



**islington college**  
(इस्लिंग्टन कॉलेज)

**Module Code & Module Title**

**CS6P05NI Project**

**Interim Report**

**25% Individual Coursework**

**Submission: Final Submission**

**AY 2025 2026**

**Academic Semester: Autumn Semester 2025**

**Credit: 30 Credit Year Long Module**

**Student Name: Bhishma Pratap Karki**

**London Met ID: 23049355**

**College ID: NP01CP4A230361**

**Assignment Due Date: Tuesday, December 30, 2025**

**Assignment Submission Date: Tuesday, December 30, 2025**

**Internal Supervisor: Shresha Rajbhandari**

**External Supervisor: Rabin Sapkota**

*I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

## Table of Contents

1. Introduction .....	1
1.1. Introduction to the topic .....	1
1.2. Current Scenario.....	1
1.3. Problem Statement.....	2
1.4. The Project as a Solution.....	2
1.5. Aim and Objectives.....	4
1.5.1. Aim.....	4
1.5.2. Objective .....	4
1.6. Report Structure .....	5
1.6.1. Background .....	5
1.6.2. Development.....	5
1.6.3. Development Till Date and Analysis of Progress .....	5
1.6.4. Future Work .....	5
2. Background .....	6
2.1. About the End User / Client .....	6
2.2. Understanding of the Project .....	6
2.3. Client Agreement Letter.....	7
2.4. Review of Similar Project.....	8
2.4.1. Similar Project 1: Expertini (Nepal) .....	8
2.5. Comparison of Similar Projects.....	9
3. Development Methodology.....	10
3.1. Software Development Life Cycle (SDLC).....	10
3.2. Considered Methodology .....	10
3.2.1. Prototype Model .....	10

3.2.2. Spiral Model .....	11
3.2.3. Agile Kanban Model .....	12
3.3. Selected Methodology .....	13
3.3.1. Agile Scrum.....	13
3.4. Work Breakdown Structure (WBS) .....	14
3.5. Gantt Chart (Revised Gantt Chart) .....	15
4. Development Till Date .....	25
4.1. System Architecture Diagram .....	25
4.2. Software Requirements Specifications (SRS).....	25
4.3. Use Case Diagram .....	26
4.4. Expanded Description.....	27
4.5. Sequence Diagram .....	27
4.5.1. Sequence Diagram for User Authentication (Access User to System).....	27
4.5.2. Sequence Diagram for Create or Update Portfolio (Candidate) .....	28
4.5.3. Sequence Diagram for Job Apply and Application Tracking .....	29
4.5.4. Sequence Diagram for AI Job Recommendations (Candidate).....	30
4.5.5. Sequence Diagram for Connection and Messaging .....	31
4.5.6. Sequence Diagram for Job Posting and Manage Application (ATS Resume scanner) .....	32
4.5.7. Sequence Diagram for Seach Candidates and View Profile .....	33
4.5.8. Sequence Diagram for Manage Users and Quizzes / Assessments .....	34
4.6. Collaboration Diagram .....	35
4.6.1. Collaboration Diagram for User Authentication (Access User to System)....	35
4.6.2. Collaboration Diagram for Create or Update Portfolio (Candidate) .....	36
4.6.3. Collaboration Diagram for Job Apply and Application Tracking.....	36

4.6.4. Collaboration Diagram for AI Job Recommendations (Candidate).....	37
4.6.5. Collaboration Diagram for Connection and Messaging .....	37
4.6.6. Collaboration Diagram for Job Posting and Manage Application (ATS Resume scanner) .....	38
4.6.7. Collaboration Diagram for Seach Candidates and View Profile .....	38
4.6.8. Collaboration Diagram for Manage Users and Quizzes / Assessments .....	39
4.7. Activity Diagram.....	40
4.7.1. Activity Diagram for User Authentication (Access User to System).....	40
4.7.2. Activity Diagram for Create or Update Portfolio (Candidate) .....	42
4.7.3. Activity Diagram for Job Apply and Application Tracking .....	43
4.7.4. Activity Diagram for AI Job Recommendations (Candidate).....	45
4.7.5. Activity Diagram for Connection and Messaging .....	46
4.7.6. Activity Diagram for Job Posting and Manage Application (ATS Resume Scanner) .....	47
4.7.7. Activity Diagram for Seach Candidates and View Profile .....	49
4.7.8. Activity Diagram for Manage Users and Quizzes / Assessments .....	50
4.8. DFD Level 0 (Context Level Diagram (Level 0)) .....	52
4.9. Flow Chart .....	53
4.9.1. Candidate System Flowchart .....	53
4.9.2. Recruiter System Flowchart .....	55
4.9.3. Admin System Flowchart.....	57
4.10. Database Collections and Relationships .....	59
4.11. Class Diagram .....	60
4.11.1. Overall, Class Diagram .....	60
4.11.2. Individual Class Diagram.....	61
4.12. Wireframes .....	68

4.12.1. Wireframe of Register Page .....	68
4.12.2. Wireframe of Login Page.....	69
4.12.3. Wireframe of Forget Password Page .....	70
4.12.4. Wireframe of Verify Account Page .....	71
4.12.5. Wireframe of New Password Page .....	72
4.13. UI Designs .....	73
4.13.1. UI Design of Register Page.....	73
4.13.2. UI Design of Login Page .....	74
4.13.3. UI Design of Forget Password Page .....	75
4.13.4. UI Design of New Password Page .....	76
4.13.5. UI Design of Verify Account Page .....	77
4.14. Development Status .....	78
4.14.1. Setup of Frontend for HireLink Project .....	78
4.14.2. Setup of Backend for HireLink Project .....	78
4.14.3. Collection of Data in MongoDB .....	79
4.14.4. Register Functionality.....	80
4.14.5. Login Functionality .....	83
4.14.6. Verification Code Functionality.....	85
4.14.7. Forget Password Functionality .....	87
4.14.8. Reset Password Functionality.....	89
4.14.9. Candidate Portfolio and Profile Page .....	91
5. Analysis of Progress.....	93
5.1. Progress Review.....	93
5.2. Progress Table .....	94
5.3. Action Plan.....	95

6. Future Work .....	95
7. Conclusion .....	96
8. References.....	97
9. Appendix .....	100
9.1. Review of Similar Projects .....	100
9.1.1. Similar Project 2: Breezy HR.....	100
9.1.2. Similar Project 3: Magnet.me .....	101
9.1.3. Similar Project 4: Torre.ai.....	102
9.1.4. Similar Project 5: Merojob AI (Nepal) .....	103
9.2. Software Development Life Cycle (SDLC).....	104
9.3. Software Requirements Specification (SRS) .....	105
9.4. Expanded Description.....	111
9.4.1. Expanded Description of Access User to the System .....	111
9.4.2. Expanded Description of Create or Update Portfolio.....	112
9.4.3. Expanded Description of Job Apply and Application Tracking .....	114
9.4.4. Expanded Description of AI Job Recommendations .....	115
9.4.5. Expanded Description of Connection and Messaging .....	117
9.4.6. Expanded Description of Job Posting and Manage Application (ATS Resume Scanner) .....	118
9.4.7. Expanded Description of Search Candidate and View Profile.....	120
9.4.8. Expanded Description of Manage Users and Quizzes / Assessments .....	122
9.5. Wireframes .....	124
9.5.1. Wireframe of Contact Us Page.....	124
9.5.2. Wireframe of About Us Page.....	125
9.5.3. Wireframe of Candidate Dashboard Page.....	126

9.5.4 Wireframe of Portfolio Page and Profile Page.....	127
9.5.5. Wireframe of Message Page.....	128
9.5.6. Wireframe of Skills Assessment and Quizzes Page.....	128
9.5.7. Wireframe of Saved Jobs Page.....	129
9.5.8. Wireframe of Job Listing Page .....	130
9.5.9. Wireframe of Individual Job Details Page .....	131
9.5.10. Wireframe of Company Name Listing Page. ....	132
9.5.11. Wireframe of Company Description Page .....	133
9.5.12. Wireframe of Home page when user is not Authenticated. ....	134
9.5.13. Wireframe of AI Job Recommendation Listing Page.....	135
9.5.14. Wireframe of Application Status Tracking Page .....	136
9.6. UI Designs .....	137
9.6.1. UI Design of Candidate Dashboard Page .....	137
9.6.2. UI Design of Portfolio and Profile Page.....	138
9.6.3. UI Design of Message Page.....	139
9.6.4. UI Design of Application Status Tracking Page.....	140
9.6.5. UI Design of Saved Job Page .....	141
9.6.6. UI Design of Quizzes and Assignments Page.....	142
9.6.7. UI Design of Job Listing Page .....	143
9.6.8. UI Design of Individual Job Details Page .....	144
9.6.9. UI Design of AI Job Recommendation Page .....	145
9.6.10. UI Design of Company Listing Page .....	146
9.6.11. UI Design of Company Description Page .....	147
9.6.12. UI Design of About Us Page .....	148
9.6.13. UI Design of Home Page when user in not authenticated .....	149

9.6.14. UI Design of Contact Us Page .....	150
9.7. Action Plan.....	151
9.8. Survey Responses.....	152
9.9. Previous Gantt Chart .....	163

## Table of Figures

Figure 1: Client Agreement Letter .....	7
Figure 2: Expertini (Expertini, 2025) .....	8
Figure 3: Prototype Model Image (Board Infinity, 2025).....	10
Figure 4: Spiral Model Image (Boehm, 1988).....	11
Figure 5: Agile Kanban Model Image (Radigan, 2025).....	12
Figure 6: Agile Scrum Image (Hubstaff tasks, 2025).....	13
Figure 7: Work Breakdown Structure .....	14
Figure 8: Revised Gantt Chart Initiation Phase .....	15
Figure 9: Revised Gantt Chart Planning Phase.....	16
Figure 10: Revised Gantt Chart Sprint Execution Phase Sprint 1 Core Setup and Authentication .....	16
Figure 11: Revised Gantt Chart Sprint Execution Phase Sprint 2 Portfolio and Profile .	17
Figure 12: Revised Gantt Chart Sprint Execution Phase Sprint 3 Job Discovery and Application Tracking .....	17
Figure 13: Revised Gantt Chart Sprint Execution Phase Sprint 4 Recruiter Features...	18
Figure 14: Revised Gantt Chart Sprint Execution Phase Sprint 5 Admin Dashboard....	18
Figure 15: Revised Gantt Chart Sprint Execution Phase Sprint 6 Notifications, Messaging and Networking .....	19
Figure 16: Revised Gantt Chart Sprint Execution Phase Sprint 7 Testing and Final Documentation .....	19
Figure 17: Revised Gantt Chart Control Phase .....	20
Figure 18: Revised Gantt Chart Closure Phase .....	20
Figure 19: Full View of Gantt Chart .....	24
Figure 20: System Architecture Diagram.....	25
Figure 21: Use Case Diagram .....	26
Figure 22: Sequence Diagram for User Authentication .....	27
Figure 23: Sequence Diagram for Create or Update Portfolio (Candidate) .....	28
Figure 24: Sequence Diagram for Job Apply and Application Tracking.....	29
Figure 25: Sequence Diagram for AI Job Recommendations (Candidate).....	30
Figure 26: Sequence Diagram for Connection and Messaging .....	31

Figure 27: Sequence Diagram for Job Posting and Manage Application (ATS Resume scanner) .....	32
Figure 28: Sequence Diagram for Seach Candidates and View Profile .....	33
Figure 29: Sequence Diagram for Manage Users and Quizzes / Assessments .....	34
Figure 30: Collaboration Diagram of Register .....	35
Figure 31: Collaboration Diagram of Login.....	35
Figure 32: Collaboration Diagram for Create or Update Portfolio (Candidate) .....	36
Figure 33: Collaboration Diagram for Job Apply and Application Tracking.....	36
Figure 34: Collaboration Diagram for Job Apply and Application Tracking.....	37
Figure 35: Collaboration Diagram for Connection and Messaging .....	37
Figure 36: Collaboration Diagram for Job Posting and Manage Application (ATS Resume scanner) .....	38
Figure 37:Collaboration Diagram for Seach Candidates and View Profile .....	38
Figure 38: Collaboration Diagram for Manage Users and Quizzes / Assessments .....	39
Figure 39: Activity Diagram for Register.....	40
Figure 40: Activity Diagram for Login .....	41
Figure 41: Activity Diagram for Create or Update Portfolio (Candidate).....	42
Figure 42:Activity Diagram for Job Apply .....	43
Figure 43:Activity Diagram for Application Tracking .....	44
Figure 44:Activity Diagram for AI Job Recommendations (Candidate).....	45
Figure 45: Activity Diagram for Connection and Messaging.....	46
Figure 46: Activity Diagram for Job Posting .....	47
Figure 47: Activity Diagram for Manage Application (ATS Resume Scanner).....	48
Figure 48: Activity Diagram for Seach Candidates and View Profile .....	49
Figure 49: Activity Diagram for Manage Users .....	50
Figure 50: Activity Diagram for Manage Quizzes / Assessments .....	51
Figure 51: DFD Level 0 (Context Level Diagram (Level 0)).....	52
Figure 52: Candidate System Flowchart 1 .....	53
Figure 53: Candidate System Flowchart 2 .....	54
Figure 54: Recruiter System Flowchart 1 .....	55
Figure 55: Recruiter System Flowchart 2 .....	56

Figure 56: Admin System Flowchart 1.....	57
Figure 57: Admin System Flowchart 2.....	58
Figure 58: UI Design of Candidate Dashboard Page .....	59
Figure 59: Overall Class Diagram .....	60
Figure 60: Class Diagram of Candidate .....	61
Figure 61: Class Diagram of Recruiter .....	62
Figure 62: Class Diagram of Admin.....	63
Figure 63: Class Diagram of Portfolio.....	63
Figure 64: Class Diagram of Job .....	64
Figure 65: Class Diagram of Application .....	64
Figure 66: Class Diagram of Message .....	65
Figure 67: Class Diagram of Connection.....	65
Figure 68: Class Diagram of AI Job Recommendation.....	66
Figure 69: Class Diagram of Resume / ATS Scan .....	66
Figure 70: Class Diagram of Quiz / Assessment.....	67
Figure 71: Class Diagram of Notification .....	67
Figure 72: Wireframe of Register Page .....	68
Figure 73: Wireframe of Login Page .....	69
Figure 74: Wireframe of Forget Password Page .....	70
Figure 75: Wireframe of Verify Account Page .....	71
Figure 76: Wireframe of New Password Page .....	72
Figure 77: UI Design of Register Page.....	73
Figure 78: UI Design of Login Page .....	74
Figure 79: UI Design of Forget Password Page .....	75
Figure 80: UI Design of New Password Page .....	76
Figure 81: UI Design of Verify Account Page .....	77
Figure 82: Setup of Frontend for HireLink Project .....	78
Figure 83: Setup of Backend for HireLink Project .....	78
Figure 84: Collection of Data in MongoDB .....	79
Figure 85: Backend codes of Registration.....	80
Figure 86: Frontend codes of Registration .....	81

