

**Recruit Right: Precision Hiring with AI Insight**

**Final Year Project Report**

**Submitted by,**

Muhammad Naeemuddin 1955-2021

Muhammad Abdullah 2206-2021

Muhammad Raza 2207-2021

**Supervisor,**

**Dr. Umer Farooq**

In partial fulfilment of the Requirements for the Degree of

Bachelor of Science in Computer Science,

2021

**Faculty of Engineering Sciences and Technology**

Hamdard University, Main Campus,

Karachi, Pakistan.

### Certificate of Approval



**Faculty of Engineering Sciences and Technology**  
Hamdard University, Karachi, Pakistan

This project “ **Recruit Right: Precision Hiring with AI Insight** ” is presented by “ **Muhammad Naeemuddin,** **Muhammad Abdullah,** **Muhammad Raza** ”under the supervision of their project advisor and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor’s degree in Computer Science.

**Close-up of a signature

Description automatically generated \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Dr. Umer Farooq In-charge FYP-Committee

(Project Supervisor)

----

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Mr. XYZ Dr. Umer Farooq

(Project Co-Supervisor) (Chairman Department of Computing)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Eng. Prof. Aamer Saleem

(Dean, FEST)

### Authors’ Declaration

We declare that this project report was carried out in accordance with the rules and regulations of Hamdard University. The work is original except were indicated by special references in the text and no part of the report has been submitted for any other degree. The report has not been presented to any other University for examination.

Dated:

Authors Signatures:

A close-up of a handwritten text

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Naeemuddin

A close-up of a signature

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Abdullah

A signature on a white background

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Raza

**Plagiarism Undertaking**

We, Muhammad Naeemuddin, Muhammad Abdullah, and Muhammad Raza, solemnly declare that the work presented in the Final Year Project Report Recruit Right has been carried out solely by ourselves with no significant help from any other person except few of those which are duly acknowledged. We confirm that no portion of our report has been plagiarized, and any material used in the report from other sources is properly referenced.

Dated:

Authors Signatures:

A close-up of a handwritten text

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Naeemuddin

A close-up of a signature

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Abdullah

A signature on a white background

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Raza

### Acknowledgment

All praises and thanks be to **Almighty ALLAH**, the Most Gracious, the Most Merciful, the Lord of all worlds, the Sustainer and the Source of all Knowledge, Wisdom, and Strength. Without His countless blessings and divine guidance, the successful completion of this project would not have been possible.

All respects and eternal salutations to the Last Prophet, **Hazrat MUHAMMAD ﷺ**, the Mercy to mankind, whose Exemplary Life, Character, and Teachings continue to enlighten and guide the entire humanity. We also extend our utmost love and reverence to **his Noble AAL (family), The Ahl-e-Bayt**, whose piety, sacrifices, and unwavering dedication to the message of Islam serve as a timeless inspiration for us in all walks of life. May Allah’s peace and blessings be upon them all.

We would like to express our sincere gratitude to the **Hamdard University, (HIET)** for providing us with an academic environment full of encouragement, resources, and the right guidance throughout the journey of this project. It is indeed a matter of great honor and pride for us to complete this project under the banner of such a prestigious institution, where excellence, ethics, and education walk hand in hand.

Our heartfelt appreciation goes to our respected supervisor, **Dr. Umer Farooq**, whose unwavering support, expert guidance, and constant encouragement were instrumental in shaping our work. His valuable insights, technical expertise, and motivating feedback helped us overcome challenges and stay focused on our goal. We are genuinely thankful for the time and effort he invested in us.

We are also indebted to **Ms. Muntaha Mehboob**, our former supervisor, for her early support and thoughtful suggestions during the initial phases of our project. Her words of encouragement remained with us throughout this journey.

We are truly grateful to our **teachers and academic mentors** who laid the foundation of knowledge, and whose teachings and values guided us throughout our academic life. Each lecture, every project discussion, and classroom interaction added layers to our understanding and development.

We would like to extend special thanks to our **classmates, batchmates, and project team members** for their cooperative spirit, valuable suggestions, constructive criticism, and for fostering a collaborative environment. Their encouragement during times of difficulty and celebration of milestones helped us remain motivated.

We cannot forget to mention the boundless love, support, and prayers of our **beloved parents and families**. Their sacrifices, moral strength, and continuous encouragement have been our biggest pillars of support. Their belief in our potential gave us the confidence to overcome every obstacle and accomplish our academic and personal goals.

Lastly, we would like to thank **all those individuals** whether named or unnamed who helped us directly or indirectly in the successful completion of this project. Every act of support, no matter how small, played a significant role in making this achievement possible.  
May Allah bless each one of them with the best of rewards in both this world and the Hereafter.

**Ameen**

### Document Information

Table 1: Document Information

|  |  |
| --- | --- |
| Customer | Tech Industries |
| Project Title | Recruit Right |
| Document | Final Year Project Report |
| Document Version | 2.1 |
| Identifier | FYP-029/FL24 - Final Report |
| Status | Final |
| Author(s) | Muhammad Naeem, Muhd. Abdullah, Muhd. Raza |
| Approver(s) | Dr. Umer Farooq |
| Issue Date | Jun 28th, 2025 |

**Definition of Terms, Acronyms, and Abbreviations***.*

Table 2: Definition of Terms, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| ATS | Applicant Tracking System |
| CSS | Cascading Style Sheets |
| CRUD | Create, Read, Update, Delete |
| CPU | Central Processing Unit |
| CSV | Comma-Separated Values |
| DB | Database |
| DBMS | Database Management System |
| FCM | Firebase Cloud Messaging |
| FYP | Final Year Project |
| GPU | Graphics Processing Unit |
| GDPR | General Data Protection Regulation |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language |
| HTTP | Hyper Text Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| IDE | Integrated Development Environment |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| KPI | Key Performance Indicator |
| MVC | Model-View-Controller |
| NER | Named Entity Recognition |
| NLP | Natural Language Processing |
| OAuth | Open Authorization |
| OS | Operating System |
| PDF | Portable Document Format |
| PII | Personally Identifiable Information |
| RESTful | Representational State Transfer |
| SDK | Software Development Kit |
| SLA | Service Level Agreement |
| SQL | Structured Query Language |
| SSL | Secure Sockets Layer |
| SMTP | Simple Mail Transfer Protocol |
| SSH | Secure Shell |
| TLS | Transport Layer Security |
| UML | Unified Modeling Language |
| UI | User Interface |
| UX | User Experience |
| VLAN | Virtual Local Area Network |
| YAML | YAML Ain’t Markup Language |

### Abstract

RecruitRight: Precision Hiring with AI Insight is an advanced recruitment platform designed to transform traditional hiring processes by addressing inefficiencies, biases, and time constraints. Leveraging the power of artificial intelligence, the platform incorporates Named Entity Recognition (NER) for automated resume analysis and asynchronous video interviews to deliver precise, unbiased candidate evaluations. These innovations streamline recruitment workflows, enabling organizations to focus on top-tier talent and make data-driven hiring decisions with confidence.

With its scalable micro services architecture, RecruitRight ensures robust performance even under high demand, making it suitable for businesses of all sizes. The platform’s user-centric design prioritizes accessibility and ease of use, offering an intuitive experience for candidates, interviewers, and organizations alike. While its core features aim to enhance efficiency and reliability, future potential integrations, such as facial expression recognition (FER), highlight its commitment to innovation and continuous improvement.

This report presents the motivations, methodologies, and technical foundations of RecruitRight, showcasing its potential to revolutionize the hiring landscape through cutting-edge technology and forward-thinking design.

**Keywords:**

* Recruitment Automation
* Artificial Intelligence (AI)
* Named Entity Recognition (NER)
* Asynchronous Video Interviews
* Scalable Micro services Architecture
* Candidate Evaluation
* Data-Driven Decision Making
* Resume Analysis
* User-Centric Design

**Table of Contents**

[Certificate of Approval 2](#_Toc202737172)

[Authors’ Declaration 1](#_Toc202737173)

[Acknowledgment 3](#_Toc202737174)

[Document Information 4](#_Toc202737175)

[Abstract 6](#_Toc202737176)

[List of Figures 10](#_Toc202737177)

[List of Tables 11](#_Toc202737178)

[CHAPTER 1 12](#_Toc202737179)

[INTRODUCTION 12](#_Toc202737180)

[1.1 Motivation 12](#_Toc202737181)

[1.2 Problem Statement 12](#_Toc202737182)

[1.3 Goals and Objectives 12](#_Toc202737183)

[1.4 Project Scope 12](#_Toc202737184)

[CHAPTER 2 13](#_Toc202737185)

[RELEVANT BACKGROUND & DEFINITIONS 13](#_Toc202737186)

[CHAPTER 3 14](#_Toc202737187)

[LITERATURE REVIEW & RELATED WORK 14](#_Toc202737188)

[Literature Review 14](#_Toc202737189)

[Related Work 14](#_Toc202737190)

[Gap Analysis 15](#_Toc202737191)

[CHAPTER 4 1](#_Toc202737192)

[1. Software Engineering Methodology: 1](#_Toc202737193)

[2. Project Methodology 1](#_Toc202737194)

[3. Phases of Project 1](#_Toc202737195)

[4. Software/Tools that Used in Project 3](#_Toc202737196)

[5. Hardware that Used in Project 3](#_Toc202737197)

[Chapter 5 4](#_Toc202737198)

[1. Proposed System Architecture/Design 4](#_Toc202737199)

[2. Functional Specifications 5](#_Toc202737200)

