Hamdard University

Department of Computing

Final Year Project

Recruit Right: Precision Hiring with AI Insight

FYP-029/FL24

Software Design Specifications

Submitted by

Muhammad Naeemuddin 1955-2021

Muhammad Abdullah 2206-2021

Muhammad Raze 2207-2021

Supervisor

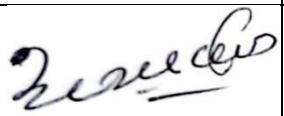
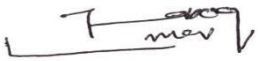
Dr. Umer Farooq

Fall 2024

Document Sign off Sheet

1.1.1 Document Information

Project Title	Recruit Right: Precision Hiring with AI Insight
Project Code	FYP-029 / FL24
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-029 / FL24-SDS
Document Status	~~Draft~~/ Final
Author(s)	Muhammad Naeemuddin, Muhammad Abdullah, Muhammad Raza
Approver(s)	Dr. Umer Farooq
Issue Date	17th January 2025

Name	Role	Signature	Date
Muhammad Naeemuddin	Team Lead		17-01-25
Muhammad Abdullah	Team Member 2		17-01-25
Muhammad Raza	Team Member 3		17-01-25
Dr. Umer Farooq	Supervisor		17-01-25
-	Co-Supervisor		
Faheem Ahmed	Project Coordinator		

Revision History

Date	Version	Description	Author
12-01-2025	1.0	Initial Draft	Muhammad Raza
17-01-2025	2.0	Final Draft	Muhammad Naeemuddin
July 6, 2025	2.1	Final Draft	Muhammad Abdullah

Definition of Terms, Acronyms, and Abbreviations

Term	Description
NLP	Natural Language Processing
AI	Artificial Intelligence
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
MongoDB	Mongo Database
SSL	Secure Sockets Layers
TLS	Transport Layers Security
HTTP	Hyper Text Transfer Protocol
API	Application Programming Interface
RESTful	Representational State Transfer
SDK	Software Development Kit

2 Table of Contents

1 Introduction

This document outlines the comprehensive design, development, and deployment process for a technical interview outsourcing platform aimed at revolutionizing the hiring process for organizations. By automating key aspects of technical interviews, the platform offers a streamlined and scalable solution for conducting personality assessments, live coding sessions, and asynchronous video interviews, providing standardized evaluations and structured feedback from experts.

1.1 Purpose of Document

The purpose of this document is to provide a detailed overview of the project, including its objectives, scope, methodology, and design considerations. This document serves as a guide for the development team, stakeholders, and other personnel involved in the project to understand the structure and functionality of the proposed system.

The intended design methodology for this project is **Object-Oriented Design (OOD)**, ensuring modularity, scalability, and ease of maintenance.

1.2 Intended Audience

This document is intended for the following groups:

- **Development Team:** Developers working on the front-end, back-end, and database systems.
- **Project Stakeholders:** Business decision-makers, recruiters, and delivery managers interested in streamlining their technical hiring processes.
- **Testing and QA Team:** Personnel responsible for ensuring the system meets functional and non-functional requirements.
- **System Administrators:** Professionals managing the deployment, scalability, and maintenance of the platform.

1.3 Document Convention

The document follows these conventions for consistency and readability:

- **Font:** Calibri
- **Font Size:** 12 for body text, 14 for headings, and 16 for section titles.
- **Formatting:**
  - Bold for section headers and key terms.
  - Italics for emphasis.
  - Numbered lists for sequential information.
  - Bulleted lists for non-sequential details.

1.4 Project Overview

The technical interview outsourcing platform automates and standardizes hiring processes by enabling organizations to:

- Conduct personality assessments, live coding sessions, and video interviews remotely.
- Use NLP-powered resume screening for job-relevant keyword detection.
- Provide candidates with real-time, collaborative tools for coding and interview preparation.
- Deliver structured feedback and generate detailed reports to help organizations make informed decisions.

The system leverages a **micro services architecture** for scalability and **Object-Oriented Programming principles** to ensure robust modular design. The project includes an intuitive user interface connected to a reliable back-end system, integrating third party APIs such as Google Calendar and Google Forms for seamless operations.

1.5 Scope

The scope of the project includes:

- **Features:**
  - Customizable interview plans created in collaboration with expert Interview Engineers.
  - A scalable network of skilled Interview Engineers for efficient candidate evaluations.
  - Reliable assessment processes using advanced algorithms and standardized evaluation rubrics.
- **Inclusions:**
  - Development of a user-friendly front-end.
  - Integration with a back-end system to handle user data and job-specific workflows.
  - Testing the system with dummy data to ensure functionality and usability.
  - Deployment on Firebase for hosting and scalability.
- **Exclusions:**
  - Integration with existing HR systems or proprietary algorithms outside the scope of public APIs.
  - Deep exploration of communication media beyond integrating Google Calender.

By focusing on these defined objectives, the project aims to deliver a functional and impactful solution that simplifies hiring for organizations while maintaining clarity and feasibility.

2 Design Considerations

This section lays the foundation for the system design by addressing critical issues, assumptions, dependencies, risks, and strategies to mitigate potential challenges during the development process.

2.1 Assumptions and Dependencies

The project operates under the following assumptions and dependencies:

- **Assumptions:**
    - Users will have access to stable internet connections to conduct video interviews and use real-time coding features.
    - The platform will be accessed via modern web browsers that support HTML, CSS, and JavaScript.
    - The backend services, including databases and third-party APIs, will be reliable and available throughout the project.
    - The platform's functionality will be tested on dummy data before actual deployment, assuming the availability of necessary test cases and datasets.
    - The core technologies (HTML, CSS, JavaScript, Node.js, MongoDB, and Python) will be compatible with the system requirements and will meet the scalability needs for initial deployment.

- **Dependencies:**
    - The project relies on third-party APIs for certain functionalities, including resume analysis and interview evaluation, which may impact the development timeline based on availability and integration complexity.
    - The project's success is dependent on the availability of expert interviewers to validate the candidate evaluations and provide feedback.
    - The deployment of the platform on Firebase is dependent on the chosen hosting plan's scalability and support for the required infrastructure.

Timely feedback and approval from the project supervisor and stakeholders are essential to ensure the project stays on track and meets its deadlines.

2.2 Risks and Volatile Areas

The following risks and potential sources of volatility could impact the design and development process:

1. **Technology Changes:**
    - **Risk:** Dependency on third-party APIs like Google Calender or Google Forms could lead to disruptions if API updates or service outages occur.
    - **Mitigation:** Use API versioning to maintain compatibility and have backup services integrated to minimize impact.
2. **Performance Bottlenecks:**
    - **Risk:** Handling high user traffic, especially during peak hiring seasons, could result in performance degradation.
    - **Mitigation:** Employ a micro services architecture with load balancers and scalable Firebase functions to manage traffic efficiently.
3. **Data Privacy and Security:**
    - **Risk:** Storing sensitive candidate and organization data exposes the platform to potential breaches.
    - **Mitigation:** Implement robust encryption mechanisms (SSL/TLS) and adhere to GDPR or other relevant data protection standards.
4. **User Adoption:**
    - **Risk:** Poor user experience or lack of trust in automation tools may hinder adoption by key stakeholders.
    - **Mitigation:** Conduct usability testing and maintain clear documentation to enhance user confidence in the platform.
5. **Evolving Requirements:**
    - **Risk:** Late-stage changes in feature requests or design specifications could disrupt timelines.
    - **Mitigation:** Use agile development practices to accommodate incremental changes without compromising project deadlines.
6. **Cheating Detection Mechanisms:**
    - **Risk:** The accuracy and reliability of algorithms for detecting cheating during coding sessions are crucial and may need refinement.
    - **Mitigation:** Regularly update the cheating detection module with new patterns and test extensively on diverse datasets.

By proactively addressing these considerations, the design will remain flexible, robust, and responsive to anticipated challenges, ensuring the platform's long-term success.

3 System Architecture

This section outlines the high-level organization of the system, focusing on the functional decomposition of its components, their interactions, and the underlying architectural strategies. The system is designed to streamline technical interviews by automating key processes and integrating with third-party tools for enhanced efficiency and scalability.

