** COMSATS University Islamabad**

**Abbottabad, Pakistan**

**AI-Based Interviewer Facilitator Application**

***By***

**Khubaib-Ur-Rehman CIIT/FA20-BCS-141/ATD**

**Danish Nasar CIIT/FA20-BCS-006/ATD**

**Hamdan Ali Tahir CIIT/FA20-BCS-008/ATD**

***Supervisor*Mam. Bushra Mushtaq**

***Bachelor of Science in Computer Science (2020-2024)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

** COMSATS University Islamabad**

**Abbottabad, Pakistan**

**AI-Based Interviewer Facilitator Application**

**A project presented to**

**COMSATS University Islamabad, Abbottabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2020-2024)***

**By**

**Khubaib-ur-Rehman CIIT/FA20-BCS-141/ATD**

**Danish Nasar CIIT/FA20-BCS-006/ATD**

**Hamdan Ali Tahir CIIT/FA20-BCS-008/ATD**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Khubaib-Ur-Rehman Hamdan Ali Tahir Danish Nasar

------------------------- ------------------------- -------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) **AI-Based Interviewer Facilitator Application** was developed by **Khubaib-ur-Rehman (CIIT/FA20c-BCS-141)**, **Hamdan Ali Tahir (CIIT/FA20-BCS-008)** Danish Nasar **(CIIT/FA20-BCS-006)**under the supervision of **Mam Bushra Mushtaq** and that in his opinion it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

-------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**EXECUTIVE SUMMARY**

The project introduces a web-based interviewer facilitator application designed to enhance the interview process for both candidates and interviewers. Through intuitive user interfaces, candidates and interviewers can register and log in, with candidates uploading their resumes in PDF format. The system employs automated text extraction from resumes and fill them in relevant fields, and any missing fields can be manually completed by the candidate.

Interviewers can create interview sessions, upload job descriptions, and retrieve candidate resumes via email. Utilizing the GPT (Generative Pre-trained Transformer) model, the system analyzes both resume data and interview transcripts, ensuring comprehensive evaluation.

During interviews, a transcription module automatically converts candidate and interviewer voices into text using Whisper API. This text data undergoes further analysis using the GPT model. A unique feature allows interviewers to generate pools of interview questions based on this analysis. With the click of a button, three sets of questions are generated, each tailored to different aspects of the candidate's profile and interview performance. Interviewers can also regenerate question pools for updated relevance. Upon completion, a candidate evaluation report is generated, providing valuable insights into the candidate's performance. Interviewers can also generate reports for previous interviews using session IDs, enabling comprehensive evaluation, and tracking of candidate progress over time.

In summary, the project presents an innovative interviewer facilitator application that significantly improves the interview process. Through intuitive interfaces, automated resume analysis, transcription capabilities, and tailored question generation using the GPT model, the system enhances efficiency and effectiveness in candidate assessment. With robust analysis tools and comprehensive reporting features, the application promises to streamline the interview experience, leading to more informed hiring decisions and improved candidate selection outcome.

**ACKNOWLEDGEMENT**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Bushra Mushtaq”. Without her personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Khubaib-ur-Rehman Hamdan Ali Tahir Danish Nasar

------------------------- ------------------------- -------------------------

**ABBREVIATIONS**

|  |  |
| --- | --- |
| UC | Use case |
| DFD | Data Flow Diagram |
| SPM | Software Project Management |
| DBMS | Database Management Systems |
| APP | Application |
| UX | User Experience |
| UI | User Interface |
| SDLC | Software Development Life Cycle |
| HCI | Human Computer Interaction |
| OS | Operating System |
| SRS | Software Requirement Specification |
| SDD | Software Design Description |
| SEC | Software Engineering Concepts |
| SRE | Software Requirements Engineering |
| SPM | Software Project Management |
| DBS | Database Systems |
| PC | Personal Computer |

**TABLE OF CONTENTS**

[**1 Introduction** 1](#_Toc520773734)

[1.1 Brief 1](#_Toc520773735)

[1.2 Relevance to Course Modules 1](#_Toc520773736)

[1.3 Project Background 2](#_Toc520773737)

[1.4 Literature Review 2](#_Toc520773738)

[1.5 Analysis from Literature Review](#_Toc520773739) 3

[1.6 Methodology and Software Lifecycle for This Project 4](#_Toc520773740)

[1.6.1 Rationale behind Selected Methodology](#_Toc520773741) 4

[**2** **Problem Definition** 5](#_Toc520773742)

[2.1 Problem Statement 5](#_Toc520773745)

[2.2 Deliverables and Development Requirements 5](#_Toc520773746)

[2.3 Current System 5](#_Toc520773747)

[**3** **Requirement Analysis** 7](#_Toc520773748)

[3.1 Use Cases Diagram(s) 7](#_Toc520773750)

[3.2 Detailed Use Case 7](#_Toc520773751)

[3.3 Functional Requirements 17](#_Toc520773752)

[3.4 Non-Functional Requirements 20](#_Toc520773753)

[**4 Design and Architecture** 22](#_Toc520773754)

[4.1 System Architecture 22](#_Toc520773759)

[4.2 Data Representation 23](#_Toc520773760)

[4.3 Process Flow/Representation 26](#_Toc520773761)

[4.4 Design Models 27](#_Toc520773762)

[**5** **Implementation** 27](#_Toc520773763)

[5.1 Algorithm 29](#_Toc520773765)

[5.2 External APIs 31](#_Toc520773766)

[5.3 AI Model Accuracy Graph 31](#_Toc520773766)

[5.4 User Interface 31](#_Toc520773767)

[**6 Testing and Evaluation** 45](#_Toc520773768)

[6.1 Manual Testing 45](#_Toc520773771)

[6.1.1 System testing 45](#_Toc520773772)

[6.1.2 Unit Testing 46](#_Toc520773773)

[6.1.3 Functional Testing 50](#_Toc520773774)

[6.1.4 Integration Testing 56](#_Toc520773775)

[**7** **Conclusion and Future Work** 5](#_Toc520773778)7

[7.1 Conclusion 57](#_Toc520773780)

[7.2 Future Work 57](#_Toc520773781)

[**8** **References** 58](#_Toc520773782)

**LIST OF FIGURES**

Fig 3.1 Use Case Diagram 7

Fig 4.1 Architectural Diagram 22

Fig 4.1 Data Representation Diagram Level (0) 23

Fig 4.2 Data Representation Diagram Level (1) 24

Fig 4.3 Data Representation Diagram Level (2) 25

Fig 4.4 Process Flow Diagram 26

Fig 4.5 Design Model 27

Fig 5.1 Home Screen of App 35

Fig 5.2 Home Screen of App 36

Fig 5.3 Interviewer Signup 37

Fig 5.4 Interviewer Login 38

Fig 5.5 Interviewer Reports page 39

Fig 5.6 Job Description Analysis 41

Fig 5.7 Interview Dashboard 42

Fig 5.8 Candidate Signup 43

Fig 5.9 Candidate Login 44

**LIST OF TABLES**

Table 1.4 Literature Review 2

Table 2.3 Component of current System 6

Table 3.1 Use Case 01 7

Table 3.2 Use Case 02 8

Table 3.3 Use Case 03 9

Table 3.4 Use Case 04 10

Table 3.5 Use Case 05 12

Table 3.6 Use Case 06 13

Table 3.7 Use Case 07 15

Table 3.8 Use Case 08 16

Table 3.9 Use Case 09 7

Table 3.10 Use Case 10 8

Table 3.11 Use Case 11 9

Table 3.12 Use Case 12 10

Table 3.13 Functional Requirement 01 17

Table 3.14 Functional Requirement 02 17

Table 3.15 Functional Requirement 03 18

Table 3.16 Functional Requirement 04 18

Table 3.17 Functional Requirement 05 19

Table 3.18 Functional Requirement 06 19

Table 3.19 Functional Requirement 07 20

Table 3.20 Functional Requirement 08 20

Table 5.1 API Description and Usage 31  
Table 6.1 Signup for candidate 45

Table 6.2 Login for candidate 46

Table 6.3 Upload Resume Pdf 46

Table 6.4 Field extraction from resume pdf in resume form 47

Table 6.5 Access to Compatibility report 48

Table 6.6 Signup for Interviewer 48

Table 6.7 Login for interviewer 49

Table 6.8 Voice Command Activation Unit Test Case 50

# Introduction

Candidates and interviewers face different problems, one of them is bias during the interview. Ai- based interviewer facilitator application is a web-based application which is inspired by these kinds of challenges that are faced by interviewers and candidates during the interview. Ai-based interviewer application offers numbers of features which includes resume analysis, Job description Analysis Real time Transcription of voice into text and Analysis it Also Provides pool of questions and Candidate Compatibility Report.

