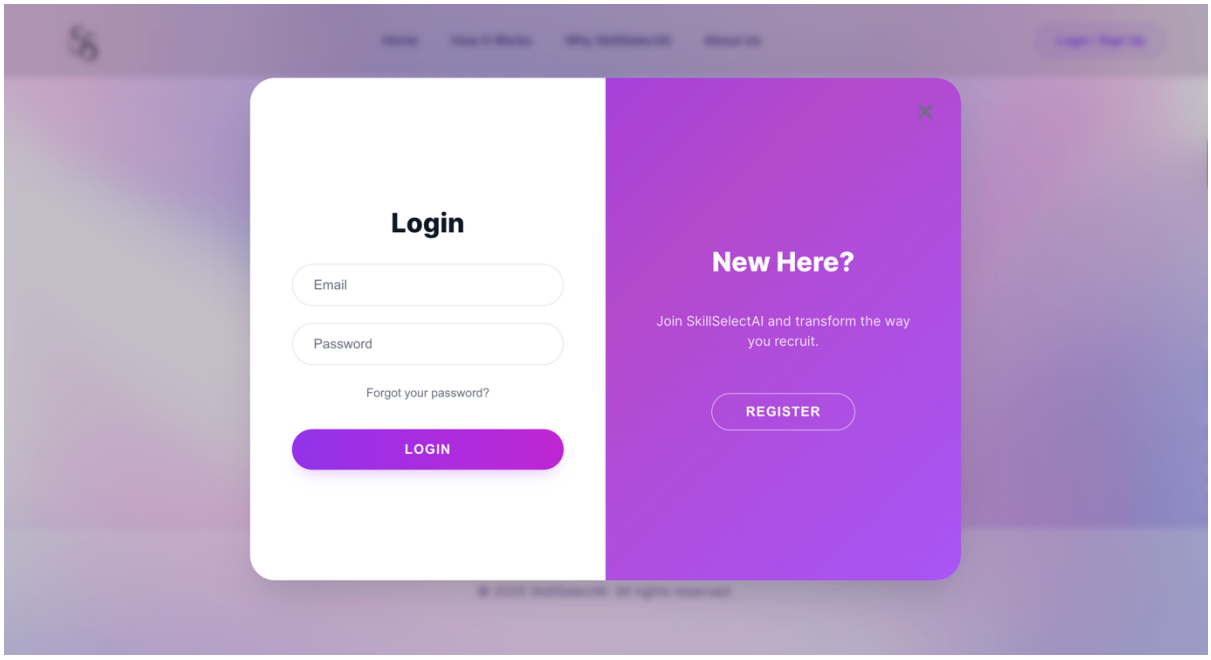


Figure 16: login Page

4.8.4 Screenshot of Forgot Password Page

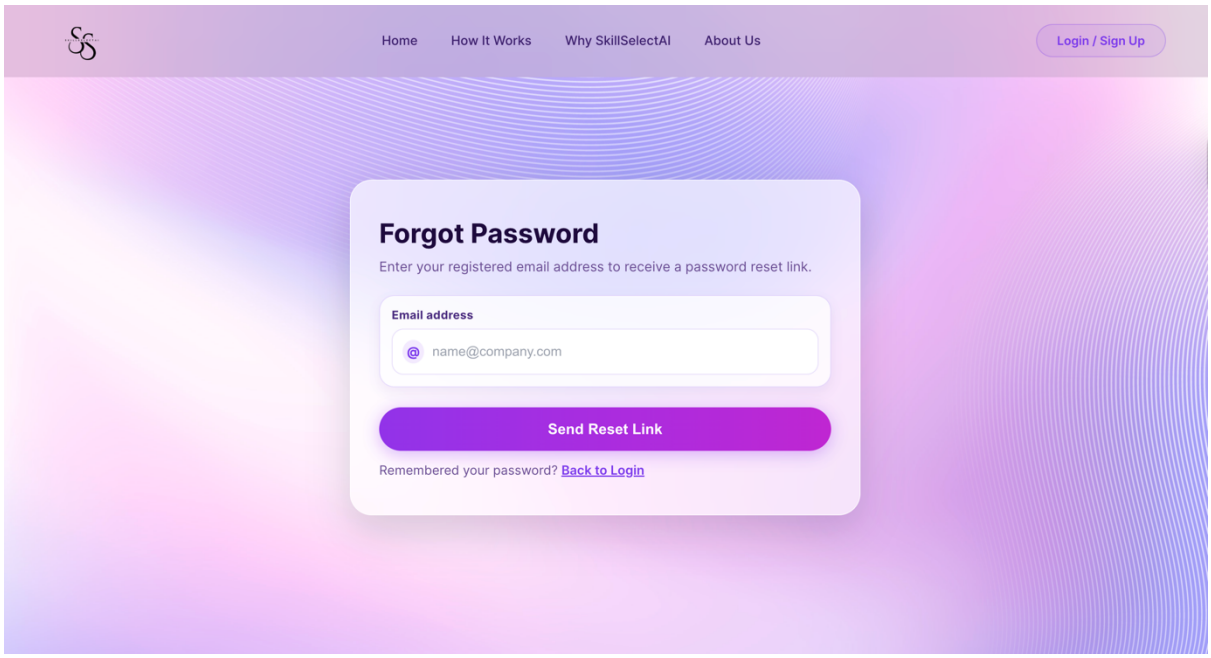


Figure 17: Forgot Password

4.8.5 Screenshot of Dashboard

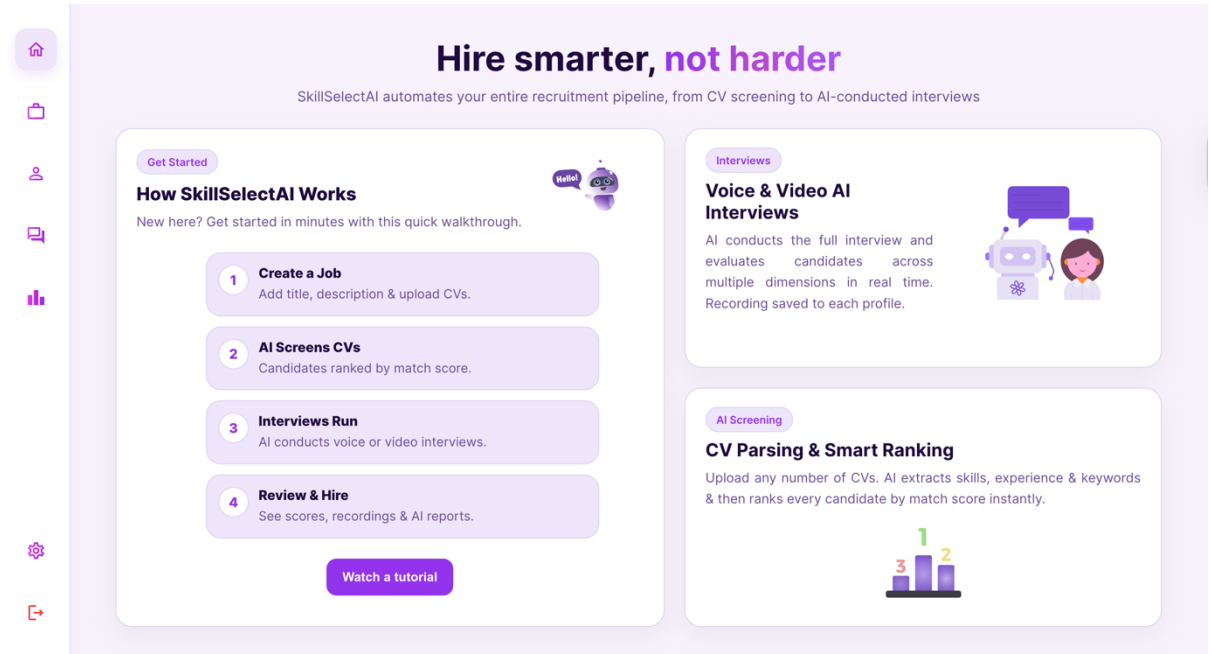


Figure 18: Dashboard

4.8.6 Screenshot of Job Management Page

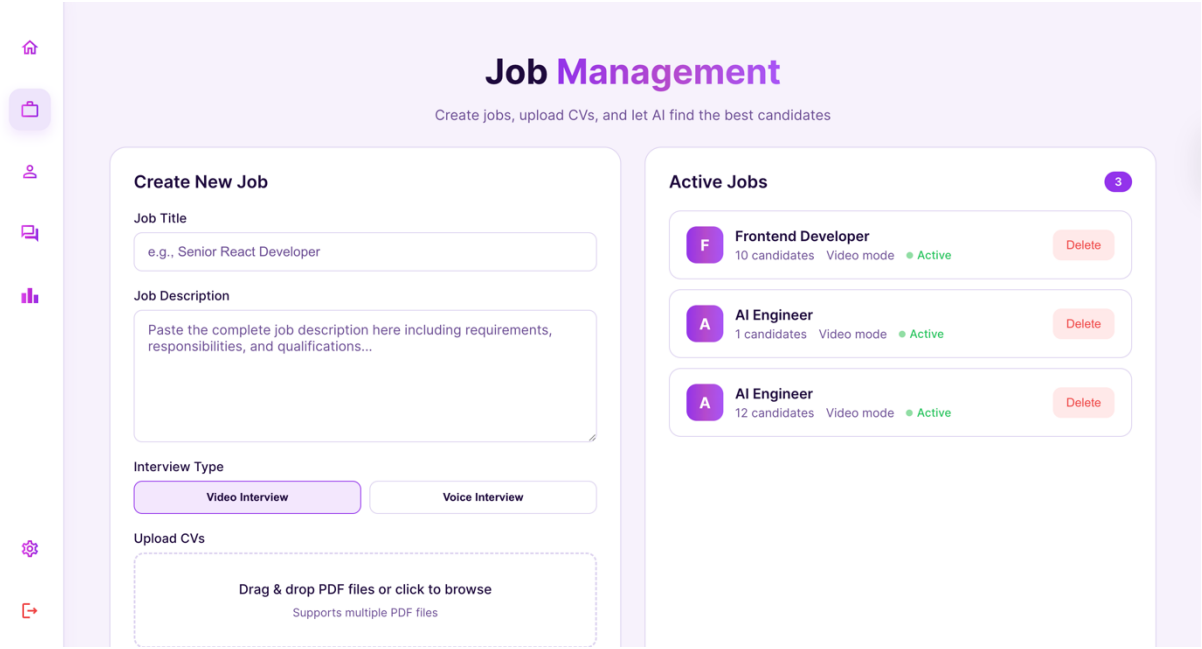


Figure 19: Job Management Page

4.8.7 Screenshot of Candidate Pool Page

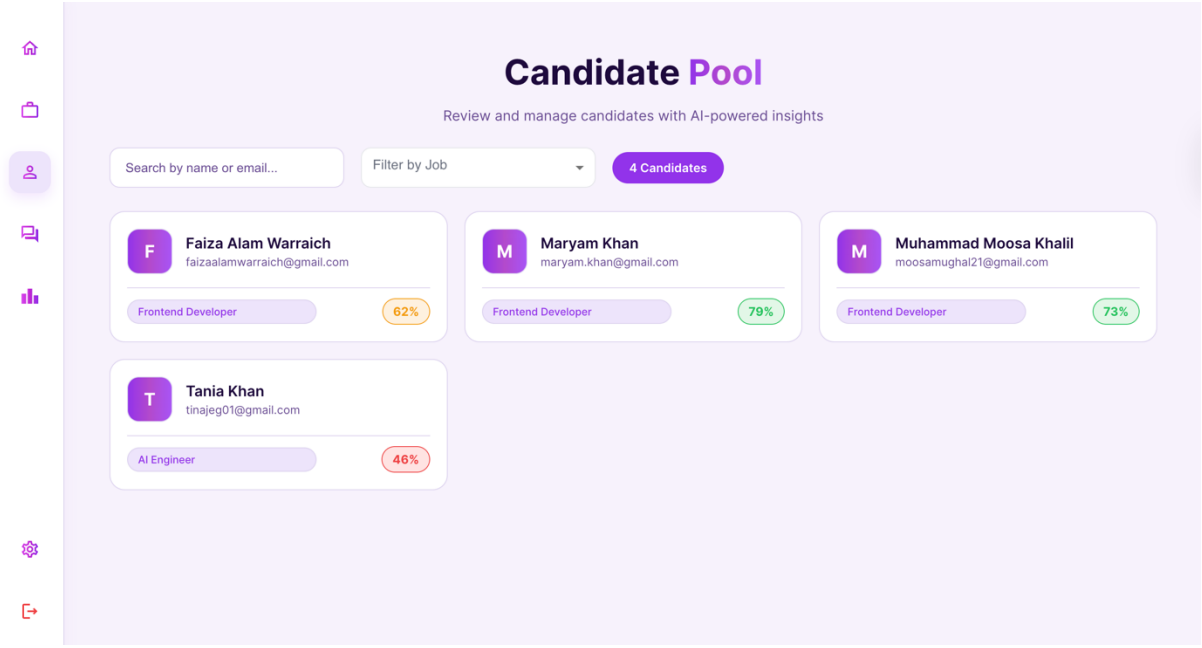


Figure 20: Candidate Pool Page

4.8.8 Screenshot of Candidate Pool Page

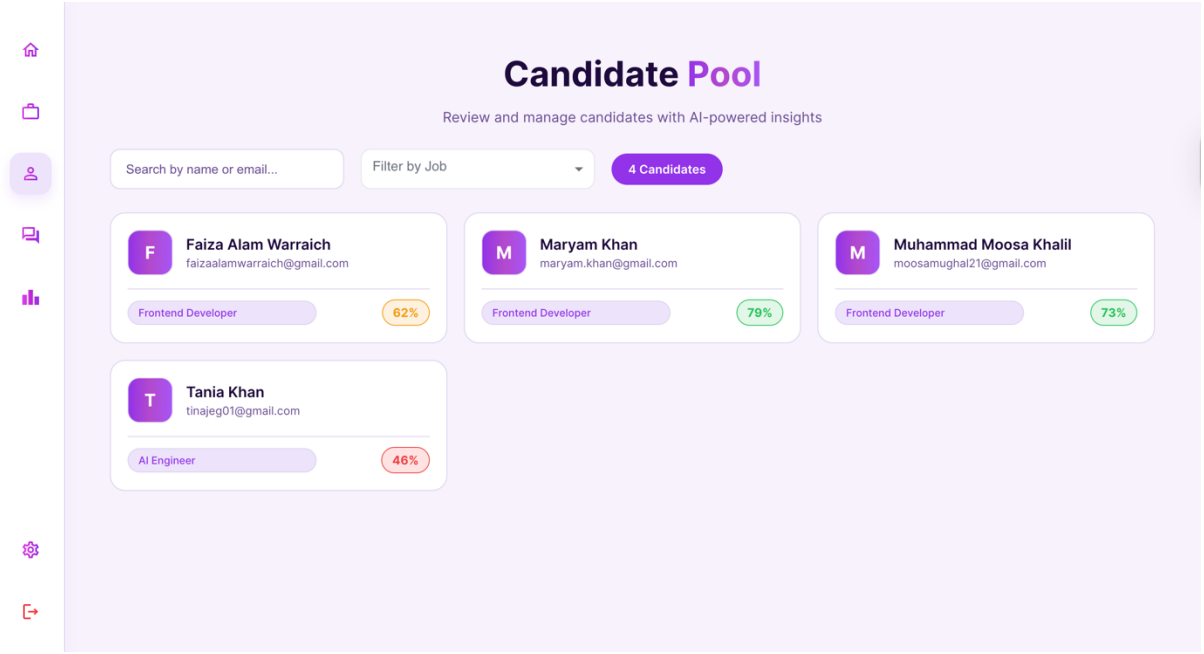


Figure 21: Candidate Pool Page

4.8.9 Screenshot of Candidate Profile