3.1 System Level Architecture

The system is decomposed into several key subsystems and components:

1. **Subsystems and Components:**
    - **User Management Subsystem:** Handles sign-ups, logins, and profile setups for all actors (Organizations, Candidates, Interview Engineers).
    - **Job Management Subsystem:** Manages job postings, resume uploads, and NER-based resume filtering.
    - **Interview Scheduling Subsystem:** Coordinates available slots for candidates and interviewers and generates meeting links.
    - **Assessment and Report Generation Subsystem:** Integrates with Google Forms for assessments and generates structured reports.
    - **Third-Party Integration Subsystem:** Interfaces with external systems like Google Calender and Google Forms.
2. **Relationships Between Elements:**
    - Data flows from User Management to other subsystems for personalized experiences.
    - Job postings and resumes feed into the Interview Scheduling and Assessment Subsystems.
    - Reports are generated and shared with Organizations and Candidates after processing.
3. **Interfaces to External Systems:**
    - Google Calender for interview coordination.
    - Google Forms API for assessments and rubric evaluations.
4. **Global Design Strategies:**
    - **Error Handling:** Implement centralized logging and retry mechanisms for third-party API calls.
    - **Scalability:** Use cloud services (Firebase) for dynamic scaling based on load.

3.2 Software Architecture

The software architecture is based on a layered design to separate concerns and enhance maintainability. Key layers include:

1. **User Interface Layer:**
   - Provides a web-based interface for all actors to perform their roles.
   - Facilitates real-time interactions, such as scheduling and joining interviews.

2. **Middle Tier:**
   - Contains business logic, including resume filtering, interview scheduling, and report generation.
   - Acts as an intermediary between the User Interface Layer and the Data Access Layer.
3. **Data Access Layer:**
   - Manages data storage and retrieval for user profiles, job details, resumes, interview schedules, and reports.
   - Utilizes Firebase as the primary database, ensuring scalability and real-time updates.
4. **Integration Layer (Optional):**
   - Handles API calls to third-party services like Google Calender and Google Forms.

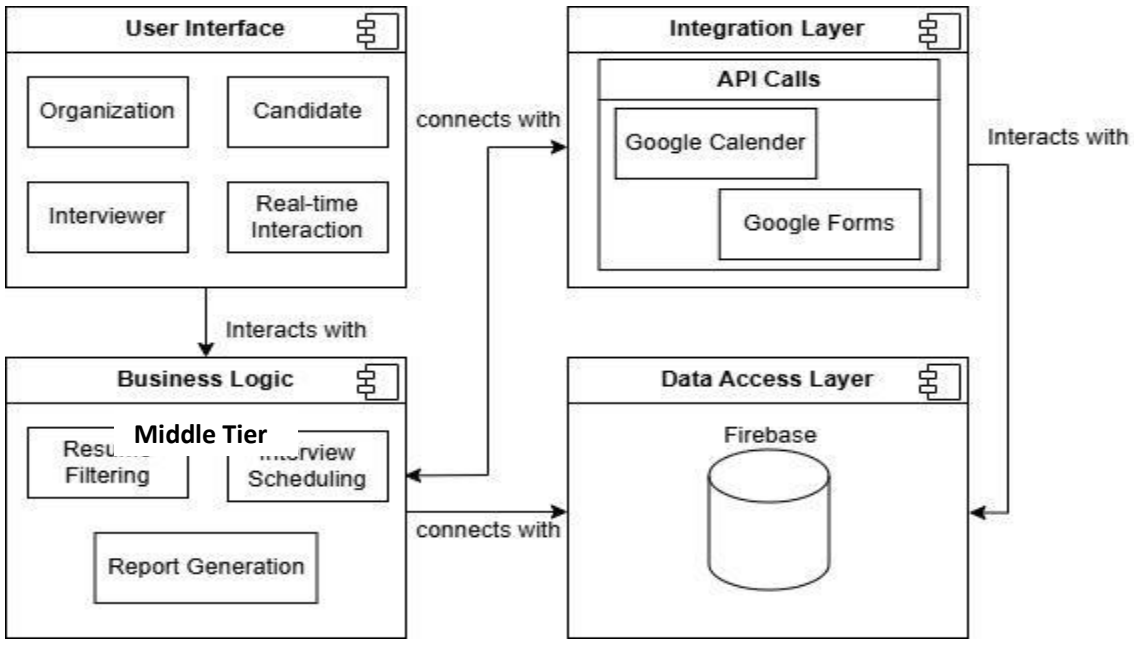


Figure 1: Software Architecture for Proposed System

4 Design Strategy

The design strategies for the system focus on achieving scalability, modularity, and maintainability while ensuring seamless integration with third-party tools. Below are the critical considerations and decisions impacting the overall organization of the system:

Future System Extension or Enhancement

- **Strategy**: The system will adopt a modular architecture using micro services to isolate functionality into discrete components (e.g., resume parsing, interview scheduling, and report generation).
    - **Reasoning**: This ensures that individual modules can be updated or replaced without affecting the entire system.
    - **Trade-off**: Initial development may take longer due to the overhead of designing modular services, but the long-term benefit of scalability outweighs this drawback.

System Reuse

- **Strategy**: Common functionalities, such as authentication, notification services, and reporting, will be implemented as reusable components or libraries.
    - **Reasoning**: This reduces redundant code and ensures consistency across different system features.
    - **Trade-off**: Some upfront effort is required to design generic, reusable components.

User Interface Paradigms

- **Strategy**: A responsive, intuitive UI will be built using **Material Design principles** to ensure cross-platform compatibility and ease of use.
    - **Reasoning**: A clean, user-friendly interface reduces onboarding time for users and improves the overall experience.
    - **Trade-off**: Emphasis on responsiveness may increase complexity in frontend development, requiring additional testing.

Data Management (Storage, Distribution, Persistence)

- **Strategy**: A NoSQL database (e.g., Firebase) will be used to store semi-structured data such as resumes, user profiles, and reports. Real-time database capabilities will handle interview scheduling and progress tracking.
    - **Reasoning**: NoSQL databases offer flexibility in handling varied data formats and support horizontal scaling to manage growing data volumes.
    - **Trade-off**: Complex querying may require additional effort to optimize, as NoSQL systems are not as query-efficient as traditional relational databases.

5 Detailed System Design

5.1 Design Class Diagram

This is the initial design of the class diagram, representing the key entities and their relationships in the system. It serves as a foundational structure and can be modified or extended as per evolving requirements.

5.1.1 Organization

- **Attributes:**
  - org_id: Unique identifier for the organization.
  - org_name: Name of the organization.
  - email: Contact email of the organization.
  - password: Encrypted password for secure access.
  - job_postings: A list of job postings created by the organization.
- **Methods:**
  - signUp(): Registers the organization in the system, storing relevant details.
  - addJobDetails(): Allows the organization to create and manage job postings, which are stored in the job_postings attribute.
- **Interactions:**
  - An Organization can create multiple JobPostings.
  - It receives reports for completed interviews through the Report class.

5.1.2 JobPosting

- **Attributes:**
  - job_id: Unique identifier for the job posting.
  - job_title: Title of the job being offered.
  - job_description: Detailed description of the job role.
  - keywords: A list of keywords extracted from the job description, used for resume filtering.
  - resumes: A list of resumes submitted for the job posting.
  - status: Status of the job posting (e.g., "Open," "Closed").
- **Methods:**
  - addJobPosting(): Adds a new job posting under an organization.
  - scanResumes(): Uses NER to filter resumes based on the keywords.
- **Interactions:**
  - A JobPosting belongs to one Organization.
  - It contains multiple Resumes, which are scanned to find suitable candidates.
  - It is linked to InterviewSessions, which involve candidates applying for the job.

5.1.3 Resume

- **Attributes:**

  - candidate_id: Identifier of the candidate who submitted the resume.
  - content: The actual content of the resume.
  - ner_score: A score derived from the NER tool based on how well the resume matches the job posting.
- **Methods:**
  - evaluateKeywords(): Analyzes the resume content for matching keywords and assigns a score.
- **Interactions:**
  - Each Resume is associated with one Candidate.
  - It is part of a JobPosting's list of submitted resumes.

5.1.4 Candidate

- **Attributes:**
  - candidate_id: Unique identifier for the candidate.
  - name: Name of the candidate.
  - email: Candidate's email address.
  - password: Encrypted password for secure access.
  - profile: A detailed profile of the candidate (e.g., experience, skills).
  - available_slots: A list of time slots when the candidate is available for an interview.
- **Methods:**
  - signUp(): Registers the candidate in the system.
  - createProfile(): Enables the candidate to add details like experience and skills.
  - selectSlots(): Allows the candidate to choose their availability for interviews.
- **Interactions:**
  - Each Candidate has one Profile.
  - Candidates submit resumes to JobPostings and are matched based on the NER score.
  - Candidates are involved in InterviewSessions and receive reports post-interview.