## Brief

The system will have a web-based user interface. It is an interviewer facilitator application that enhances the interview process for the candidate and interviewer both. Both interviewer and candidate will have their portal, first of all, they will register and login after then candidate will upload his resume by uploading the pdf file in text format and there will be a form which will be automatically filled by using the extract fields if some fields remains unfilled will be filled by candidate manually then at interviewer's side he will create a session for an interview and upload job description pdf and to fetch resume of candidate he will add the email of candidate to fetch his resume information then both of these things are analyzed by using gpt model.

After that interviewer will start interview and the transcription module will start automatically in which voice of candidates and interviewer is changed into text using Whisper Api and then it will be analyzed by using gpt model. And then there will be a button to generate pool of questions when the interviewer will click that button three pool of question based on three analysis will be generated and there will also be a button to regenerate questions by using that the pool of questions will be updated and then when interview is finished a candidate evaluation report will be generated and interviewer can also generate reports of previous interviews by using there session id.

## Relevance to Course Modules:

This project is closely related to several courses studied during a Bachelor of Computer Science (BSCS) program: Software Engineering Concepts:The project involves the application of software engineering principles, methodologies, and best practices to develop a complex Android application. It includes requirements analysis, design, implementation, testing and maintenance phases. Mobile Application Development**.**

The project involves developing a mobile application for the Android platform using tools like Android Studio and frameworks like TensorFlow. Knowledge of mobile development concepts, UI/UX design, and platform- specific features is applied. Human-Computer Interaction (HCI) Considerations for designing an accessible and user-friendly interface are addressed, with a focus on accommodating the needs of visually impaired users. Database Management Systems**:**

Understanding database concepts is crucial for designing the data storage and retrieval mechanisms within the app. Knowledge of MySQL lite and database design principles is essential for managing user profiles and emergency contacts during Registration.

## Project Background

The project introduces a web-based application powered by AI, specifically leveraging the GPT model, to optimize the interview process. It conducts thorough analysis of resumes, job descriptions, and interview transcripts to provide comprehensive insights. No Table features include the generation of tailored question pools and the production of detailed evaluation reports for candidates. Ultimately, the goal is to enhance decision-making capabilities and minimize biases throughout the interviewing experience.

## Literature Review:

There is few software out there that provide same features but not all below are the software’s

that we study for our project.

*Table 1. 1 Literature Review*

|  |  |  |
| --- | --- | --- |
| Application Name | Weakness | Solution in AI-Based Interviewer Facilitator Application |
| Hire Vue  (2014) | Lack of personalization, potential biases | Utilizes AI to personalize interviews, rigorous bias testing, transparent algorithms |
| Interview Buddy  (2016) | Limited adaptability, generic assessments | Iterative development model, flexible and customizable interview process, adaptive feedback. |

## 1.5. Analysis from Literature Review:

In comparison with existing literature on AI-driven interview facilitation and related applications, the AI-Based Interviewer Facilitator Application incorporates several key features and advancements in modernizing the interview process. Here's an analysis:

### 1.5.1. Resume and Job Description Analysis:

Recognizes the importance of analyzing resumes and job descriptions to match candidates with job requirements effectively. Integrates AI for resume and job description analysis, providing insights into candidate qualifications and job compatibility.

### 1.5.2. Real-Time Transcription and Analysis:

Discusses the benefits of real-time transcription and analysis in capturing interview content and dynamics. Utilizes AI-driven transcription and analysis to capture interview conversations, enabling continuous assessment and tailored question generation.

### 1.5.3. Objective Candidate Assessment:

Highlights the need for objective candidate assessments to mitigate bias and improve decision-making. Offers objective candidate assessments through AI analysis, reducing the influence of subjective evaluations and promoting fairness.

### 1.5.4. Structured Follow-Up Questions:

Suggests the use of structured follow-up questions to delve deeper into candidate responses and motivations. Generates structured follow-up questions based on AI analysis, ensuring comprehensive interviews and insightful interactions.

### 1.5.5. Integration of AI and User-Friendly Interface:

Emphasizes the integration of AI technologies and user-friendly interfaces for enhanced usability. Integrates AI for advanced functionalities while maintaining a user-friendly interface, facilitating ease of use for both candidates and interviewers.

### 1.5.6. Data Security and Privacy Measures:

Stresses the importance of data security and privacy in handling sensitive user information. Ensures robust data security and privacy measures, safeguarding user data throughout the interview process.

## 1.6. Methodology and Software Lifecycle for This Project

For our project, we have opted for an iterative incremental software development model, which is well-suited to the nature of our AI-based interviewer facilitator application. This methodology allows us to iteratively build and refine our application in stages, incorporating user feedback, adapting to changing requirements, minimizing risks, and maintaining high levels of motivation and productivity throughout the project lifecycle.

### 1.6.1. Rationale behind Selected Methodology

The decision to employ an iterative incremental model for our project is grounded in several key factors. Early and Frequent Feedback With the iterative approach, we can gather feedback from users and stakeholders early in the development process. This ensures that the application evolves to meet their needs effectively, particularly crucial for enhancing the interview experience for both candidates and interviewers. Flexibility to Adapt The iterative model provides us with the flexibility to adapt to changing requirements and incorporate new ideas seamlessly. This agility allows us to respond promptly to emerging needs without disrupting the overall development process significantly. Reduced Risk of Failure By breaking down the development process into smaller, manageable iterations, we mitigate the risk of major failures or setbacks. Any issues or challenges encountered can be addressed and corrected early on, reducing the likelihood of project failure. Increased Motivation and Productivity the iterative model fosters a sense of accomplishment and progress within the development team. Seeing tangible results at the end of each iteration boosts motivation and encourages continued effort, leading to sustained productivity throughout the project lifecycle.

# 2. Problem Definition

The project addresses the challenges encountered in traditional job interviews, such as biases, inefficiencies, and subjective evaluations. These issues lead to inconsistent questioning, limited insights, and time-consuming assessments, impacting both candidates and interviewers negatively. The AI-Based Interview Facilitator Application aims to overcome these obstacles by providing objective candidate assessments, structured interview processes, real-time insights, and enhanced decision- making capabilities, ultimately improving the overall interview experience for all parties involved.

## 2.1. Problem Statement

The project seeks to tackle the inherent biases and inefficiencies present in traditional job interviews by introducing an AI-Based Interview Facilitator Application. The problem lies in the subjective nature of the interview process, leading to biased evaluations, inconsistent questioning, and limited insights. The outcome is to create a platform that leverages AI technologies to provide objective candidate assessments, enhance interview depth, and streamline the overall interview process for both candidates and interviewers.

## 2.2. Deliverables and Development Requirements

The project will yield the following deliverables upon completion:

### 2.2.1. AI-Based Interview Facilitator Application

A fully functional web-based application facilitating interviews for candidates and interviewers, featuring modules for resume analysis, job description analysis, real-time transcription, conversation analysis, follow-up question generation, and candidate evaluation reporting.

### 2.2.2. Scope Document:

A comprehensive document outlining the scope, objectives, features, and functionalities of the AI-Based Interview facilitator Application.

### 2.2.3. Software Requirement Specification (SRS) Document:

Detailed documentation specifying the functional and non-functional requirements of the AI-Based Interview Facilitator Application, including system architecture, data models, user interfaces, and security requirements.

### 2.2.4 System Testing Documentation:

Documentation detailing the testing plans, procedures, and results for the AI-Based Interview Facilitator Application to ensure its reliability, functionality, and security.

## . Current System

*Table 2.1 Literature review*

|  |  |  |
| --- | --- | --- |
| Application Name | Weakness | Solution in AI-Based Interviewer Facilitator Application |
| Hire Vue  (2014) | Lack of personalization, potential biases | Utilizes AI to personalize interviews, rigorous bias testing, transparent algorithms |
| Interview Buddy  (2016) | Limited adaptability, generic assessments | Iterative development model, flexible and customizable interview process, adaptive feedback. |

There are few Applications out there that provide same features.

# 3. Requirement Analysis

This chapter includes use case diagram, detailed use cases, functional requirements and nonfunctional requirements.