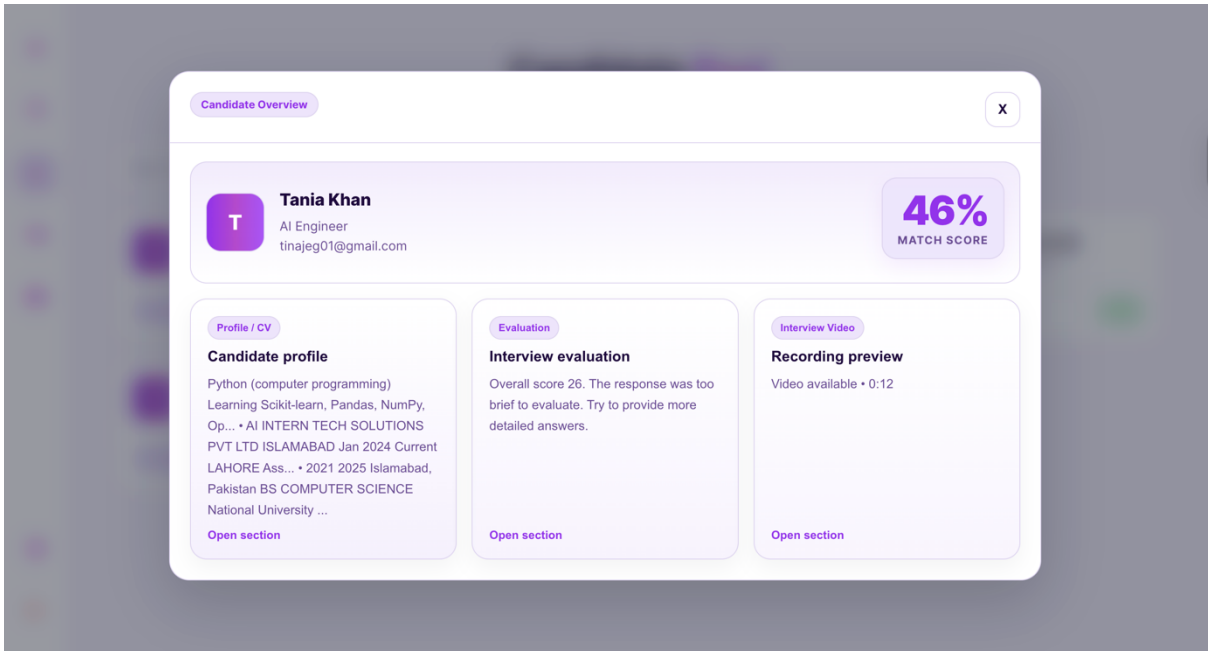


Figure 22: Candidate Profile

4.8.10 Screenshot of Interview Management Page

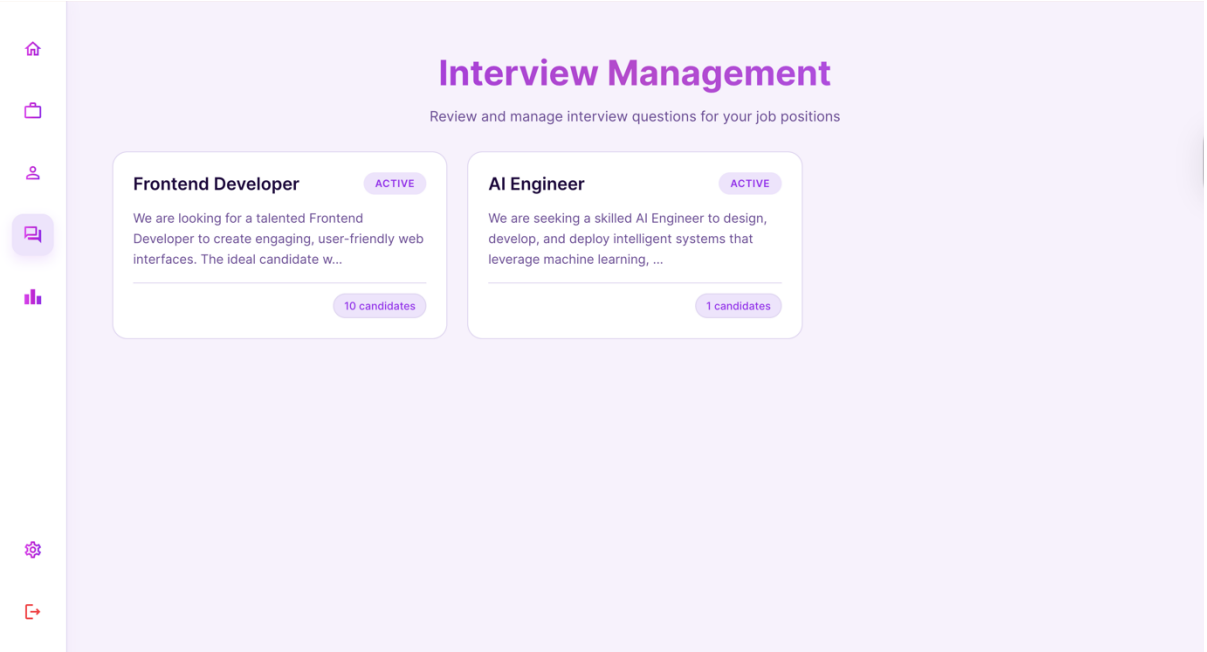


Figure 23: Interview Management Page

4.8.11 Screenshot of Questions Page

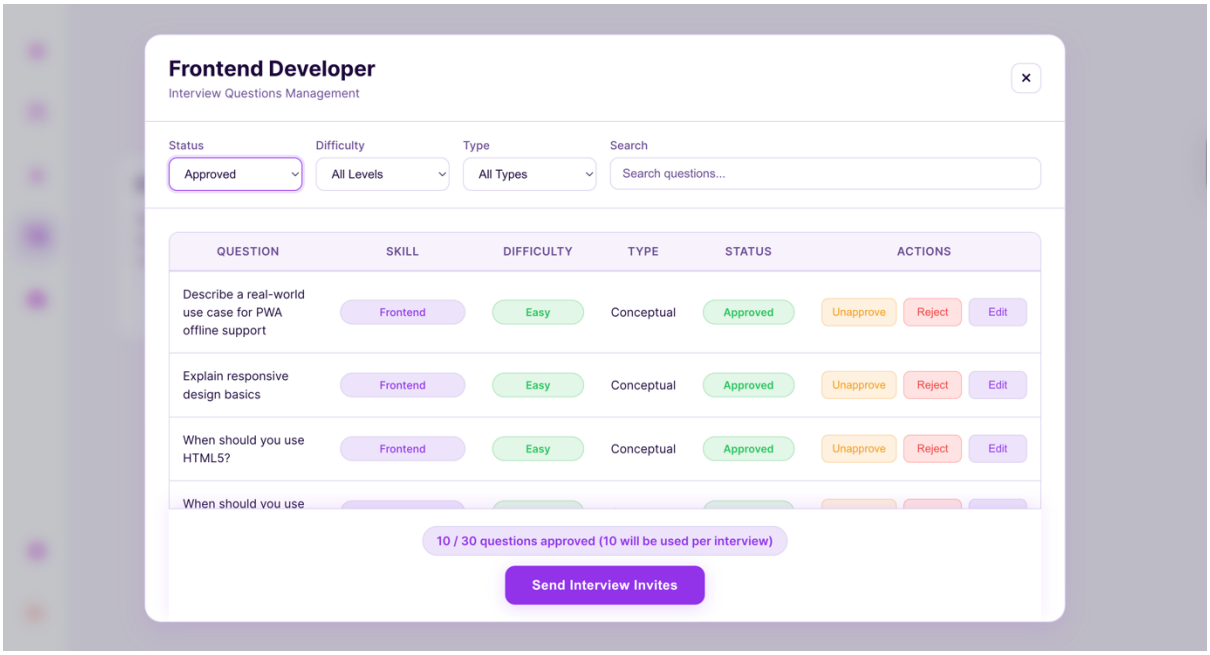


Figure 24: Question Page

4.8.12 Screenshot of Interview Landing Page

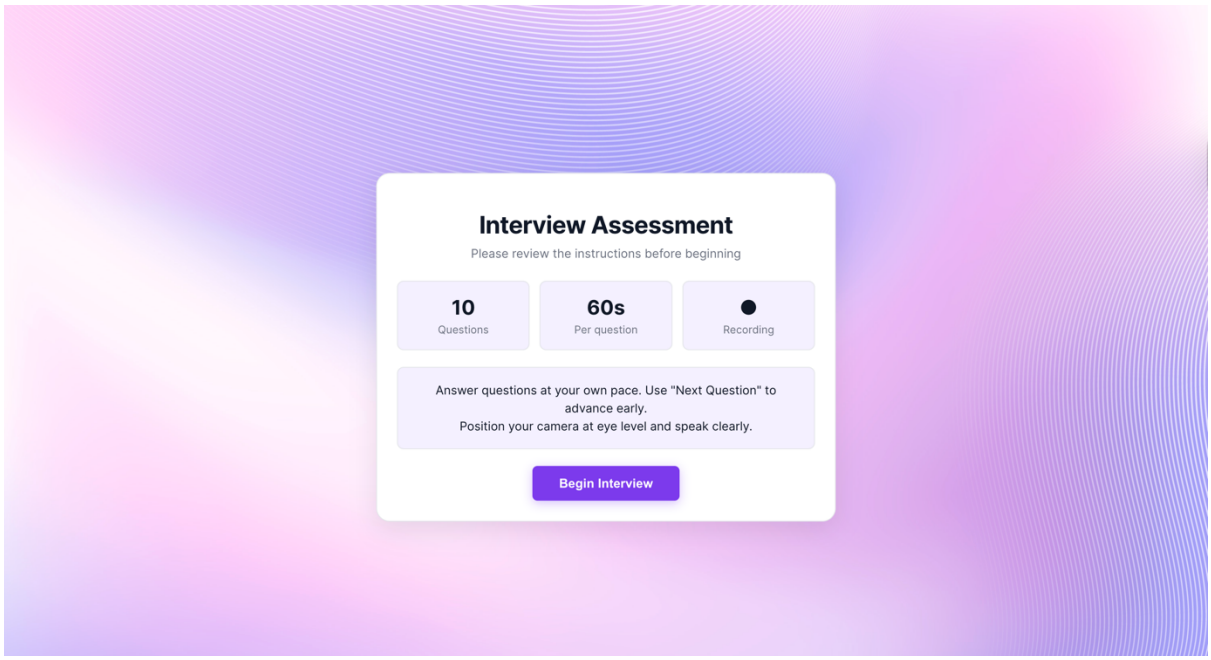


Figure 25 :Interview Landing Page

4.8.13 Screenshot of Interview Conduction Page

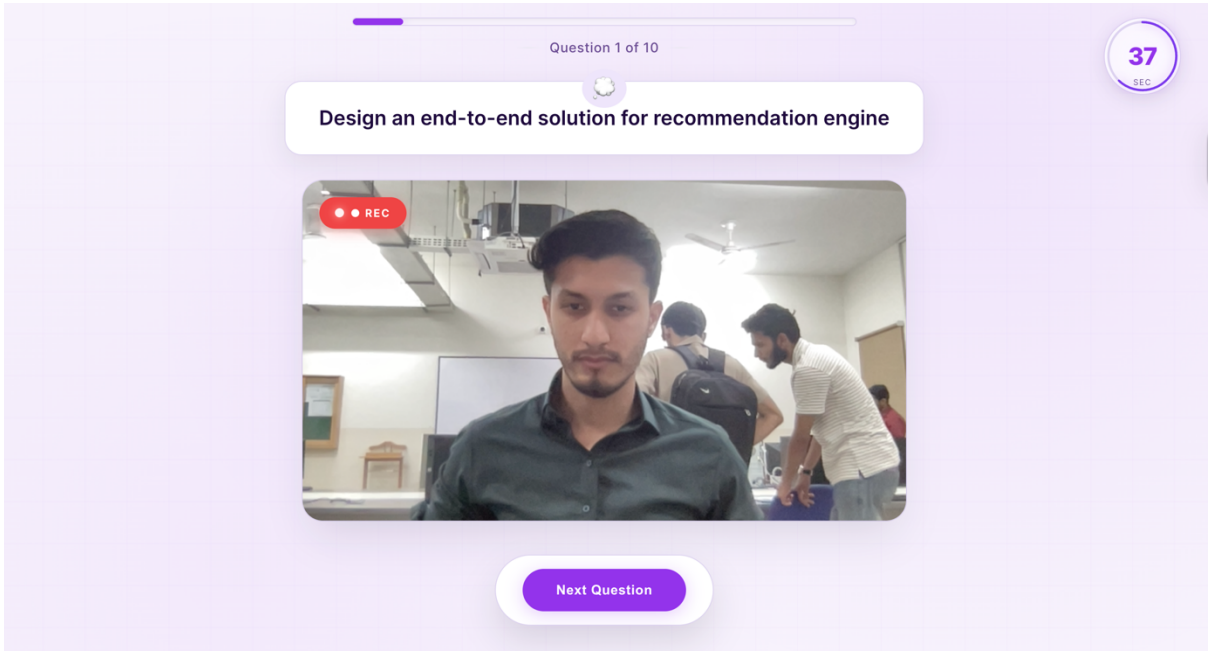


Figure 26: Interview Conduction

4.8.14 Screenshot of Settings Page

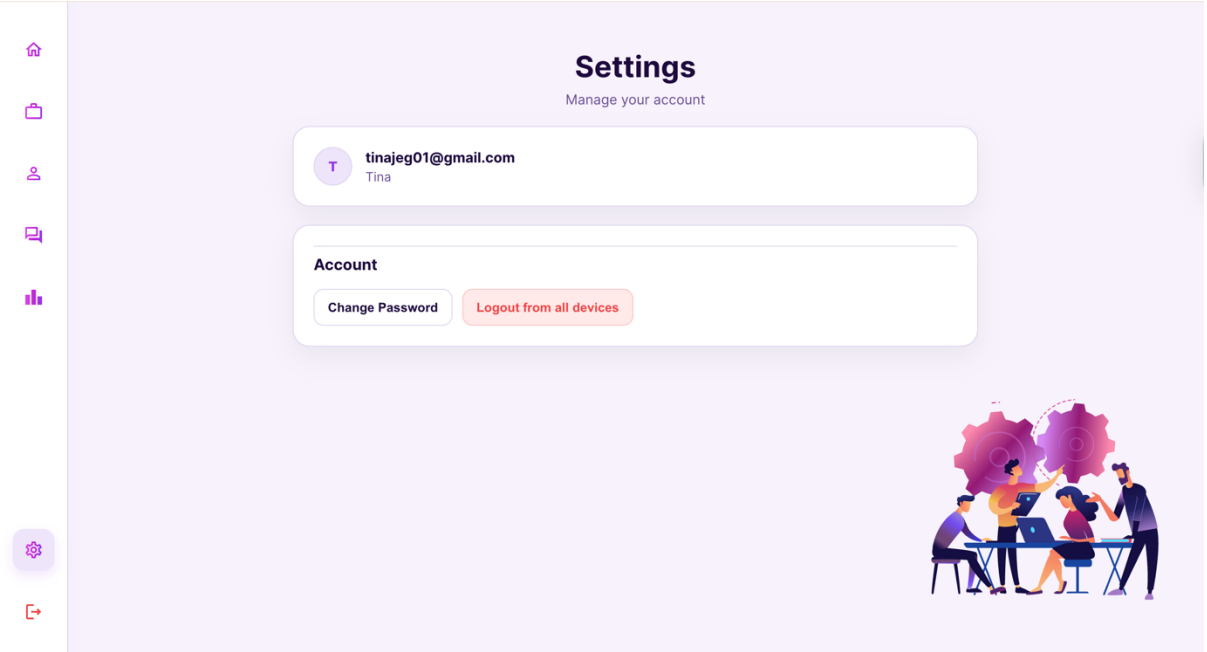


Figure 27: Settings Page

4.8.15 Screenshot of Admin Login Page