5.1.5 Profile

- **Attributes:**
  - experience: Candidate's work experience.
  - skills: A list of the candidate's skills.
- **Interactions:**
  - The Profile belongs to a Candidate and is used during resume filtering and interview processes.

5.1.6 Interviewer

- **Attributes:**
  - interviewer_id: Unique identifier for the interviewer.
  - name: Name of the interviewer.
  - email: Contact email for the interviewer.
  - password: Encrypted password for secure access.
  - specialty: The interviewer's area of expertise.
  - assigned_interviews: A list of interviews the interviewer is scheduled to conduct.
- **Methods:**
  - signUp(): Registers the interviewer in the system.
  - selectCandidate(): Enables the interviewer to pick a candidate for an interview.
- **Interactions:**
  - Interviewers conduct multiple InterviewSessions, which are linked to Candidates and JobPostings.

5.1.7 InterviewSession

- **Attributes:**
  - session_id: Unique identifier for the interview session.
  - candidate: Reference to the Candidate being interviewed.
  - interviewer: Reference to the Interviewer conducting the session.
  - job_posting: Reference to the JobPosting the candidate applied for.
  - meeting_link: Link to the Google Meet/Zoom interview session.
  - scheduled_at: Scheduled date and time of the interview.
  - assessment_form_link: Link to the Google Form for assessment.
  - report: The report generated after the interview.
- **Methods:**
  - schedule(): Schedules an interview session for a Candidate and Interviewer.
  - conductInterview(): Tracks the interview process.
  - generateReport(): Generates a detailed report based on the interviewer's assessment.
- **Interactions:**
  - Links Candidates, Interviewers, and JobPostings for interviews.
  - Generates a Report that is shared with the Organization and Candidate.

5.1.8 Report

- **Attributes:**
  - report_id: Unique identifier for the report.
  - content: Detailed assessment of the interview.
  - organization: Reference to the Organization that posted the job.
  - candidate: Reference to the Candidate who was interviewed.

- **Methods:**
  - generate(): Creates the report based on the Google Form assessment.
  - send(): Shares the report with the Organization and Candidate.
- **Interactions:**
  - Each Report is associated with one InterviewSession.
  - It is sent to the Organization and Candidate for further evaluation.

5.1.9 Key Interactions Between Classes

1. **Organization and JobPosting:**
   - Organizations create multiple JobPostings.
   - JobPostings contain Resumes submitted by Candidates.
2. **Candidate and Profile:**
   - Each Candidate has one Profile, which stores their skills and experience.
3. **JobPosting and Resume:**
   - JobPostings contain a list of Resumes.
   - Resumes are filtered using NER to match job requirements.
4. **Candidate, Interviewer, and InterviewSession:**
   - Interviewers select available Candidates for InterviewSessions.
   - Each InterviewSession involves one Candidate, one Interviewer, and a JobPosting.
5. **InterviewSession and Report:**
   - Each InterviewSession generates a Report, which is shared with the Organization and Candidate.

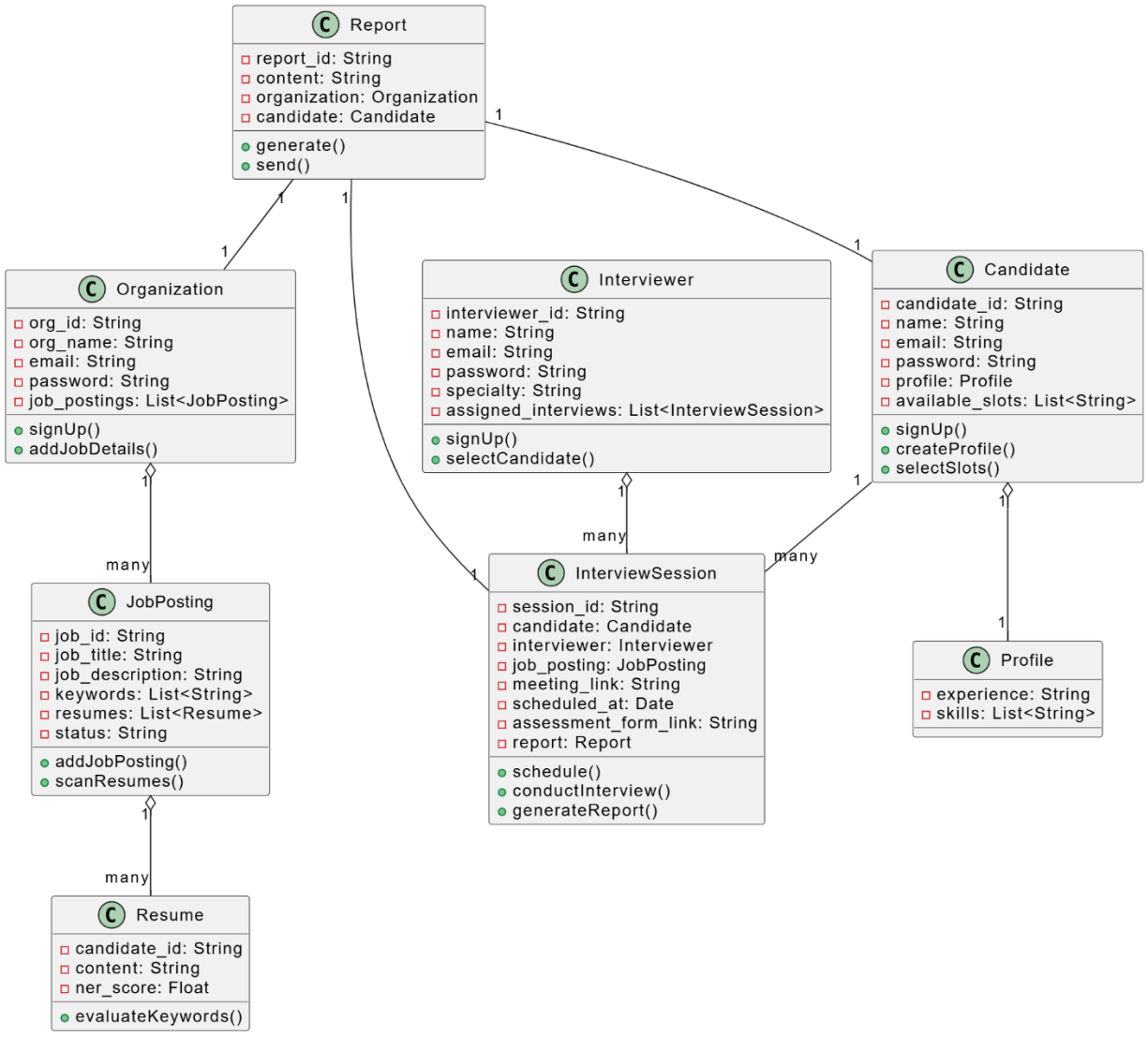
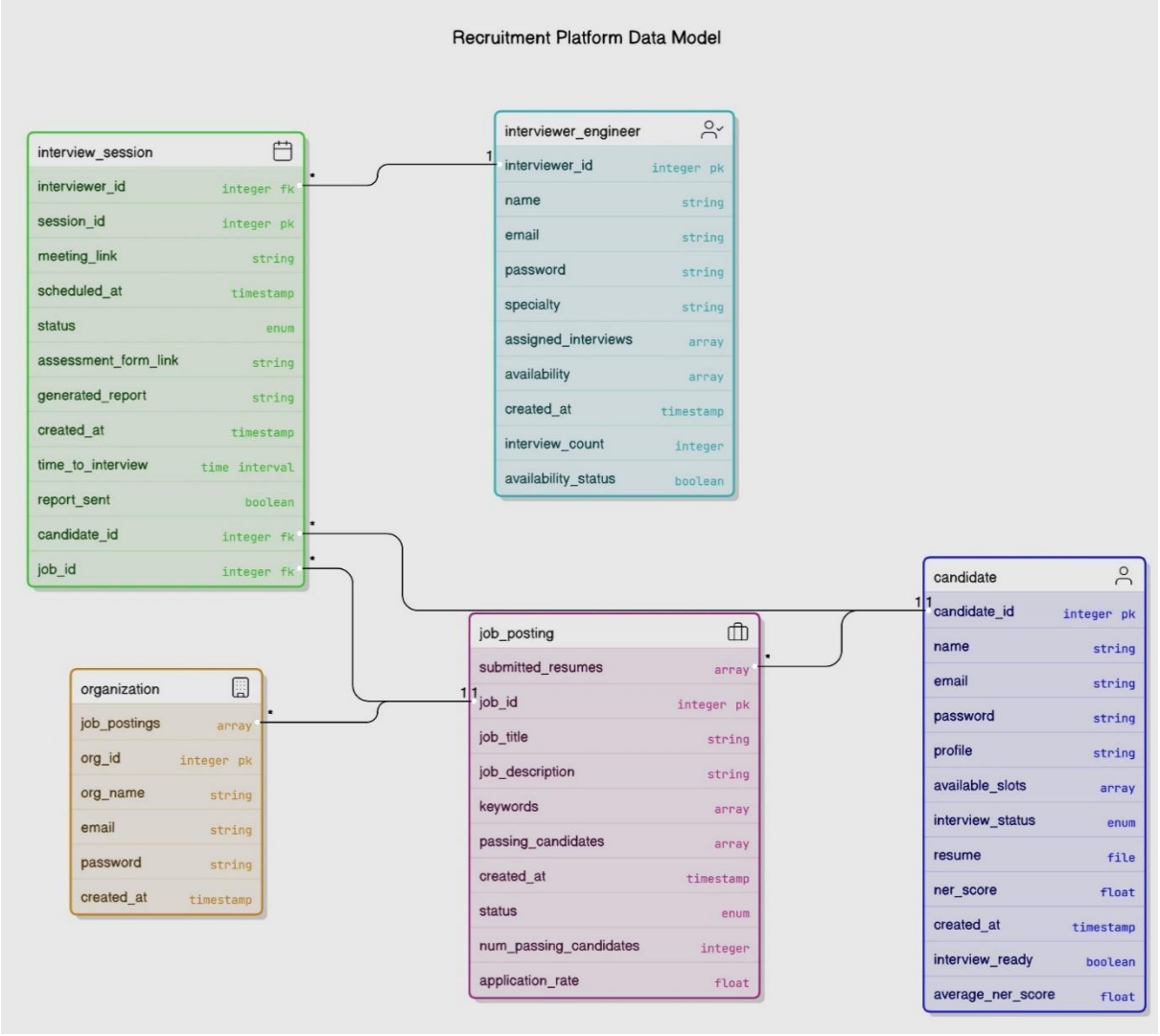