[3. Non-Functional Specifications 6](#_Toc202737201)

[4. Testing 9](#_Toc202737202)

[5. Purpose of Testing 9](#_Toc202737203)

[Chapter 6 10](#_Toc202737204)

[EXPERIMENTAL EVALUATIONS & RESULTS 10](#_Toc202737205)

[1. Evaluation Testbed 10](#_Toc202737206)

[2. Results and Discussion 11](#_Toc202737207)

[2.1. Resume Parsing Accuracy 11](#_Toc202737208)

[2.2. AI Job Matching Results 11](#_Toc202737209)

[2.3. User Feedback 12](#_Toc202737210)

[Table 6: User Feedback 12](#_Toc202737211)

[2.4. System Performance 12](#_Toc202737212)

[2.5. Discussion 12](#_Toc202737213)

[CHAPTER 7 13](#_Toc202737214)

[CONCLUSION AND DISCUSSION 13](#_Toc202737215)

[1. Strength of this Project 13](#_Toc202737216)

[2. Limitations and Future Work 13](#_Toc202737217)

[Limitations: 13](#_Toc202737218)

[Future Work: 14](#_Toc202737219)

[3. Reasons for Failure – If Any 14](#_Toc202737220)

[REFERENCES 15](#_Toc202737221)

[APPENDICES 16](#_Toc202737222)

[A1A. PROJECT PROPOSAL AND VISION DOCUMENT 17](#_Toc202737223)

[A1B. COPY OF PROPOSAL EVALUATION COMMENTS BY JURY 17](#_Toc202737224)

[A2. REQUIREMENT SPECIFICATIONS 18](#_Toc202737225)

[A3. DESIGN SPECIFICATIONS 18](#_Toc202737226)

[A4. OTHER TECHNICAL DETAIL DOCUMENTS 19](#_Toc202737227)

[Test Cases Document 19](#_Toc202737228)

[Test Case 2 – Job Application Flow 20](#_Toc202737229)

[Test Case 3 – Status-Change Notification 21](#_Toc202737230)

[Test Case 4 – Interview Scheduling 21](#_Toc202737231)

[Test Case 5 – Interview Reminder (30 min) 22](#_Toc202737232)

[Test Case 6 – “Start” Button Timing 22](#_Toc202737233)

[Test Case 7 – Recruiter Feedback Submission 23](#_Toc202737234)

[Test Case 8 – Security & Access Control 23](#_Toc202737235)

[Test Case 9 – Bulk Email Queue 24](#_Toc202737236)

[Test Case 10 – Logout & Session Termination 24](#_Toc202737237)

[UI/UX Detail Document 25](#_Toc202737238)

[Coding Standards Document 25](#_Toc202737239)

[Project Policy Document 25](#_Toc202737240)

[User Manual Document 25](#_Toc202737241)

[A5. FLYER & POSTER DESIGN 26](#_Toc202737242)

[A6. COPY OF EVALUATION COMMENTS 27](#_Toc202737243)

[COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION 27](#_Toc202737244)

[A7. MEETINGS’ MINUTES & Sign-Off Sheet 28](#_Toc202737245)

## List of Figures

**Figure No Page No.**

Table of Contents

**Figure 4.1 ER Diagram15**

**Figure 5.1 System Architecture Diagram17**

**Figure 5.2 Use Case Diagram 17**

**Figure 5.3 Sequence Diagram 20**

**Figure 5.4 Class Diagram20**

## List of Tables

**Table No. Page No.**

Table of Contents

**Table 1: Document Information 12**

**Table 2: Definition of Terms, Acronyms and Abbreviation 21**

**Table 3: Gap Analysis 32**

**Table 4: Software Test Plan 23**

**Table 5: Resume Parsing Accuracy 44**

**Table 6: User Feedback 23**

**Table 7: Summary of Software Test Plan 99**

**Table 8: Test Cases Document 15**

**Test Case 1 – Login & Sign Up 15**

**Test Case 2 – Job Application Flow 16**

**Test Case 3 – Status-Change Notification 17**

**Test Case 4 – Interview Scheduling 17**

**Test Case 5 – Interview Reminder (30 min) 18**

**Test Case 6 – “Start” Button Timing 18**

**Test Case 7 – Recruiter Feedback Submission 19**

**Test Case 8 – Security & Access Control 19**

**Test Case 9 – Bulk Email Queue 20**

**Test Case 10 – Logout & Session Termination**

**Table 19: FYP – I Evaluation, Comments by Jury**

**Table 20: Doc Change Record**

# CHAPTER 1

# INTRODUCTION

## Motivation

Time-consuming manual labor, subjective assessments, and inefficiencies are common in the traditional hiring process, which can make it more difficult for businesses to find the best candidates. A streamlined, objective, and automated recruitment solution is becoming more and more necessary as a result of the quick advancement of technology and the rising need for qualified people.  
  
By utilizing cutting-edge technologies like artificial intelligence and natural language processing, the RecruitRight: Precision Hiring with AI Insight project seeks to address these issues and revolutionize the hiring procedure. This platform gives businesses a dependable and effective method to make data-driven hiring decisions by automating personality tests, asynchronous video evaluations, and real-time coding interviews. In the end, this reduces bias, saves time, and improves recruitment results.

## Problem Statement

Conventional technical hiring procedures are ineffective and result in longer turnaround times, erratic candidate evaluations, and higher operating expenses. These issues impede the expansion and efficiency of the organization. To maximize hiring productivity, shorten time-to-hire, and guarantee uniform candidate evaluation, an automated and streamlined solution is essential.

## Goals and Objectives

The project aims to automate and streamline the technical hiring process. The primary objective is to free up internal hiring teams by automating the technical interview stages, allowing them to focus on selecting the best candidates for the role. The platform will feature a network of pre-vetted expert interviewers who will conduct tailored interviews based on specific job requirements, ensuring consistent and objective candidate evaluations. By offering flexible, scalable interview packages, the platform will cater to businesses of all sizes. It will also generate detailed, data-driven post-interview reports, providing insights into candidates' strengths, weaknesses, and performance. Additionally, the project aims to enhance the candidate’s experience by offering a smooth, transparent interview process, ultimately reducing time-to-hire and improving the overall recruitment efficiency.

## Project Scope

Conventional technical hiring procedures are ineffective and result in longer turnaround times, erratic candidate evaluations, and higher operating expenses. These issues impede the expansion and efficiency of the organization. To maximize hiring productivity, shorten time-to-hire, and guarantee uniform candidate evaluation, an automated and streamlined solution is essential.

# CHAPTER 2

# RELEVANT BACKGROUND & DEFINITIONS

The **RecruitRight: Precision Hiring with AI Insight** project draws inspiration from existing innovative platforms that aim to revolutionize the hiring process through technology. Two notable websites relevant to this work are **Intervue** and **Karat**, which have pioneered the integration of advanced tools to streamline and enhance recruitment practices.

**Intervue** is a collaborative interview platform designed to simplify the technical hiring process by providing tools for live interviews. It emphasizes seamless interaction between interviewers and candidates while enabling quick assessments of technical skills.

**Karat** focuses on conducting professional technical interviews at scale. The platform employs trained interview engineers to evaluate candidates and provide comprehensive reports, helping organizations make informed hiring decisions. Karat’s emphasis on scalability and accuracy in candidate evaluations serves as a benchmark for modern recruitment systems.

Building on these strategies, RecruitRight wants to provide a comprehensive platform that combines asynchronous video interviews, AI-driven personality analysis, and thorough expert reviews. Through the introduction of sophisticated automation, performance evaluation, and scalable solutions catered to the requirements of businesses in a competitive hiring market, this project seeks to push the limits of current technologies.

# CHAPTER 3

# LITERATURE REVIEW & RELATED WORK

# Literature Review

|  |  |
| --- | --- |
| [1] | G. O. Narendra and S. Hashwanth, "Named Entity Recognition based Resume Parser," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT),* 2022. |
| [2] | M. Mehboob, S. Ali, S. ul Islam and S. Ali, "Evaluating Automatic CV Shortlisting Tool for Job Recruitment Based on Machine Learning Techniques," *2022 Mohammad Ali Jinnah University International Conference on Computing (MAJICC),* 2022. |
| [3] | Intervue, "Intervue: Simplifying Remote Hiring," [Online]. Available: https://www.intervue.io. [Accessed 21 Jan 2025]. |
| [4] | Karat, "Karat: Interviewing Cloud for Hiring Engineers," [Online]. Available: https://karat.com. [Accessed 21 Jan 2025]. |

## Related Work

Several platforms and studies have laid the foundation for enhancing technical recruitment through innovative technologies. Two prominent platforms, **Intervue** and **Karat**, have significantly influenced modern hiring practices.

**Intervue** provides a collaborative interview environment tailored for technical interviews. It enables organizations to conduct live assessments, evaluate candidates' problem-solving abilities, and foster seamless interaction between interviewers and candidates. The platform’s focus on efficient evaluation has proven to reduce time-to-hire and enhance the accuracy of technical skill assessments.

**Karat** specializes in scaling technical interviews by leveraging a network of trained interview engineers. It ensures consistency and accuracy in candidate evaluations through standardized interview formats and detailed reporting. Karat’s emphasis on data-driven insights and scalability has demonstrated its effectiveness for large organizations requiring efficient technical hiring solutions.

In addition to these platforms, research on automated recruitment systems has highlighted the importance of incorporating AI and machine learning for unbiased candidate evaluations. Studies on natural language processing for resume screening and facial gesture analysis for confidence assessment have further showcased the potential of advanced technologies in recruitment.