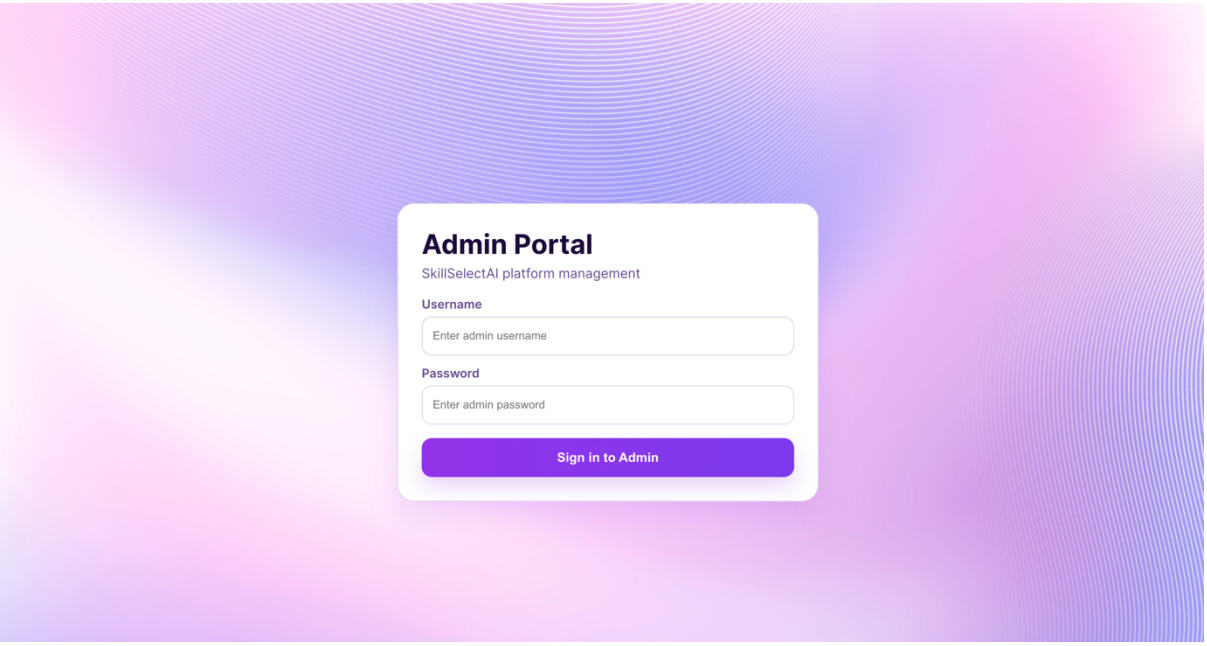


Figure 28: Admin Login Page

4.8.16 Screenshot of Admin Dashboard

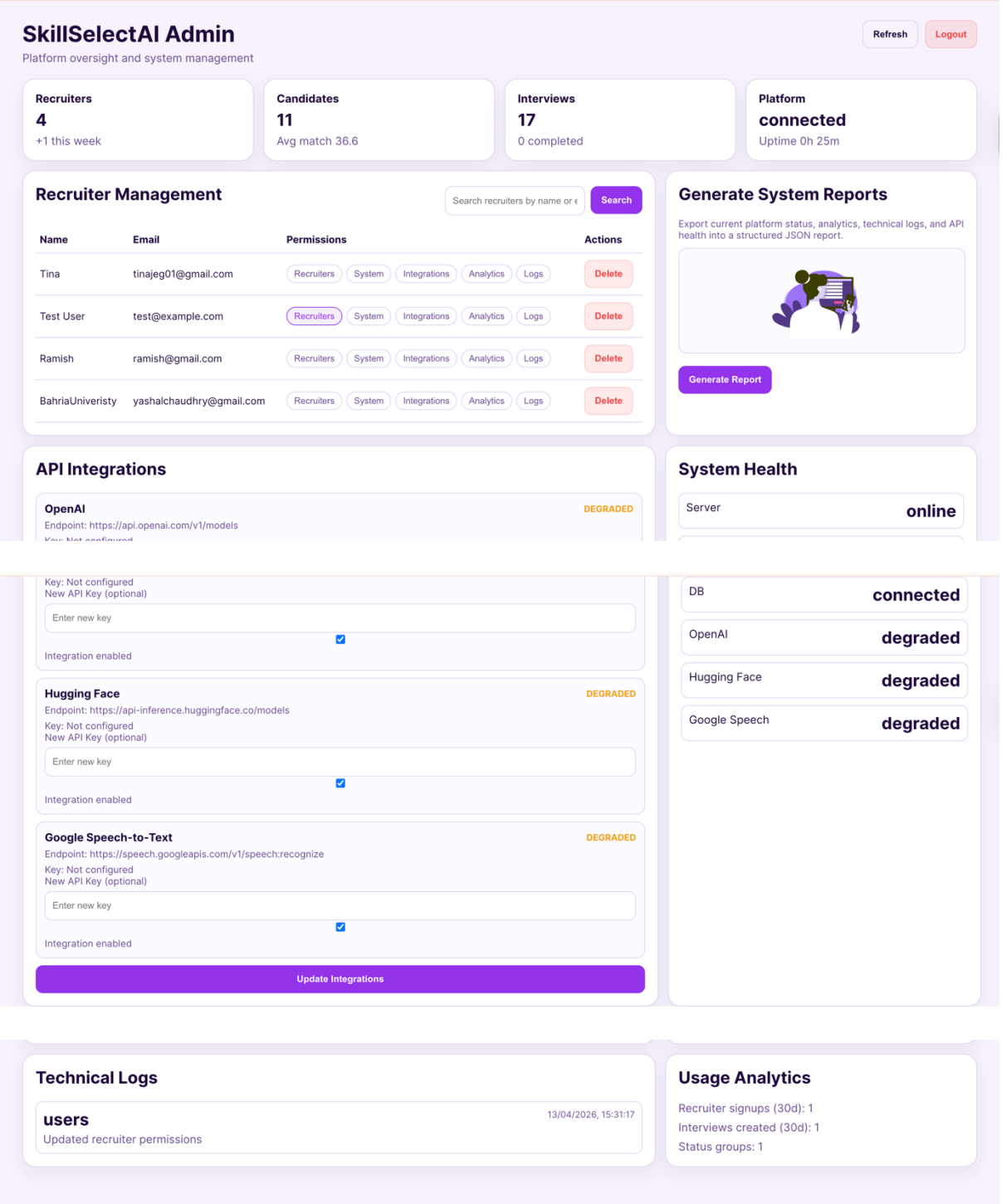


Figure 29: Admin Dashboard

4.9 Conclusion

We have explained about the design approach, architecture, and interface of our application in this chapter.

Chapter # 5

System Implementation

Chapter 5

System Implementation

In the present chapter, we will consider the practical implementation of SkillSelectAI and its tools, technologies, and general approach used to develop the software product successfully. On the whole, the platform was designed to automate technical recruiting and incorporate various aspects, including resume parsing, generation of interview questions for specific positions, and the design of recruiters' workflow procedures.

Additionally, scheduling interviews and conducting a comprehensive assessment based on visual, audio, and natural language processing data were among other essential features of the platform. Thanks to the application of a scoring matrix, the software ensures consistency in the recruitment process and considerably simplifies the decision-making process regarding hiring. Consequently, with the help of the full-scale deployment of SkillSelectAI, all key functions of the software product that were initially outlined during the planning phase have been covered.