## 3.1 Use Cases Diagram:

## ksckcn,mcxnxz,m

Fig 3. 1 Use-Case

## Detailed Use Case

Following tables shows the detailed use cases .

**Use Case 01**

The following Table 3.01 shows the process of candidate Registration.

Table 3.1 Use case 1

|  |  |
| --- | --- |
| Use Case ID: | UC-01 |
| Use Case Name: | Candidate Registration |
| Actors: | Candidate |
| Description: | A candidate creates an account by providing necessary information and facilitating access to additional features and interview processes. |
| Trigger: | The candidate selects "Register" on the application. |
| Preconditions: | The candidate is not registered. |
| Postconditions: | The candidate has a registered account. |
| Normal Flow: | The candidate launches the application and selects the "Register" option. The application displays a registration form.  The candidate enters personal details, including name, email, and password. The candidate submits the registration form.  The system validates the information and creates a new candidate account. |
| Alternative Flows: | If the candidate cancels the registration process, the application returns to the main menu.  If the entered email is already in use, the application displays an error  message and allows the candidate to retry. |

**Use Case 02**

The following Table 3.02 shows the process of candidate login.

Table 3.2 Use case 2

|  |  |
| --- | --- |
| Use Case ID: | UC-02 |
| Use Case Name: | Candidate Login |
| Actors: | Candidate |
| Description: | A registered candidate logs in to access personalized features and participate in interviews. |
| Trigger: | Candidate selects "Log In" on the application. |
| Preconditions: | Candidate is not logged in. |
| Postconditions: | Candidate is logged in. |
| Normal Flow: | Candidate selects "Log In" on the application. The application displays a login form.  Candidate enters their email and password.  The system validates the credentials and logs in the candidate. |
| Alternative Flows: | If the candidate cancels the login process, the application returns to the main menu.  If the entered credentials are incorrect, the application displays an error  message and allows the candidate to retry. |

**Use Case 03**

The following Table 3.03 shows the process of resume submission and analysis.

Table 3.3 Use case 3

|  |  |
| --- | --- |
| Use Case ID: | UC-03 |
| Use Case Name: | Submit Resume |
| Actors: | Candidate |
| Description: | A candidate submits their resume through the application, providing additional details relevant to their work experience. |
| Trigger: | The candidate selects "Submit Resume" on the application. |
| Preconditions: | The candidate is logged in. |
| Postconditions: | Resume is successfully submitted. |
| Normal Flow: | Candidate selects "Submit Resume" on the application.  The application displays a form with fields such as registration number, previous experience, and a description of work experience.  Candidate fills in the required information. Candidate submits the resume form.  The system stores the submitted resume data. |
| Alternative Flows: | If the candidate cancels the submission process, the application returns to the main menu. |

**Use Case 04**

The following Table 3.04 shows the process extracting data from resume pdf and filling it is resuming form.

Table 3.4 Use case 4

|  |  |
| --- | --- |
| Use Case ID: | UC-04 |
| Use Case Name: | Resume form fields extraction |
| Actors: | Candidate |
| Description: | Candidate will click on fields extraction button and the resume form will be filled with relevant data. |
| Trigger: | The candidate will select field extraction. |
| Preconditions: | The candidate has uploaded resume pdf. |
| Postconditions: | Resume fields are filled with relevant data. |
| Normal Flow: | Candidate selects "extract fields " on the application. The resume form will be filled with relevant data.  Which fields are not filled are filled manually by the candidate. After filling all the fields, the candidate will submit the form. |
| Alternative Flows: | If candidate does not use field extraction then he will have to fill all fields manually |
| Business Rules | Candidates can use field extraction module if resume pdf is uploaded. |
| Assumptions: | Candidate has uploaded the resume pdf already. |

**Use Case 05**

The following Table 3.05 shows the process of interviewer Registration.

Table 3.5 Use case 5

|  |  |
| --- | --- |
| Use Case ID: | UC-05 |
| Use Case Name: | Interviewer Registration |
| Actors: | Interviewer |
| Description: | Users create an account by providing personal information (e.g., name, email, password). The app validates the email and creates an account. |
| Trigger: | The user selects "Register" |
| Preconditions: | The user is not logged in |
| Postconditions: | The user has a new account and is logged in |
| Normal Flow: | The user selects "Register".  The app displays a form for personal information. The user enters information and submits.  The app creates the account and logs in |
| Alternative Flows: | If user cancels, return to main menu. If email is in use, display error. |
| Business Rules | Candidates can use field extraction module if resume pdf is uploaded. |

**Use Case 06**

The following Table 3.06 shows the process of interviewer Login.

Table 3.6 Use case 6

|  |  |
| --- | --- |
| Use Case ID: | UC-06 |
| Use Case Name: | Interviewer Login |
| Actors: | Interviewer |
| Description: | Registered users log in to their accounts by providing their credentials. The app verifies the credentials and grants access. |
| Trigger: | User selects "Log In" |
| Preconditions: | User is Registered |
| Postconditions: | User enter his account can perform different functions. |
| Normal Flow: | User selects "Log In". App displays a login form.  User enters email and password.  App validates the credentials and logs in. |
| Alternative Flows: | If user cancels, return to main menu.  If credentials are incorrect, display an error. |
| Business Rules | Interviewers can log in only with valid registered accounts and internet access. The system validates interviewer credentials during the login process. |

**Use Case 07**

The following Table 3.07 shows how new session is started.

Table 3.7 Use case 7

|  |  |
| --- | --- |
| Use Case ID: | UC-07 |
| Use Case Name: | Start session |
| Actors: | Interviewer |
| Description: | Interviewers initiate interviews session by providing a session email of the candidate. The app identifies the interview session for future reference. |
| Trigger: | Interviewer selects "Start Interview" |
| Preconditions: | Interviewer is logged in |
| Postconditions: | Interview session is created |
| Normal Flow: | The interviewer selects "Start Interview".  App prompts the interviewer to enter a session name and candidate information. The interviewer provides the session details.  App creates the interview session. |
| Business Rules | Interviewers can start an interview session only if they are logged into the system.  Interview sessions are identified and created by providing a session name and  basic candidate information. |

**Use Case 08**

The following Table 3.08 shows how Job description submission takes place.

Table 3.8 Use case 8

|  |  |
| --- | --- |
| Use Case ID: | UC-08 |
| Use Case Name: | Job Description Submission |
| Actors: | Interviewer |
| Description: | Interviewers submit job descriptions for analysis. The app processes and stores the job requirements. |
| Trigger: | Interviewer selects "Submit Job Description" |
| Preconditions: | Interviewer is logged in and started the session |
| Postconditions: | Job description is submitted and stored |
| Normal Flow: | Interviewer selects "Submit Job Description". App displays a form for job description.  Interviewer enters job details and submits.  App stores the job description. |
| Business Rules | Interviewers can submit job descriptions for analysis.  The system processes and stores job requirements submitted by interviewers. |
| Assumptions: | Interviewer is logged in and have internet access. |

**Use Case 09**

The following Table 3.09 shows how to Start interview.

Table 3.9 Use case 9

|  |  |
| --- | --- |
| Use Case ID: | UC-09 |
| Use Case Name: | Start Interview |
| Actors: | Interviewer |
| Description: | Interviewers initiate interviews by providing a session name and basic  candidate information. The app identifies the interview session for future reference. |
| Trigger: | Interviewer selects "Start Interview" |
| Preconditions: | Interviewer is logged in |
| Postconditions: | Interviewer is logged in |
| Normal Flow: | The interviewer selects "Start Interview".  App prompts the interviewer to enter a session name and candidate information. The interviewer provides the session details.  App creates the interview session |
| Business Rules | Interviewers can start an interview session by providing a session name and  basic candidate information.  The system creates the interview session for future reference. |

**Use Case 10**

The following Table 3.10 shows how to refresh the pool of question manually.

Table 3.10 Use case 10

|  |  |
| --- | --- |
| Use Case ID: | UC-10 |
| Use Case Name: | Refresh Question Pool |
| Actors: | Interviewer |
| Description: | The interviewer can refresh the pool of interview questions during an ongoing interview session. |
| Trigger: | Interviewer selects "Refresh Question Pool" during an interview. |
| Preconditions: | Interviewer is conducting an interview. |
| Postconditions: | Question pool is refreshed. |
| Normal Flow: | The interviewer, during an ongoing interview, selects the option to "Refresh Question Pool."  The system processes the request and updates the pool of interview questions based on the current context, candidate responses, and analysis.  The refreshed pool of questions is displayed to the interviewer for  consideration. |
| Business Rules | Interviewers can refresh the pool of interview questions during an ongoing interview session.  The system updates the pool of questions based on the current context |

**Use Case 11**

The following Table 3.11 shows how interview ends and compatibility report is generated.