Figure 87: UI after Frontend code of Registration .....	82
Figure 88: Backend codes of Login .....	83
Figure 89: Frontend codes of Login .....	83
Figure 90: Backend Code of Verification Code .....	85
Figure 91: Frontend Code of Verification Code .....	85
Figure 92: UI after frontend code of Verification Code .....	86
Figure 93: Backend Code of Forget Password.....	87
Figure 94: Frontend Code of Forget Password .....	87
Figure 95: UI after frontend code of Forget Password Page .....	88
Figure 96: Backend Code of Reset Password.....	89
Figure 97: Frontend Code of Reset Password .....	89
Figure 98: UI after Frontend code of Reset Password .....	90
Figure 99: Backend Code of Portfolio and Profile Page .....	91
Figure 100: Frontend Code of Portfolio and Profile Page .....	91
Figure 101: UI after Frontend Code of Portfolio and Profile Page .....	92
Figure 102: Jira Dashboard Showing Progress .....	93
Figure 103: Progress Table.....	94
Figure 104: Similar Project 1 (Breezy HR, 2025) .....	100
Figure 105: Similar Project 2 Magnet.me .....	101
Figure 106: Similar Project Torre.ai.....	102
Figure 107: MerojobAI (merojob, 2025).....	103
Figure 108: Wireframe of Contact Us Page.....	124
Figure 109: Wireframe of About Us Page.....	125
Figure 110: Wireframe of Contact Candidate Dashboard Page .....	126
Figure 111: Wireframe of Portfolio Page and Profile Page.....	127
Figure 112: Wireframe of Message Page .....	128
Figure 113: Wireframe of Skills Assessment and Quizzes Page.....	128
Figure 114: Wireframe of Saved Jobs Page.....	129
Figure 115: Wireframe of Job Listing Page .....	130
Figure 116: Wireframe of Individual Job Details Page.....	131
Figure 117: Wireframe of Company Name Listing Page .....	132

Figure 118: Wireframe of Company Description Page .....	133
Figure 119: Wireframe of Home page when user is not Authenticated. ....	134
Figure 120: Wireframe of AI Job Recommendation Listing Page .....	135
Figure 121: Wireframe of Application Status Tracking Page .....	136
Figure 122: UI Design of Candidate Dashboard Page .....	137
Figure 123: UI Design of Portfolio and Profile Page .....	138
Figure 124: UI Design of Message Page.....	139
Figure 125: UI Design of Application Status Tracking Page.....	140
Figure 126: UI Design of Saved Job Page .....	141
Figure 127: UI Design of Quizzes and Assignments Page.....	142
Figure 128: UI Design of Job Listing Page .....	143
Figure 129: UI Design of Individual Job Details Page .....	144
Figure 130: UI Design of AI Job Recommendation Page .....	145
Figure 131: UI Design of Company Listing Page .....	146
Figure 132: UI Design of Company Description Page .....	147
Figure 133: UI Design of About Us Page .....	148
Figure 134:UI Design of Home Page when user in not authenticated .....	149
Figure 135: UI Design of Contact Us Page .....	150
Figure 136: Survey Response 1 .....	152
Figure 137: Survey Response 2 .....	152
Figure 138: Survey Response 3 .....	153
Figure 139: Survey Response 4 .....	154
Figure 140: Survey Response 5 .....	154
Figure 141: Survey Response 6 .....	155
Figure 142: Survey Response 7 .....	156
Figure 143: Survey Response 8 .....	156
Figure 144: Survey Response 9 .....	157
Figure 145: Survey Response 10 .....	158
Figure 146: Survey Response 11 .....	159
Figure 147: Survey Response 12 .....	160
Figure 148: Survey Response 13 .....	161

Figure 149: Survey Response 14 .....	161
Figure 150: Previous Gantt Chart Initiation Phase .....	163
Figure 151: Previous Gantt Chart Planning Phase.....	163
Figure 152: Previous Gantt Chart Sprint 1 Core Setup and Authentication Phase.....	164
Figure 153: Previous Gantt Chart Sprint 2 Portfolio and Profile Phase .....	164
Figure 154: Previous Gantt Chart Sprint 3 Job Discovery and Application Tracking Phase .....	165
Figure 155: Previous Gantt Chart Sprint 4 Recruiter Features Phase.....	165
Figure 156: Previous Gantt Chart Sprint 5 Admin Dashboard Phase.....	166
Figure 157: Previous Gantt Chart Sprint 6 Notifications, Messaging, and Networking Phase .....	166
Figure 158: Previous Gantt Chart Sprint 7 Testing and Final Documentation Phase ..	167
Figure 159: Previous Gantt Chart Control Phase .....	167
Figure 160: Previous Gantt Chart Closure Phase .....	168
Figure 161: Previous Gantt Chart Full View .....	172

## **Table of Tables**

Table 1: Comparison of Similar Projects .....	9
Table 2: Expanded Description of Access User to the System .....	111
Table 3: Expanded Description of Create or Update Portfolio.....	112
Table 4: Expanded Description of Job Apply and Application Tracking.....	114
Table 5: Expanded Description of AI Job Recommendations .....	115
Table 6: Expanded Description of Connection and Messaging .....	117
Table 7: Expanded Description of Job Posting and Manage Application (ATS Resume Scanner).....	118
Table 8: Expanded Description of Search Candidate and View Profile .....	120
Table 9: Expanded Description of Manage Users and Quizzes / Assessments .....	122

## 1. Introduction

### 1.1. Introduction to the topic

The recruitment process in today's job market is highly competitive, with millions of candidates seeking suitable employment and recruiters needing to identify the right talent efficiently. Traditional methods such as manual resume reviews or keyword-based shortlisting are often slow and may overlook qualified candidates. As a result, candidates struggle to present their true skills and experience, while recruiters face challenges in accurately evaluating applicants (Chavan et al., 2024).

Online tools and platforms have improved this situation by allowing candidates to create detailed portfolios, showcase projects, and present certifications. Features such as AI-based job recommendations and real-time application tracking make it easier for candidates to find suitable opportunities. Recruiters benefit from streamlined application management, candidate ranking, and skill assessments. The Smart Job and Skill Portfolio web application integrates candidates, recruiters, and administrators on a single platform, enhancing transparency, reducing delays, and ensuring better job-candidate matching (Peicheva, 2022) (International Labour Organization, 2023).

### 1.2. Current Scenario

The current job market is highly competitive, with over 220 million unemployed candidates worldwide (International Labour Organization, 2023). Many job seekers struggle to find opportunities that match their skills and often do not receive timely updates on their applications (CIPD, 2024). Recruiters face challenges in efficiently shortlisting large volumes of resumes and frequently rely on traditional methods such as manual review or keyword-based screening, which may overlook qualified candidates (Hatti et al., 2025). These conventional processes often lack transparency and efficiency, causing delays and dissatisfaction for both candidates and recruiters (Tuttle & Critchlow, 2024).

To overcome these challenges, online recruitment platforms and portfolio management tools are increasingly adopted. They allow candidates to present skills, projects, and certifications in a structured format. Features like AI-based job

recommendations and real-time application tracking enable candidates to find suitable roles faster. Recruiters benefit from improved candidate management, automated shortlisting, and analytics to make informed hiring decisions, ensuring a more efficient and transparent recruitment process (Peicheva, 2022) (Taya et al., 2023).

### **1.3. Problem Statement**

Although online recruitment platforms are widely used, many candidates still struggle to showcase their skills, projects, and certifications in a clear and structured way. Traditional resumes mainly focus on academic qualifications and work experience, often failing to reflect practical skills and real-world accomplishments. This limitation can result in capable candidates being overlooked during the selection process, even if they meet job requirements (Nikolaou, 2021).

Recruiters face increasing pressure from the high volume of applications received for each job opening. Manual evaluation and basic keyword-based filtering are time-consuming and may not accurately identify the most suitable candidates. The lack of advanced tools like ATS resume scanners limits recruiters' ability to rank candidates efficiently (Bogen & Rieke, 2025). Additionally, admins often lack a centralized system to manage assessments, quizzes, and user content, creating challenges in maintaining transparency and control. This highlights the need for a unified smart job and skill portfolio platform that supports candidates, recruiters, and admins effectively (Köchling & Wehner, 2020).

### **1.4. The Project as a Solution**

The Smart Job and Skill Portfolio web application provides a unified platform for candidates, recruiters, and admins to address challenges in the current recruitment process. Candidates can create detailed profiles to showcase their skills, projects, and certifications in an organized way. Real-time application tracking allows candidates to monitor their applications, while AI-based job recommendations suggest positions that best match their skills and portfolio.

Recruiters benefit from automated candidate management, job posting, and ranking applicants using an ATS resume scanner, which extracts key information and ranks candidates fairly. They can also assign skill assessments and quizzes to evaluate candidates more effectively, improving hiring quality and speed. Admins can manage quizzes, content, and overall platform activity, ensuring transparency and consistency. By integrating the needs of all users, the platform reduces delays, enhances communication, and improves the match between candidates and suitable job opportunities.

## **1.5. Aim and Objectives**

### **1.5.1. Aim**

To develop a Smart Job and Skill Portfolio web application that allows candidates to showcase their skills and find suitable jobs efficiently, while enabling recruiters and admins to manage applications, assessments, and platform activities effectively.

### **1.5.2. Objective**

- i. To enable job seekers to create, update, and maintain portfolios with projects, certificates, and skills in a structured format by the end of January 05, 2025.
- ii. To provide candidates with real-time status updates on their applications, ensuring timely notifications by the end of January 24, 2026.
- iii. To implement an AI-driven system that suggests suitable jobs to candidates based on their skills, portfolio, and assessment results by January 24, 2026, improving job-candidate matching.
- iv. To allow recruiters to post job openings, search for candidates, and manage applications efficiently through the platform by Sprint February 11, 2026.
- v. To develop a local ATS (Application Tracking System) resume scanner that extracts, scores, and ranks candidates fairly based on their resume by February 11, 2026.
- vi. To provide admins with tools to manage quizzes, assessments, platform content, and monitor user activities effectively by February 27, 2026, ensuring transparency and control.
- vii. To deliver dashboards with key performance indicators and activity analytics for candidates, recruiters, and admins by February 27, 2026, supporting data-driven decisions.
- viii. To enable secure networking and messaging between candidates and recruiters by March 15, 2026, enhancing engagement and communication.
- ix. To ensure a secure, user-friendly, and accessible interface, complete final testing, and provide full documentation by Sprint March 31, 2026.

## 1.6. Report Structure

### 1.6.1. Background

The background section introduces the Smart Job and Skill Portfolio application and its main users including candidates, recruiters, and admins. It explains common recruitment challenges such as inefficient shortlisting and scattered portfolio information and how the system aims to solve them. Similar platforms are reviewed to identify gaps that this project addresses.

### 1.6.2. Development

The development section explains the overall approach used to build the Smart Job and Skill Portfolio application. It describes the development methodologies reviewed and justifies the selected methodology for the project. The section outlines the work breakdown structure, showing how tasks are organized into phases and sprints. A Gantt Chart is included to illustrate the project timeline, task dependencies, and planned schedule for implementation.

### 1.6.3. Development Till Date and Analysis of Progress

This section summarizes the work completed to date, including system architecture design, UML diagrams, flowcharts, wireframes, and UI design. It also presents evidence of implemented features and evaluates progress using progress tables and action plans. The section compares completed tasks with the project plan to ensure alignment with sprint goals. Core features for candidates, recruiters, and admins are partially implemented, with testing and improvements still in progress.

### 1.6.4. Future Work

This section explains the remaining tasks needed to complete the Smart Job and Skill Portfolio web application. It covers completing all user features, adding analytics dashboards, and implementing secure messaging and networking. Full system testing, documentation, and UI improvements based on feedback are also included.

## 2. Background

### 2.1. About the End User / Client

The primary users of the Smart Job and Skill Portfolio web application are candidates, recruiters, and admins, each with distinct roles. Candidates use the platform to showcase skills, projects, certifications, and experience in a structured portfolio, while benefiting from AI-based job recommendations, real-time application tracking, and secure communication with employers.

Recruiters use the system to source, evaluate, and rank applicants efficiently. They manage applications, assess candidates fairly using the ATS resume scanner, assign skill tests, and make informed hiring decisions with analytics and dashboards.

Admins oversee the platform to ensure smooth operation, content management, and user monitoring. The client, Excel Development Bank Ltd., especially its HR department, aims to modernize recruitment by connecting candidates, recruiters, and admins efficiently, reducing delays, and improving talent acquisition.

### 2.2. Understanding of the Project

The Smart Job and Skill Portfolio web application improves recruitment by providing a unified platform for candidates, recruiters, and admins. Candidates can create detailed profiles with skills, projects, certifications, and experience in a structured format, while benefiting from AI-based job recommendations and real-time application tracking to find suitable opportunities efficiently.

Recruiters can manage job postings, fairly evaluate and rank candidates using the ATS resume scanner, and assign skill assessments, supported by analytics and dashboards for informed hiring decisions. Admins manage quizzes, assessments, content, and system settings to ensure smooth operation, fairness, and security. By combining portfolio management, automated evaluation, and real-time tracking, the platform reduces recruitment delays and enhances overall hiring efficiency.

## 2.3. Client Agreement Letter

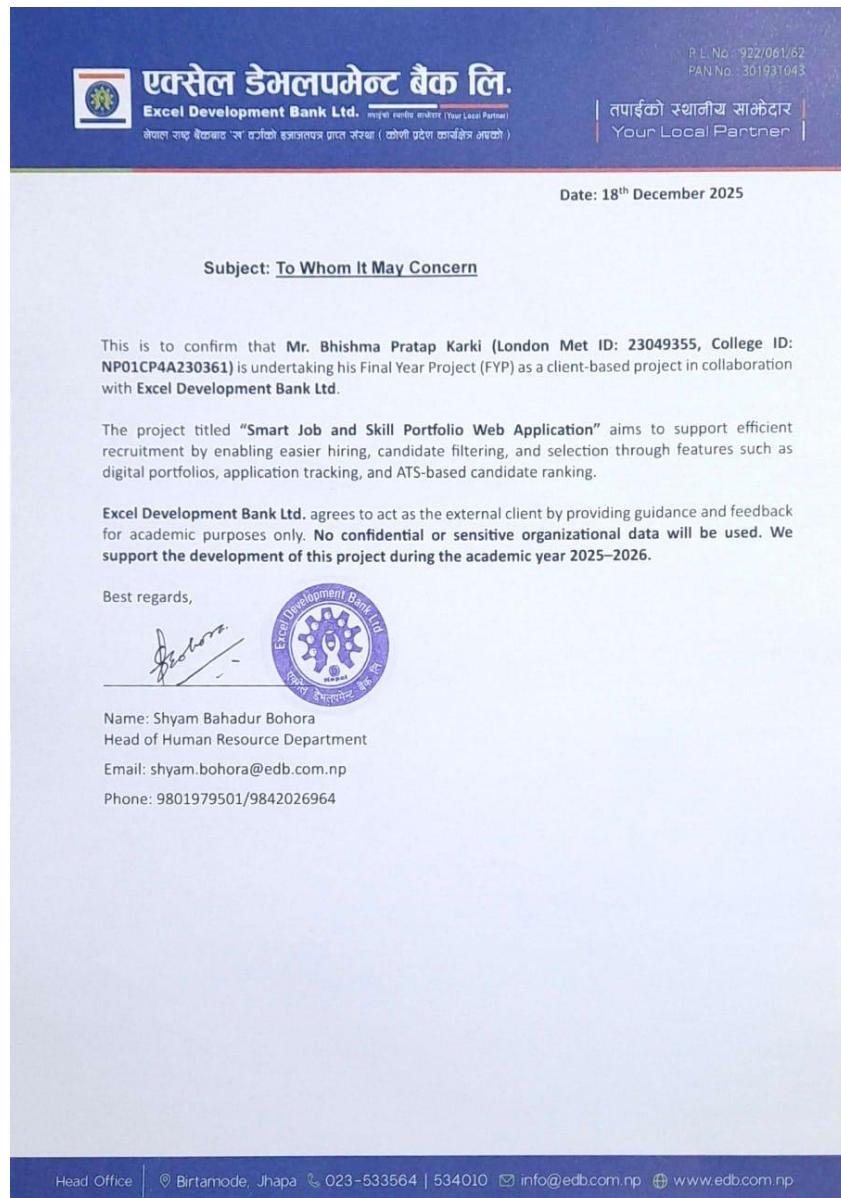


Figure 1: Client Agreement Letter

## 2.4. Review of Similar Project

### 2.4.1. Similar Project 1: Expertini (Nepal)

Expertini is a career-focussed platform widely used in Nepal and beyond that provides job search tools, application tracking, AI-powered job matching, resume scoring, and employer dashboards for posting and managing jobs. The system includes a built-in ATS, smart job matching algorithms, personalized job alerts, and analytics for both job seekers and employers, offering a robust solution for candidate-employer interactions within the Nepalese market (Expertini, 2025).