5.1 Strategy

Our method in the design of SkillSelectAI was through modular programming. It is basically the idea that all major functionalities in the hiring process were modules that worked on their own while communicating with other modules. This helped us accelerate development and made debugging easier, along with making it possible to expand later on.

Here are the methods we used to implement the approach:

- **Requirement-driven module breakdown:** In this case, we broke down the requirements into major modules, the recruiting workflow, job and interview management, resume parsing, questions generation, and finally interview evaluation itself.
- **Backend-first service design:** We decided to implement our backend first, starting from creating REST services and database models. As a result, we got the ability to handle our jobs, applicants and interview data without working on the user interface at all.

- **Frontend integration and workflow validation:** When our backend was ready, we integrated it with an interface created using React framework. Thanks to this solution, recruiters were able to start interacting with our product.
- **AI pipeline integration:** Finally, after implementing a working solution, we added our custom AI services to evaluate applications.

1. Recognition of behaviors (visual-based)
2. Sound and voice analysis
3. Analysis of content (NLP)
4. Content quality assessment in terms of its relevance to certain position

- **Question generation logic:** We ensured that the question selection logic would be mapped to particular job positions with a balanced level of difficulty to ensure fair treatment for all candidates and technical relevance.
- **Incremental testing and refinement:** We didn't rely only on post-development testing; we tested each component independently first and then the entire process chain to validate the usability.

In this way, we managed to maintain SkillSelectAI's feasibility in terms of real-world applications and a strong reliance on automation.

5.2 Tools and Technologies:

Table 42: Tools and Technologies

Tools	Version	Rationale
MS Word	2016	Documentation
MS PowerPoint	2016	Presentations
Visual Studio Code	2016	Code Editor
Visual Paradigm	17.0	UML Diagrams Design
Git & GitHub	Latest	Version Control

Postman	Latest	API Testing
Figma	Web-Based	UI/UX Design
Technologies	**Version**	**Rationale**
React	19.2.0	Frontend UI development
React Router DOM	7.13.1	Client-side routing and navigation
Axios	1.13.x	API communication between frontend and backend
Material UI (MUI)	7.3.7	UI components and consistent interface design
Node.js	20.x (project target)	Backend runtime environment
Express.js	4.18.2	RESTful API development
MongoDB (via Mongoose)	Mongoose 8.19.1	NoSQL data storage and schema-based modeling
JSON Web Token (JWT)	9.0.3	Authentication and secure session handling
Multer	1.4.5-lts.1	CV/document upload handling
TensorFlow	2.15.0	AI/ML model execution for analysis tasks
MediaPipe	0.10.32	Facial landmark and visual behavior analysis
DeepFace	0.0.98	Emotion recognition from interview video
OpenCV	4.8.1.78	Video frame processing for visual pipeline

OpenAI Whisper	20250625	Speech-to-text transcription for interview audio
Librosa	0.11.0	Audio feature extraction (pace, silence, pitch)
Sentence-Transformers	5.2.2	Semantic similarity and response relevance scoring
TextBlob	0.19.0	Sentiment-related NLP feedback

Table 43: Models

Models	Purpose
MediaPipe Face Landmarker	This is used to detect facial landmarks and estimate things like gaze and eye contact while the interview video is being analyzed.
OpenCV Haar Cascade Face Detector	We used this for a quick, lightweight face detection pass on every frame before the more complex visual scoring starts.
DeepFace (Emotion Analysis)	This identifies the main emotions a candidate is showing from sampled video frames to get a sense of their emotional engagement.
OpenAI Whisper (Base)	This handles the audio-to-text transcription, basically turning the candidate's spoken answers into text that we can analyze later.
Sentence-BERT (all-MiniLM-L6-v2)	This calculates how similar a candidate's answer is to the "ideal" content we expected, which helps with scoring accuracy.
TextBlob	We used this for basic NLP signals, like sentiment analysis, to provide extra feedback on the quality of the response.

Librosa Signal Features	This extracts specific audio metrics like how much silence there was, the energy of the voice, and general speaking patterns.
Gemini 2.0 Flash	This is the brain that evaluates the final transcripts for technical accuracy, relevance, and communication style, even offering tips for improvement.

5.2.1 Explanation

1. **MS Word:** MS Word was the tool of choice when preparing all forms of documentation, including requirement specification, mid-term presentations, and finally, this very project report.
2. **MS PowerPoint:** For creating presentation slides for project proposal, mid-terms, and final evaluations, we used MS PowerPoint.
3. **Visual Studio Code:** Our primary IDE was Visual Studio Code, which is fast, highly extendable through its vast library of plugins and has an integrated console.
4. **Visual Paradigm:** In case we had to create UML diagrams such as use cases, activity diagram, or sequence diagram, we used Visual Paradigm. It was also helpful when modeling the system architecture.
5. **Git & GitHub:** To control the versions of the source code files, Git & GitHub were employed, which allowed collaboration, tracking of changes, and branching the repository without overwriting any team member's contributions.
6. **Figma:** Figma was chosen as the tool for designing the User Interface/User Experience. It allows for convenient prototyping by designing wireframes and mockups before actual coding of the frontend part begins.
7. **React**: The React framework was selected for developing the frontend part of the project. The use of components made it easy to build reusable UI elements, which provided a seamless experience when navigating from one page to another.
8. **Express.js and Node.js**: The combination of both frameworks was used as a foundation of backend creation. Both were intended for implementation of all necessary APIs and

business logic. Express.js and Node.js make possible an efficient work with authentication and interviewing.

9. **MongoDB and Mongoose**: Working with dynamic data like job offers, candidate profiles, interviews, and so on, NoSQL database is needed. That is why we have chosen to use MongoDB and Mongoose for schema management in our case..
10. **JWT and Bcrypt**: Security is of critical importance; therefore, it was necessary to involve two modules for token-based login protection and hash passwords with bcrypt in order to secure user accounts.
11. **Multer**: This module is intended for uploading files to the server. That is why we could process CVs of users and the videos of interviews uploaded by them..
12. **OpenCV**: With the help of this tool, we managed to build up the algorithm of video files analysis in order to track candidates' behavior during interviews.
13. **MediaPipe**: To detect the facial landmarks and eye tracking, we introduced the MediaPipe framework. This allows the AI to estimate the level of engagement, such as whether the candidate makes eye contact during the interview.
14. **DeepFace**: The emotion detection was done by using the DeepFace technology. It processes each frame of the video interview to give the emotion analysis of the candidate.
15. **Whisper**: It was used to transform candidates' oral answers into texts that can then be used for further NLP evaluations.
16. **Librosa**: With librosa we were able to work with audio files directly, analyzing their content and extracting features such as speech activity and vocal characteristics necessary for evaluations of communication quality.
17. S**entence-BERT**: To evaluate whether candidates' answers matched what was expected from them, we utilized "semantic similarity" function that can be calculated using the Sentence-BERT algorithm.
18. **TextBlob**: The final element of our stack is TextBlob, which helped us calculate sentiment scores for candidate's answers transcripts.
19. **Gemini**: Gemini helped us immensely while evaluating candidates' answers. We were not only able to evaluate technical and communication competences of the candidates but also make recommendations for further improvement.
20. **Dotenv**: While deploying the system, we used dotenv library to store any sensitive data in environment variables safely.

21. **Nodemon** Our tool of choice to make working process more efficient during development. With nodemon utility, we could automatically restart server after making changes in source code.

5.3 Conclusion

In its actual implementation, SkillSelectAI demonstrated that we have managed to successfully develop a fully functional and modular recruitment platform using artificial intelligence. The integration of the front-end application for recruiters with the back-end automation and AI-driven assessments was successful, as per our approach where we first design and implement the essential aspects of the recruitment process and then add advanced functionality.

The selection of technologies like React and Node.js, along with the implementation of Python-based AI modules, turned out to be the most optimal way for performing resume parsing, building interview questions based on job requirements, and candidate assessments. Besides, our proposed architecture is scalable from the perspective of data management and storage.

Chapter # 6

System Testing and Evaluation

Chapter 6

System Testing & Evaluation

This section will cover the implementation and testing processes that ensure the stability of the program, as it does not crash every few minutes. All the tests from unit and component testing to integration and system testing, as well as use case testing, were done to ensure that all parts work together efficiently and do what they should.

Our first objective in this stage was to show that the functionality of our modules, such as the CV parser and the interview recorder, works individually. Once we made sure that these individual modules do their part, we needed to ensure that they worked together. It was important for us to confirm that our overall process, from the posting of the recruitment advertisement by recruiters to the end analysis using AI, does provide relevant output. In addition, this section will discuss the observations made during our evaluation process, the limitations of the system and how testing contributes to high-quality software development.

6.1 Test Strategy

SkillSelectAI's test plan has basically been designed to ensure that all aspects of the software are accurate, reliable, and easy to operate. This approach used a multi-level approach from testing at a function level all the way up to integration level. At the end, we performed system-level testing as well as tested different use cases.

However, our primary concern was making sure all the features did what they were supposed to, whether they were working under regular conditions or even under more difficult situations. Our primary focus in testing the application was its key components, such as user authentication, job creation, questions creation, and AI management of interviews and scores.

We have also taken into consideration that the results actually make sense within the context of the real world. We have considered whether the questions that were generated by the algorithms made sense relative to the jobs that the recruiters advertised, that the APIs did not crash, and that the scores given to candidates make sense.

6.2 Component Testing

First, we did our unit testing for each of the main sub-systems separately, in order to make sure that each individual module worked on its own, before attempting to integrate everything. When testing the back end, we focused primarily on issues such as route handling, validation, and proper database usage.