Table 3.11 Use case 11

|  |  |
| --- | --- |
| Use Case ID: | UC-11 |
| Use Case Name: | End Interview and Generate Report (Automated) |
| Actors: | Interviewer |
| Description: | Interviews conclude, and the app automatically generates a report summarizing the interview. |
| Trigger: | Interviewer selects "End Interview" |
| Preconditions: | Interviewer is conducting an interview |
| Postconditions: | Interview report is generated |
| Normal Flow: | I Interviewer selects "End Interview". App concludes the interview session.  App automatically generates a report summarizing the interview. |
| Business Rules | Interviewers can end an interview session, and the system automatically generates a report summarizing the interview.  Internet access is required for the automated report generation process. |
| Assumptions: | Interviewer is conducting an interview and has internet access. |

**Use Case 12**

The following Table 3.12 shows how previous interview reports are saved and viewed.

Table 3.12 Use case 12

|  |  |
| --- | --- |
| Use Case ID: | UC-12 |
| Use Case Name: | View Previous Interview Reports |
| Actors: | User |
| Description: | The interviewer can access and view reports and information from previous interview sessions conducted through the system. |
| Trigger: | The interviewer selects the "View Previous Interviews" option in the application. |
| Preconditions: | The interviewer is logged into the system. |
| Postconditions: | The interviewer has successfully viewed the selected interview report. |
| Normal Flow: | The interviewer navigates to the "View Previous Interviews" section in the application.  The system displays a list of previous interview sessions, organized by candidate name, date, or other relevant criteria.  The interviewer selects a specific interview session to view detailed information and the generated report.  The system presents the interviewer with a detailed report, including candidate details, interview questions, responses, and the overall assessment.  The interviewer has the option to download or print the report for reference or sharing. |

## 3.2 Functional Requirements:

All the Major Functional Requirements are itemized below

### Functional requirements 01

The following Table 3.3.1 shows the Functional Requirements 01

Table 3.13 FR-1

|  |  |
| --- | --- |
| Identifier | FR-01 |
| Title | User Registration and Login |
| Requirement | The system shall allow both candidates and interviewers to register  and log in to access personalized features and functionalities specific to their roles. |
| Source | Candidate and Interviewer Registration and Login Use Cases |
| Rationale | To enable users to access system features relevant to their roles and facilitate the interview process. |
| Business Rule | Users must provide valid information for registration and valid credentials for login. |
| Priority | High |

### Functional requirements 02

The following Table 3.3.2 shows the Functional Requirements 02

Table 3.14 FR-2

|  |  |
| --- | --- |
| Identifier | FR-02 |
| Title | Candidate Resume Submission |
| Requirement | The system shall allow candidates to submit their resumes. |
| Source | Submit Resume Use Case |
| Rationale | To collect candidate details for the interview process. |
| Business Rule | Candidates must fill in the required information for resume submission. |
| Dependencies | FR-01 (User Registration and Login) |
| Priority | Medium |

### Functional requirements 03

The following Table 3.3.3 shows the Functional Requirements 03

Table 3.15 FR-3

|  |  |
| --- | --- |
| Identifier | FR-03 |
| Title | Job Description Submission |
| Requirement | The system shall allow interviewers to submit job descriptions for  analysis. |
| Source | Job Description Submission Use Case |
| Rationale | To analyze job requirements for the interview process. |
| Business Rule | Interviewers must be logged in to submit job descriptions . |
| Dependencies | FR-01 (User Registration and Login) |
| Priority | Medium |

### Functional requirements 04

The following Table 3.3.4 shows the Functional Requirements 04

Table 3.16 FR-4

|  |  |
| --- | --- |
| Identifier | FR-04 |
| Title | Resume Form field extraction. |
| Requirement | The system shall allow candidates to click the button of field  extraction after the submission of resume pdf. |
| Source | Resume Submission Use Cases. |
| Rationale | To fill the fields of resume form by extracting data from resume pdf. |
| Business Rule | Candidate should have submit resume pdf already. |
| Dependencies | FR-02 (Resume submission) |
| Priority | Medium |

### 

### Functional requirements 05

The following Table 3.3.5 shows the Functional Requirements 05

Table 3.17 FR-5

|  |  |
| --- | --- |
| Identifier | FR-05 |
| Title | Interview Session Management |
| Requirement | The system shall enable interviewers to start interview sessions , conduct interviews with candidates and automatically generate reports  at the end of interviews. |
| Source | Start Session, Start Interview, End Interview and Generate Report Use  Cases |
| Rationale | To streamline the interview process from initiation to conclusion. |
| Business Rule | Interviewers must be logged in to start and conduct interviews. |
| Dependencies | FR-01 (User Registration and Login) |
| Priority | Medium |

### 

### Functional requirements 06

The following Table 3.3.6 shows the Functional Requirements 06

Table 3.18 FR-6

|  |  |
| --- | --- |
| Identifier | FR-06 |
| Title | Dynamic Question Generation and Management |
| Requirement | The system shall provide an initial set of questions based on  resume and job description analysis and allow interviewers to refresh the question pool during an interview . |
| Source | Initial Question Generation, Refresh Question Pool Use Cases |
| Rationale | To adapt interview questions based on analysis and ongoing session dynamics. |
| Business Rule | None specified. |
| Dependencies | Interview session must be ongoing. |
| Priority | High |

### 

### Functional requirements 07

The following Table 3.3.6 shows the Functional Requirements 06

Table 3.19 FR-8

|  |  |
| --- | --- |
| Identifier | FR-07 |
| Title | User Profile and Interview Report Access |
| Requirement | The system shall allow users to manage their profiles and access interview reports, including viewing reports from previous sessions . |
| Source | User Profile Management, View Previous Interview Reports Use  Cases |
| Rationale | To enable users to manage their information and review past interview  sessions. |
| Business Rule | Users must be logged in. |
| Dependencies | User must be logged in. |
| Priority | High |

### 

### Functional requirements 08

The following Table 3.3.6 shows the Functional Requirements 06

Table 3.20 FR-8

|  |  |
| --- | --- |
| Identifier | FR-08 |
| Title | Interview Transcription and Analysis |
| Requirement | The system shall enable transcription of interviews as they start and utilize the transcribed data for analysis and question pool updates. |
| Source | Start Interview Use Case |
| Rationale | To document and analyze the interview conversation for dynamic question updating. |
| Business Rule | None specified. |
| Dependencies | The interview must be started. |
| Priority | Medium |

## Non-Functional Requirements

### 3.3.1. Usability-1: User-Friendly Interface

The system must have a user-friendly interface that is easy to navigate and understand. This requirement is crucial to ensure a positive user experience and facilitate easy adoption of the system. Verification will be achieved through user testing and feedback.

### 3.3.2. Usability-2: Clear Instructions

The system must provide clear and concise instructions for using each feature. This clarity is essential to facilitate user understanding and reduce the learning curve, ultimately leading to a smoother experience. Verification will be conducted through documentation review and user feedback.

### 3.3.3. Usability-3: Intuitiveness

The system must be intuitive, allowing users to easily perform tasks and access information. An intuitive design enhances user productivity and satisfaction. This aspect will be verified through user feedback and usability testing.

### 3.3.4. PER-1: Fast Loading and Execution

The system must be optimized for fast loading and execution to provide a responsive user experience. Ensuring quick response times is vital for maintaining user engagement. Verification will involve measuring system response times.

### 3.3.5. PER-2: Scalability

The system must be capable of handling a high volume of requests without experiencing significant delays. This scalability is important to accommodate potential growth in usage over time. Verification will be performed through load testing and response time measurements.

### 3.3.6. PER-3: Large Data Handling

The system must effectively handle large data sets to ensure efficient processing of substantial amounts of data. This capability is critical for maintaining performance under heavy loads. Verification will occur through performance testing with large datasets.

# . Design and Architecture

The following parts of Software Design Description (SDD) report should be included in this chapter.