The **RecruitRight: Precision Hiring with AI Insight** project builds upon these existing works by integrating features such as AI-powered personality assessments, asynchronous video interviews, and scalable micro services architecture. Unlike its predecessors, RecruitRight emphasizes a holistic approach, combining multiple automated tools into a single platform to streamline hiring processes and deliver precise insights for improved decision-making.

## Gap Analysis

There is currently no single platform in the technical hiring environment that uses Named Entity Recognition (NER), one of the more sophisticated natural language processing (NLP) approaches, to automate resume analysis. While existing platforms like **Karat** and **Intervue** seek to standardize assessments and interviews, they are unable to automate the extraction of vital candidate data like credentials, experience, and abilities.

Because hiring managers and recruiters must manually review resumes, which takes time and is prone to inconsistencies, this gap leads to inefficiencies in the recruitment process. During the candidate evaluation process, biases and missed insights may arise from the lack of automated, data-driven resume analysis.

**How RecruitRight Addresses the Gap:**

The platform can now automatically extract important candidate data including skills, experience, certifications, and qualifications thanks to **RecruitRight's** introduction of NER as a basic feature. **RecruitRight** guarantees uniformity and equity in candidate assessments by decreasing manual labor and enhancing resume analysis precision.

In conclusion, **RecruitRight** distinguishes itself by incorporating cutting-edge natural language processing (NLP) methods into the hiring process, resolving the shortcomings of competing platforms, and expediting the hiring process.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Intervue** | **Incruiter** | **Karat** | **RecruitRight** |
| **NER for Resume Analysis** | Not available | Not available | Not available | Integrated to scan and analyze resumes automatically. |
| **Interview scheduling** | Basic real-time scheduling | Advanced scheduling with interviewer assignment. | Basic scheduling with limited flexibility. | Candidate-driven slot selection dynamic scheduling. |
| **Scalability for high users** | Moderate | High scalability | High scalability | Designed for dynamic scalability to handle peak loads. |
| **Candidate feedback integration** | Not available | Available | Not available | Feedback included in generated reports for candidates’ review. |
| **Target audience** | Tech companies focusing on coding assessments | Wide range of industries | Large enterprises focusing on high-volume technical hiring. | Comprehensive hiring needs for diverse industries. |
| **Customization options** | Minimal customization | Provides tailed assessments. | Standardized evaluation methods. | Customizable workflows for job posting, resume screening, and interview processes. |
| **Multi-industry support** | Focused on tech | Supports multiple industries. | Primarily enterprise-level clients. | Designed for diverse industry requirements. |

Table 3: Gap Analysis

# CHAPTER 4

**PROJECT DISCUSSION**

## Software Engineering Methodology:

For the development of this recruitment website, the **Agile Software Development Methodology** was adopted. Agile methodology promotes iterative development, where requirements and solutions evolve through collaboration between cross-functional teams. The flexibility of Agile allowed us to frequently adapt to changing client requirements and add new features like candidate matching, interview scheduling, and employer dashboards. Regular sprint meetings ensured constant feedback, timely updates, and efficient progress tracking.

## Project Methodology

The overall project was structured using the **Incremental Model**, where the system was built and improved upon in pieces. Each functional component—such as the job seeker module, employer module, and admin panel—was developed and tested incrementally. This approach allowed for early deployment of core features, which provided users with immediate value while additional modules were being developed and integrated.

## Phases of Project

The project was executed in the following phases:

* **Requirement Gathering:**

Conducted competitor research to finalize system requirements, including job posting, resume parsing, user authentication, and AI-powered job matching.

* **System Design:**

Designed the system architecture, database schema (Firebase Real time Database). Frontend-backend flow and APIs were also planned.

Diagram is on the Next Page.

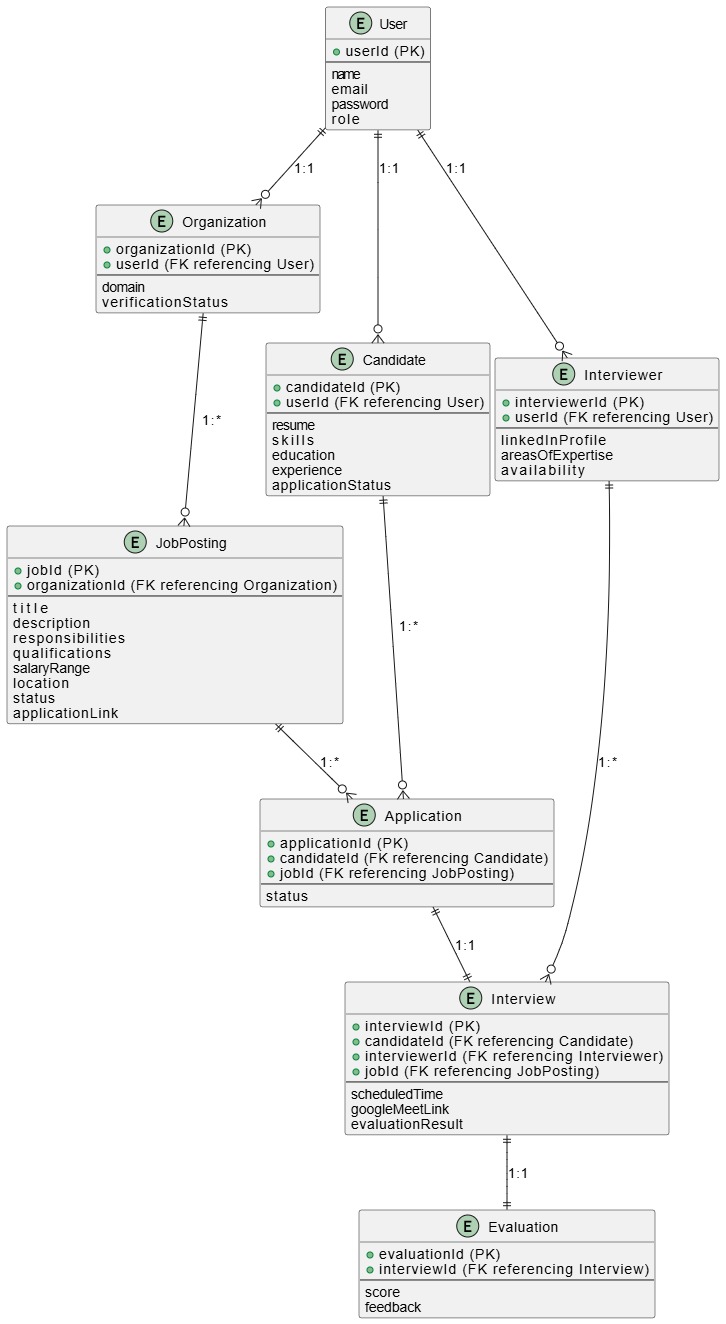


Figure 4.1 ER Diagram

* **Development:**

Implemented frontend pages using HTML, CSS, JavaScript, and Python. Backend logic included Firebase integration, login/signup handling, resume parsing using spaCy, and job matching algorithms.

* **Testing & Debugging:**

Carried out unit testing for each module, followed by integration testing and usability testing. Bugs related to user sessions, API response handling, and form validation were resolved.

* **Deployment:**

The system was hosted on a live server, and Google Firebase was used for user authentication, database storage, and cloud functions. Google Calendar API was integrated for interview scheduling.

## Software/Tools that Used in Project

* **Frontend:**
  + HTML5, CSS3, JavaScript
* **Backend:**
  + Python (for AI/ML resume parser)
* **Database and Cloud Services:**
  + Firebase Realtime Database
  + Firebase Authentication
  + Google Cloud Functions
  + Google Calendar API
* **AI Tools & Libraries:**
  + spaCy (for Named Entity Recognition)
  + scikit-learn (for model training and evaluation)
* **Others:**
  + Visual Studio Code / Cursor (Code Editor)
  + Postman (API Testing)
  + Git & GitHub (Version Control)

## Hardware that Used in Project

* **Development Machine:**
  + Processor: Intel Core i7 4th gen
  + RAM: 8 GB
  + Storage: SSD 256 GB
  + OS: Windows 10

# Chapter 5

**IMPLEMENTATION**

## 1. Proposed System Architecture/Design

The recruitment website follows a **modular, client-server architecture** integrating both frontend and backend components. The system consists of four core user roles: **Job Seeker/Candidate, Employer/Organization**, **Interviewer** and **Administrator**. It uses **Firebase** as the backend for real-time data operations and authentication, while the **AI model (Python-based spaCy)** processes resumes for intelligent job matching.