In regards to the frontend, the emphasis was put on testing the behavior of the forms, as well as checking if there were no problems with state updates in the UI and whether the communication between the front-end and the API worked smoothly for the recruiters during their workflow. Given the importance of the AI elements, which form such a big chunk of the interview process module, these also got a separate test.

In essence, all of the components were tested by supplying them with data to ensure that they generated the output in the proper form, and that the processing did not take too long. By testing them this way, problems with one were not camouflaged by problems with another.

6.3 Unit Testing

The unit tests involved going into the details of the program. The particular pieces of the code such as those that make the program tick were chosen. These are the parts where the mathematics behind the calculation of scores is done, the difficulty level of the questions to ask, and basic user login validation.

In this case, our main concern was ensuring that all functions produce consistent output regardless of whether we provide them with appropriate parameters or deliberately test their resistance to invalid input. In particular, special attention was dedicated to deterministic components like scoring criteria and filters because these areas were the most vulnerable to errors when implementing changes to the system. Testing on such a low level makes the entire development process more sustainable because any minor modifications in one component do not interfere with other files' functionality.

6.4 Integration Testing

The integration tests were carried out in order to see whether all the individual pieces of the software communicated with one another in the way they were supposed to do. While each individual module might be working well by itself, the story is completely different when these modules need to communicate and share their information with each other.

The primary objective of our testing was ensuring that the flow of data through the system is flawless and that no "schema mismatches" occur or dependencies break during the process. This was achieved by executing API test cases to validate whether a particular action, such as creation of a job position, triggers the corresponding behavior at the other end, for example, generation of questions for the interview or registration of a candidate's data. Another aspect considered important was ensuring that the results obtained from the analysis are properly returned and stored following completion of the interview process.

6.5 System Testing

The system testing process involved the assembly of the entire SkillSelectAI solution stack and deployment into a production-like environment where all of the components were integrated as they would have been in the real-world scenario. In contrast to the component-based testing approach, the complete application solution was taken under test as a whole unit. The key scenarios such as logging in, job posting, and CV uploading were tested to verify the overall stability of the system and proper execution of the recruiter workflows. The next steps included verifying whether the system could generate adequate questions for particular roles, send interview invitations to candidates, and analyze their responses. One of the important tasks during the testing was to validate the fault tolerance of the system and determine how the software behaves when a wrong request or missing data are supplied by the user.

6.6 Use Case Testing

6.6.1 User Management and Authentication:

Test Case: 01

Table 44: Test Case of Signup with Valid Details

Test Case	**01**
Objective	**Signup with Valid Details**
Pre-Condition	Signup page should be open
Flow	1. Recruiter opens the app 2. Clicks on Signup 3. Enters valid name, email, password. 4. Clicks Signup button.
Expected Output	Account created successfully
Actual Output	Account created and redirected to dashboard Page
Status	Passed

Test Case: 02

Table 45: Test Case of Signup with Already Registered Email

Test Case	**02**
Objective	**Signup with Already Registered Email**
Pre-Condition	Signup Page should be open
Flow	1. Enter details with an email that already exists in the system 2. Clicks Signup button.
Expected Output	Error message: “Email already registered”
Actual Output	Email already in use
Status	Passed

Test Case: 03

Table 46: Test Case of Signup with Already Registered Username

Test Case	**03**
Objective	**Signup with Already Registered Username**
Pre-Condition	Signup Page should be open
Flow	1. Enter details with a company name of user that already exists in the system 2. Clicks Signup button.
Expected Output	Error message: “User already exists”
Actual Output	Account created
Status	Passed

Test Case: 04

Table 47: Test Case of Signup with Missing fields.

Test Case	**04**
Objective	**Signup with Missing fields.**
Pre-Condition	Signup Page should be open
Flow	1. Recruiter opens the app 2. Clicks on Signup 3. Leaves email field empty 4. Clicks Signup button.
Expected Output	Error message “Email is required”
Actual Output	Please fill all required fields
Status	Passed

Test Case: 05

Table 48: Test Case of Signup with Invalid Email

Test Case	**05**
Objective	**Signup with Invalid Email**
Pre-Condition	Signup Page should be open
Flow	1. Recruiter enters invalid email format 2. Clicks Signup button.
Expected Output	Error message “Invalid Email Format”
Actual Output	Error message displayed
Status	Passed

Test Case: 06

Table 49: Test Case of Login with Valid Credentials

Test Case	**06**
Objective	**Login with Valid Credentials**
Pre-Condition	Login Page should be open
Flow	1. Recruiter enters registered email and correct password 2. Clicks Login
Expected Output	Redirected to dashboard Page
Actual Output	Redirected
Status	Passed

Test Case: 07

Table 50: Test Case of Login with Incorrect Credentials

Test Case	**07**
Objective	**Login with Incorrect Credentials**
Pre-Condition	Login Page should be open
Flow	1. Enter incorrect email and password 2. Click Login
Expected Output	Error message “Invalid Credentials”
Actual Output	Error displayed
Status	Passed

6.6.1.8 Test Case: 08

Table 51: Test Case of Login with Incorrect Email

Test Case	**08**
Objective	**Login with Incorrect Email but correct password**
Pre-Condition	Login Page should be open
Flow	1. Enter incorrect email and correct password 2. Click Login
Expected Output	Error message “Invalid Credentials”
Actual Output	Error displayed
Status	Passed

Test Case: 09

Table 52: Test Case of Login with Incorrect Password

Test Case	**09**
Objective	**Login with Incorrect Password but correct email**
Pre-Condition	Login Page should be open
Flow	1. Enter incorrect password and correct email. 2. Click Login
Expected Output	Error message “Invalid Credentials”
Actual Output	Error displayed
Status	Passed

Test Case: 10

Table 53: Test Case of Forgot Password with Valid Email

Test Case	10
Objective	**Forgot Password - Valid Email**
Pre-Condition	1-Login Page should be open. 2-Forgot password option should be visible
Flow	1. Recruiter Enters valid email and incorrect password. 2. Click on Forgot Password 3. Enter valid registered email 4. Receive Reset Link 5. Reset Password
Expected Output	Password Reset Successful
Actual Output	Password Reset Successful
Status	Passed

Test Case: 11

Table 54: Test Case of Forgot Password with Invalid Email

Test Case	11
Objective	**Forgot Password - Invalid Email**
Pre-Condition	1-Login Page should be open. 2-Forgot password option should be visible
Flow	1. Enter unregistered email 2. Click submit
Expected Output	“Email not found” error
Actual Output	Error message displayed
Status	Passed

Test Case: 12

Table 55: Test Case of Logout

Test Case	12
Objective	**Logout**
Pre-Condition	Recruiter must be logged in
Flow	1. Click Logout
Expected Output	Redirect to login Screen
Actual Output	Redirected
Status	Passed

6.6.2 CV Parsing

Test Case 13

Table 56: Test Case of CV upload

Test Case	13
Objective	**Upload CV successfully**
Pre-Condition	Recruiter is logged in
Flow	1. User navigates to CV Upload section 2. Selects a valid CV file 3. Clicks Upload
Expected Output	CV is uploaded and stored successfully
Actual Output	CV uploaded successfully
Status	Passed

Test Case 14

Table 57: Test Case of CV Extraction

Test Case	14
Objective	Extract information from CV
Pre-Condition	CV is uploaded
Flow	1. System sends CV to parser 2. Parser extracts data 3. Data displayed
Expected Output	Candidate name, email, skills, and experience extracted correctly
Actual Output	Data extracted successfully
Status	Passed

Test Case 15

Table 58: Test Case of candidate upload unsupported file format

Test Case	15
Objective	Handle unsupported file format
Pre-Condition	User uploads non-supported file (e.g., .txt)
Flow	1. User selects unsupported file 2. Clicks Upload
Expected Output	Error message displayed “Unsupported file format”
Actual Output	Error shown correctly
Status	Passed

Test Case 16

Table 59: Test Case of Match CV with job requirements

Test Case	16
Objective	Match CV with job requirements
Pre-Condition	CV and job description available
Flow	1. System compares CV skills with job skills 2. Matching algorithm runs
Expected Output	Matching score or result displayed
Actual Output	Matching performed successfully
Status	Passed

6.6.3 Interview Module:

Test Case: 17

Table 60: Test Case of Sending Interview Links to Shortlisted Candidates

Test Case	17
Objective	**Send Interview Link to Shortlisted Candidates**
Pre-Condition	Recruiter is logged in; job exists; shortlisted candidates are available
Flow	1. Recruiter opens job details page 2. Clicks Send Interview Invites 3. System generates interview links and schedules interviews
Expected Output	Interview links are generated and sent to shortlisted candidates successfully
Actual Output	Links generated and invites sent
Status	Passed

Test Case: 18

Table 61: Test Case of Candidate Interview Start via Invite Link

Test Case	18
Objective	**Candidate Starts Interview Using Valid Link**
Pre-Condition	Recruiter has sent interview link; candidate has valid tokenized URL
Flow	1. Candidate opens received interview link 2. System validates token/session 3. Candidate enters interview page
Expected Output	Interview session starts successfully for candidate
Actual Output	Candidate accessed and started interview
Status	Passed

Test Case: 19

Table 62: Test Case of Recording Submission and Analysis Trigger

Test Case	19
Objective	**Store Candidate Interview Recording and Analysis Trigger**
Pre-Condition	Candidate completed interview through valid session
Flow	1. Candidate submits interview response 2. Recording is uploaded 3. Analysis pipeline is triggered
Expected Output	Recording saved and analysis initiated
Actual Output	Upload successful and analysis triggered
Status	Passed

Test Case: 20

Table 63: Test Case of Recruiter Result Review

Test Case	20
Objective	**Recruiter Views Candidate Interview Result**
Pre-Condition	Recruiter is logged in; candidate interview analysis exists
Flow	1. Recruiter opens candidate/interview results page 2. Selects candidate 3. System fetches analysis report
Expected Output	Visual, audio, NLP, and final score report is displayed
Actual Output	Full report shown correctly
Status	Passed