While Expertini provides strong job search, resume optimization, and applicant tracking features, it does not offer user portfolios with skill badges, admin-controlled quizzes and assessments, or a dedicated professional networking feature between candidates and recruiters. These are key differentiators of the Smart Job and Skill Portfolio platform.

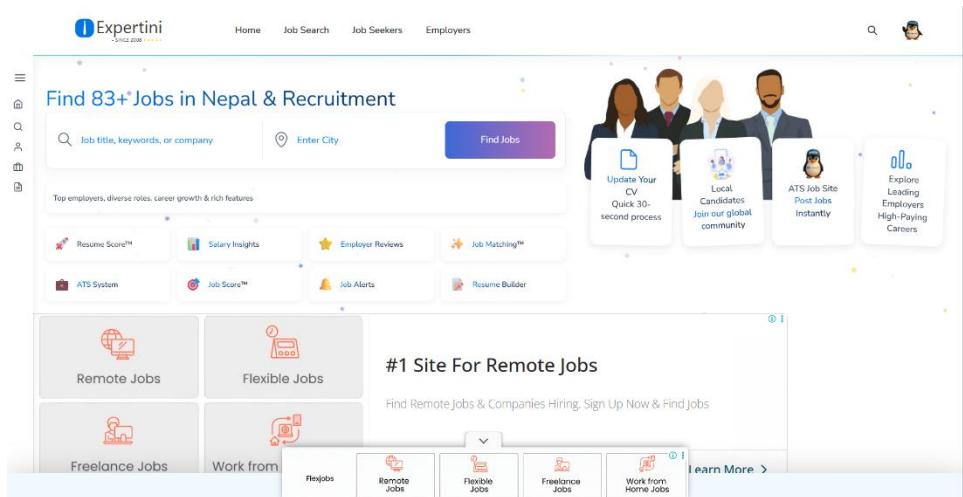


Figure 2: Expertini (Expertini, 2025)

For more detailed research on similar projects, see the Appendix.

## 2.5. Comparison of Similar Projects.

*Table 1: Comparison of Similar Projects*

S.N	Features	Expertini (Nepal)	Breezy HR	Magnet.me	Torre.ai	Merojob AI (Nepal)	Smart Job & Skill Portfolio
1.	Candidate Portfolio Creation	Limited	Limited	Yes	Limited	No	Yes
2.	AI-based Job Recommendations	Yes	No	Yes	Yes	Limited	Yes
3.	ATS Resume Scanner	Yes	Yes	No	No	Yes	Yes
4.	Resume Scoring and Ranking	Yes	No	No	No	Yes	Yes
5.	Skill Assessments and Quizzes	No	No	No	No	No	Yes
6.	Admin Control and Monitoring	Limited	Limited	Limited	No	Limited	Yes
7.	Real-time Application Tracking	Yes	No	Limited	No	Yes	Yes
8.	Dashboards and Analytics	Yes	Yes	Limited	Limited	Yes	Yes
9.	Secure Messaging and Networking	No	Limited	Limited	No	No	Yes

### 3. Development Methodology

#### 3.1. Software Development Life Cycle (SDLC)

[For detailed explanation of Software Development Life Cycle \(SDLC\), refer to Appendix](#)

#### 3.2. Considered Methodology

##### 3.2.1. Prototype Model

The Prototype Model involves developing an initial simplified version of the system to help clarify requirements and gather early user feedback (Pressman, 2010). This approach is particularly useful when system requirements are unclear or likely to evolve during development. By interacting with a prototype, stakeholders can identify missing features, usability issues, and requirement gaps at an early stage (Sommerville, 2011).

However, the Smart Job and Skill Portfolio platform requires strong architectural planning and integration of multiple complex modules such as AI-based job recommendations and ATS resume analysis. Relying heavily on prototyping could lead to insufficient documentation and weak system structure. Therefore, while limited prototyping may be applied at the user interface level, the Prototype Model was not selected as the primary methodology.

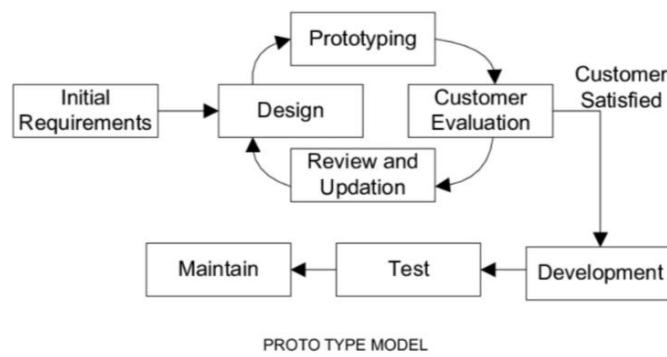


Figure 3: Prototype Model Image (Board Infinity, 2025)

### 3.2.2. Spiral Model

The Spiral Model is a risk-driven methodology that combines iterative development with systematic risk analysis (Boehm, 1988). Each iteration focuses on identifying risks, planning solutions, development, and evaluation, making it suitable for large-scale and high-risk systems.

Although the Spiral Model provides strong risk management, it requires extensive documentation, continuous risk assessment, and significant management effort. Given that this project is an individual academic project with a clearly defined scope and fixed timeline, the Spiral Model was considered too complex and resource-intensive for effective implementation.

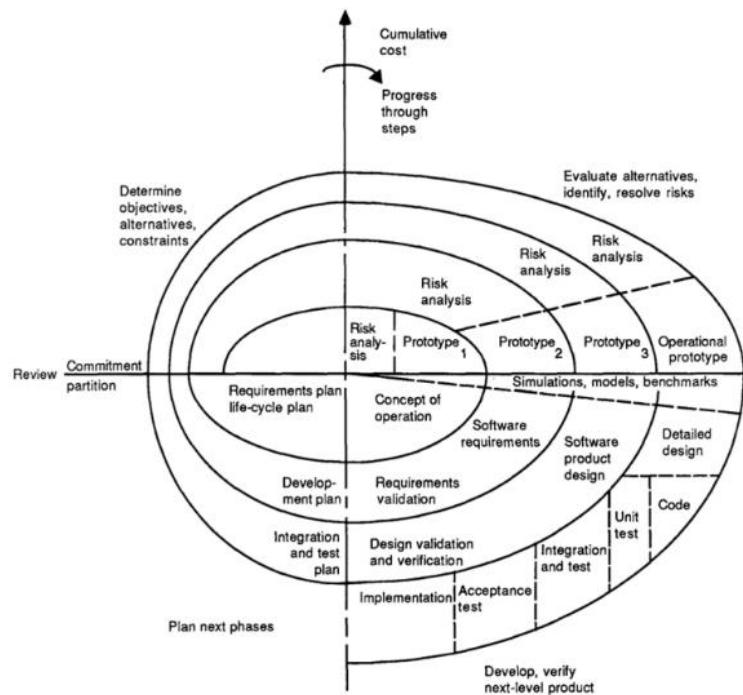


Figure 4: Spiral Model Image (Boehm, 1988)

### 3.2.3. Agile Kanban Model

Agile Kanban is a lightweight and visual development approach that focuses on continuous delivery and workflow optimization using task board (Radigan, 2025). It allows flexibility in handling changing priorities and managing multiple tasks simultaneously.

Despite its flexibility, Kanban lacks defined sprint structures, formal review cycles, and clear milestones-based deliverables. Since this project requires structured progress tracking and sprint-wise feature delivery, Agile Kanban alone was not considered sufficient as the primary methodology.

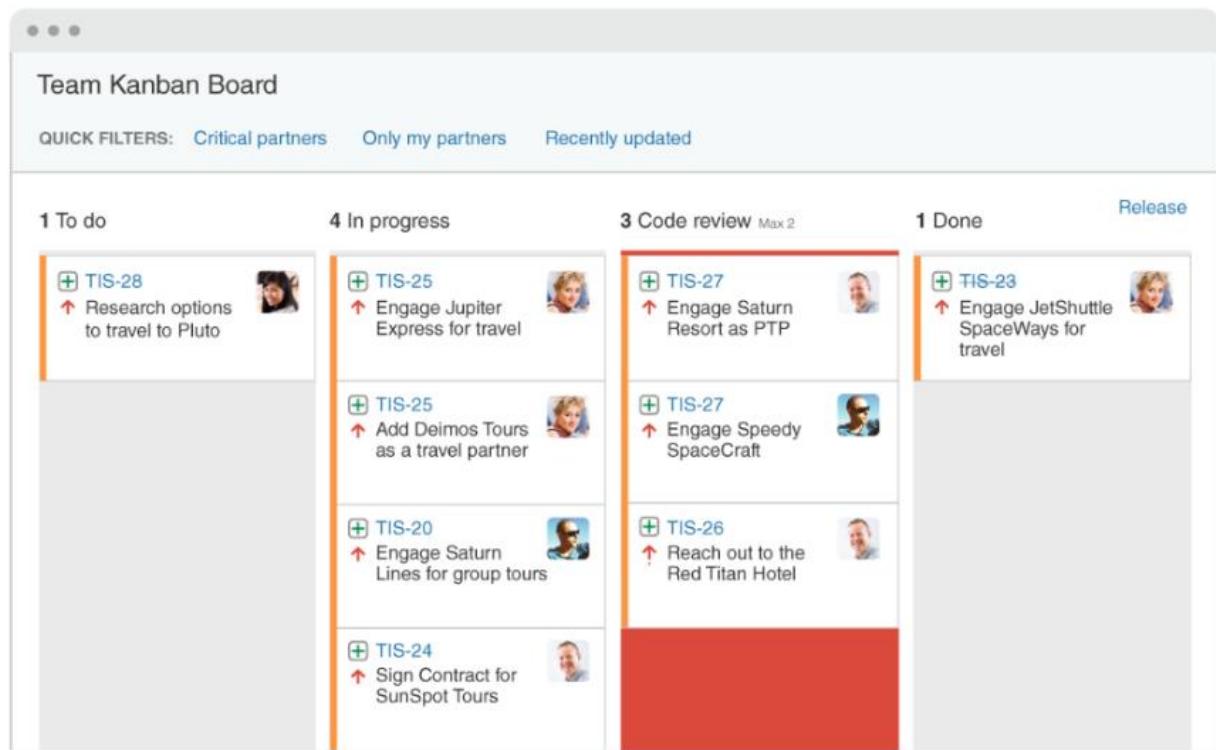


Figure 5: Agile Kanban Model Image (Radigan, 2025)

### 3.3. Selected Methodology

#### 3.3.1. Agile Scrum

Agile Scrum is an iterative and incremental development framework that divides the project into short sprints, allowing features to be developed, tested, and reviewed regularly. This ensures that the project can adapt to changes, incorporate feedback quickly, and deliver functional modules in a structured manner.

Scrum was chosen for this project because it provides clear sprint planning, review cycles, and milestone-based deliverables, which are essential for managing multiple interconnected modules. Its iterative approach helps track progress effectively, identify issues early, and maintain the quality of the system throughout development.

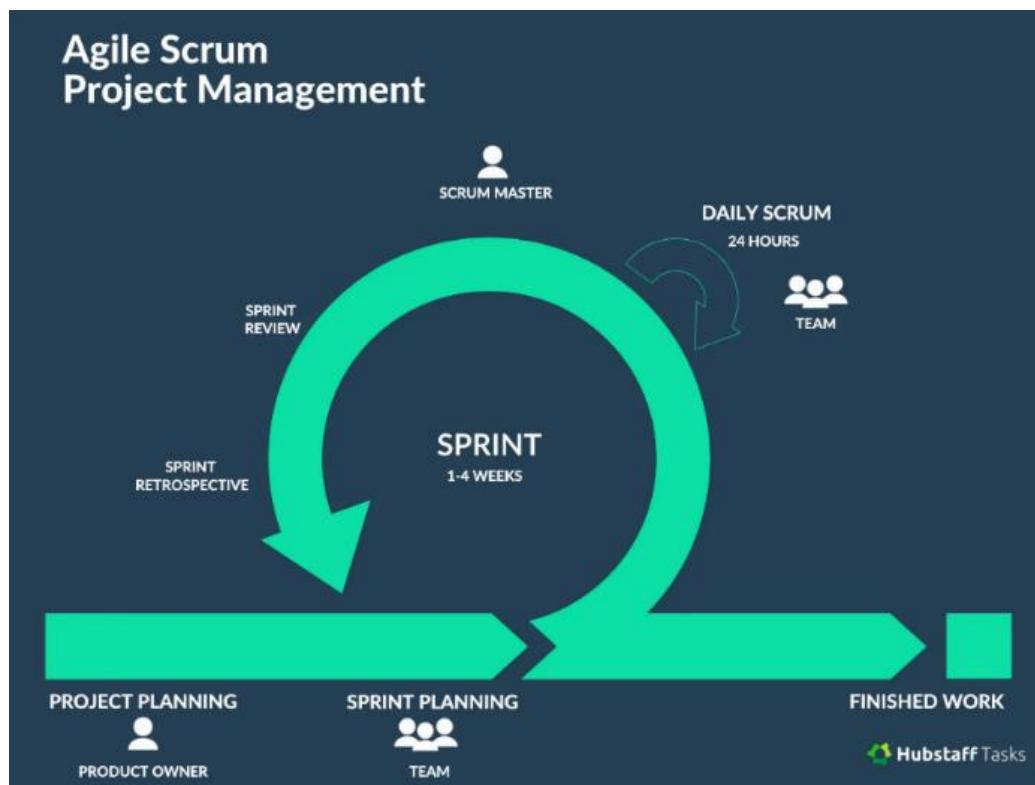


Figure 6: Agile Scrum Image (Hubstaff tasks, 2025)

### 3.4. Work Breakdown Structure (WBS)

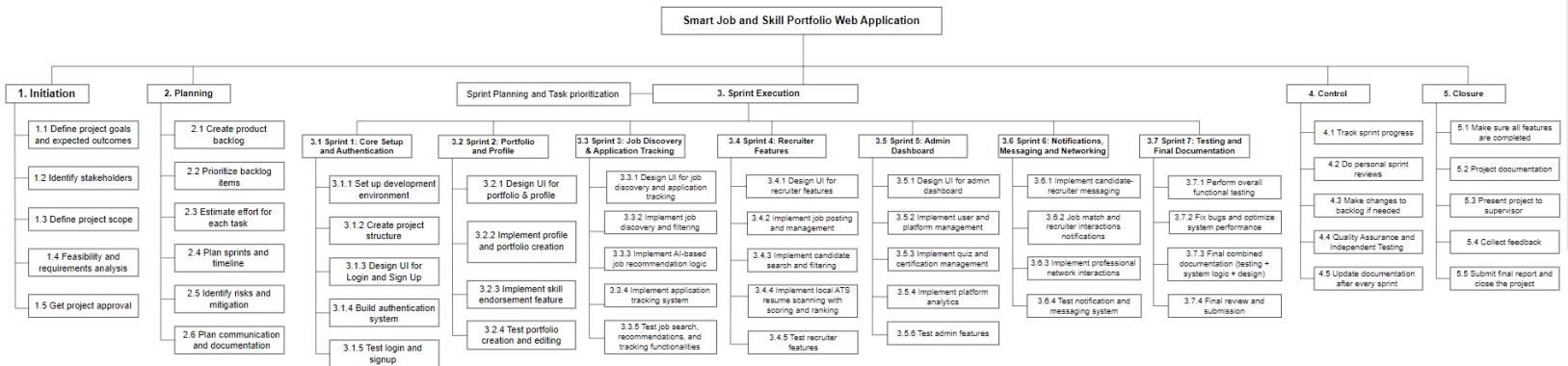


Figure 7: Work Breakdown Structure

[For clear and larger view of WBS, Click here.](#)

### 3.5. Gantt Chart (Revised Gantt Chart)

The project timeline was revised because Sprint 2 required more time than expected. This sprint includes creating portfolio and profile features for three different users (Candidate, Recruiter, and Admin), which took extra effort to design and implement properly. Along with this, academic workload and other submissions also affected the available development time. Therefore, the schedule was adjusted to ensure the work is completed correctly and with good quality.