**Architecture Overview:**

* **Client Side (Frontend):**
  + HTML, CSS, JavaScript
  + Responsive UI for user interactions
* **Server Side (Backend):**
  + Python for server-side logic
  + Firebase Cloud Functions for backend automation
  + Python AI model (resume parsing and scoring)
* **Database:**
  + Firebase Realtime Database (NoSQL)
* **APIs:**
  + Google Calendar API for interview scheduling

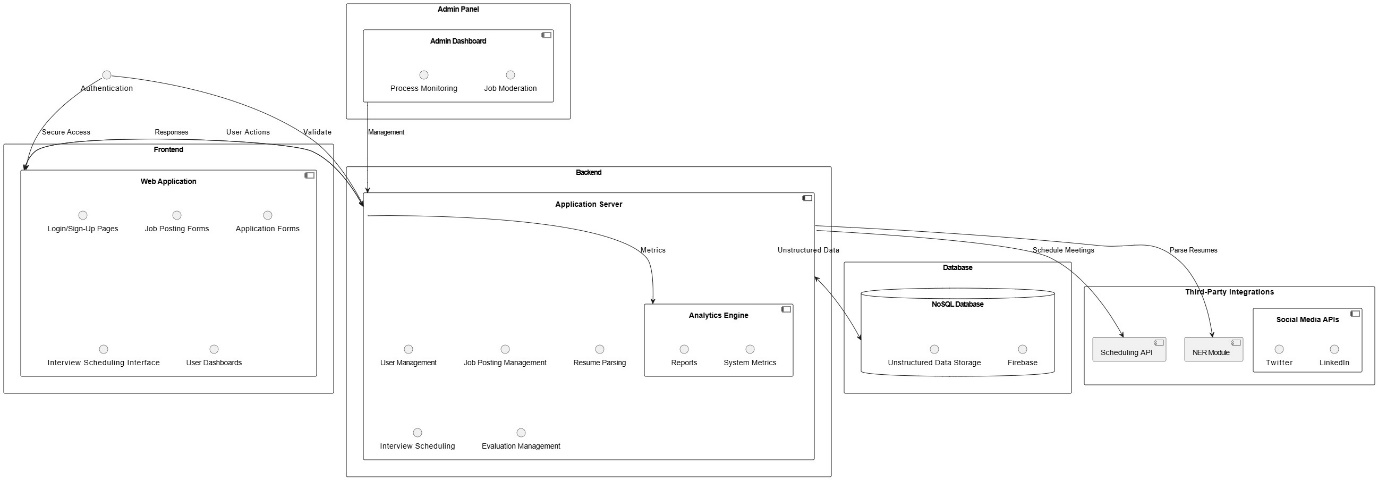


Figure 5.1 System Architecture Diagram

* **Use-Case Model**

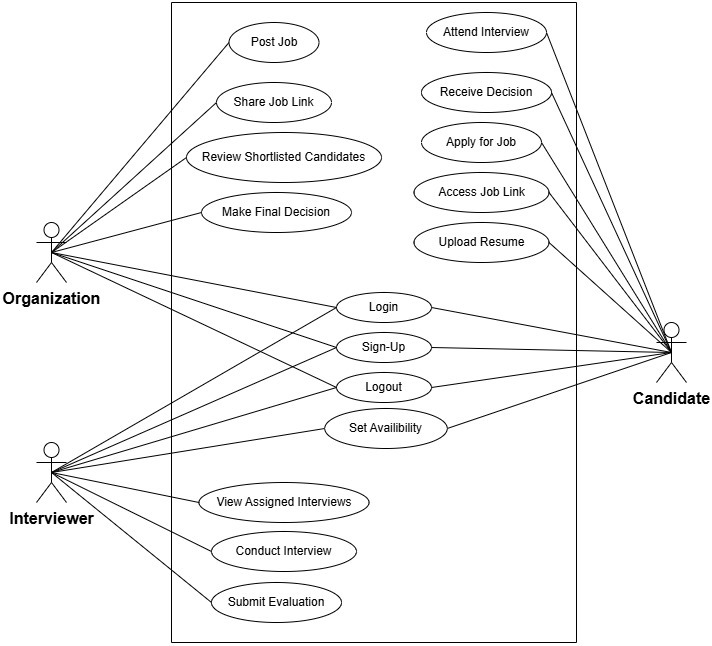
****

Figure 5.2 Use Case Diagram

## 2. Functional Specifications

The functional requirements define the core operations that the system must perform:

* User Registration and Login (Manual & Google OAuth)
* Role-based Dashboard (Job Seeker, Employer, Admin)
* Resume Upload and Parsing (PDF/Doc)
* Job Search and Filter Functionality
* AI-based Resume Matching with Job Requirements
* Interview Scheduling and Calendar Integration
* Notifications via Google Calendar

## 3. Non-Functional Specifications

These specifications define the quality attributes of the system:

* **Performance:**  
  Fast resume parsing and response under 2 seconds for 90% of requests
* **Scalability:**  
  Firebase and cloud-based architecture allows easy scalability
* **Security:**
  + Firebase Authentication for secure login
  + Input validation and session management
  + Role-based access control
* **Usability:**  
  User-friendly design and mobile-responsive interface
* **Maintainability:**  
  Modular code structure and API-based architecture ease updates and debugging
* **Availability:**  
  In future will be hosted on a live server with >99% uptime via cloud services
* **Dynamic Behaviour (Sequence Diagrams)**