Test Case: 21

Table 64: Test Case of Interview Video Retrieval for Recruiter

Test Case	21
Objective	**Recruiter Fetches Interview Video Playback**
Pre-Condition	Recruiter is logged in; candidate interview video exists
Flow	1. Recruiter opens candidate details 2. Clicks view interview recording 3. System fetches interview video endpoint
Expected Output	Candidate interview video is streamed/loaded successfully
Actual Output	Video loaded successfully
Status	Passed

Test Case: 22

Table 65: Test Case of candidate start video based interview

Test Case	22
Objective	**candidate starts Video-Based Interview**
Pre-Condition	User is logged in; camera & microphone permissions granted
Flow	1. User click Video Mode interview link 2. Allows camera access 3. Clicks Start Interview
Expected Output	Video interview starts with camera recording
Actual Output	Video interview started successfully
Status	Passed

Test Case: 23

Table 66: Test Case of candidate start video based interview

Test Case	23
Objective	**User starts Voice-Based Interview**
Pre-Condition	User is logged in; microphone permissions granted
Flow	1. User click Voice Mode interview link 2. Allows microphone access 3. Clicks Start Interview
Expected Output	Voice interview starts with microphone recording
Actual Output	voice interview started successfully
Status	Passed

6.6.4 AI service:

Test Case: 24

Table 67: Test Case of system generate interview question

Test Case	24
Objective	**System generates interview questions**
Pre-Condition	Interview session is active
Flow	1. System fetches question from Templates 2. Question displayed to user
Expected Output	Relevant interview question is displayed
Actual Output	Question generated correctly
Status	Passed

Test Case: 25

Table 68: Test Case of candidate submits answer for video Interview

Test Case	25
Objective	**Candidate submits answer in video Interview**
Pre-Condition	Question is displayed; camera & microphone permissions accessed
Flow	1. User record answer for every question 2. Clicks Submit
Expected Output	Answer is recorded and sent for evaluation
Actual Output	Answer submitted successfully
Status	Passed

Test Case: 26

Table 69: Test Case of candidate submits answer for voice Interview

Test Case	26
Objective	**Candidate submits answer in voice Interview**
Pre-Condition	Question is displayed; microphone permissions accessed
Flow	1. User record answer for every question 2. Clicks Submit
Expected Output	Answer is recorded and sent for evaluation
Actual Output	Answer submitted successfully
Status	Passed

Test Case: 27

Table 70: Test Case of System evaluates user response

Test Case	27
Objective	**System evaluates user response**
Pre-Condition	candidate has submitted answer
Flow	1. System sends response to AI 2. AI analyzes answer 3. Feedback generated
Expected Output	Feedback with score and suggestions is displayed
Actual Output	short Feedback generated
Status	Passed

Test Case: 28

Table 71: Test Case of End interview session

Test Case	28
Objective	**End Interview Session**
Pre-Condition	Interview session is ongoing
Flow	1. candidate clicks End Interview
Expected Output	Interview stops and final report is generated
Actual Output	Interview ended successfully
Status	Passed

Test Case: 29

Table 72: Test Case of Generate Feedback with failed API request

Test case	29
Objective	**Generate Feedback with failed API request.**
Pre-Condition	candidate has completed interview
Flow	1. System collects all data (body language, speech) 2. Calls final feedback API 3. API fails: Show message “Feedback unavailable”
Expected Output	Show message “Feedback unavailable”
Actual Output	Feedback unavailable
Status	Passed

6.6.5 Ranking and scoring

Test Case: 30

Table 73: Test Case of View Analytics Successfully

Test Case	30
Objective	**View result from ranking dashboard**
Pre-Condition	Recruiter logged in
Flow	1.Go to Dashboard 2. select job 3.View results from Dashboard
Expected Output	Candidate rankings are shown
Actual Output	Same as expected
Status	Passed

Test Case: 31

Table 74: Test Case of calculate candidate

Test Case	**31**
Objective	Calculate candidate score after interview
Pre-Condition	Candidate has completed interview
Flow	1. System collects responses 2. Sends to AI evaluator 3. Score is calculated
Expected Output	Candidate score is generated accurately
Actual Output	Score calculated successfully
Status	Passed

Test case: 32

Table 75: Test Case of generate ranking for multiple candidate

Test Case	32
Objective	Generate ranking for multiple candidates
Pre-Condition	Multiple candidates have completed interviews
Flow	1. System retrieves all scores 2. Sorts candidates based on scores
Expected Output	Candidates ranked from highest to lowest score
Actual Output	Ranking generated correctly
Status	Passed

Test case: 33

Table 76: Test Case of display ranking for recruiter

Test Case	33
Objective	**Display candidate ranking to recruiter**
Pre-Condition	Ranking has been generated
Flow	1. Recruiter opens ranking dashboard 2. System fetches ranking data
Expected Output	Ranked list of candidates displayed
Actual Output	Ranking displayed correctly
Status	Passed

6.7 Conclusion

The entire testing and evaluation process indicated that the SkillSelectAI application was production-ready. It was fully functional and the components were stable enough to perform the technical recruitment processes without experiencing any malfunctions. The application underwent unit, component, and integration testing to verify its functionality at each level, as well as assess how all components interacted.

In general, the evaluation demonstrated the validity and reliability of the SkillSelectAI application. It was consistent in performing its designated tasks and was reliable during the regular use within the established context. Additionally, it provided a stable basis for further development of the application in the future.

Chapter # 7

Conclusion

Chapter 7

Conclusion

Recruitment is one of those things that drags on forever with all the manual work involved. Sifting through resumes and ranking candidates just seems so inefficient a lot of the time. We came up with SkillSelectAI to handle the boring stuff, like quickly scanning CVs and sorting people based on skills. It even gives interview tips or something like that. Recruiters can then focus on actual conversations instead of getting buried in paperwork. I hope it speeds everything up and improves decisions, but it is hard to say for sure how much yet.

The tech side felt a bit tricky to put together. For the front end, we used React because it manages the user interface without too many issues. Then on the back end, Node.js paired with MongoDB handles the running parts and data storage. The AI features, you know, parsing those CVs and checking performance, that is where Python comes in with Flask to serve it and TensorFlow for the machine learning. Mixing different languages was not always straightforward, it got messy sometimes.

In the report, we start by explaining why we built this in the first place. There are already hiring platforms out there, but they mostly do the basic things and skip the deeper analysis. From there, it covers the whole process, figuring out needed features early on, then designing, coding, and testing. We added stuff on databases and use cases, along with some diagrams to make it easier for anyone reading later. That technical explanation part stands out, even if it comes off kind of rough.

It all wrapped up on schedule and works about how we wanted. The system seems to scale okay, I think, and it really cuts down hassle for hiring teams. Some tweaks might come later, but it feels like a decent beginning overall.

7.1 Contributions

We finally got everything up and running, you know, all the functional parts and even the technical bits behind it. It was pretty much exactly like we wanted.

SkillSelectAI basically cleans up the whole recruitment mess. Recruiters can track candidates without losing their minds, and it pulls in AI to handle interviews or check performance stuff. That seems helpful.

The annoying hiring problems, like sorting through tons of CVs by hand, it deals with those. Or the way feedback after interviews ends up all over the place. AI steps in for analysis, so things feel more fair, I think, and it does not take forever anymore.

7.2 Reflections

7.2.1 Strengths

So, here's the breakdown of what the platform actually does. We basically focused on the features that would save the most time:

- **CV Parsing:** The system reads through all the resumes and ranks the candidates automatically. This way you do not have to scan every single CV by yourself which is a huge pain.
- **Auto-Scheduling:** It handles all that annoying back and forth of setting up interviews. It also sends out secure links to everyone involved so you dont have to worry about it.
- **AI Interview Tech:** This part records the interviews and then uses AI to help evaluate how the candidate actually did during the process.
- **Recruiter Dashboard:** This is a single spot where you can see all of your job postings and keep track of where every single candidate is at in the process.
- **Admin Panel:** A separate area for managing all the users and just keeping an eye on how the whole platform is running overall.
- **Scaleable Setup:** We built the architecture in a way so it wont crash or slow down if you are dealing with huge amounts of data or thousands of applicants at once.

Overall, it's just a way more organized way to hire people without getting bogged down in the tiny details.

7.2.2 Weaknesses

The system is currently limited to using a web browser. Additionally, for the AI processing to function properly, a reliable internet connection is essential; if the Wi-Fi is inconsistent, the entire "smart" aspect of things may have trouble.

7.2.3 Disciplined Project Management

To ensure that SkillSelectAI was completed on schedule, we maintained organization. In essence, we divided the entire process into smaller components, such as the admin dashboard, ranking, and CV parsing. We prevented last-minute panics and ensured that all the various components fit together correctly by the deadline by closely monitoring our preparation and meeting our milestones.

7.2.4 Importance of Team Communication

Probably the most crucial aspect of the evolution was their regular communication with one another. Regular conversations and feedback sessions were quite helpful when we ran into technical difficulties. It ensured that everyone was on the same page and that the finished system seemed more like a cohesive project than a collection of disparate pieces put together.

7.3 Future Work

7.3.1 Other Platforms

Right now SkillSelectAI is only available as a web based system which work fine but it definitely have its limits. Moving forward one of the main goals are to develop dedicated mobile apps for both Android and iOS. This would be a huge plus for recruiters because it let them keep a eye on candidate progress or check in on interviews even when they isnt stuck at they're desks. Making the platform more portable just feel like the natural next step for making the whole recruitment process even more flexible.