## System Architecture:

The system employs a client-server architecture, with React for the frontend and Flask for the backend, utilizing Firestore as the database. React provides an intuitive user interface, while Flask handles backend processing and interaction with the database. Firestore ensures efficient data management with real-time synchronization and scalability. This architecture enables seamless communication between client and server components, facilitating real- time data exchange and smooth user interactions.

## 4.2 Data Representation

## Level 0 DFD

A white circle with black text

Description automatically generated

Fig 4 1 Level 0 DFD

## 

## Level 1 DFD

## A diagram of a workflow Description automatically generated with medium confidence

Fig 4 2 Level 1 DFD

## Level 2 DFD

## A diagram of a data flow Description automatically generated

Fig 4 3 Level 2 DFD

## Process Flow/Representation:

## A screenshot of a computer Description automatically generated

Fig 4.4

Fig 4 4Process Flow Diagram

## Design Models

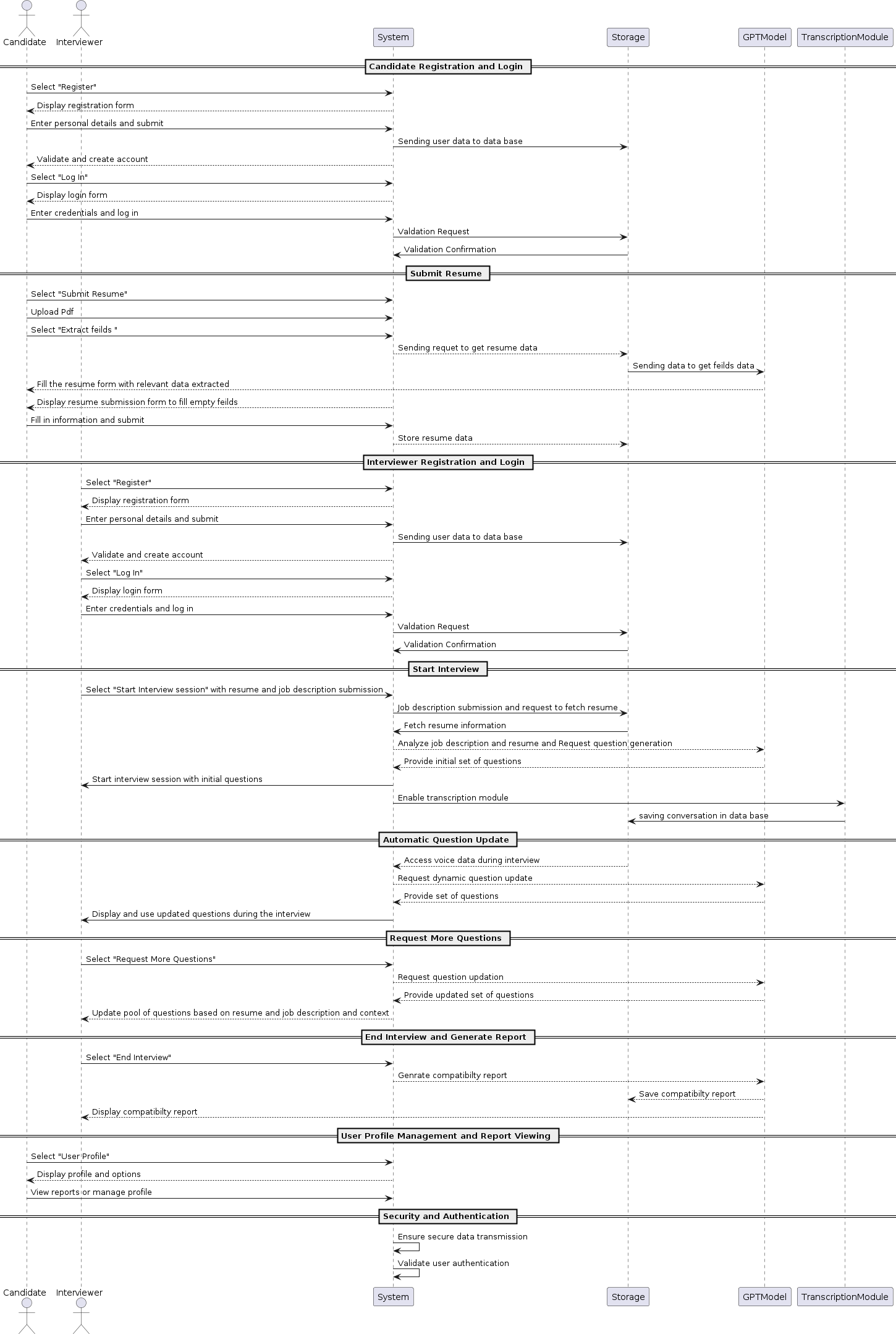


Fig 4.5

Fig 4 5 Design Models

* The sequence diagram presents the interaction between Ai-based Interviewer Facilitator Application and components, from user registration to specific feature utilization like Submit Resume , Resume and Job Description Analysis and so on.
* It captures the app's response to user commands, highlighting the backend processes for delivering accessible information and assistance.
* This diagram serves as a step-by-step guide for developers to understand the order of operations and system responses within the Ai-based Interviewer Facilitator Application.

# Implementation

Implementation includes the algorithms that used to build the project and user interface through which the user interacts with the system.

## Algorithm

An algorithm is a step-by-step procedure or set of instructions designed to solve a specific problem or perform a particular task. Following are the algorithms that used in My Kanban Tool:

**5.1.1. User Registration and Login**

// Pseudo-code for User Registration

function registerUser(username, password, userType) {

// Validate user input

if (isValid(username, password, userType)) {

// Persist user data in database database.insertUser(username, password, userType); return true; // Registration successful

}

return false; // Registration failed

}

// Pseudo-code for User Login

function loginUser(username, password) {

// Check if credentials are valid

if (isValidCredentials(username, password)) {

// Generate authentication token token = generateToken(username);

return token; // Return authentication token

}

return null; // Login failed

}

**Explanation:** These functions handle the registration and login processes. registerUser validates and stores user information, while loginUser verifies credentials and returns an authentication token upon successful login.

**5.1.2. Candidate Resume Submission**

// Pseudo-code for Resume Submission

function submitResume(candidateId, resumePdf) {

// Store resume in candidate's profile if (isValidPdf(resumePdf)) {

database.storeResume(candidateId, resumePdf); return true; // Submission successful

}

return false; // Invalid PDF format

}

**Explanation:** Submit Resume verifies the PDF format and stores it in the database associated with the candidate's profile.

* + 1. **Job Description Submission**

Pseudo-code for Job Description Submission function submitJobDescription(jobDescriptionPdf) {

// Store job description in database if (isValidPdf(jobDescriptionPdf)) {

database.storeJobDescription(jobDescriptionPdf); return true; // Submission successful

}

return false; // Invalid PDF format

}

**Explanation:** Submit Job Description validates and stores the job description PDF in the system.

**5.1.4. Resume Form Field Extraction**

// Pseudo-code for Resume Form Field Extraction function extractResumeFields(resumePdf) {

// Extract fields from resume PDF extractedFields = extractFields(resumePdf); return extractedFields; // Return extracted data

}

**Explanation:** Extract Resume Fields extracts data fields from the submitted resume PDF.

**5.1.5. Interview Session Management**

// Pseudo-code for Interview Session Management function startInterview(candidateEmail) {

// Retrieve candidate's resume

resume = database.fetchResume(candidateEmail);

// Conduct interview and generate report interviewResult = conductInterview(resume); return interviewResult; // Return interview result

}

**Explanation:** Start Interview retrieves the candidate's resume and initiates the interview process.

**5.1.6. Dynamic Question Generation and Management**

// Pseudo-code for Dynamic Question Generation

function generateDynamicQuestions(resumeAnalysis, jobDescriptionAnalysis) {

// Generate initial set of questions

initialQuestions = generateInitialQuestions(resumeAnalysis, jobDescriptionAnalysis); return initialQuestions; // Return initial set of questions

}

// Pseudo-code for Refreshing Question Pool function refreshQuestionPool() {

// Generate new set of questions

newQuestions = generateNewQuestions();

return newQuestions; // Return refreshed questions

}

**Explanation:** Generate Dynamic Questions creates questions based on resume and job description analyses. Refresh Question Pool updates the question pool during an interview session.