[For Previous Gantt Chart, refer to Appendix.](#)

#### i. Initiation Phase

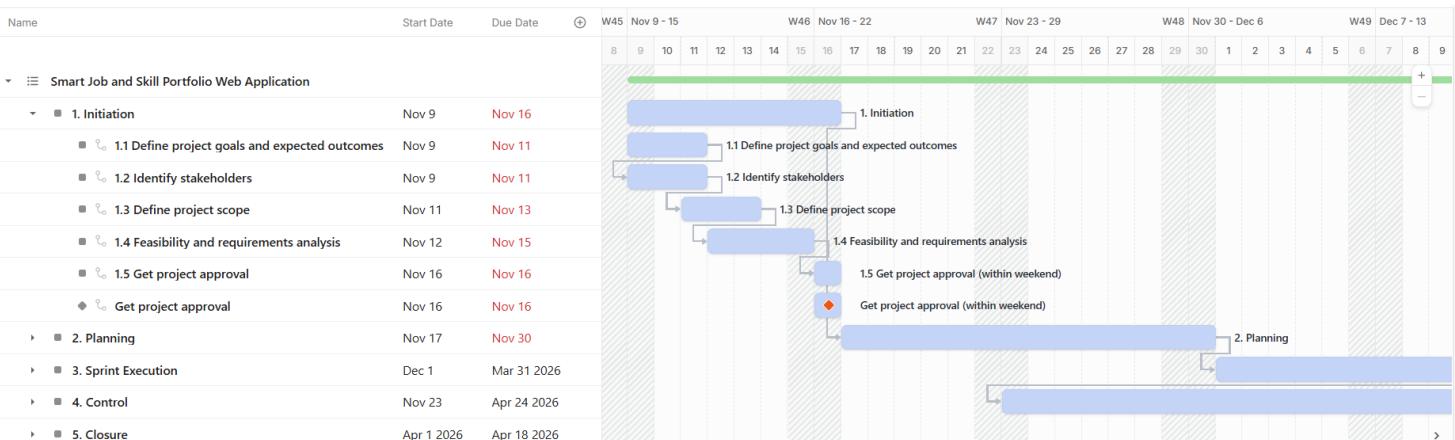


Figure 8: Revised Gantt Chart Initiation Phase

## ii. Planning Phase

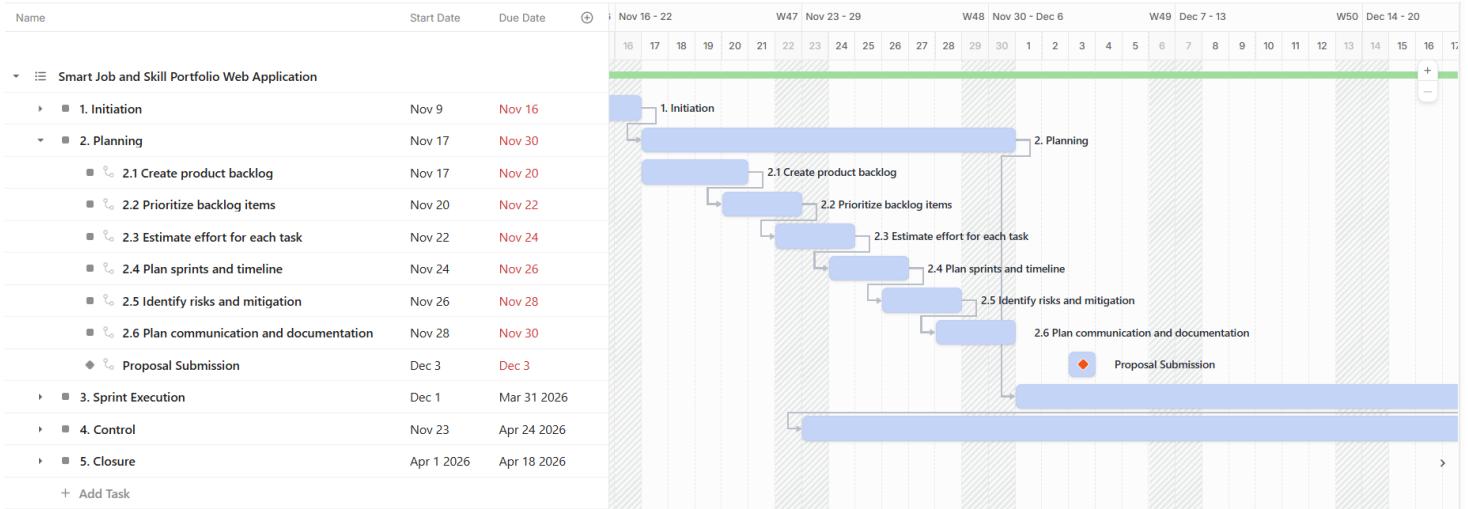


Figure 9: Revised Gantt Chart Planning Phase

## iii. Sprint Execution Phase (Sprint 1: Core Setup and Authentication)

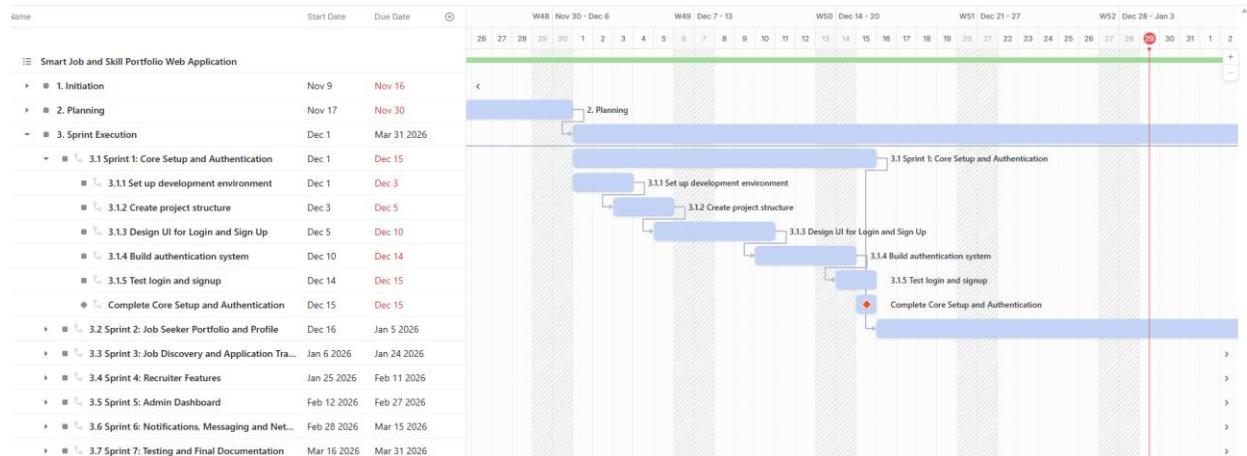


Figure 10: Revised Gantt Chart Sprint Execution Phase Sprint 1 Core Setup and Authentication

#### iv. Sprint Execution Phase (Sprint 2: Portfolio and Profile)

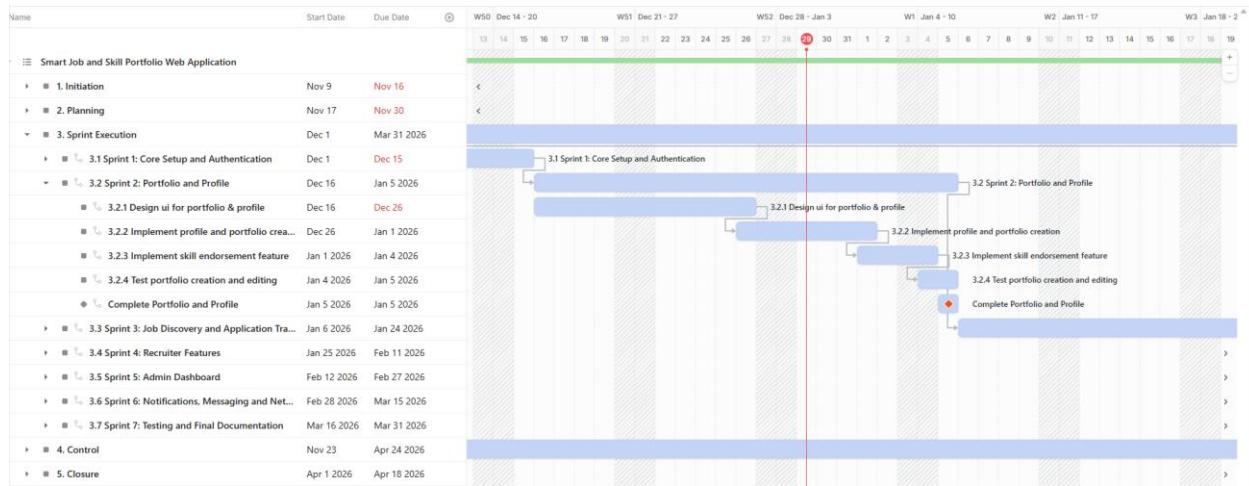


Figure 11: Revised Gantt Chart Sprint Execution Phase Sprint 2 Portfolio and Profile

#### v. Sprint Execution Phase (Sprint 3: Job Discovery and Application Tracking)

##### Tracking

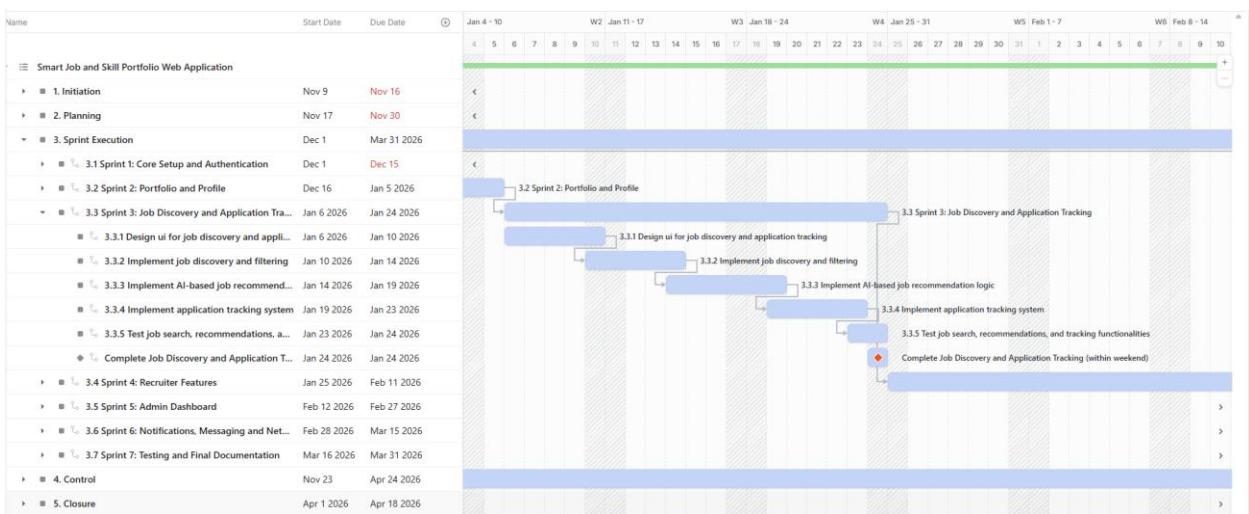


Figure 12: Revised Gantt Chart Sprint Execution Phase Sprint 3 Job Discovery and Application Tracking

## vi. Sprint Execution Phase (Sprint 4: Recruiter Features)

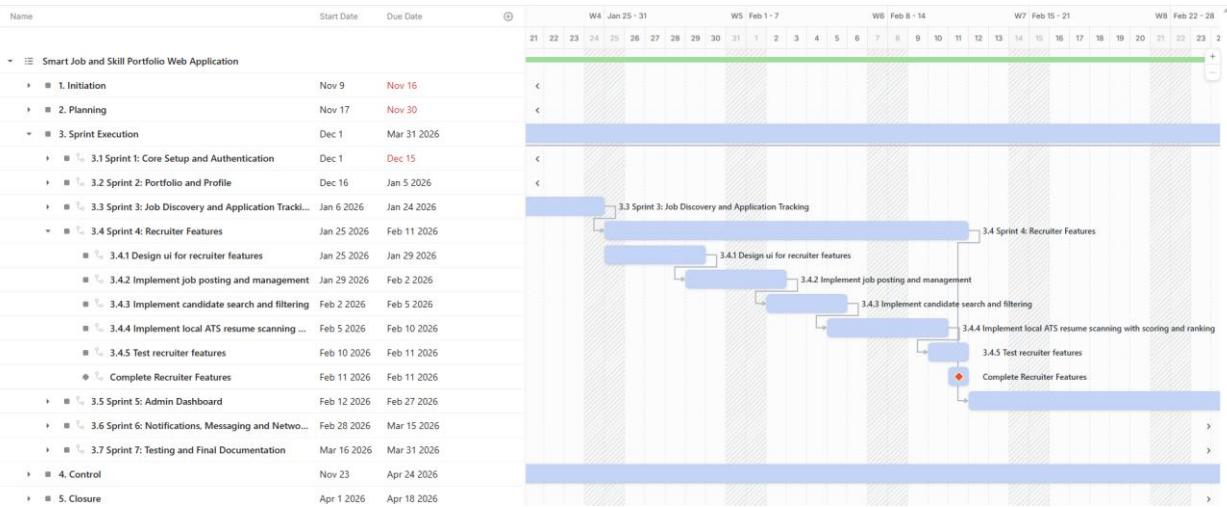


Figure 13: Revised Gantt Chart Sprint Execution Phase Sprint 4 Recruiter Features