Figure 2: Initial Class Diagram for Recruit Right

5.2 Database Design

This is the initial design of the database, outlining the core entities and their attributes. It serves as a foundational structure and can be modified or extended as needed to accommodate additional requirements or system enhancements.

5.2.1 ER Diagram



5.2.2 Data Dictionary: Organization

5.2.2.1 Data 1: Organization

Name	Organization
Alias	Company, Employer
Where-used/how-used	Used in job postings as the creator (input). Used in authentication and for organization-level data retrieval (store).
Content description	org_id = org_name + email + password + job_postings {created_at}

5.2.2.2 Column Details

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
org_id	Unique identifier for the organization	Integer	10	No	Auto-Increment	PK
org_name	Name of the organization	String	255	No	NULL	
email	Contact email address	String	255	No	NULL	
password	Encrypted password	String	255	No	NULL	
job_postings	Array of job postings associated with the organization	JSON Array	N/A	Yes	NULL	
created_at	Timestamp of when the organization signed up	Timestamp	N/A	No	CURRENT_TIMESTAMP	

5.2.3 Data Dictionary: Job Posting

5.2.3.1 Data 2: Job Posting

Name	Job Posting
Alias	Listing, Vacancy
Where-used/how-used	Used as input for resume submission (input). Used to filter candidates via NER and pass results (store).
Content description	job_id = job_title + job_description + keywords + submitted_resumes + passing_candidates + {created_at, status}

5.2.3.2 Column Details

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
job_id	Unique identifier for the job posting	Integer	10	No	Auto-Increment	PK
job_title	Title of the job	String	255	No	NULL	
job_description	Detailed description of the job	String	1000	No	NULL	
keywords	Extracted keywords for NER matching	JSON Array	N/A	Yes	NULL	
submitted_resumes	Array of candidate resumes submitted	JSON Array	N/A	Yes	NULL	
passing_candidates	Array of candidate IDs that passed NER filtering	JSON Array	N/A	Yes	NULL	
created_at	Timestamp of job creation	Timestamp	N/A	No	CURRENT_TIMESTAMP	
status	Current status (e.g., "Open," "Closed")	Enum	N/A	No	"Open"	

5.2.4 Data Dictionary: Candidate

5.2.4.1 Data 3: Candidate

Name	Candidate
Alias	Applicant, Job Seeker
Where-used/how-used	Used as input for resume submission (input). Used to track interview status (store).
Content description	candidate_id = name + email + password + profile + available_slots + {resume, ner_score, created_at}

5.2.4.2 Column Details

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
candidate_id	Unique identifier for the candidate	Integer	10	No	Auto-Increment	PK
name	Candidate's name	String	255	No	NULL	
email	Candidate's email address	String	255	No	NULL	
password	Encrypted password	String	255	No	NULL	
profile	Detailed profile of the candidate (experience, skills)	JSON Object	N/A	Yes	NULL	
available_slots	Array of available time slots	JSON Array	N/A	Yes	NULL	
resume	Uploaded resume	File (PDF)	N/A	Yes	NULL	
ner_score	Derived score from NER matching	Float	N/A	Yes	NULL	
created_at	Timestamp of sign-up	Timestamp	N/A	No	CURRENT_TIMESTAMP	

5.3 Application Design

This section provides a detailed overview of the application's design, focusing on the flow of interactions and state transitions within the system. The following diagrams and accompanying explanations outline the key processes and states that drive the application's functionality.

5.3.1 Sequence Diagram

1. **Actors**:
    - Organization
    - Candidate
    - Interviewer Engineer
2. **Modules/Components**:
    - **User Interface (UI)**: For user interaction.
    - **Authentication Module**: To verify credentials and grant access.
    - **Scheduling Module**: For booking interview slots and notifying users.
    - **Integration Module**: To connect with external tools like Google Meet/Zoom.
    - **Report Generation Module**: For creating and sharing interview reports.
    - **System**: Backend logic and session handling.

Main Processes (from left to right):