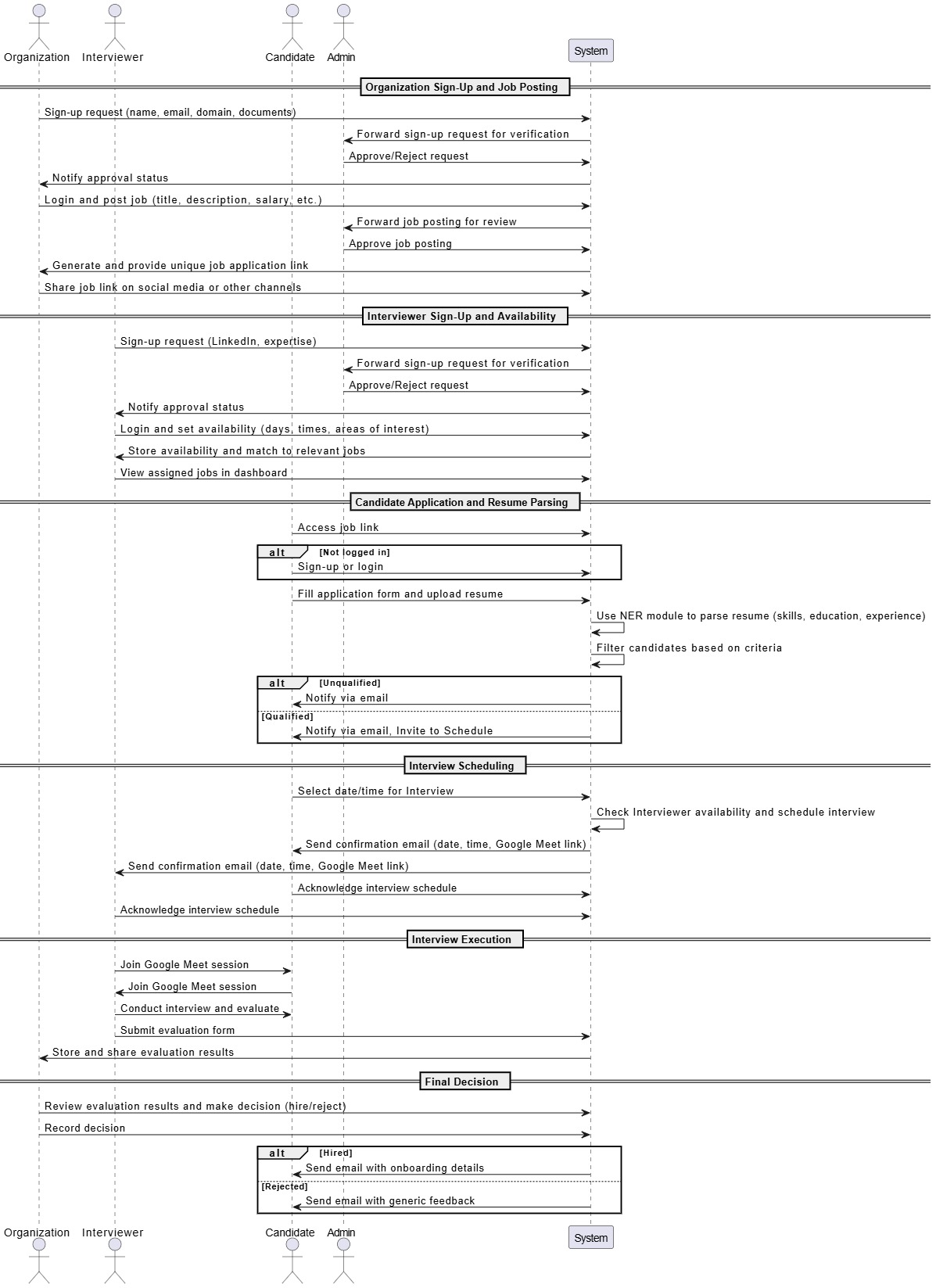


Figure 5.3 Sequence Diagram

* **Class Diagram**

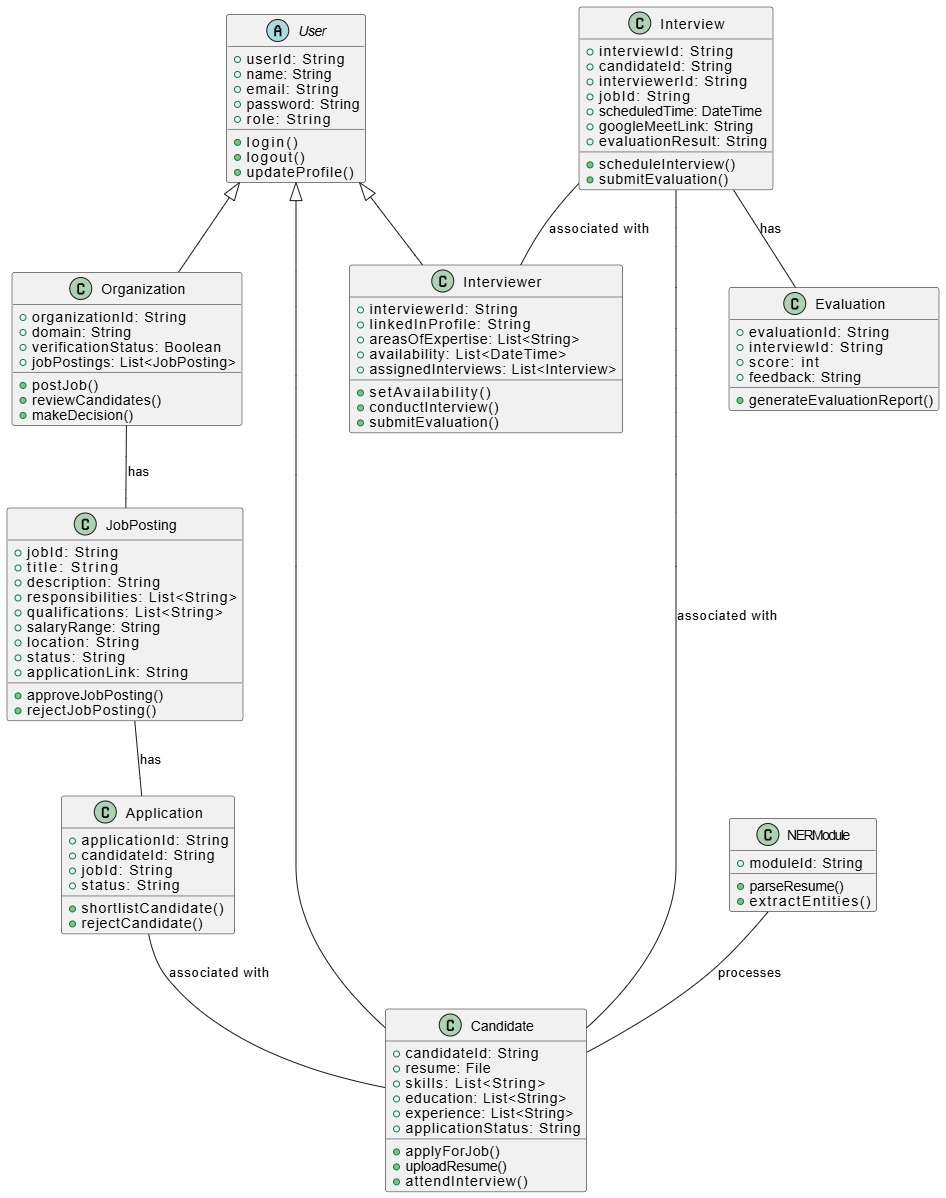


Figure 5.4 Class Diagram

## 4. Testing

Testing was conducted in multiple phases to ensure the system’s reliability and functionality. The following types of testing were performed:

* **Unit Testing:**
  + Each module (e.g., login, resume parser, job match score) was tested individually.
* **Integration Testing:**
  + Verified data flow and logic between frontend forms, backend Python logic, Firebase, and the AI model.
* **System Testing:**
  + End-to-end test of full workflows including user registration, job application, and interview scheduling.
* **User Acceptance Testing (UAT):**
  + Conducted with real users to validate usability and correctness.

## 5. Purpose of Testing

The purpose of testing was to:

* Identify and eliminate bugs or logic errors
* Ensure the system meets the functional and non-functional requirements
* Confirm that user journeys (e.g., applying for jobs, scheduling interviews) are seamless
* Validate the AI model’s matching accuracy and resume parsing reliability
* Ensure mobile responsiveness and real-time data sync via Firebase.

**6. Software Test Plan**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Expected Output** |
| TC01 | Login & Sign-Up | Account successfully created / user logged in |
| TC02 | Job Application Flow | |  | | --- | |  |   Application submitted / confirmation message shown |
| TC03 | Status-Change Notification | Resume parsed; entities extracted |
| TC04 | Interview Scheduling | Available interview slots displayed |
| TC05 | Interview Reminder Notification | Notification sent to candidate and interviewer |
| TC06 | Interview “Start” Button | |  | | --- | |  |   Google Meet/Zoom link opened & event created on calendar |
| TC07 | Feedback Submission | Feedback form submitted successfully |
| TC08 | Security & Access Control | Unauthorized access blocked / Admin sees full user list |
| TC09 | Bulk Email Dispatch | Emails sent to all selected recipients |
| TC10 | Logout & Session Termination | User session ended / redirected to login page |

Table 4: Software Test Plan

# Chapter 6

# EXPERIMENTAL EVALUATIONS & RESULTS

## 1. Evaluation Testbed

To evaluate the effectiveness and performance of the recruitment platform, a controlled test environment was set up with the following components:

* **Development Machine Specs:**
  + Processor: Intel Core i5 10th Gen
  + RAM: 8 GB
  + OS: Windows 10 + Ubuntu (for Python model)
  + Browsers: Chrome, Firefox (for cross-browser testing)
* **Testing Tools:**
  + Postman (for API testing)
  + Browser Developer Tools (for frontend validation)
  + Firebase Console (for backend monitoring)
  + Visual Studio Code (development and debugging)
  + Python environment for spaCy model testing
* **Dataset Used:**
  + 100 anonymized resumes in PDF format
  + 50 job descriptions from different domains
  + Custom evaluation scripts to test AI model accuracy in parsing and matching resumes
* **Users Involved:**
  + 5 Employers
  + 10 Job Seekers
  + 2 Admins
  + Test sessions conducted to collect feedback and measure system performance

## 2. Results and Discussion

#### Resume Parsing Accuracy

The custom-trained **spaCy Named Entity Recognition (NER)** model achieved high accuracy in extracting essential resume fields:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Precision** | **Recall** | **F1 Score** |
| Name | 97% | 96% | 96.5% |
| Education | 93% | 91% | 92% |
| Experience | 90% | 88% | 89% |
| Skills | 95% | 94% | 94.5% |
| Job Preference | 89% | 86% | 87.5% |

Table 5: Resume Parsing Accuracy

This shows the model is reliable in extracting key information, which is essential for accurate job-resume matching.

#### AI Job Matching Results

The ATS (Applicant Tracking System) logic scored resumes against job descriptions. Out of 50 jobs and 100 resumes, the system was able to:

* Recommend top 5 matched candidates with >85% accuracy in 82% of cases.
* Show meaningful skill alignment and gaps to help employers filter applicants faster.

#### User Feedback

Feedback collected via forms and interviews revealed the following:

|  |  |
| --- | --- |
| **Feature** | **User Satisfaction** |
| UI/UX Design | 9.1/10 |
| Resume Upload & Parsing | 8.8/10 |
| Job Matching Logic | 8.5/10 |
| Interview Scheduling Feature | 9.2/10 |
| Notifications | 9.0/10 |

#### Table 6: User Feedback

#### System Performance

* **Average Resume Parsing Time:** 1.8 seconds
* **Job Match Score Calculation:** < 1 second
* **Interview Scheduling API Response:** ~2 seconds
* **Notification Delivery via FCM:** < 1.5 seconds

#### Discussion

The results confirm that the proposed recruitment platform functions effectively and meets the project goals. The AI model reliably parses resumes and calculates meaningful match scores. Firebase-based backend ensures fast, real-time updates and data syncing. Minor limitations were observed in job preference extraction due to inconsistent formatting in some resumes. However, these can be improved through additional training data and normalization techniques.

# CHAPTER 7

# CONCLUSION AND DISCUSSION

# 1. Strength of this Project

The recruitment platform successfully achieves its core objective of simplifying the hiring process through AI and automation. Major strengths of this project include:

* **AI-Powered Resume Parsing:**

The integration of a custom-trained spaCy NER model enables accurate extraction of resume data and meaningful job-resume matching based on skills, experience, and job preferences.

* **Role-Based Functionalities:**

Separate dashboards for Job Seekers, Employers, and Admins allow each user to access relevant tools and information, improving the overall user experience.

* **Firebase Integration:**

The use of Firebase for authentication, real-time database operations, and cloud messaging provides scalability, real-time performance, and ease of deployment.

* **Interview Scheduling & Notifications:**

Google Calendar API integration streamlines the scheduling process, also ensures users are instantly notified.

* **User-Friendly UI:**

A clean and responsive frontend ensures accessibility across devices and browsers, enhancing usability.

* **Security and Data Handling:**

Secure login, session handling, and structured data storage ensure reliability and trustworthiness of the platform.

# 2. Limitations and Future Work

While the system performs well, a few limitations were observed:

#### Limitations:

* **Resume Format Dependency**

The AI parser may struggle with non-standard resume layouts or scanned images that lack proper text structure.

* **Limited Admin Features:**

Current admin functionality is basic and can be extended with analytics, activity logs, and more granular user controls.

* **Job Suggestion Logic**

Although effective, the job matching system can be enhanced by incorporating more complex natural language understanding and ranking algorithms.

* **No Live Chat or Interview Room:**

The system does not yet support live chat or in-platform interviews.

#### Future Work:

* Train a more advanced NLP model using transformers (e.g., BERT) for deeper resume-job matching.
* Implement a **chatbot assistant** to help users find jobs or post listings more efficiently.
* Add **in-app video interview capabilities**.
* Develop a **mobile application** for easier access on smartphones.
* Include **analytics dashboards** for employers and admins.

# 3. Reasons for Failure – If Any

The project did not experience complete failure in any aspect. However, there were some **initial development delays** and **integration challenges**:

* **Firebase and Python Integration Complexity:**

Connecting Firebase with Python-based AI scripts and ensuring smooth data flow required additional research and debugging.