## vii. Sprint Execution Phase (Sprint 5: Admin Dashboard)

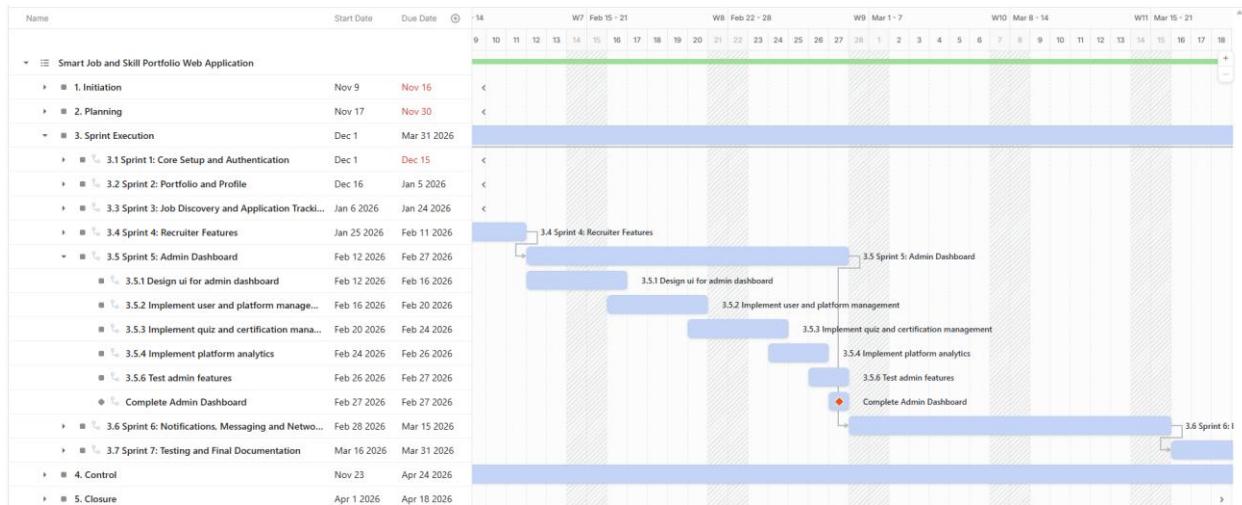


Figure 14: Revised Gantt Chart Sprint Execution Phase Sprint 5 Admin Dashboard

### viii. Sprint Execution Phase (Sprint 6: Notifications, Messaging and Networking)

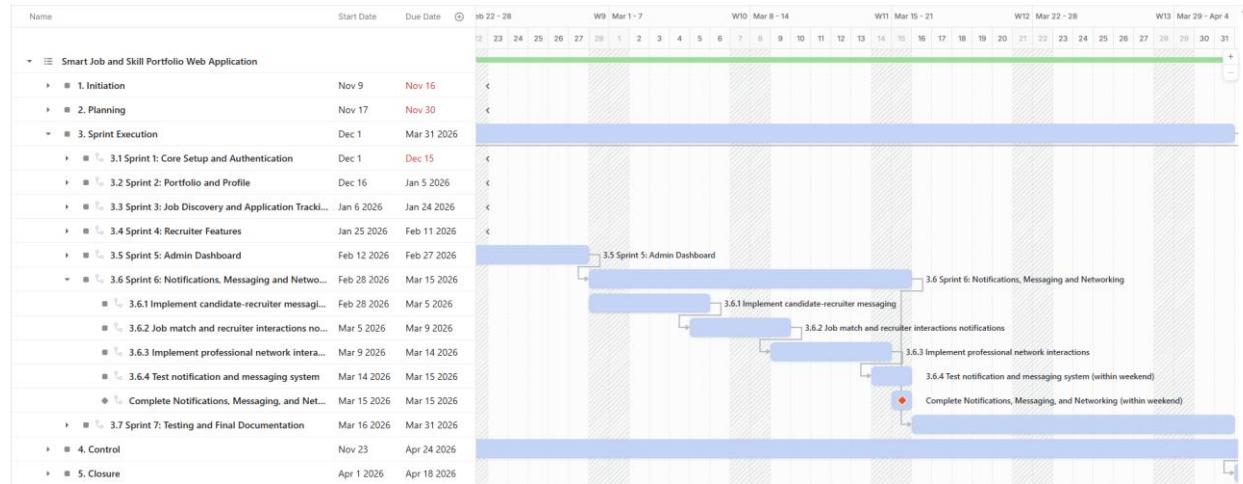


Figure 15: Revised Gantt Chart Sprint Execution Phase Sprint 6 Notifications, Messaging and Networking

### ix. Sprint Execution Phase (Sprint 7: Testing and Final Documentation)

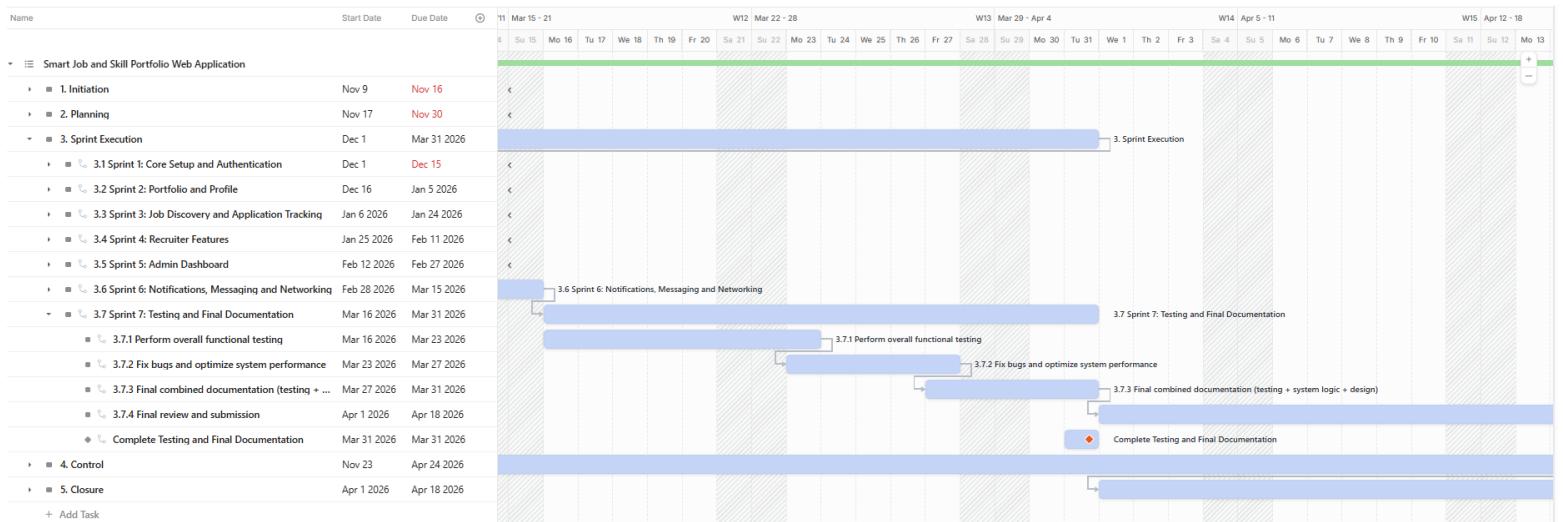


Figure 16: Revised Gantt Chart Sprint Execution Phase Sprint 7 Testing and Final Documentation