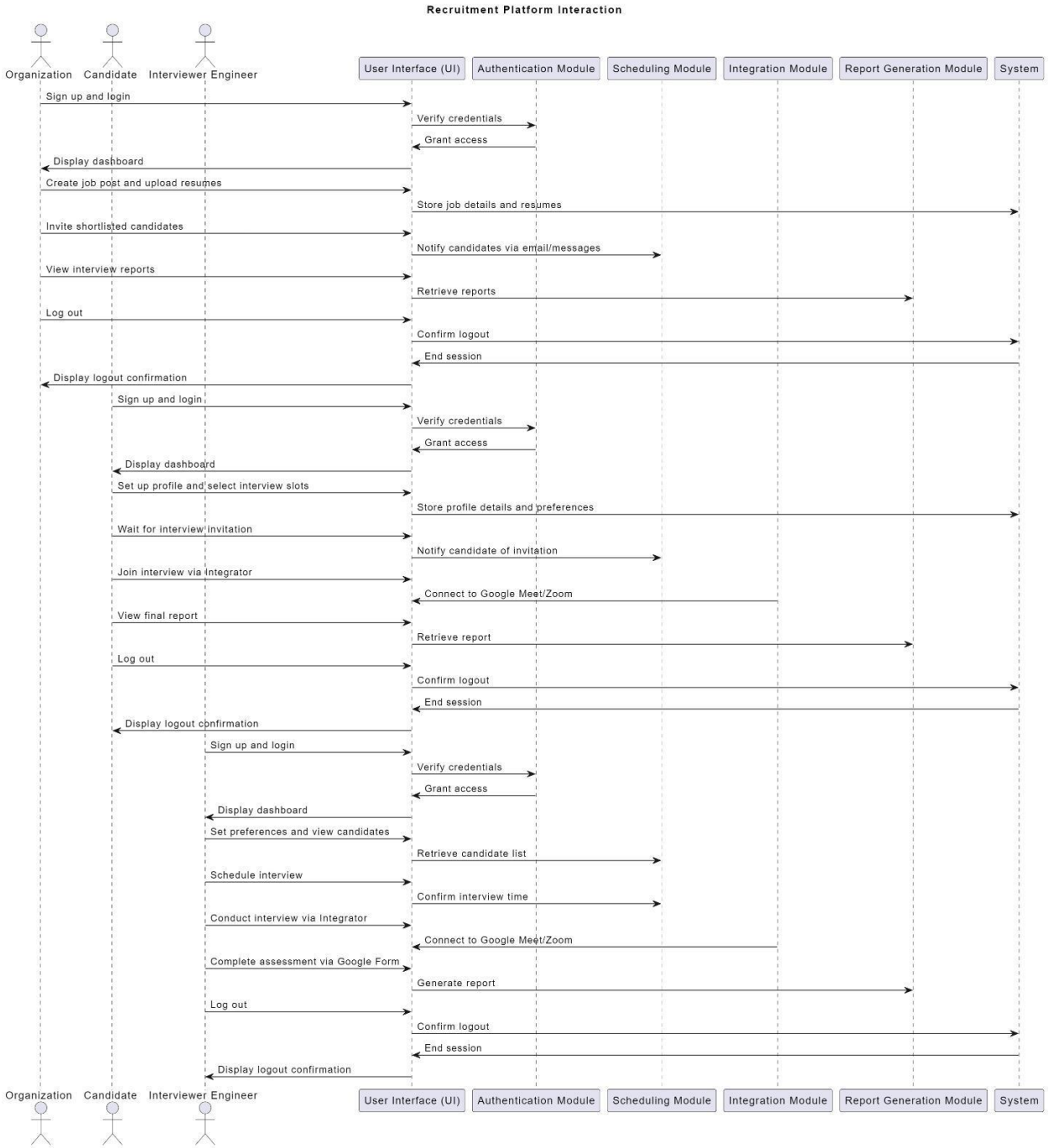
1. **Organization**:
    - Sign-up/login and dashboard display.
    - Job post creation and resume upload.
    - Invite shortlisted candidates (notification system triggered).
    - View interview reports.
    - Logout and confirmation.
2. **Candidate**:
    - Sign-up/login and dashboard display.
    - Profile setup and interview slot selection.
    - Wait for interview invitation (notification system triggers).
    - Join interview (via integration with Google Meet/Zoom).
    - View final report.
    - Logout and confirmation.
3. **Interviewer Engineer**:
    - Sign-up/login and dashboard display.
    - Preference setup and candidate selection.
    - Schedule interview and conduct it via integrator (Google Meet/Zoom).
    - Complete the assessment (Google Form).
    - Generate a report (triggered via system).
    - Logout and confirmation.

Flow of Communication:

- Each action triggers operations in respective modules:
    - Authentication Module verifies credentials for sign-in.
    - Scheduling Module handles interview slot booking and notifications.
    - Integration Module connects to external platforms like Google Meet/Zoom.
    - Report Generation Module retrieves/generates reports.
- The system also handles session management during logouts.

General Purpose:

This sequence diagram details how users (organization, candidate, and interviewer) interact with the platform across different modules and how the system coordinates various functionalities like authentication, scheduling, integration, and reporting.


Recruitment Platform Interaction
Organization
Candidate
Interviewer Engineer
User Interface (UI)
Authentication Module
Scheduling Module
Integration Module
Report Generation Module
System
Sign up and login
Verify credentials
Grant access
Display dashboard
Create job post and upload resumes
Store job details and resumes
Invite shortlisted candidates
Notify candidates via email/messages
View interview reports
Retrieve reports
Log out
Confirm logout
End session
Display logout confirmation
Sign up and login
Verify credentials
Grant access
Display dashboard
Set up profile and select interview slots
Store profile details and preferences
Wait for interview invitation
Notify candidate of invitation
Join interview via Integrator
Connect to Google Meet/Zoom
View final report
Retrieve report
Log out
Confirm logout
End session
Display logout confirmation
Sign up and login
Verify credentials
Grant access
Display dashboard
Set preferences and view candidates
Retrieve candidate list
Schedule interview
Confirm interview time
Conduct interview via Integrator
Connect to Google Meet/Zoom
Complete assessment via Google Form
Generate report
Log out
Confirm logout
End session
Display logout confirmation
Organization
Candidate
Interviewer Engineer
User Interface (UI)
Authentication Module
Scheduling Module
Integration Module
Report Generation Module
System

5.3.2 State Diagram

5.3.3 Components:

1. **Recruitment Platform** (Central System):
    - Encompasses key modules:
        1. **Job Management**:
            - Handles job details and resumes from the organization.
        2. **Candidate Management**:
            - Manages candidate profiles and interactions.
        3. **Interview Scheduling & Execution**:
            - Coordinates interview scheduling and execution processes (likely integrates with tools like Google Meet/Zoom).
        4. **Report Generation**:
            - Creates and shares final interview reports.
2. **Data Storage and Retrieval**:
    - Supports the recruitment platform by storing and retrieving information such as job details, resumes, candidate profiles, and reports.

5.3.4 Interaction Flow:

1. **Organization**:
    - Interacts with the **Job Management** module to input job details and resumes.
    - Receives final interview reports via the **Report Generation** module.
2. **Candidate**:
    - Managed by the **Candidate Management** module for profile creation and updates.
    - Receives interview invites via the **Interview Scheduling & Execution** module.
    - Receives final reports through the **Report Generation** module.
3. **Interview Engineer**:
    - Interacts with the **Interview Scheduling & Execution** module to handle interview invites.
    - Possibly accesses the **Candidate Management** module to view candidate information before conducting interviews.

5.3.5 Data Flow:

- Data flows between the **Recruitment Platform** and the **Data Storage and Retrieval** system to store and retrieve relevant information for all modules.

5.3.6 Purpose:

This architecture illustrates the functional breakdown of the recruitment platform and how different user roles (Organization, Candidate, and Interview Engineer) interact with the system's modules. It highlights the modular structure and centralized data management supporting the platform's operations.

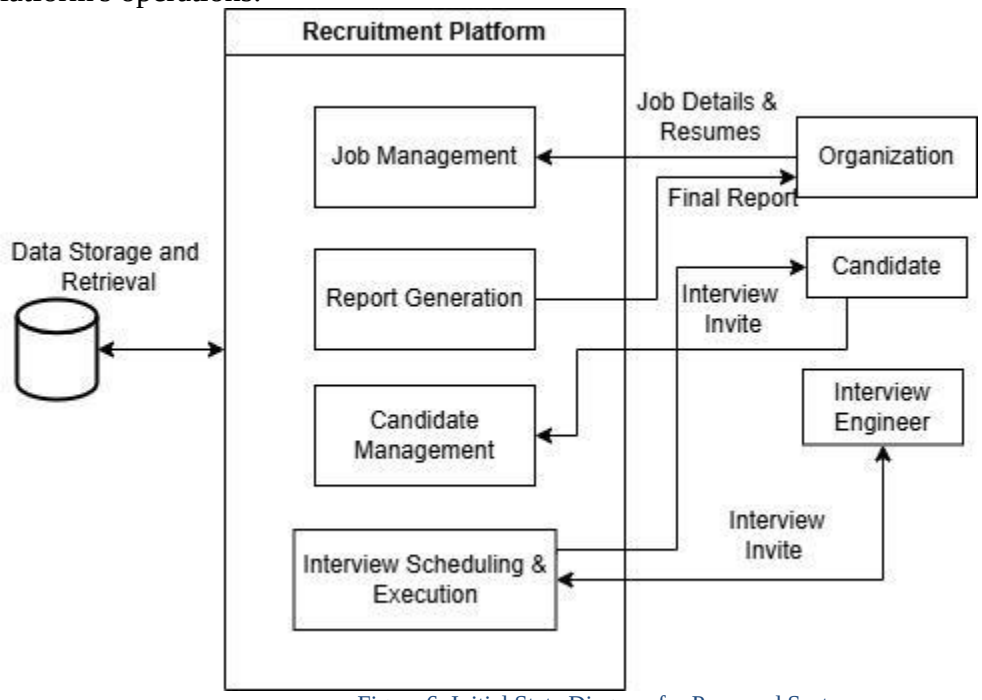


Figure 6: Initial State Diagram for Proposed System

5.4 GUI Design

The initial design of this GUI serves as a foundation for further development. It represents our current understanding of the user's needs and the system's functionality. We recognize that this design may evolve as we gather more user feedback and iterate on the system. Continuous improvement is crucial to ensure the GUI remains user-friendly, efficient, and effective in supporting the desired workflows.