* **Session Management Issues:**

During the early testing phase, proper session handling for users was not implemented, causing temporary login issues.

* **Redirect Errors in Oath (Google Login):**

Incorrect redirect URIs in Firebase caused authentication errors during deployment, which were later resolved.

Despite these hurdles, each issue was addressed in time, and no feature had to be dropped due to failure. The project is considered a success in both implementation and user satisfaction.

# REFERENCES

[1] M. Grinberg, Flask Web Development: Developing Web Applications with Python. 2014. [Online]. Available: https://dl.acm.org/citation.cfm?id=2621997

[2] L. Moroney, “The Firebase Realtime database,” in Apress eBooks, 2017, pp. 51–71. doi: 10.1007/978-1-4842-2943-9\_3.

[3] “Git - reference.” https://git-scm.com/docs

[4] “GitHub.com help documentation,” GitHub Docs. https://docs.github.com/

[5] “GitHub Actions documentation - GitHub Docs,” GitHub Docs, Jul. 01, 2001. https://docs.github.com/en/actions

[6] “RFC 6749: The OAUTH 2.0 Authorization Framework,” IETF Datatracker. https://tools.ietf.org/html/rfc6749

[7] Auth, “JSON Web Token Introduction - JWt.io,” JSON Web Tokens - jwt.io, Nov. 30, 2024. https://jwt.io/introduction/

[8] “Making network requests with JavaScript - Learn web development | MDN,” MDN Web Docs, May 22, 2025. https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX

[9] “Cross-Origin Resource Sharing (CORS) - HTTP | MDN,” MDN Web Docs, Jul. 04, 2025. https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

[10] “Python 3.13 documentation,” Python Documentation. https://docs.python.org/3/

[11] “What is REST?: REST API Tutorial,” REST API Tutorial, Apr. 01, 2025. https://restfulapi.net/

[12] “OpenAPI Specification - Version 3.1.0 | Swagger.” https://swagger.io/specification/

[13] “JSON.” https://www.json.org/

[14] “Django documentation | Django documentation,” Django Project. https://docs.djangoproject.com/

[15] “Newest questions,” Stack Overflow. https://stackoverflow.com/

[16] “IEEE Xplore.” https://ieeexplore.ieee.org/

[17] “3.11.12 documentation.” https://docs.python.org/3.11/

# APPENDICES

**List of Appendices**

A1a. Project Proposal and Vision Document

A1b. Copy of Proposal Evaluation Comments by Jury

A2. Requirement Specifications

A3. Design Specifications

A4. Other Technical Details

Test cases

UI/UX Details

Coding Standards

Project Policy

A5. Flyer & Poster Design

A6. Copy of Evaluation Comments

Copy of Evaluation Comments by Jury for Project – I End Semester Evaluation

A7. Meetings’ Minutes

A8. Document Change Record

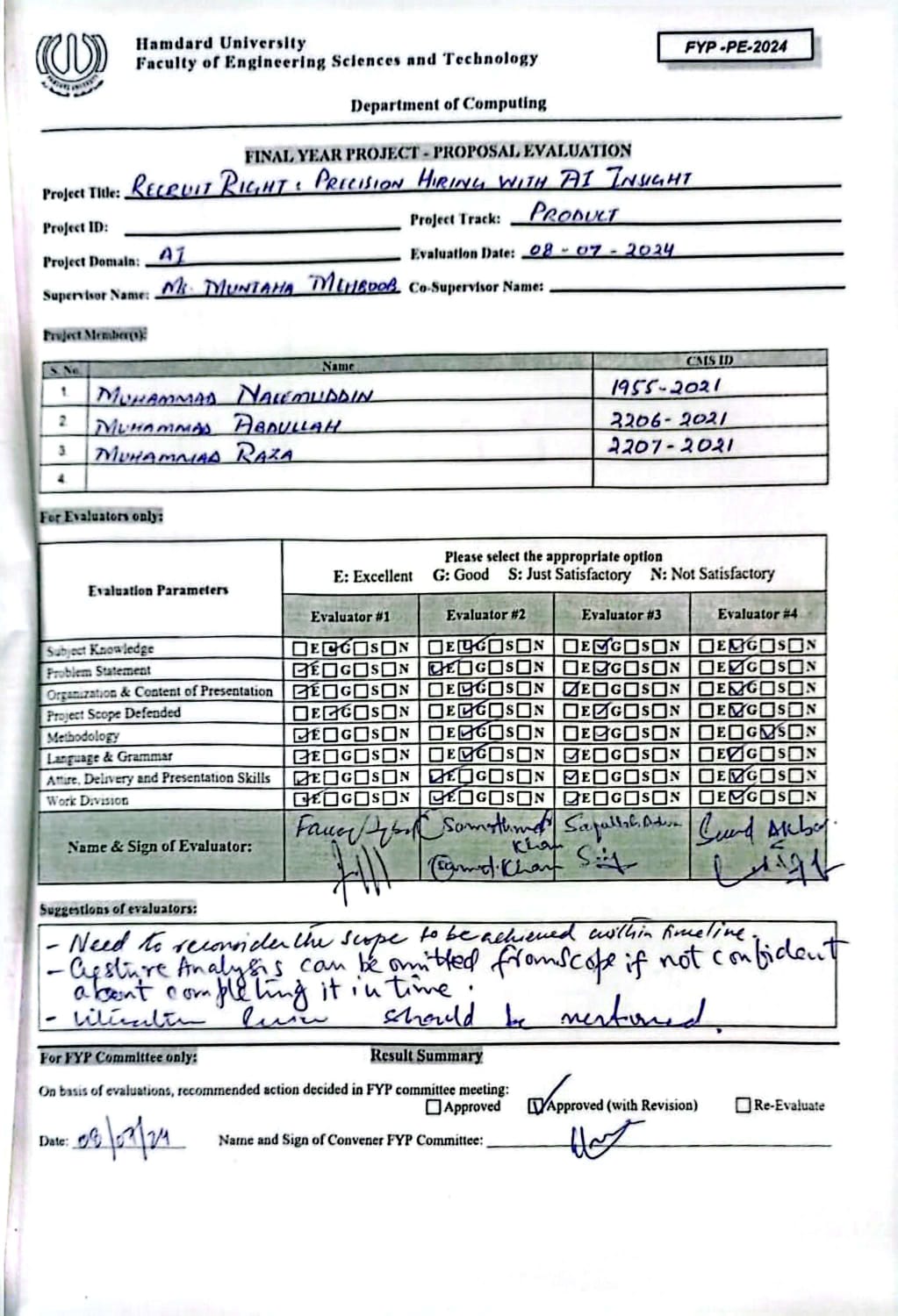
A9. Project Progress

# A1A. PROJECT PROPOSAL AND VISION DOCUMENT

Below is the link of the Project Proposal Document:

[https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/1.ProjectProposal](https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/1.%20Project%20Proposal)

# A1B. COPY OF PROPOSAL EVALUATION COMMENTS BY JURY



# A2. REQUIREMENT SPECIFICATIONS

Below is the link of the Software Requirement Specification Document:

[https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/2.SRSDocument](https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/2.%20SRS%20Document)

# A3. DESIGN SPECIFICATIONS

Below is the link of the Software Design Specification Document:

[https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/3.SDSDocument](https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/3.%20SDS%20Document)

# A4. OTHER TECHNICAL DETAIL DOCUMENTS

## Test Cases Document

**Software Test Plan and Test Cases**

**Test Plan:**

The following table presents the detailed test plan, including the test schedule for each functional component of Recruit Right platform.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Description** | **Tested by** | **Start date** | **End Date** |
| 1 | Login & SignUp | Naeem | 01-Jul-2025 | 01-Jul-2025 |
| 2 | Job Application Flow | Naeem | 01-Jul-2025 | 01-Jul-2025 |
| 3 | Status-Change Notification | Naeem | 02-Jul-2025 | 02-Jul-2025 |
| 4 | Interview Scheduling | Naeem | 02-Jul-2025 | 02-Jul-2025 |
| 5 | Interview Reminder Notifi. | Naeem | 03-Jul-2025 | 03-Jul-2025 |
| 6 | Interview “Start” Button | Naeem | 03-Jul-2025 | 03-Jul-2025 |
| 7 | Feedback Submi | Naeem | 03-Jul-2025 | 03-Jul-2025 |
| 8 | Security & Access Control | Naeem | 03-Jul-2025 | 03-Jul-2025 |
| 9 | Bulk Email Dispatch | Naeem | 03-Jul-2025 | 03-Jul-2025 |
| 10 | Logout&Session Termination | Naeem | 03-Jul-2025 | 03-Jul-2025 |

Table 7: Summary of Test Plan

**Test Case 1 – Login & Sign-Up  
Project Name:** Recruit Right  
**Module Name:** Login & Sign-Up  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_001  
**Test Engineer:** Naeem  
**Test Case Description:** Testing user login, registration, and session handling.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Open Login Page | – | Login page displays correctly | Login page loaded | Pass |
| 2 | Enter email & password | Valid credentials | Redirects to Candidate Dashboard | Redirected successfully | Pass |
| 3 | Open Sign-up Page | – | Sign-up form displays | Sign-up form loaded | Pass |
| 4 | Register new user | Name, email, password | Account created & redirected to dashboard | Account created | Pass |
| 5 | Logout user | – | Session ends & returns to login page | Logout successful | Pass |