## x. Control Phase

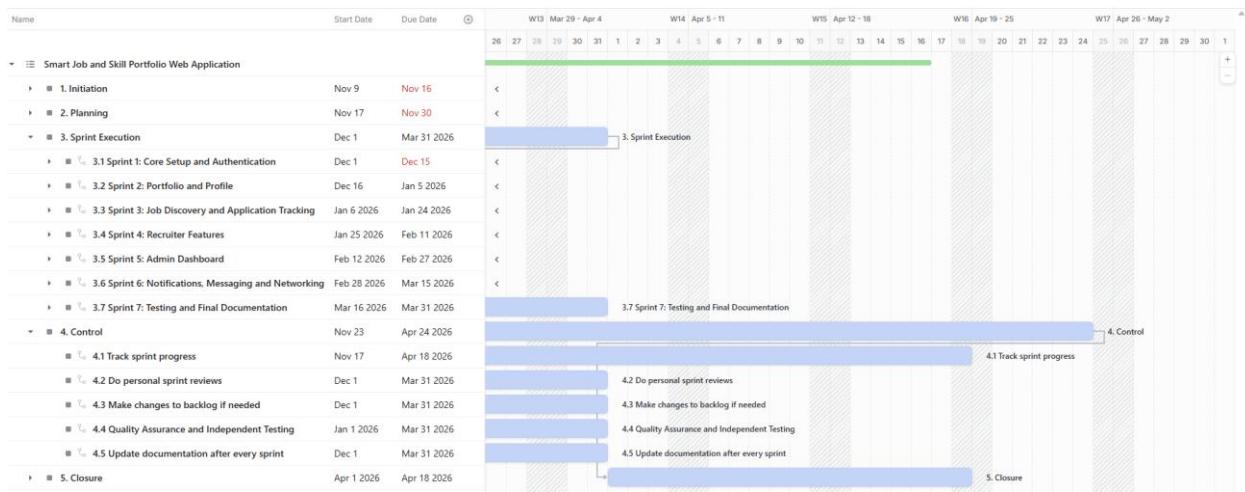


Figure 17: Revised Gantt Chart Control Phase

## xi. Closure Phase

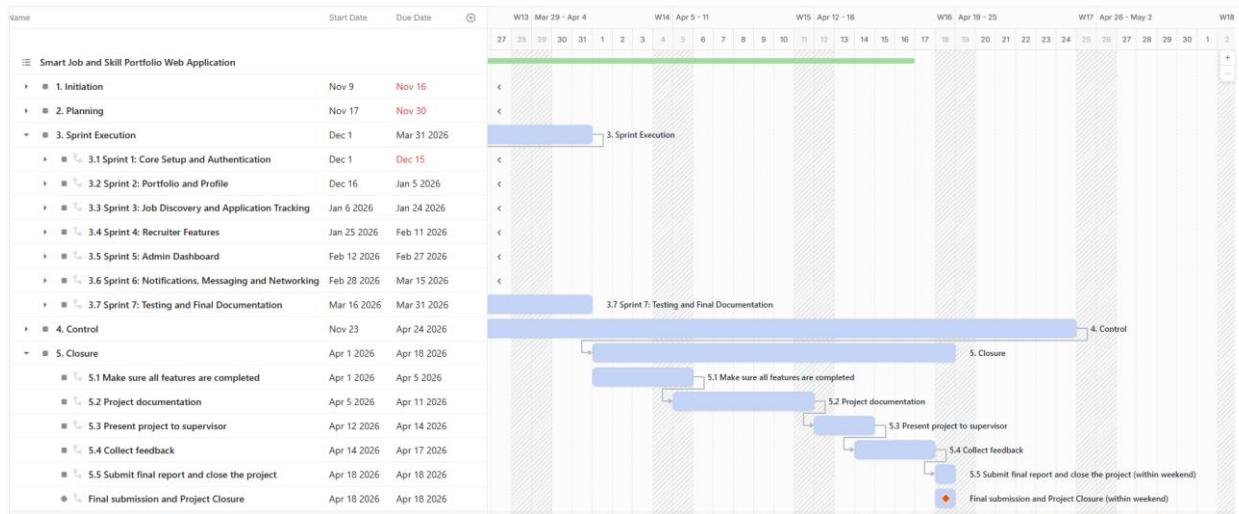
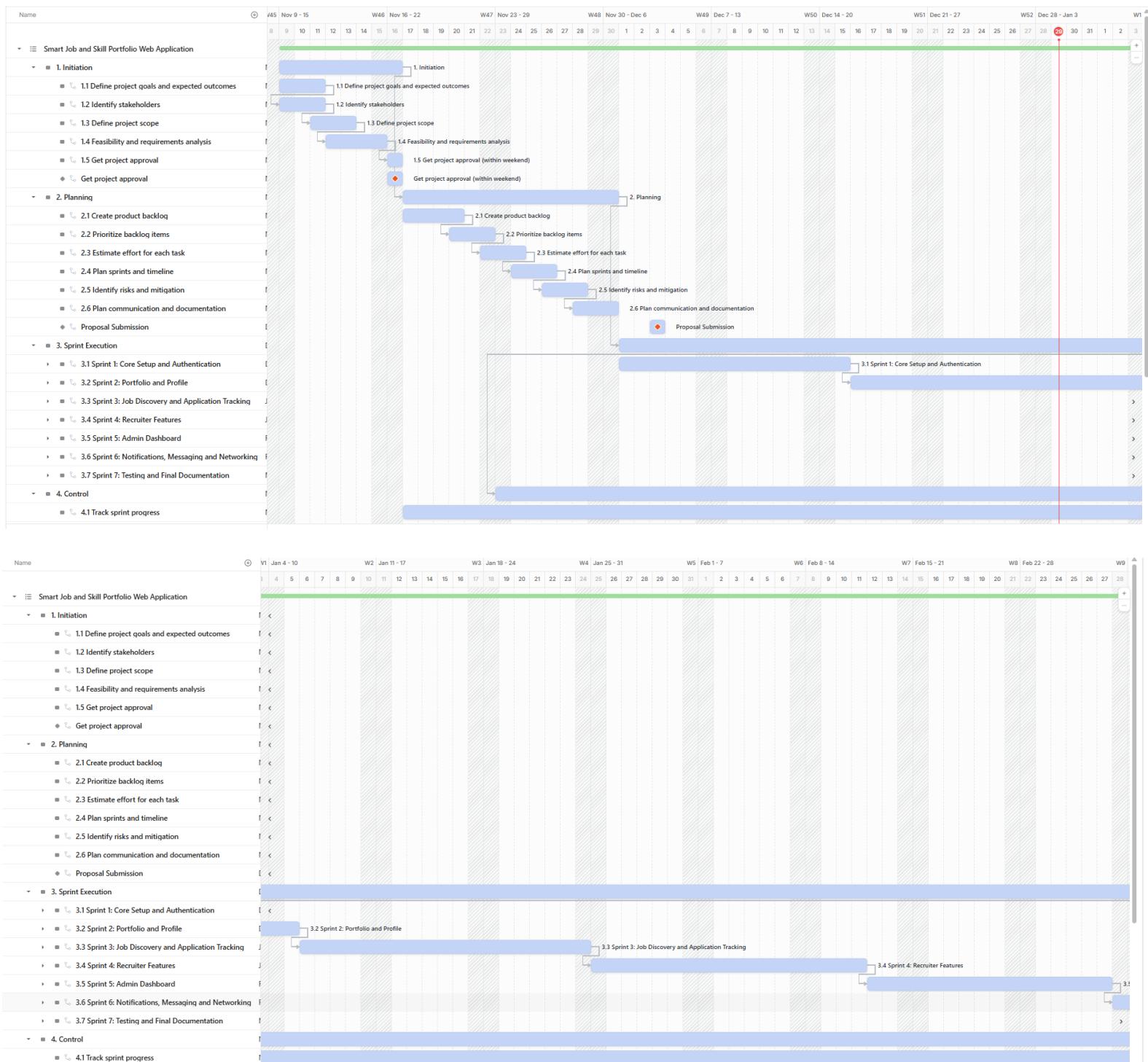
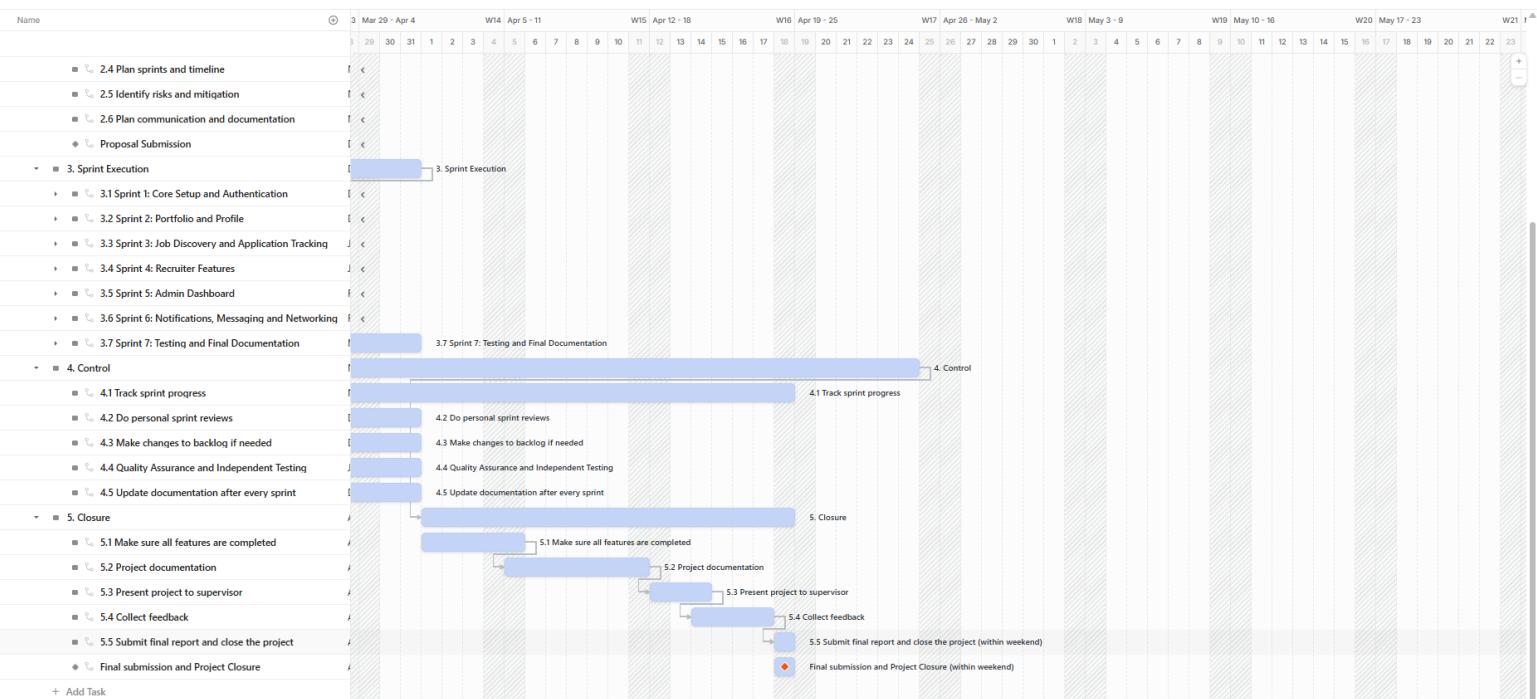
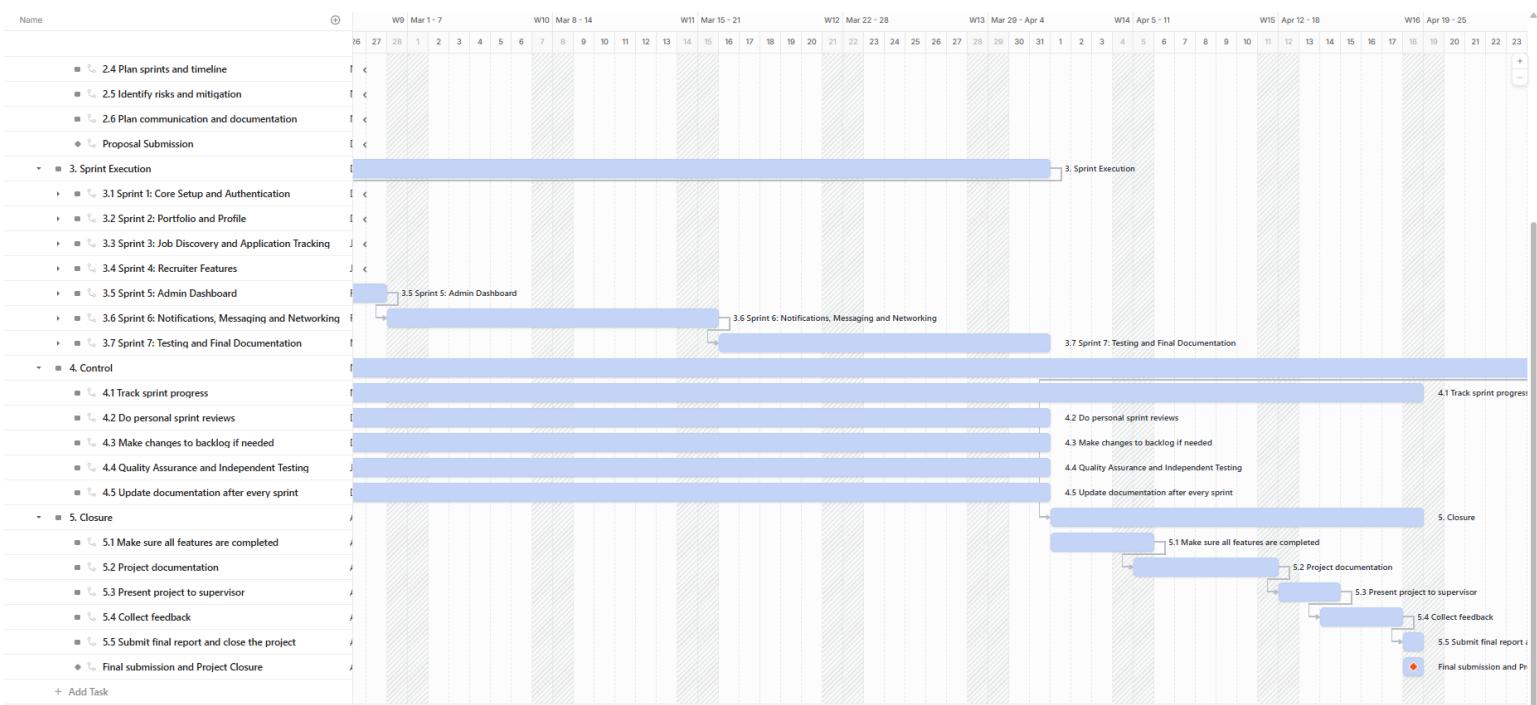
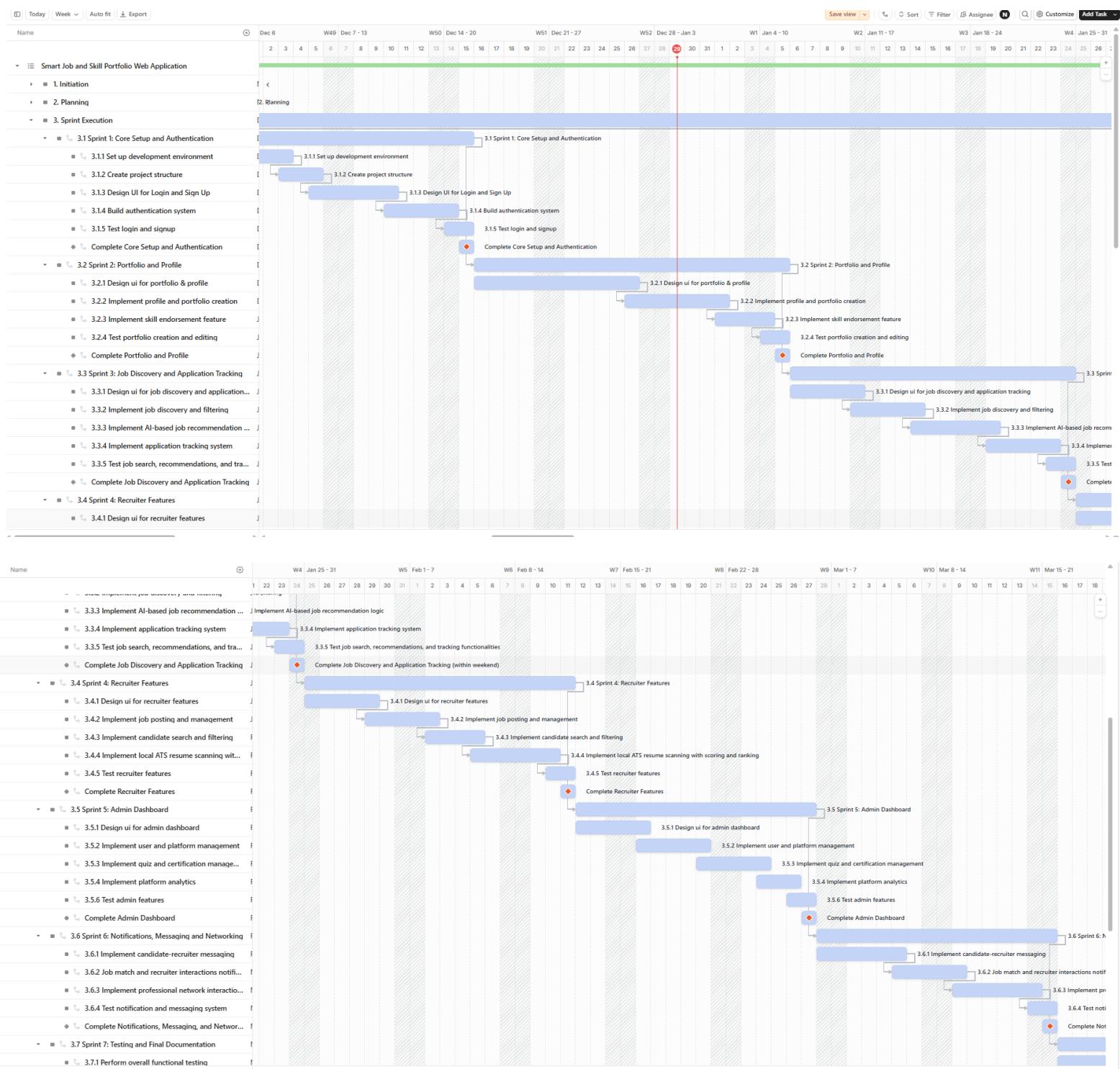


Figure 18: Revised Gantt Chart Closure Phase

## xii. Full View of Gantt Chart







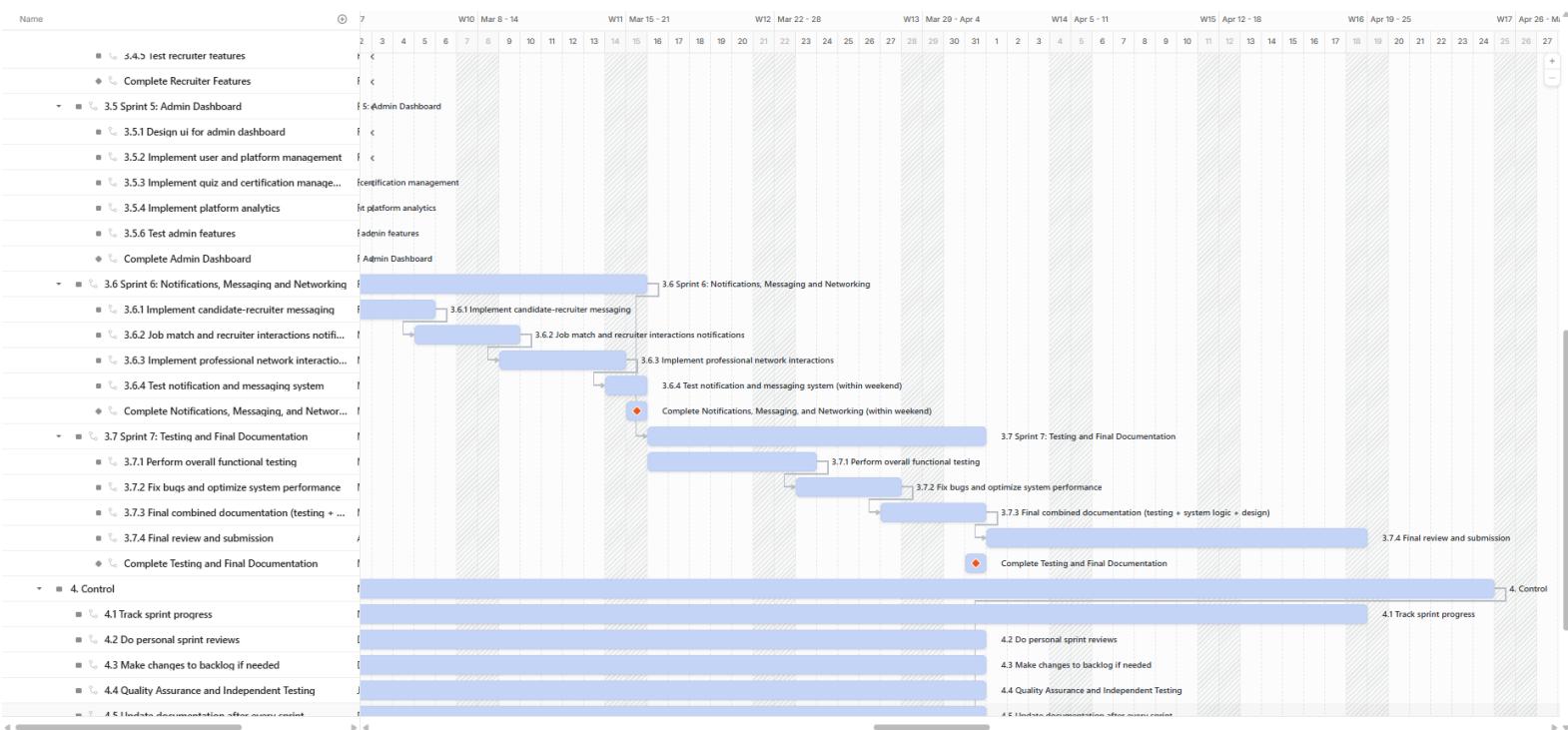


Figure 19: Full View of Gantt Chart

[For clear and larger view of Gantt Chart, Click here.](#)

## 4. Development Till Date

### 4.1. System Architecture Diagram

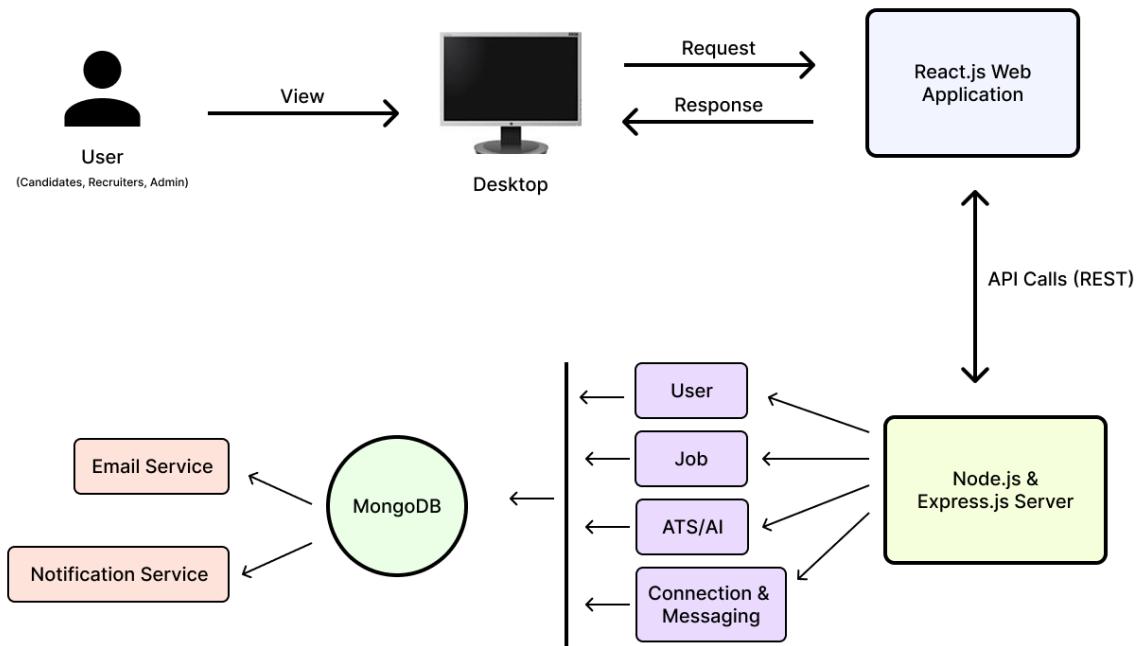


Figure 20: System Architecture Diagram

### 4.2. Software Requirements Specifications (SRS)

[For a detailed explanation of Software Requirement Specifications \(SRS\), refer to Appendix.](#)