**5.1.7 User Profile and Interview Report Access**

// Pseudo-code for User Profile Management

//

function manageUserProfile(userId) {

// Fetch user profile details

userProfile = database.fetchUserProfile(userId); return userProfile; // Return user profile

}

// Pseudo-code for Viewing Previous Interview Reports function viewInterviewReports(userId) {

// Fetch interview reports for user

interviewReports = database.fetchInterviewReports(userId); return interviewReports; // Return interview reports

}

**Explanation:** Manage User Profile retrieves and updates user profile details. View Interview Reports fetches and displays interview reports for the user.

**5.1.8 Interview Transcription and Analysis**

// Pseudo-code for Interview Transcription function transcribeInterview(audioRecording) {

// Convert audio to text

transcribedText = convertAudioToText(audioRecording); return transcribedText; // Return transcribed text

}

// Pseudo-code for Interview Analysis function analyzeInterview(transcribedText)

// Analyze transcribed text

analysisResult = performAnalysis(transcribedText); return analysisResult; // Return analysis result

}

**Explanation:** Transcribe Interview converts interview audio to text format. Analyze Interview processes the transcribed text to generate analysis results.

## 5.2 External APIs

The project leverages several APIs to facilitate data handling and user interaction. As shown in Table 5.1 REST API Description and Usage

*Table 5. 1 External Api used*

|  |  |  |  |
| --- | --- | --- | --- |
| Name of API | Description | Purpose of Usage | Function/Class Name |
| Gpt Api | For performing different analysis and generating compatibility report. | For performing different analysis and generating compatibility report. | Resume Analysis Job description Analysis Conversational Analysis Compatibility report Field extraction |
| Whisper Api | For transcription of voice to text | For transcription of voice to text | Transcription |

## 5.3 User Interface

The user interface for our system typically consists of a web application that represents the workflow and tasks, along with additional features to facilitate task management and collaboration.

**5.3.1 Web Application**

The User Interface for Web application is designed user friendly. You can easily see and access all features of the application.

1. **Home**

A group of people in a room

Description automatically generated

Fig 5. 1 Home Screen of App

A screenshot of a video game

Description automatically generated

Fig 5. 2 Home Screen of App

1. **Interviewers sign up**

A close-up of a login form

Description automatically generated

Fig 5. 3 Interviewer Signup

1. **Interviewer Login**

**A login screen with people in the background

Description automatically generated**

Fig 5. 4 Interviewer Login

1. **Interviewer reports page**

A screenshot of a computer

Description automatically generated

Fig 5. 5 Interviewer Reports page

**A login box in a room

Description automatically generated**

Fig 5. 6 Job Description Analysis

1. **Interview dashboard**

A screenshot of a computer

Description automatically generated

Fig 5. 7 Interview Dashboard

1. **Candidate Signup**

**A screenshot of a login screen

Description automatically generated**

Fig 5. 8 Candidate Signup

1. **Candidate Login**

**A close-up of a login form

Description automatically generated**

Fig 5. 9 Candidate Login

# Testing and Evaluation

Testing in our application involves validating the functionality, performance, and usability of the system. It includes various types of testing, such as functional testing to ensure that the our applications features work as intended, integration testing to verify seamless integration with other systems or tools.

## .Manual Testing

Manual testing is a type of software testing where the testing activities are carried out manually by human testers without the use of automated testing tools. It involves the manual execution of test cases. Following are the types of manual testing that we performed:

* + 1. **System Testing**

After the Ai based interviewer facilitator has been constructed, system testing must be carried out to confirm that it performs as expected and conforms to the requirements that were originally established. The system's smooth and dependable operation is ensured by the testing phase's identification of any defects or issues that the user might not notice at first. All testing efforts (unit testing, functional testing, and integration testing) must be finished before releasing the Ai based interviewer facilitator system to users. This lessens the likelihood of introducing bugs or missing features that might negatively affect the user experience or get in the way of effective project management.

* + 1. **Unit Testing**

Unit testing involves creating test cases for each unit of code and executing them independently. Following are the Tables of Unit Testing:

**Unit testing 1:** Sign up for Candidate

**Testing Objective:** To validate that a Candidate can successfully sign up for the system. Table 6.1 illustrates the unit testing of signup.

Table 6. 1 Signup for candidate

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| Validate that the Candidate is successfully signup and directed to the Candidate’s home page and record is save din database in encrypted form. | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234  Confirm Password: Test1234 | The user should be successfully signed up and directed to the Candidate’s home page and success message should be displayed, confirming the successful sign up. | Pass |

**Unit testing 2:** Login for Candidate

**Testing Objective:** To validate that a Candidate shall successfully login for the system. Table illustrates the unit testing of login for candidate.

*Table 6. 2 Login for candidate*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| validate that the Candidate is successfully logged in and the Candidate is authenticated and has  access to their account | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234 | The user should be successfully logged in and directed to the Candidate’s home page. | Pass |

**Unit testing 3:** Upload Resume Pdf

**Testing Objective:** To verify that candidate can upload Resume in pdf form so that it can be analyzed and used for field extraction in future. Table 6.3 illustrates the unit testing of Upload Resume Pdf.

*Table 6. 3 Upload Resume Pdf*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| Verify that candidate can upload Resume in the pdf from so that it can be analyzed and used for field  extraction in future | Resume in form of Pdf file. | The Resume pdf should be uploaded successfully, and message should be shown for successful submission | Pass |

**Unit testing 4:** Field extraction from resume pdf in resume form

**Testing Objective:** To verify that after uploading resume pdf candidate can extract fields from pdf in resume form. Table 6.4 illustrates the unit testing of Field extraction from resume pdf for resume form.

*Table 6. 4 Field extraction from resume pdf in resume form*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To verify that after uploading resume pdf candidate can extract fields from pdf in resume form. | After clicking button of field extraction fields of resume form should be filled with relevant data. | Pass |

**Unit testing 5:** Access to Compatibility report.

**Testing Objective:** To verify that after giving the interview candidate will have access to compatibility report. Table 6.5 illustrates the unit testing of Access to Compatibility report.

*Table 6. 5 Access to Compatibility report*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To verify that after giving the interview candidate will have access to compatibility report | After giving interview candidate can download and view his compatibility report. | Pass |

**Unit testing 6:** Sign up for Interviewer.

**Testing Objective:** To validate that a Candidate can successfully sign up for the system. Table 6.6 illustrates the unit testing of signup.

*Table 6. 6 Signup for Interviewer.*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| Validate that the Interviewer is successfully signup and directed to the Interviewer’s home page and record is save din database in  encrypted form. | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234  Confirm Password: Test1234 | The user should be successfully signed up and directed to the Interviewer’s  home page and success message should be displayed, confirming the  successful sign up. | Pass |

**Unit testing 7:** Login for Interviewer

**Testing Objective:** To validate that a Interviewer shall successfully login for the system. Table 6.7 illustrates the unit testing of login for candidate

*Table 6. 7 Login for interviewer*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| validate that the Interviewer is successfully logged in and the Interviewer is authenticated and has  access to their account | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234 | The user should be successfully logged in and directed to the Interviewer’s home page. | Pass |

**Unit testing 8:** Start interview session by giving candidates email

**Testing Objective:** To validate that interviewer will add candidate’s email start interview and fetch candidate’s resume. Table 6.8 illustrates the unit testing of login for candidate

*Table 6. 8: Start interview session by giving candidates email*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| To validate that interviewer will add candidate’s email start interview and fetch  candidate’s resume | Email: [abc@gmail.com](mailto:abc@gmail.com) | Session for interview will be started and the resume data will be fetched from record of candidate present  in the database. | Pass |

**Unit testing 9:** Analyze Candidate Resume

**Testing Objective:** To validate that Candidate resume is analyzed by Ai model and analysis is stored in database in encrypted form. Table 6.8 Analyze Candidate Resume .

*Table 6. 9 Analyze Candidate Resume*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that Candidate resume is analyzed by Ai model and analysis is stored in database in encrypted form. | After clicking the button of Resume Analysis the content of resume will be given to Ai model and it will provide analysis on it and will be stored in  database I encrypted form | Pass |

**Unit testing 10:** Upload Job Description Pdf.

**Testing Objective:** To Verify that Interviewer can upload Job description in the pdf from so that it can be analyzed. Table 6.10 illustrates the unit testing of Upload Resume Pdf.

*Table 6. 10 Upload Job Description Pdf*

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Attribute and value | Expected result | Result |
| Verify that Interviewer can upload Job description in the pdf from so that it can be analyzed. | Job Description in form of Pdf file. | The Job Description pdf should be uploaded successfully, and message should be shown for successful submission | Pass |

**Unit testing 11:** Analyze Job Description

**Testing Objective:** To validate that Job description is analyzed by Ai model and analysis is stored in database in encrypted form. Table 6.11 Analyze Job Description.