Table 8: Test Plan - 1

## Test Case 2 – Job Application Flow

**Project Name:** Recruit Right  
**Module Name:** Job Application Flow  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_002  
**Test Engineer:** Naeem  
**Test Case Description:** Verify candidates can view job listings and apply.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Navigate to Jobs page | – | List of active jobs appears | 10 active jobs shown | Pass |
| 2 | Click job card | Job ID 123 | Job detail page/modal opens | Job 123 details shown | Pass |
| 3 | Click “Apply” | Resume, Cover Letter | Application stored with status = “pending” | DB row inserted, status pending | Pass |
| 4 | Open Applications tab | – | Newly applied job listed with pending status | Card shows “pending” badge | Pass |

Table 9: Test Plan - 2

## Test Case 3 – Status-Change Notification

**Project Name:** Recruit Right  
**Module Name:** Status-Change Notification  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_003  
**Test Engineer:** Naeem  
**Test Case Description:** Confirm dashboard + email alerts on application status change.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Recruiter sets status → “short-listed” | Application ID 77 | Notification service triggered | Webhook logged 200 OK | Pass |
| 2 | Candidate opens dashboard | – | Banner/toast shows status update | Green toast “Short-listed” displayed | Pass |
| 3 | Check candidate email | Candidate email | Email with same message received | Mail received within 30 s | Pass |

Table 10: Test Plan - 3

## Test Case 4 – Interview Scheduling

**Project Name:** Recruit Right  
**Module Name:** Interview Scheduling  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_004  
**Test Engineer:** Naeem  
**Test Case Description:** Ensure recruiter can schedule an interview and candidate sees it.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Open “Schedule Interview” form | Job ID 123, Candidate ID 45 | Form loads | Form rendered | Pass |
| 2 | Save date/time & medium | 09-Jul-2025 10:00 AM, Zoom link | Session stored with status “scheduled” | Session ID 88 created | Pass |
| 3 | Candidate → Upcoming Interviews | – | New interview card shows correct details | Card shows 09-Jul 10:00 AM | Pass |

Table 11: Test Plan - 4

## Test Case 5 – Interview Reminder (30 min)

**Project Name:** Recruit Right  
**Module Name:** Interview Reminder Notification  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_005  
**Test Engineer:** Naeem  
**Test Case Description:** Validate that reminder emails/alerts go out 24 h before interview.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Simulate T-30 min scheduler | Session ID 88 | Reminder job triggers | Reminder job ran | Pass |
| 2 | Check candidate email | – | “Interview tomorrow” email received | Email timestamp correct | Pass |
| 3 | Check dashboard bell | – | Reminder badge/notification present | Bell shows 1 unread | Pass |

Table 12: Test Plan - 5

## Test Case 6 – “Start” Button Timing

**Project Name:** Recruit Right  
**Module Name:** Interview “Start” Button  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_006  
**Test Engineer:** Naeem  
**Test Case Description:** “Start” button activates only at interview start time.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Open dashboard 30 min early | – | “Start” button disabled/greyed | Greyed-out button | Pass |
| 2 | Reach exact start time | – | Button becomes active | Turned blue, clickable | Pass |
| 3 | Click “Start” | – | Redirects to Zoom meeting link | Redirected to Zoom | Pass |

Table 13: Test Plan - 6

## Test Case 7 – Recruiter Feedback Submission

**Project Name:** Recruit Right  
**Module Name:** Feedback Submission  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_007  
**Test Engineer:** Naeem  
**Test Case Description:** Verify recruiter can submit feedback; candidate cannot view it.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Open feedback form | Session ID 88 | Form fields load | Form ready | Pass |
| 2 | Submit rating & notes | Score 8, “Strong React skills” | Feedback saved to DB | Row added in Feedback table | Pass |
| 3 | Candidate refreshes apps | – | Status shows “Interview Completed”, feedback hidden | Feedback not visible to candidate | Pass |

Table 15: Test Plan - 7

## Test Case 8 – Security & Access Control

**Project Name:** Recruit Right  
**Module Name:** Security & Access Control  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_008  
**Test Engineer:** Naeem  
**Test Case Description:** Prevent cross-role and cross-organization data access.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Candidate hits /recruiter/dashboard | Candidate token | HTTP 403 or redirect to login | 403 JSON error | Pass |
| 2 | Recruiter fetches other org’s applicant data | Org mismatch | API returns 401/403 | API 403 | Pass |
| 3 | Expire JWT and reload | – | User forced to log in again | Redirect to /login | Pass |

Table 16: Test Plan - 8

## Test Case 9 – Bulk Email Queue

**Project Name:** Recruit Right  
**Module Name:** Bulk Email Dispatch  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_009  
**Test Engineer:** Naeem  
**Test Case Description:** Ensure notification API handles 50 rejection emails in parallel.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Bulk-reject 50 apps | Status → “rejected” | 50 emails queued | Queue shows 50 jobs | Pass |
| 2 | Monitor worker | – | All emails marked “sent” | Mail log confirms 50 sent | Pass |
| 3 | Spot-check inboxes | 3 random candidates | Emails received with correct template | All three emails correct | Pass |

Table 17: Test Plan - 9

## 

## Test Case 10 – Logout & Session Termination

**Project Name:** Recruit Right  
**Module Name:** Logout & Session Termination  
**Date:** 05-Jul-2025  
**Test Case Id:** TC\_010  
**Test Engineer:** Naeem  
**Test Case Description:** Verify logout completely destroys session tokens and cookies.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Steps | Input Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Click “Logout” | – | Redirects to login page | Redirected successfully | Pass |
| 2 | Press browser Back | – | Dashboard remains inaccessible (redirect) | Redirects back to login | Pass |
| 3 | Inspect cookies | – | Auth tokens/cookies removed | Storage empty | Pass |

Table 18: Test Plan - 10

Below is the link for Test Cases Document:

[https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/8.SoftwareTestPlan&TestReport](https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/tree/main/Documents/8.%20Software%20Test%20Plan%20%26%20Test%20Report)

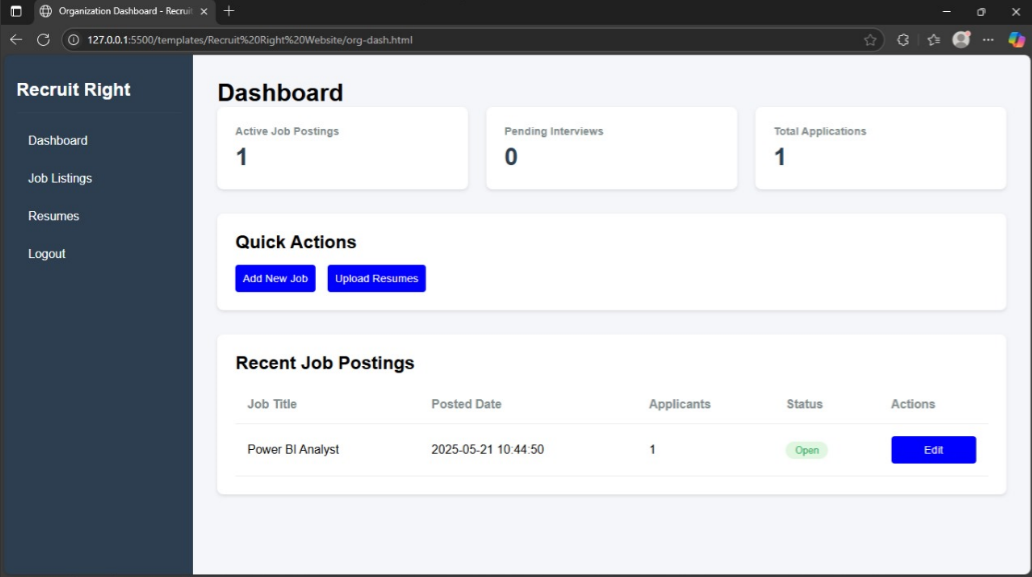
## UI/UX Detail Document

Below is the link of the UI/UX Detail Document:

Link, <https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/blob/main/Documents/extra_documents/ui-ux_detailed_document.pdf>

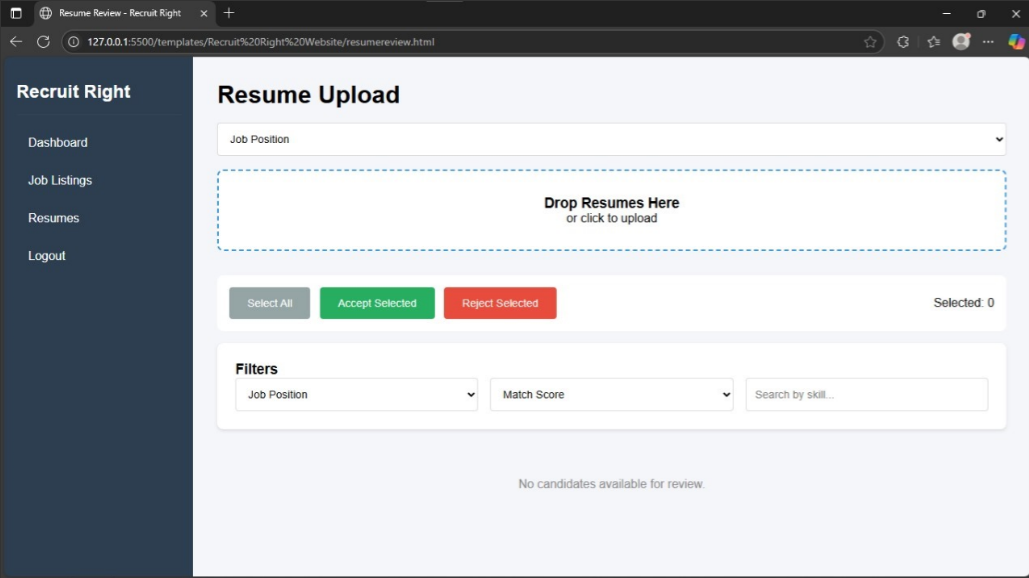
Here are some samples of the Project.