### 4.3. Use Case Diagram

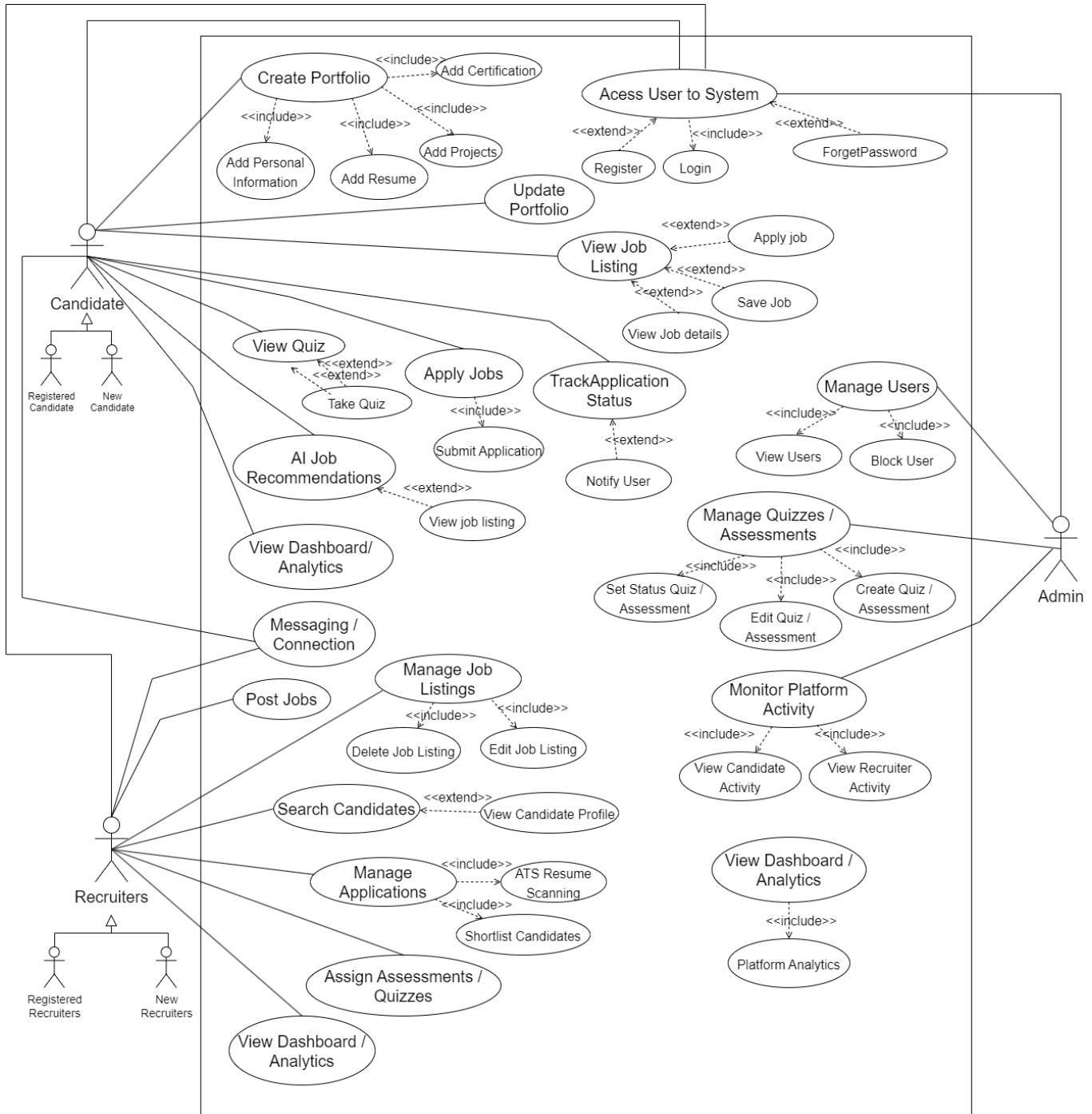


Figure 21: Use Case Diagram

## 4.4. Expanded Description

[For Expanded Description, refer to Appendix](#)

## 4.5. Sequence Diagram

### 4.5.1. Sequence Diagram for User Authentication (Access User to System)

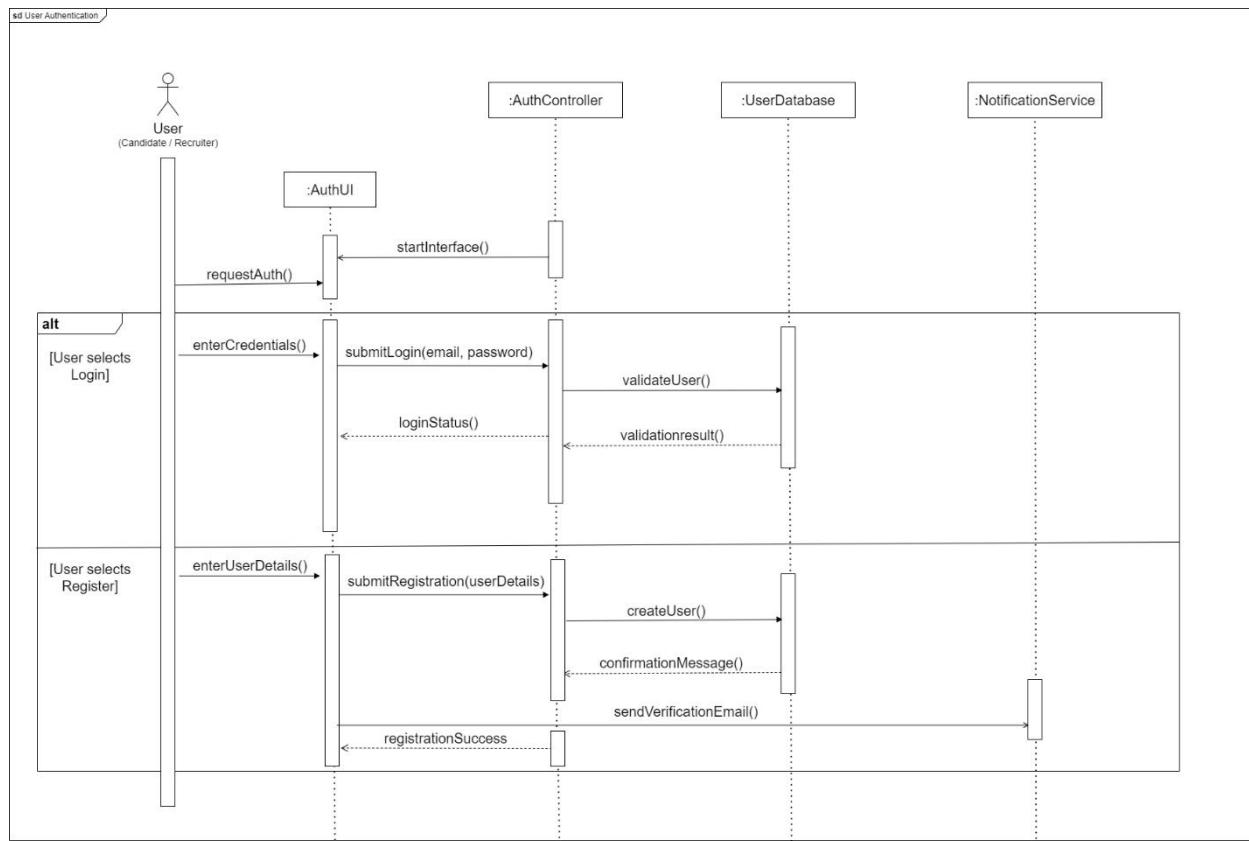


Figure 22: Sequence Diagram for User Authentication

#### 4.5.2. Sequence Diagram for Create or Update Portfolio (Candidate)

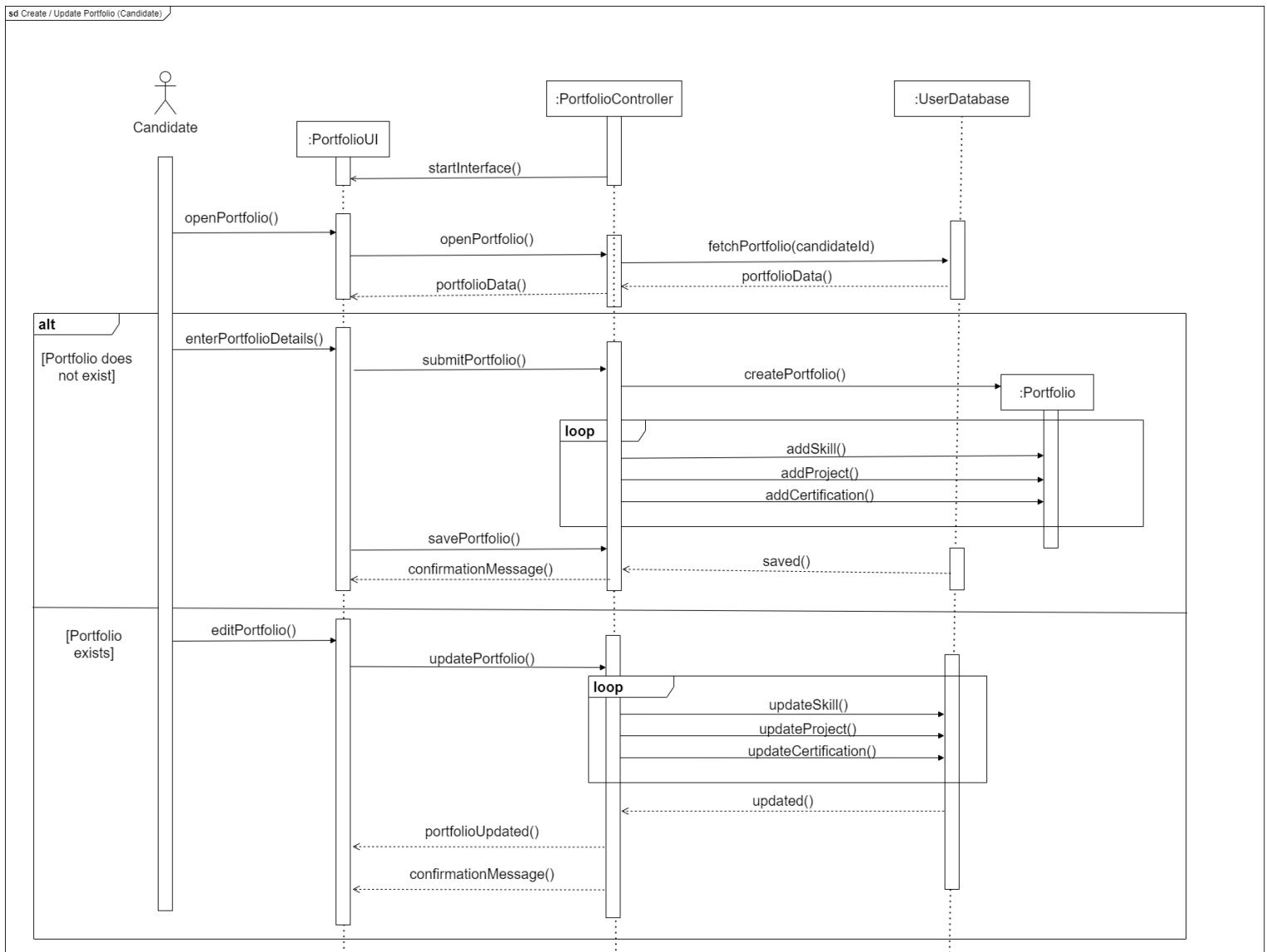


Figure 23: Sequence Diagram for Create or Update Portfolio (Candidate)