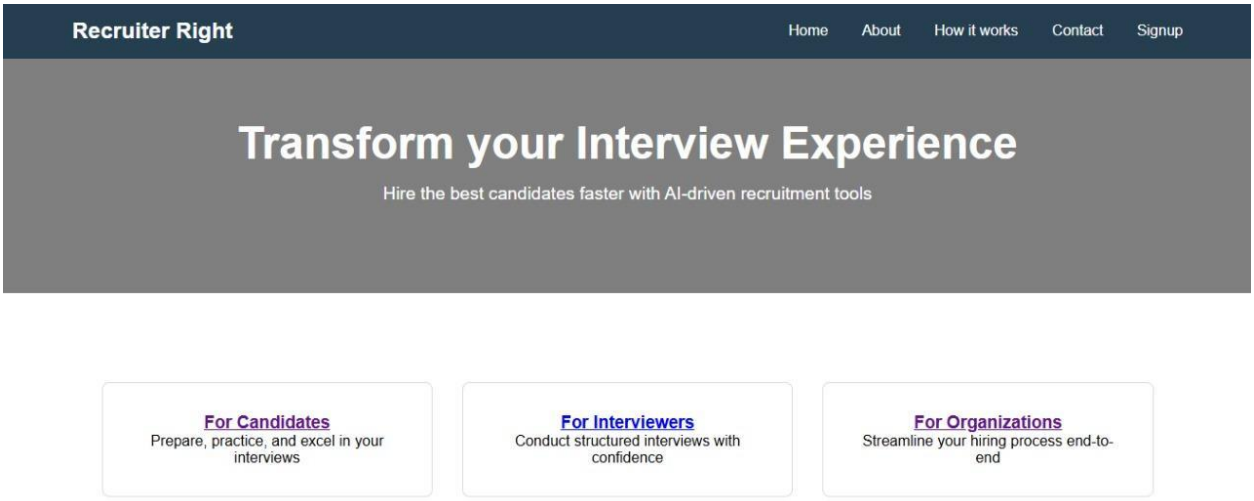
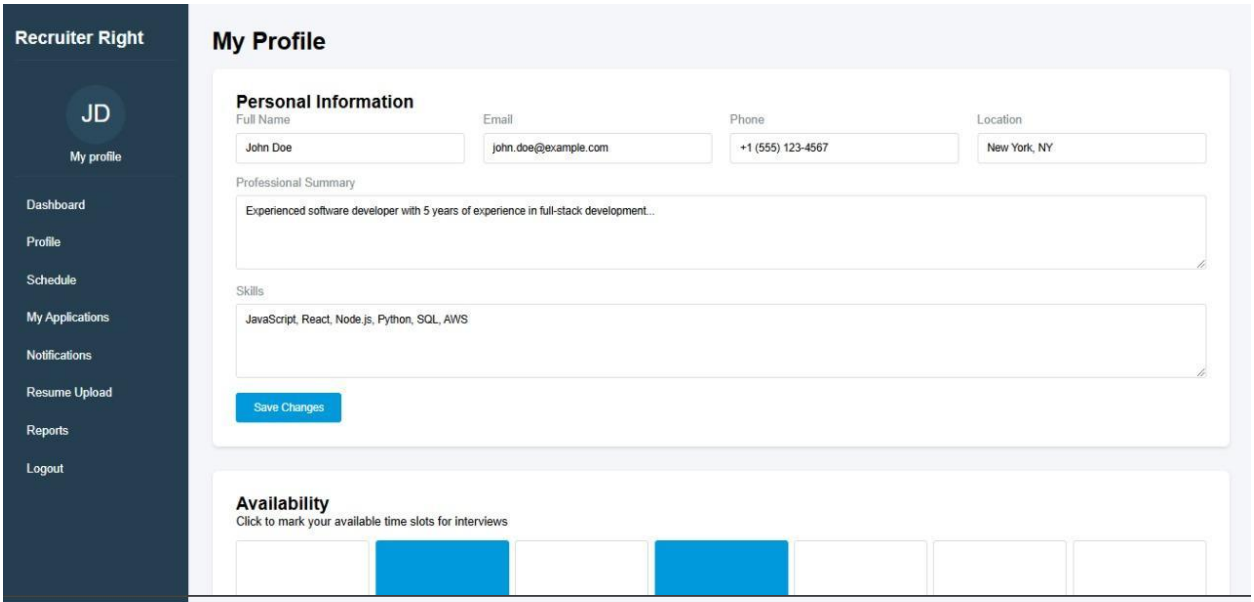
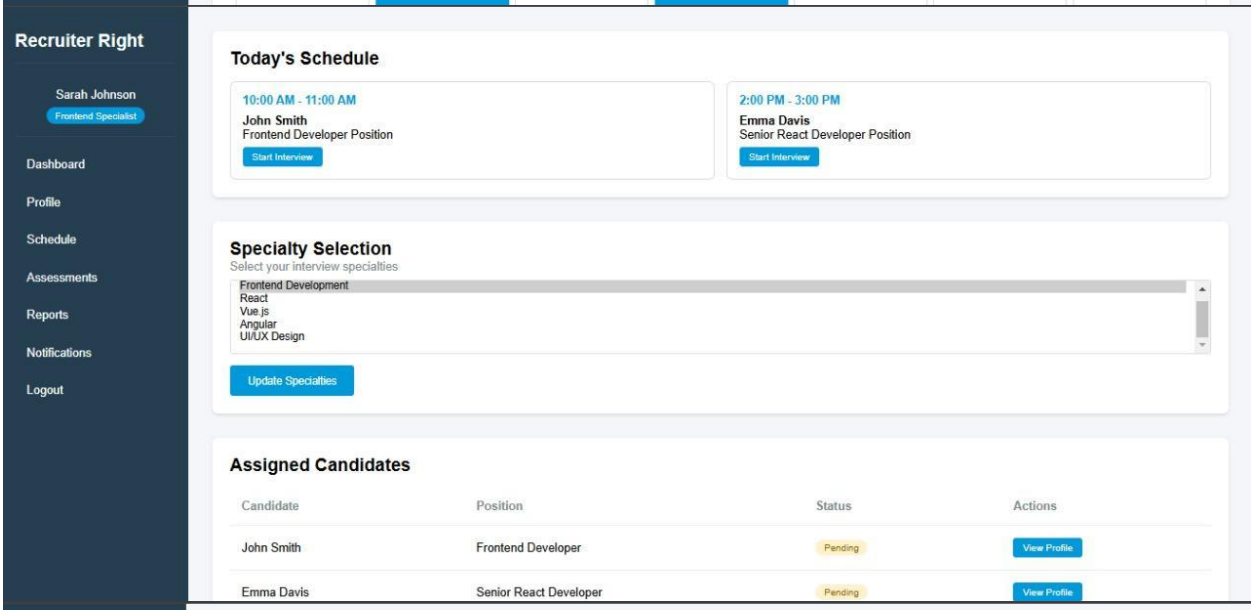
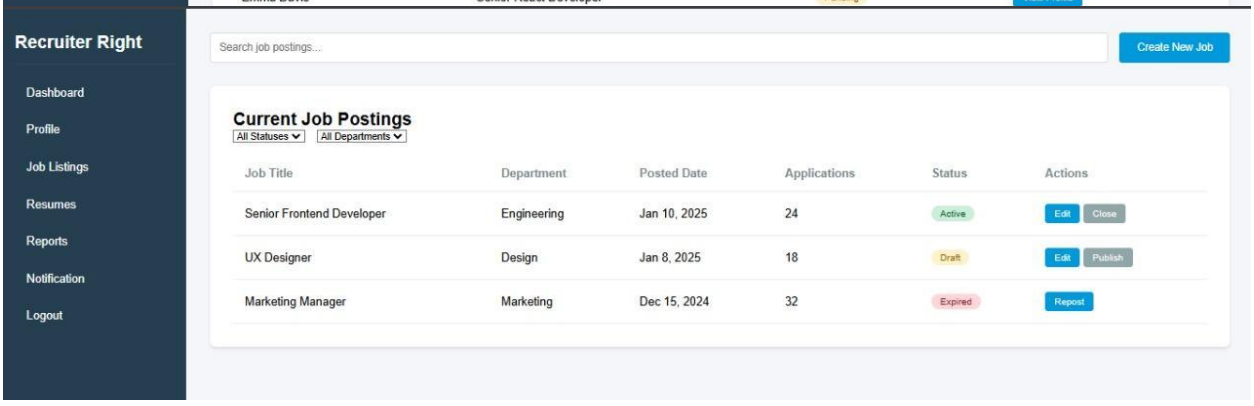
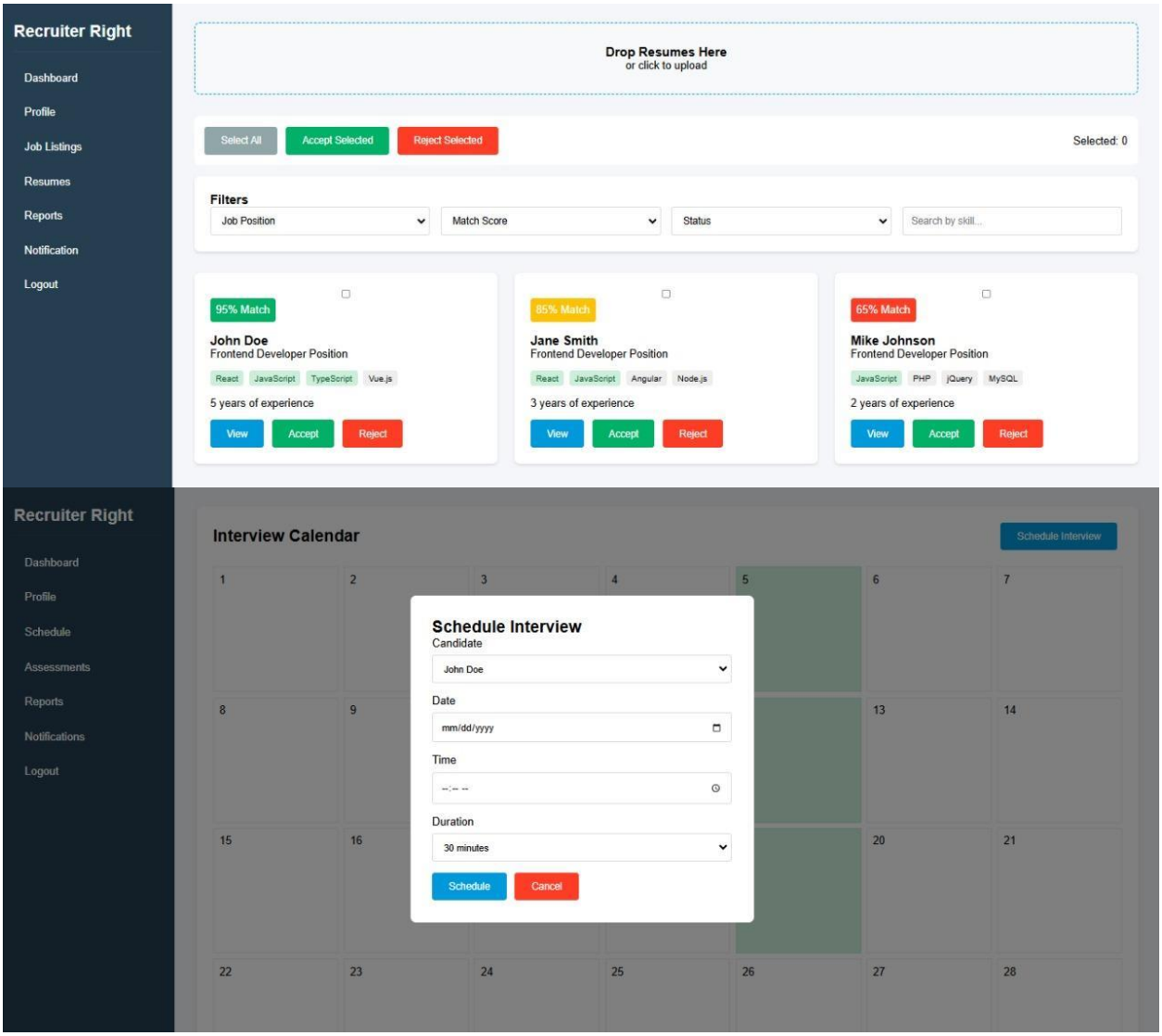