7.3.2 Additional Features

There is a whole lot of room for grow with SkillSelectAI and we already got some ideas for what come next. Some of the main things we looking at for future versions includes:

- **Job Portal Integration:** It would be beneficial to have integrations with other systems, including LinkedIn or Indeed. This would allow for the automated process of searching for candidates without having to depend on their own resumes.
- **Real-time Analytics:** Having real-time analytics that display statistics about each interview would be a huge step forward. This would help track hiring processes and spot potential problems as they arise.
- **Multi-language Support:** Currently, there are only a few language options available within the system. Adding support for multilingual interviews would greatly enhance its value for companies worldwide.

References

- [1] Fatima, W. (2023). Intelligent Recruitment and Automated Selection System. Proceedings of the International Conference on Information and Communication Technology (IBRAS). Jinnah University for Women. https://www.juw.edu.pk/conference/ibras-2023/4-%20CSSE/6%20Wajiha%20Fatima.pdf
- [2] Development of an AI-Based Interview System for Remote Hiring. (2021). International Journal of Advanced Research in Engineering and Technology (IJARET), 12(3), 543-552. https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_12_ISSUE_3/IJARET_12_03_060.pdf
- [3] Liu, Z. (2025). Interview AI-ssistant: Real-Time Human-AI Collaboration. arXiv preprint. https://arxiv.org/abs/2504.13847
- [4] AI-Powered Mock Interview System with Real-Time Voice & Emotion Analysis. (2025). International Journal of Novel Research and Development (IJNRD). https://www.ijnrd.org/papers/IJNRD2502318.pdf
- [5] Smart Hire: AI Interview Platform. (2025). International Journal for Research in Applied Science & Engineering Technology (IJRASET). https://www.ijraset.com/best-journal/smarthire-ai-interview-platform
- Zhou, F. (2024). AI system report: HireVue's AI-driven assessment tool. *Innovation in Science and Technology, 3*(4), 109–112. https://www.paradigmpress.org/ist/article/view/1204
- Lee, B. C., & Kim, B. Y. (2021). *Development of an AI based interview system*. *International Journal of Advanced Research in Engineering and Technology (IJARET), 12*(3), 573–580. https://d1wqtxts1xzle7.cloudfront.net/66683706/IJARET_12_03_060-libre.pdf

APPENDIX A

turnitin Page 1 of 140 - Cover Page Submission ID trn:oid:::3618:135418080

turnitin Page 2 of 140 - AI Writing Overview Submission ID trn:oid:::3618:135418080

20% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

39 AI-generated only 20%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

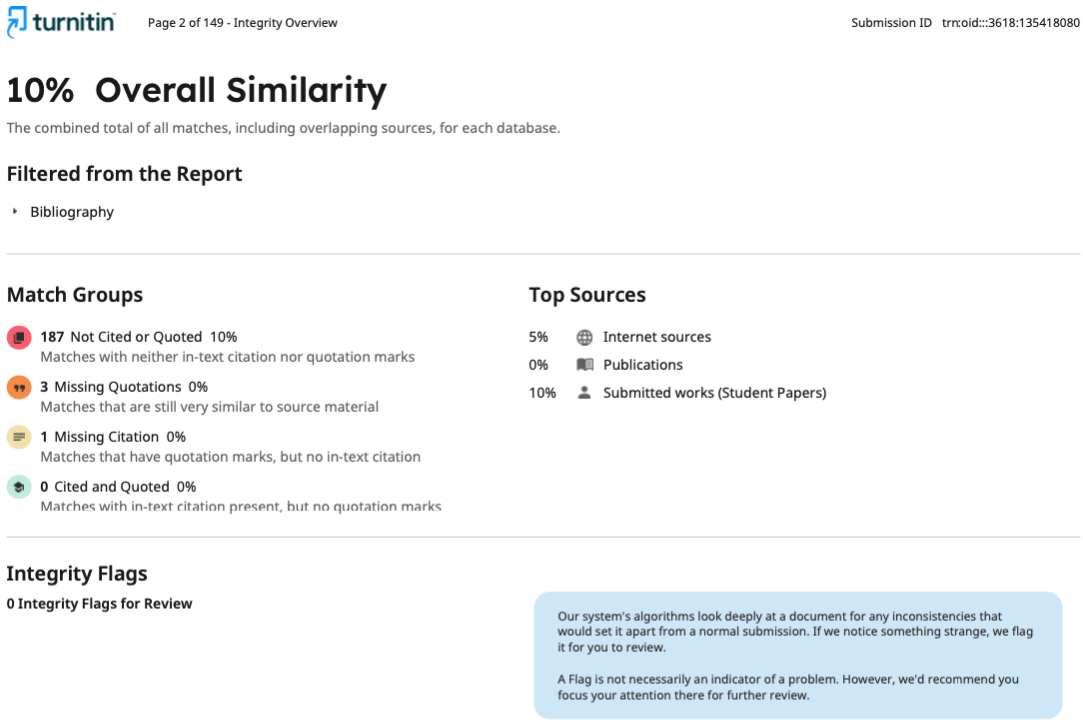
AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

APPENDIX B



turnitin Page 2 of 149 - Integrity Overview Submission ID trn:oid:::3618:135418080

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

187 Not Cited or Quoted 10%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 0%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
0% Publications
10% Submitted works (Student Papers)

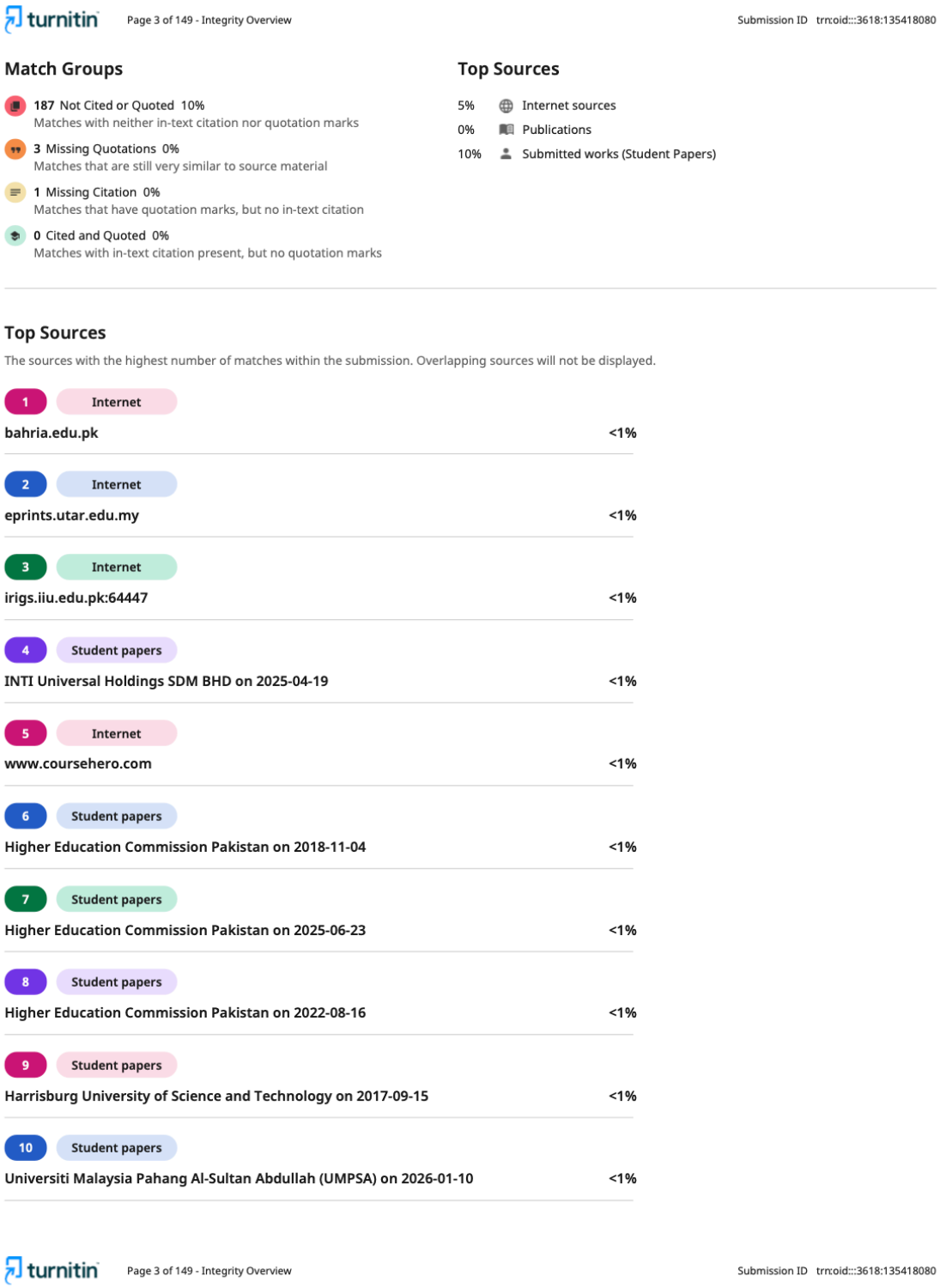
Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

APPENDIX C



turnitin Page 3 of 149 - Integrity Overview Submission ID trn:oid:::3618:135418080

Match Groups

- **187** Not Cited or Quoted 10%
  Matches with neither in-text citation nor quotation marks
- **3** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
0% Publications
10% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	bahria.edu.pk <1%
2	Internet	eprints.utar.edu.my <1%
3	Internet	irigs.iiu.edu.pk:64447 <1%
4	Student papers	INTI Universal Holdings SDM BHD on 2025-04-19 <1%
5	Internet	www.coursehero.com <1%
6	Student papers	Higher Education Commission Pakistan on 2018-11-04 <1%
7	Student papers	Higher Education Commission Pakistan on 2025-06-23 <1%
8	Student papers	Higher Education Commission Pakistan on 2022-08-16 <1%
9	Student papers	Harrisburg University of Science and Technology on 2017-09-15 <1%
10	Student papers	Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) on 2026-01-10 <1%

turnitin Page 3 of 149 - Integrity Overview Submission ID trn:oid:::3618:135418080