*Table 6. 11 Analyze Candidate Resume*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that Job description is analyzed by Ai model and analysis is stored in database in encrypted form. | After clicking the button of Job description Analysis, the content of Job description will be given to Ai model and it will provide analysis on it and will be stored in database encrypted form | Pass |

**Unit testing 12:** Start Interview

**Testing Objective:** To validate that after starting the interview transcription takes place and the voice of candidate and interviewer is transcribed into text automatically. Table 6.12 Start Interview.

*Table 6. 12 Start interview*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that after starting the interview transcription take place and voice of candidate and  interviewer is transcribed into text  automatically | Voice of candidate and interviewer will be transcribed into text. | Pass |

**Unit testing 13:** Conversation Analysis.

**Testing Objective:** To validate that conversation of candidate and interviewer which is transcribed into text is given to Ai model and it provides analysis on it and that analysis is stored into database in encrypted form. Table 6.13 Conversation Analysis.

*Table 6. 13 Conversation Analysis*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that conversation of  Candidate and interviewer which is transcribed into text is given to Ai model and it provides analysis on it and that analysis is stored into database in  encrypted form | Analysis of conversation between the interviewer and candidate is saved into database in encrypted form. | Pass |

**Unit testing 14:** Generation of questions

**Testing Objective:** To validate that after interview is started and conversational, resume and job description analysis is done then 3 sets of question containing one question for each (easy, medium and high) difficulty level of each analysis is generated and shown on frontend Table 6.14 Conversation Analysis.

*Table 6. 14 Generation of questions.*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that after interview is started and conversational, resume and job description analysis is done then 3 sets of question containing one question for each (easy, medium and high) difficulty level of each analysis is generated and shown on frontend | 3 sets of question containing one question for each (easy, medium and high) difficulty level of each analysis is generated and shown on frontend | Pass |

**Unit testing 15:** Automatic updating of questions

**Testing Objective:** To validate that after every 30 seconds new sets of questions are generated for each type of analysis for each complexity and shown on frontend Table 6.15 Conversation Analysis.

*Table 6. 15 Automatic updating of questions.*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that after every 30 seconds new sets of questions are generated for each type of analysis for each complexity | Updated 3 sets of question containing one question for each (easy, medium and high) difficulty level of each analysis is generated and shown on frontend  after every 30 second | Pass |

**Unit testing 16:** Dynamic updation of questions.

**Testing Objective:** To validate that when interviewer click on the button for updation of pool of questions new set of questions is generated for each type of analysis for each complexity 6.16 Conversation Analysis.

Table 6. 16 *Dynamic updation of questions.*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that when interviewer click on the button for updating of pool of questions new set of questions is generated for each type of analysis for  each complexity | Updated 3 sets of question containing one question for each (easy, medium and high) difficulty level of each analysis is generated and shown on frontend after clicking button of updating of question pool. | Pass |

**Unit testing 17:** End interview with compatibility report generation.

**Testing Objective:** To validate that when the interviewer ends interview then a pdf for compatibility Is generated having result according to scoring scheme defined 6.17 End interview compatibility report generation.

*Table 6. 17 End interview with compatibility report generation .*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| To validate that when the interviewer ends interview then a pdf for compatibility Is generated having result according to scoring scheme  Defined | Compatibility report generated for candidate | Pass |

**Unit testing 18:** Candidate Logout

**Testing Objective:** To validate that a Candidate can successfully log out for the system. Table 6.18 illustrates the unit testing of Candidate logout.

*Table 6. 18 Candidate Logout*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| Verify that system allows Candidate to logoutafter click on the ‘Logout’ button | System will allow the Candidate to logout successfully | Pass |

**Unit testing 19:** Interviewer Logout

**Testing Objective:** To validate that a Interviewer can successfully log out for the system. Table 6.19 illustrates the unit testing of Interviewer logout.

*Table 6. 19 Interviewer Logout*

|  |  |  |
| --- | --- | --- |
| Test case | Expected result | Result |
| Verify that system allows Interviewer to logoutafter click on the ‘Logout’ button | System will allow the Interviewer to logout successfully | Pass |

**6.1.3 Functional Testing**

It involves testing the system against its functional requirements to ensure that it performs the intended operations correctly. This Section Shows the Functional Testing

**Functional testing 1:** Candidate and interviewer sign up and sign in

**Testing Objective:** To Validate that candidate and interviewer both can register their accounts and an login through their credentials. Table 6.14 illustrates the functional testing of signup.

Table 6. 20 *Candidate and interviewer sign up and sign in*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test case | Attribute and value | Expected result | Result |
| 1. | Validate that the Candidate is successfully signup and directed to the Candidate’s home page. | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234  Confirm Password: Test1234 | The Candidate should be successfully signed up and directed to the Candidate’s home page and success message should be displayed, confirming the  successful sign up. | Pass |
| 2. | Validate that the Candidate is successfully logged in and the Candidate is authenticated and has access to their account | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: Test1234 | The candidate should be successfully logged in and directed to the Candidate’s home page. | Pass |
| 3 | Validate that the interviewer is successfully signup and directed to the interviewer’s home page. | Email: [def@gmail.com](mailto:def@gmail.com) Password: Test1234 Confirm Password: Test1234 | The interviewer should be successfully signed up and directed to the interviewer’s home page and success message should be displayed, confirming the successful  sign up. | Pass |

**Functional testing 2:** Resume Analysis

**Testing Objective:** To Validate that candidate, submit resume in pdf form then fields are extracted from that pdf and filled into resume form and then the interviewer will fetch that resume information by using email of the candidate and then that information is given to Ai model and that Ai model will provide analysis on it and that analysis is stored in database in encrypted form. Table 6.14 illustrates the functional testing of Resume Analysis.

*Table 6. 21Resume Analysis*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test case | Attribute and value | Expected result | Result |
| 1. | Verify that candidate can upload Resume in the pdf from so that it can be analyzed and used for field extraction in future | Resume in form of Pdf file. | The Resume pdf should be uploaded successfully and message should be shown for successful submission. | Pass |
| 2. | Validate that the To verify that after uploading resume pdf candidate can extract fields from pdf in resume form. |  | After clicking button of field extraction fields of resume form should be filled with relevant data. | Pass |
| 3 | To validate that interviewer will add candidate’s email to fetch candidate’s resume | Email: [abc@gmail.com](mailto:abc@gmail.com) | The resume data will be fetched from record of candidate present in the database. | Pass |
| 4 | To validate that Candidate resume is analyzed by Ai model and analysis is stored in database in encrypted form. |  | After clicking the button of Resume Analysis the content of resume will be given to Ai model and it will provide analysis on it and will be stored in database in encrypted form | Pass |

**Functional testing 3:** Job description Analysis

**Testing Objective:** To Validate that interviewer upload the pdf of job description after successful upload the information will be given to the Ai model and that Ai model will provide analysis on job description and that analysis will be stored into database in encrypted form. Table 6.14 illustrates the Job description Analysis.

Table 6.22 :Job Description Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test case | Attribute and value | Expected result | Result |
| 1. | Verify that Interviewer can upload Job description in the pdf from so that it can be analyzed. | Job Description in form of Pdf file. | The Job Description pdf should be uploaded successfully, and message should be shown for successful submission | Pass |
| 2. | To validate that Job description is analyzed by Ai model and analysis is stored in database in encrypted form. |  | Ater clicking he button f job description analysis the information will be given to the Ai model and that Ai model will provide analysis on job description and that analysis will be  stored into database in | Pass |

**Functional testing 4:** Transcriptional Analysis

**Testing Objective:** To Validate that when interviewer starts interview the transcriptional module start working the conversation of candidate and interviewer should be transcribed into text and given to A model in patches and it will provide analysis on it and it that analysis will be stored in database in encrypted form. Table 6.14 illustrates the Transcriptional Analysis.