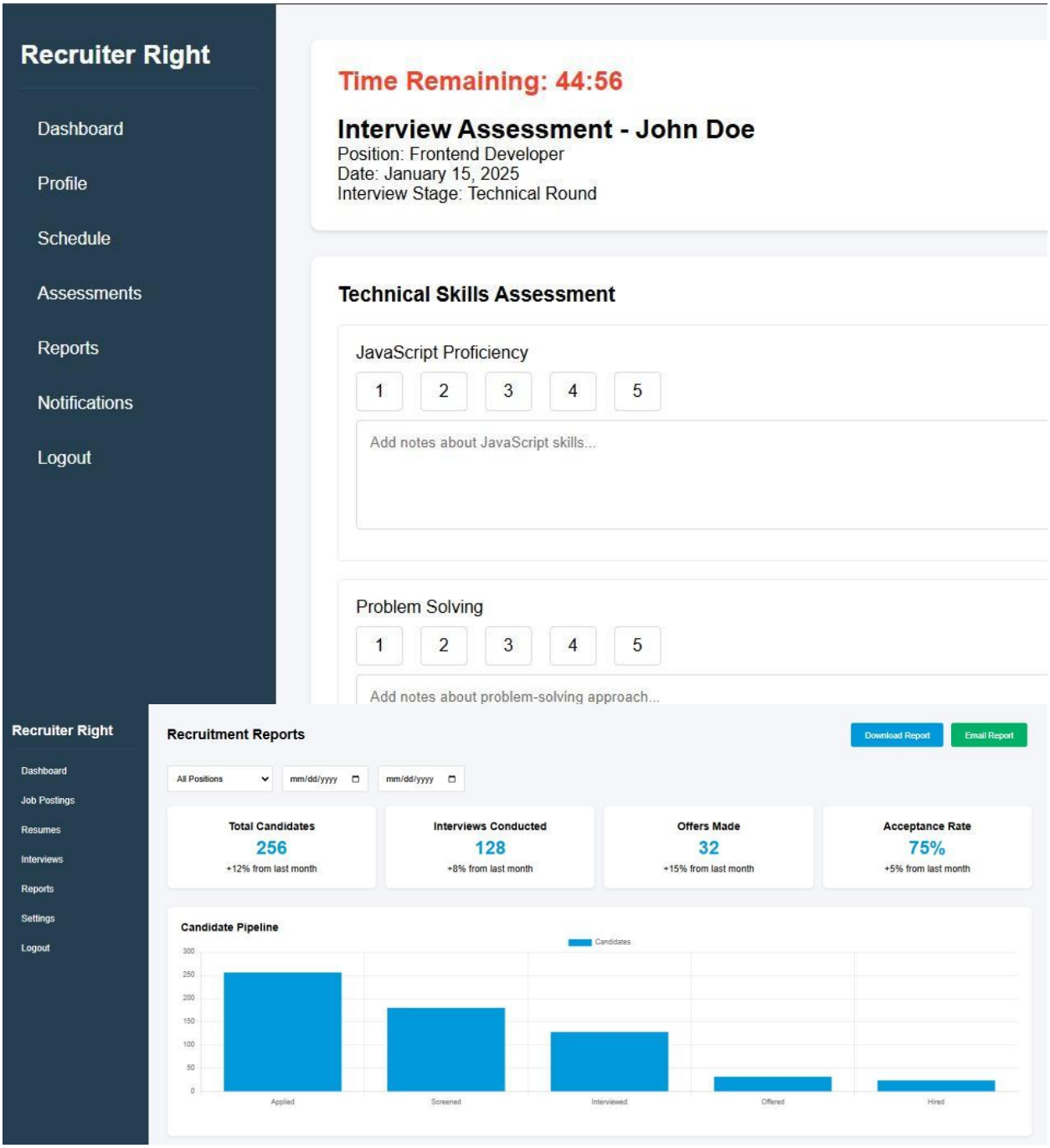
Figure 7: Landing Page

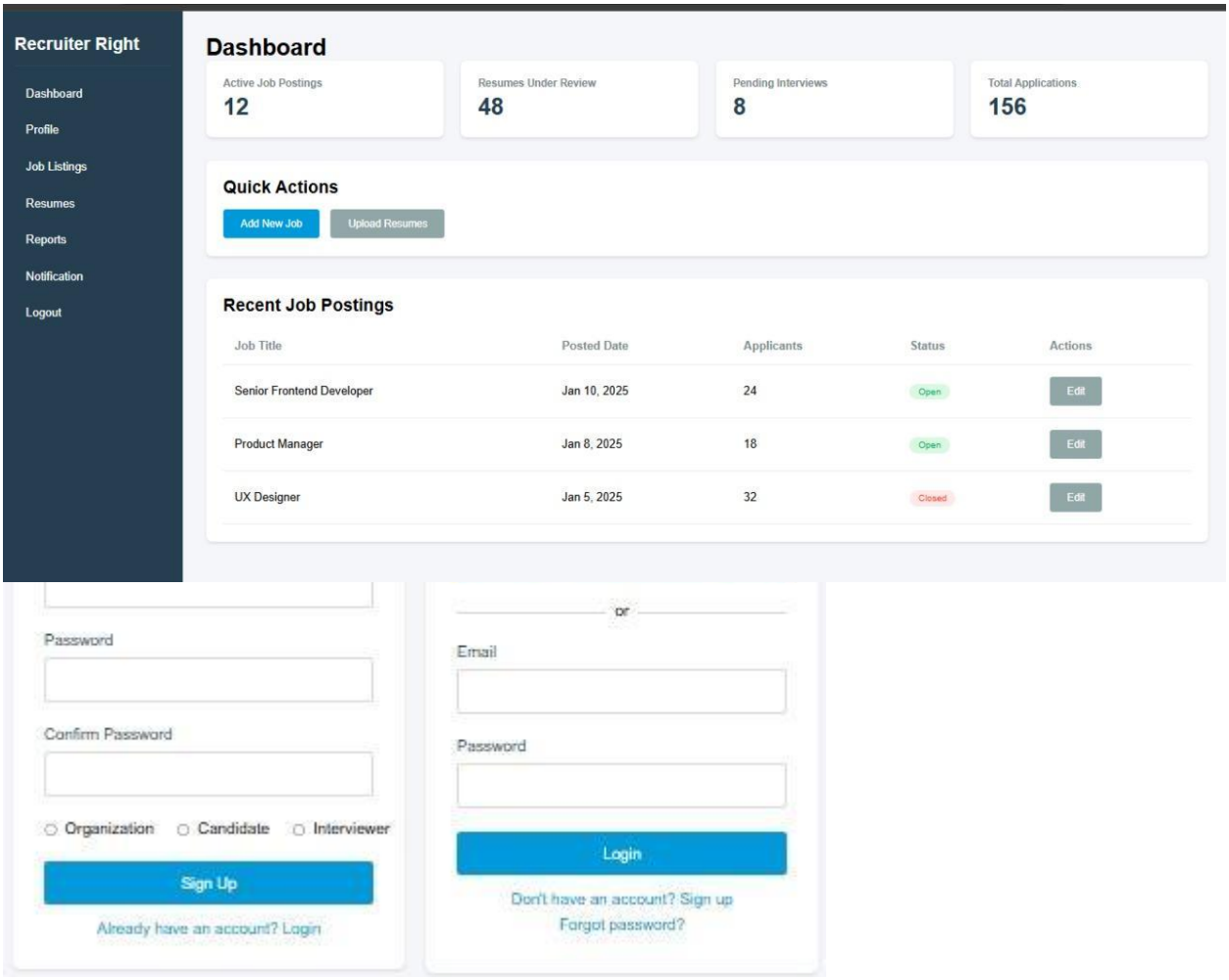

Recruiter Right
JD
My profile
Dashboard
Profile
Schedule
My Applications
Notifications
Resume Upload
Reports
Logout
My Profile
Personal Information
Full Name
John Doe
Email
john.doe@example.com
Phone
+1 (555) 123-4567
Location
New York, NY
Professional Summary
Experienced software developer with 5 years of experience in full-stack development...
Skills
JavaScript, React, Node.js, Python, SQL, AWS
Save Changes
Availability
Click to mark your available time slots for interviews


Recruiter Right
Sarah Johnson
Frontend Specialist
Dashboard
Profile
Schedule
Assessments
Reports
Notifications
Logout
Today's Schedule
10:00 AM - 11:00 AM
John Smith
Frontend Developer Position
Start Interview
2:00 PM - 3:00 PM
Emma Davis
Senior React Developer Position
Start Interview
Specialty Selection
Select your interview specialties
Frontend Development
React
Vue.js
Angular
UI/UX Design
Update Specialties
Assigned Candidates
Candidate
Position
Status
Actions
John Smith
Frontend Developer
Pending
View Profile
Emma Davis
Senior React Developer
Pending
View Profile


Recruiter Right
Dashboard
Profile
Job Listings
Resumes
Reports
Notification
Logout
Search job postings...
Create New Job
Current Job Postings
All Statuses
All Departments
Job Title
Department
Posted Date
Applications
Status
Actions
Senior Frontend Developer
Engineering
Jan 10, 2025
24
Active
Edit
Close
UX Designer
Design
Jan 8, 2025
18
Draft
Edit
Publish
Marketing Manager
Marketing
Dec 15, 2024
32
Expired
Repost


Recruiter Right
Dashboard
Profile
Job Listings
Resumes
Reports
Notification
Logout
Drop Resumes Here
or click to upload
Select All
Accept Selected
Reject Selected
Selected: 0
Filters
Job Position
Match Score
Status
Search by skill...
95% Match
John Doe
Frontend Developer Position
React
JavaScript
TypeScript
Vue.js
5 years of experience
View
Accept
Reject
85% Match
Jane Smith
Frontend Developer Position
React
JavaScript
Angular
Node.js
3 years of experience
View
Accept
Reject
65% Match
Mike Johnson
Frontend Developer Position
JavaScript
PHP
jQuery
MySQL
2 years of experience