**Organization Interface:**

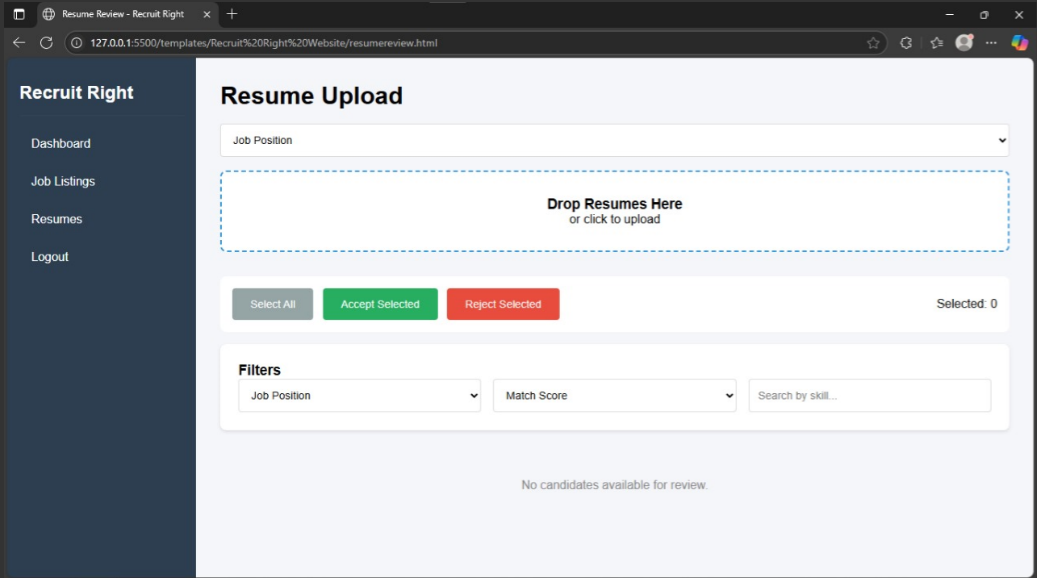


Description:

The dashboard provides an overview for an organization user managing job postings and applications.

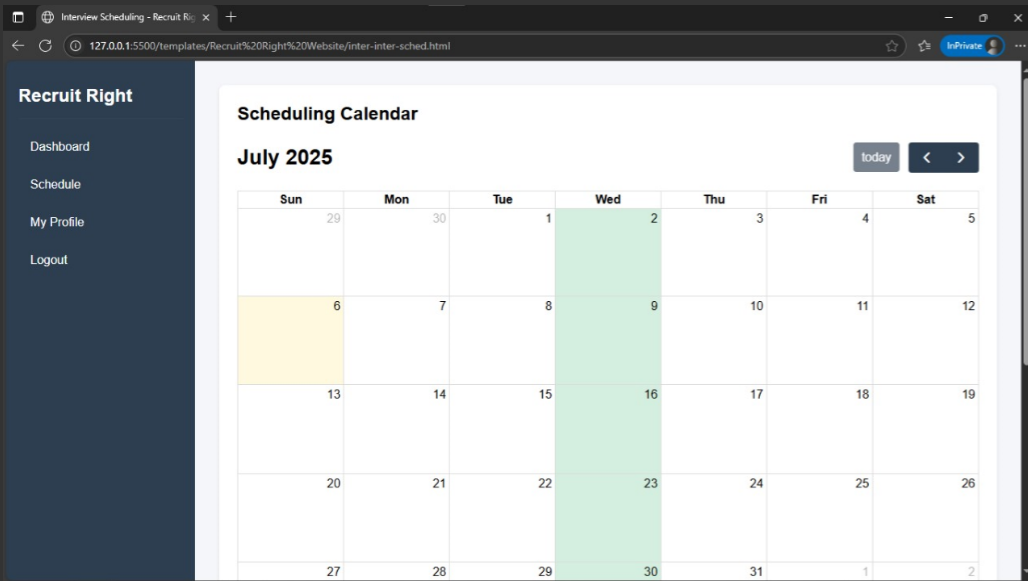


This is the organization's Resume Upload page where users can upload, filter, and manage candidate resumes for specific job positions.



Organization Resume

Interviewer Interface:



Interviewer Scheduling

## Coding Standards Document

Below is the link of the Coding Standards Document:

<https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/blob/main/Documents/extra_documents/coding_standards_document.pdf>

## Project Policy Document

Below is the link of the Project Policy Document:

<https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/blob/main/Documents/extra_documents/project_policy_document.pdf>

## User Manual Document

Below is the link of the User Manual Document:

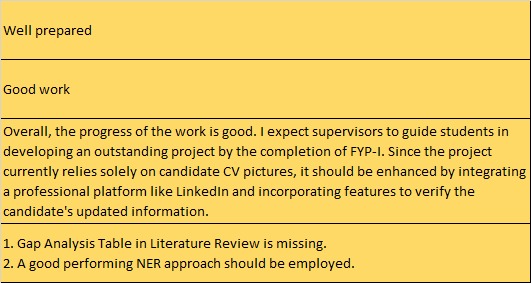
<https://github.com/bluevit/Recruit-Right-Precision-Hiring-with-AI-Insight/blob/main/Documents/extra_documents/user_manual_document.pdf>

# A5. FLYER & POSTER DESIGN



# A6. COPY OF EVALUATION COMMENTS

## COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION

  
  
Table 19: FYP-I Evaluation, Comments of Jury

# A7. MEETINGS’ MINUTES & Sign-Off Sheet

Below is the link to all the minutes of meetings for FYP-I & FYP-II

<https://drive.google.com/drive/folders/1oW36MQwTiqBpVSJ0MTyuVi0depGYpvjT?usp=sharing>

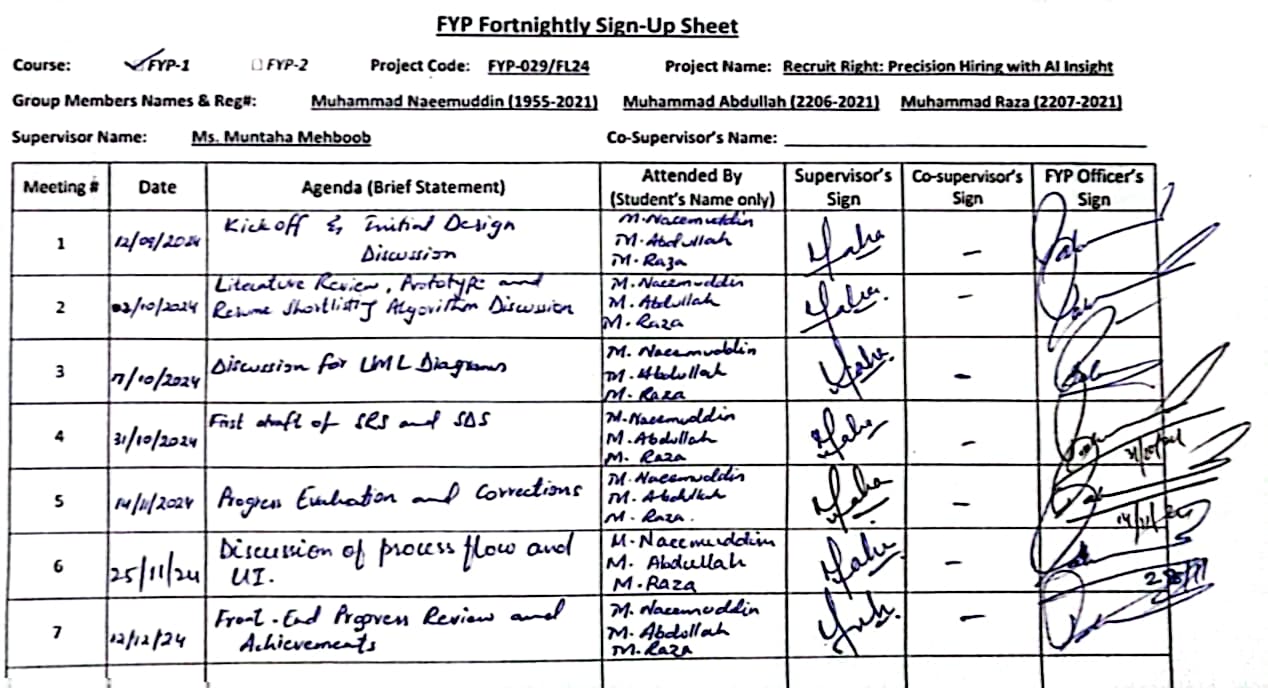
**A8. DOCUMENT CHANGE RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| Jan 18th, 2025 | 1.0 | Muhammad Abdullah | First Draft for FYP-I (First 3 Chapters) |
| Jun 31st, 2025 | 2.0 | Muhammad Abdullah | Remaining Chapters |
| July 5th, 2025 | 2.1 | Muhammad Abdullah | Addition of Test Case |

Table20: Doc Change Record

**A9. PROJECT PROGRESS**

**FYP-I:**



**FYP-II:**