*Table 6. 23: Transcriptional Analysis*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test case | Attribute and value | Expected result | Result |
| 1. | To validate that interviewer will add candidate’s email and start session . | Email: [abc@gmail.com](mailto:abc@gmail.com) | The session of interview will start. | Pass |
| 2. | To validate that analysis for both job description and resume are done and saved into database in  encrypted form | Resume and job description pdf. | After giving information to ai model about job description and resume the analysis will be provided and saved into database in  encrypted form | Pass |
| 3. | To validate that transcription module start working and the voice of interviewer and candidate is  transcribed into text |  | The voice of interviewer and candidate will transcribed into text. |  |

**Functional testing 5:** Automatic and dynamic generation of questions .

**Testing Objective:** To Validate that when three analysis are given to Ai model it generates three sets of questions each with one question of (easy, medium and difficult) difficulty level and they will be updated after each 30 seconds automatic on the basis of conversation between the candidate and interviewer and if interviewer does not like pool of question he can also update them by clicking the update pool of question button then new set of question will be generated for each analysis type. Table 6.14 illustrates the Automatic and dynamic generation of questions.

*Table 6. 24 Automatic and dynamic generation*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test case | Expected result | Result |
| 1. | To validate that all the three analysis are saved in database so that they can be passed to ai model to generate three sets of respective  Questions | All the three analysis will be given to the Ai model to provide sets of questions for each type. | Pass |

**Functional testing 6:** End of interview and compatibility report generation.

**Testing Objective:** To Validate that when interviewer ends the interview the compatibility report is generated on basis of conversational analysis and the matches between job description and resume according to defined scoring scheme and it will be visible to both candidate and interviewer after end of interview in section of result and evaluation respectively. Table 6.14 illustrates the End of interview and compatibility report generation.

*Table 6. 25: End of interview and compatibility report generation.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test case | Expected result | Result |
| 1. | To Validate that when interviewer ends the interview the compatibility report is generated on basis of conversational analysis and the matches between job description and resume according to defined  scoring scheme | The compatibility report of the candidate is generated. | Pass |

* + 1. **Integration Testing**

Integration testing involves testing the interfaces and interactions between various software modules, subsystems, or external systems to ensure that they work correctly together This Section shows the Integration Testing:

**Integration testing 1:** Check User Authentication Flow

**Testing Objective:** Validate that the signup and login functionalities interact correctly with user (interviewer and candidate) authentication mechanisms. Table 6.18 illustrates the integration testing 1.

*Table 6. 26: Check User Authentication Flow*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute and value | Expected result | Result |
| 1. | Perform Candidate sign up | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: 1234 Confirm password:  1234 | Candidate is registered successfully | Pass |
| 2. | Verify candidate sign in. | Email: [abc@gmail.com](mailto:abc@gmail.com) Password: 1234 | Candidate is login successfully | Pass |

**Integration testing 2:** Resume Handling and AI Analysis

**Testing Objective:** Verify integration between resume upload, AI model for field extraction, and database storage.

*Table 6. 27: Resume Handling and AI Analysis*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute and value | Expected result | Result |
| 1. | Upload resume pdf | Resume Pdf | Candidates resume pdf is uploaded successfully | Pass |
| 2. | Extraction of fields in resume form |  | Fields are extracted Properly in resume form | Pass |
| 3. | Resume Analysis |  | Resume content is given to Ai model and after analyzing it analysis is stored in database | Pass |

**Integration testing 3:** Job description Handling and AI Analysis

**Testing Objective:** Verify integration between job description upload and AI model for analysis and database storage. Table 6.19 illustrates Job description Handling and AI Analysis.

Table 6. 28 Jo*b description Handling and AI Analysis*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case/Test script | Attribute and value | Expected result | Result |
| 1. | Upload job description pdf | Job description Pdf | Job description pdf is uploaded successfully | Pass |
| 2. | Job description Analysis |  | Job description content is given to Ai model and after analyzing it analysis is stored in database | Pass |

**Integration testing 4:** Interview Session Management.

**Testing Objective:** Validate integration between candidate resume retrieval, job description analysis, AI model interaction, and transcription module. Table 6.19 illustrates Interview Session Management.

*Table 6. 29: Interview Session Management*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Test case | Attribute and value | Expected result | Result |
| 1. | Start interview | Email: [abc@gmail.com](mailto:abc@gmail.com) | After giving candidates email interview starts and resume data  is fetched | Pass |
| 2. | Job description Analysis |  | Job description content is given  to Ai model and after analyzing it analysis is stored in database | Pass |
| 3. | Transcriptional module |  | After starting interview the conversation between interviewer and candidate is  transcribed into text |  |

**Integration testing 5:** Automatic and dynamic question generation.

**Testing Objective:** Verify integration between AI analysis outputs and automatic and dynamic question generation module. Table 6.19 illustrates **:** Automatic and dynamic questions generation.

Table 6. 30: *Automatic and dynamic question generation*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test case/Test script | Expected result | Result |
| 1. | Analysis are performed | Content of resume , job description and real time conversation between candidate and interviewer is given to model and analysis are stored into  database | Pass |
| 2. | Automatic Question generation | All these analysis are given to Ai model and it generates 3 questions for each analysis and they are updated after every 30  seconds | Pass |
| 3. | Dynamic question generation | As interviewer click on update question pool button mew pool of question is generated for each  type of analysis | Pass |

**Integration testing 6:** Compatibility Report Generation

**Testing Objective:** Ensure integration between interview conclusion, AI analysis, and compatibility report generation. Table 6.19 illustrates Compatibility Report Generation.

Table 6. 31 Compatibility Report Generation

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test case/Test script | Expected result | Result |
| 1. | End interview session | Interview session is ended by interviewer and transcriptional  module is stopped | Pass |
| 2. | Report generation | All the analysis are given to ai model and it provides a compatibility report according to  the scoring scheme defined | Pass |

# 7. Conclusion and Future Work:

## 7.1 Conclusion

The AI-Based Interviewer Facilitator Application represents a significant advancement in the recruitment process, leveraging artificial intelligence to enhance efficiency, objectivity, and user experience. The application integrates various AI technologies, including resume and job description analysis, real-time transcription, and dynamic question generation, to streamline the interview process and provide comprehensive candidate evaluations. Key benefits observed include Enhanced Efficiency The application automates many time-consuming tasks such as resume screening, job description analysis, and question generation, significantly reducing the time required for interview preparation and execution. Assessments By utilizing AI for candidate evaluation, the application minimizes biases and ensures a more objective assessment of candidates' qualifications and performance. User-Friendly Interface The system's intuitive design ensures ease of use for both interviewers and candidates, facilitating smooth interactions and improving the overall user experience. Comprehensive Reporting Detailed evaluation reports generated by the application provide valuable insights into candidates' strengths and areas for improvement, aiding in more informed hiring decisions.

## 7.2. Future Work

While the AI-Based Interviewer Facilitator Application has demonstrated considerable success, several areas for future improvement and expansion have been identified Enhanced AI Models Further refinement of AI models to improve the accuracy and reliability of resume and job description analysis, as well as real-time transcription and question generation. Integration with Other Systems Expanding the application's compatibility with other HR and recruitment systems, such as Applicant Tracking Systems (ATS) and Human Resource Management Systems (HRMS), to provide a more seamless experience. Bias Mitigation Implementing advanced techniques to detect and mitigate any potential biases in AI algorithms, ensuring fairness and equity in candidate evaluations. User Feedback Mechanisms Developing robust feedback mechanisms to continuously gather and incorporate user feedback, enhancing the application's functionality and user satisfaction. Scalability and Performance Optimizing the system for scalability to manage larger volumes of data and users without compromising performance. Advanced Security Measures Strengthening data security and privacy measures to protect sensitive user information and comply with relevant regulations. Globalization and Localization Adapting the application for use in different regions and languages, making it accessible to a broader audience.

# References

* 1. Interviewer.AI. (2024). "Interviewer.AI | #1 End-to-End AI Video Interview Platform."
  2. Oorwin. (2024). "How AI Powers and Optimizes the Recruitment Process."
  3. Polygon Technology. (2024). "Top 6 AI Recruitment Tools Overview."
  4. ScienceDirect. (2023). "Applicants' perception of artificial intelligence in the recruitment process."
  5. Korn Ferry. (2024). "AI in Recruiting 2024: Pros and Cons."
  6. TechTarget. (2024). "6 challenges of AI in recruitment."
  7. WeAreDevelopers. (2024). "How AI Recruiting Will Improve Hiring."
  8. Adam Fard UX Studio. (2024). "Mastering AI Tools for Recruiting Success."
  9. ResearchGate. (2024). "AI Based Recruitment Tool."
  10. Hiring LinkedIn. (2024). "The Top Recruitment Trends of 2024: Embracing AI and Smart