View
Accept
Reject
Recruiter Right
Dashboard
Profile
Schedule
Assessments
Reports
Notifications
Logout
Interview Calendar
Schedule Interview
Schedule Interview
Candidate
John Doe
Date
mm/dd/yyyy
Time
--:-- --
Duration
30 minutes
Schedule
Cancel


Recruiter Right
Dashboard
Profile
Schedule
Assessments
Reports
Notifications
Logout
Time Remaining: 44:56
Interview Assessment - John Doe
Position: Frontend Developer
Date: January 15, 2025
Interview Stage: Technical Round
Technical Skills Assessment
JavaScript Proficiency
1 2 3 4 5
Add notes about JavaScript skills...
Problem Solving
1 2 3 4 5
Add notes about problem-solving approach...
Recruiter Right
Dashboard
Job Postings
Resumes
Interviews
Reports
Settings
Logout
Recruitment Reports
Download Report
Email Report
All Positions
mm/dd/yyyy
mm/dd/yyyy
Total Candidates
256
+12% from last month
Interviews Conducted
128
+8% from last month
Offers Made
32
+15% from last month
Acceptance Rate
75%
+5% from last month
Candidate Pipeline
Candidates
300
250
200
150
100
50
0
Applied
Screened
Interviewed
Offered
Hired


Recruiter Right
Dashboard
Profile
Job Listings
Resumes
Reports
Notification
Logout
Dashboard
Active Job Postings
12
Resumes Under Review
48
Pending Interviews
8
Total Applications
156
Quick Actions
Add New Job
Upload Resumes
Recent Job Postings
Job Title
Posted Date
Applicants
Status
Actions
Senior Frontend Developer
Jan 10, 2025
24
Open
Edit
Product Manager
Jan 8, 2025
18
Open
Edit
UX Designer
Jan 5, 2025
32
Closed
Edit
Password
Confirm Password
Organization
Candidate
Interviewer
Sign Up
Already have an account? Login
or
Email
Password
Login
Don't have an account? Sign up
Forgot password?

6 References

[1] Internet Web page: Author. “karat”. https://karat.com/.

[2] Internet Web page: Author. “intervue.io”. https://www.intervue.io/.

[3] Mehboob, Muntaha & Ali, Saad & ul Islam, Saif & Ali, Syed. (2022). Evaluating Automatic CV Shortlisting Tool For Job Recruitment Based On Machine Learning Techniques. 1-4. 10.1109/MAJICC56935.2022.9994112.

[4] G O, Narendra & S., Hashwanth. (2022). Named Entity Recognition based Resume Parser and Summarizer. 2581-9429. 10.48175/IJARSCT-3029.