#### 4.5.3. Sequence Diagram for Job Apply and Application Tracking

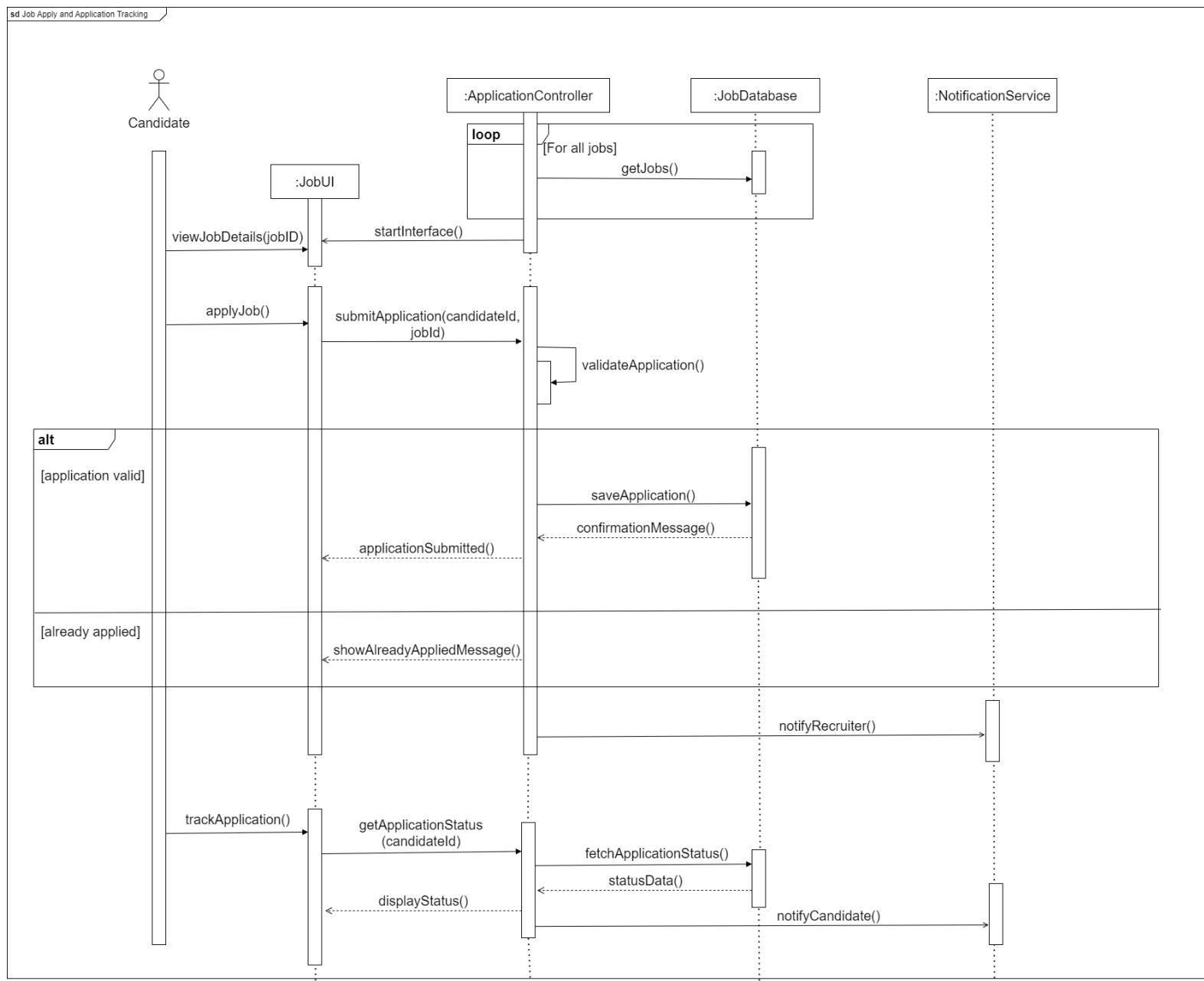


Figure 24: Sequence Diagram for Job Apply and Application Tracking

#### 4.5.4. Sequence Diagram for AI Job Recommendations (Candidate)

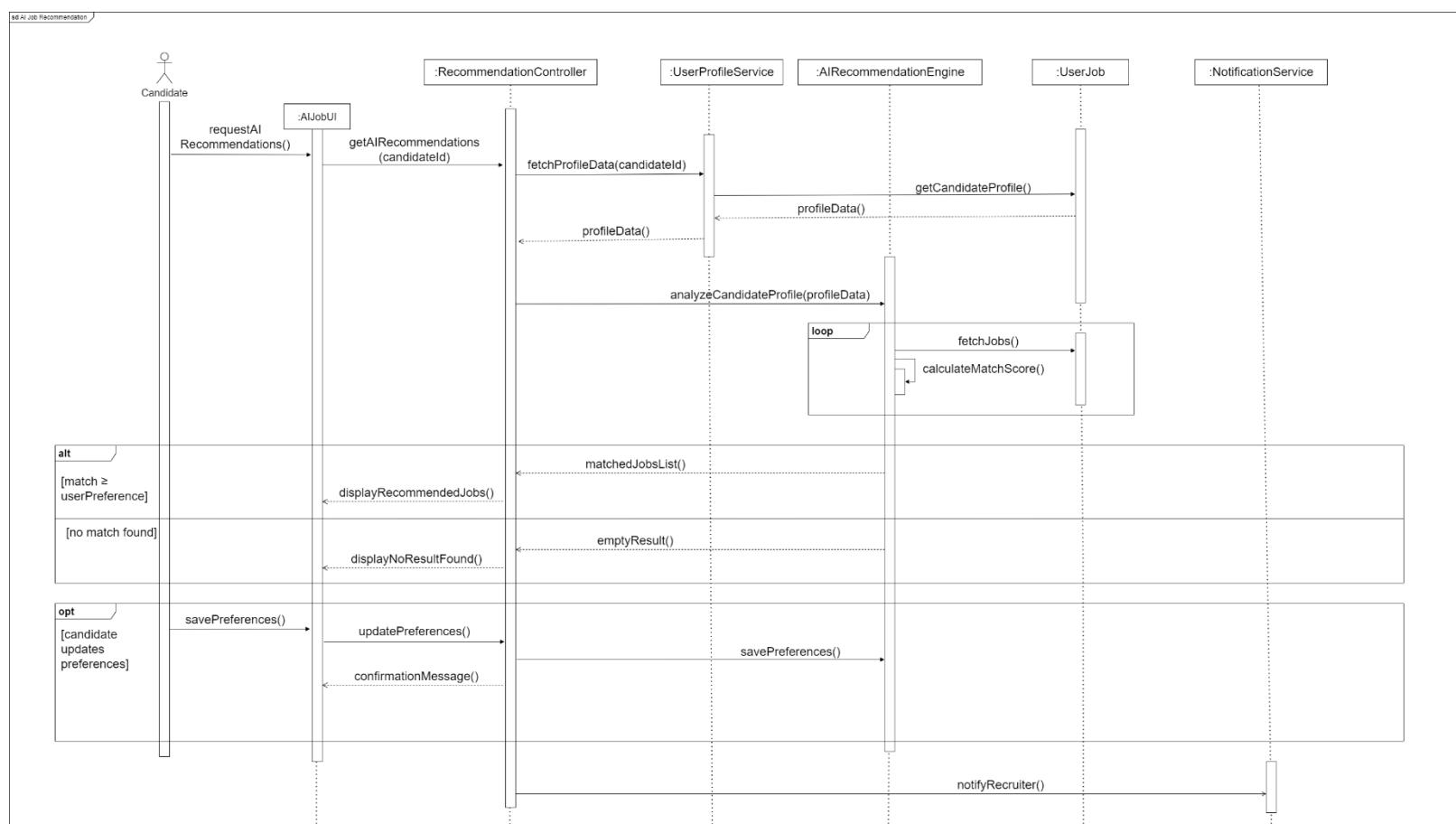


Figure 25: Sequence Diagram for AI Job Recommendations (Candidate)

#### 4.5.5. Sequence Diagram for Connection and Messaging

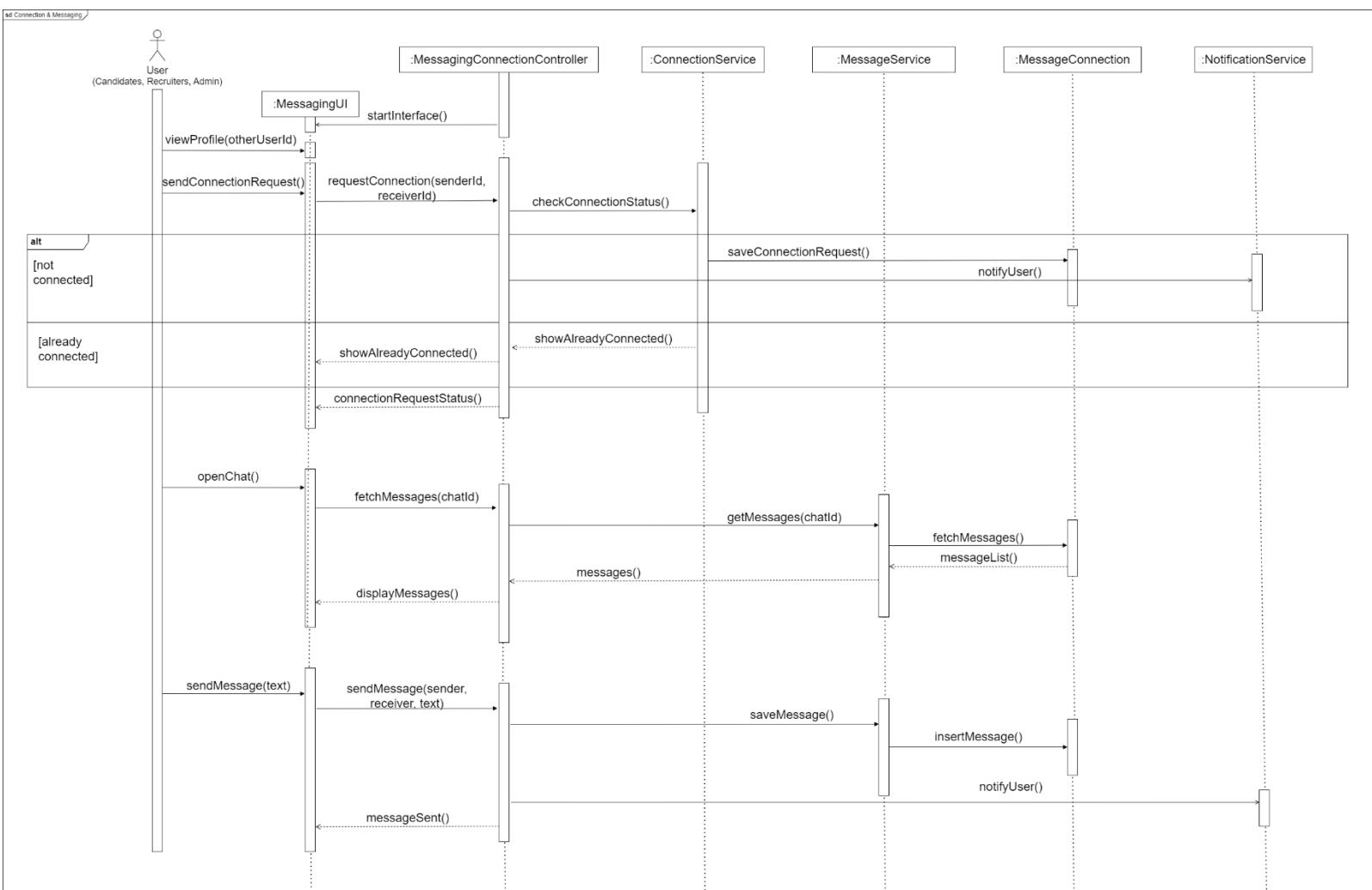


Figure 26: Sequence Diagram for Connection and Messaging

#### 4.5.6. Sequence Diagram for Job Posting and Manage Application (ATS Resume scanner)

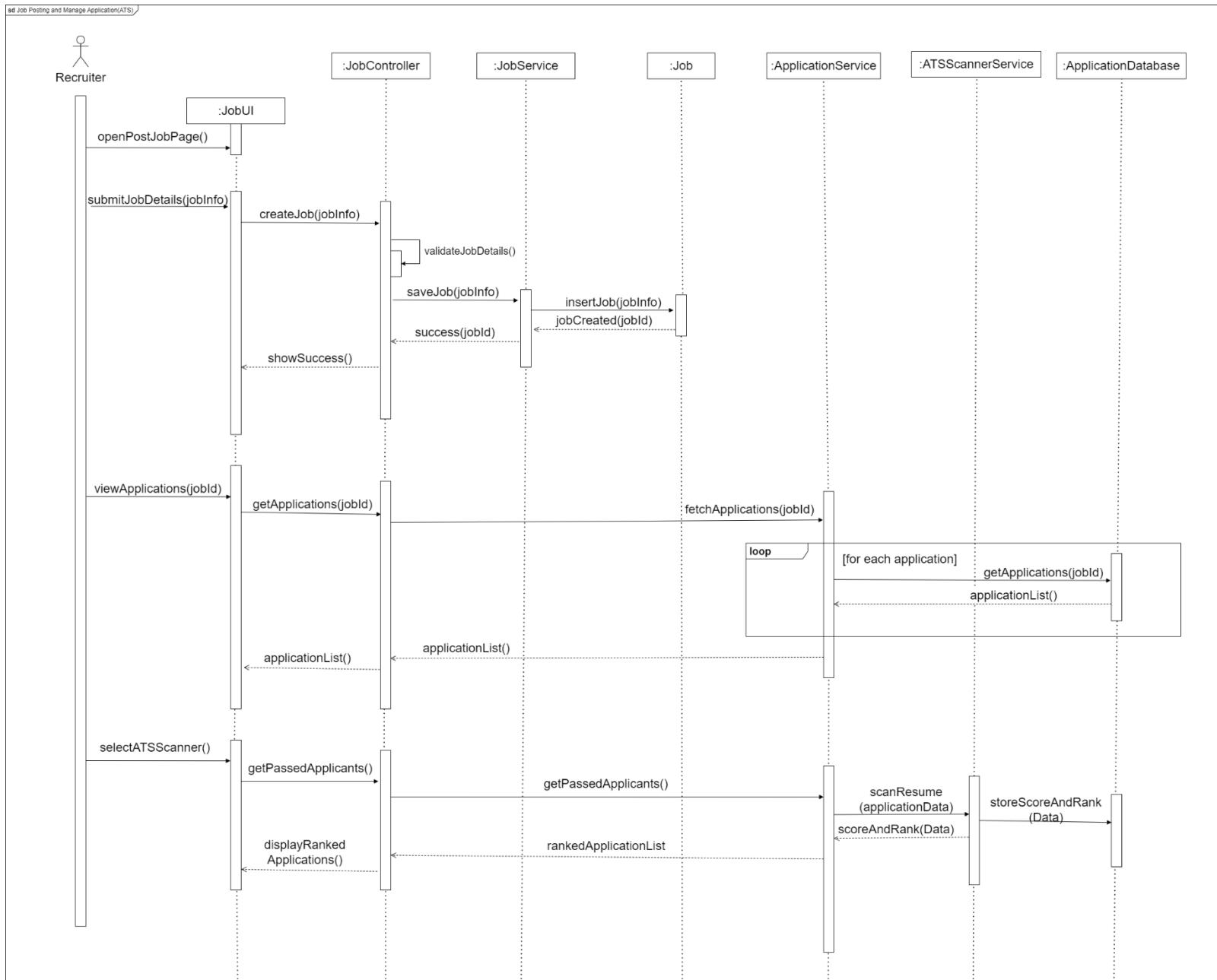


Figure 27: Sequence Diagram for Job Posting and Manage Application (ATS Resume scanner)

#### 4.5.7. Sequence Diagram for Search Candidates and View Profile

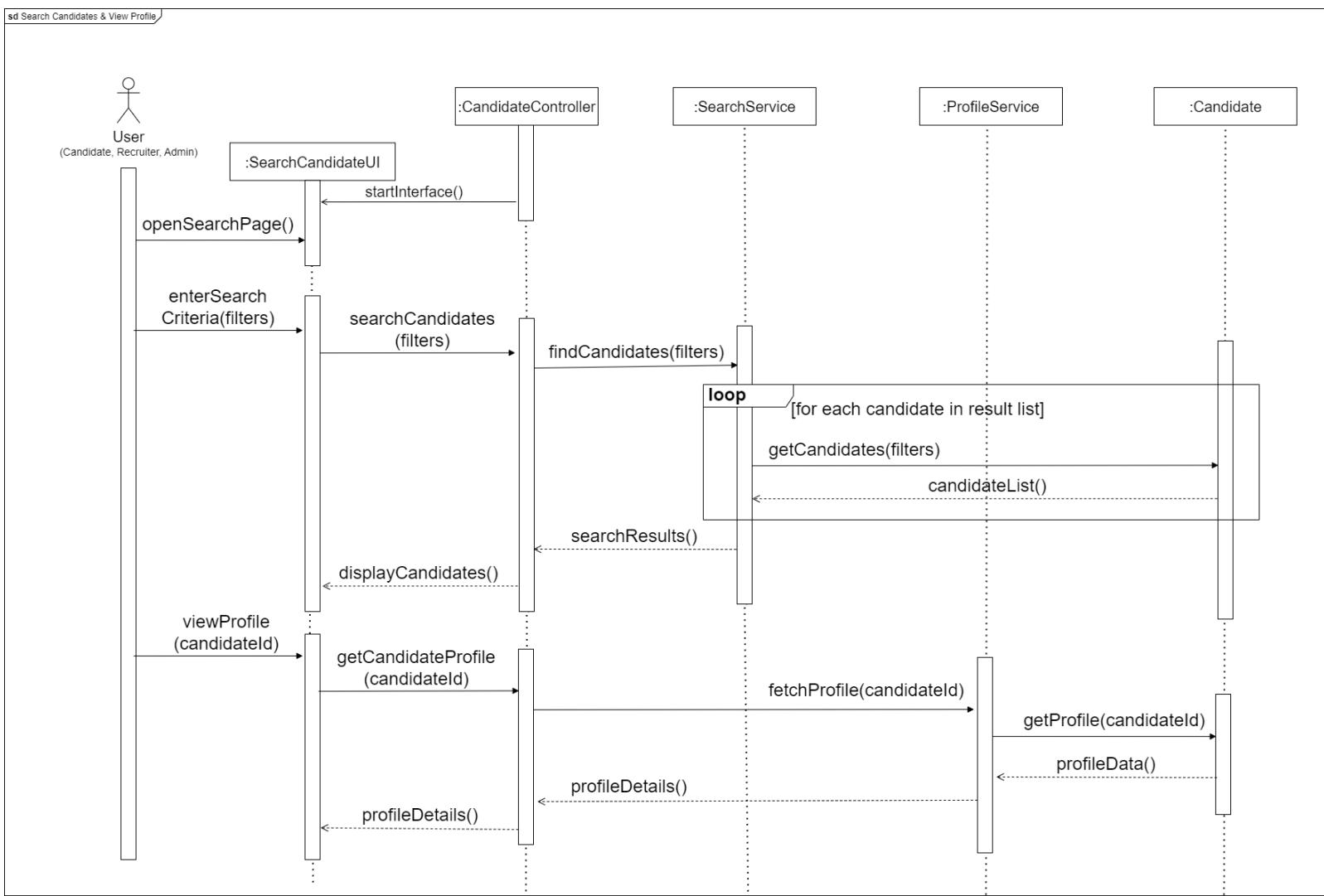


Figure 28: Sequence Diagram for Search Candidates and View Profile

#### 4.5.8. Sequence Diagram for Manage Users and Quizzes / Assessments

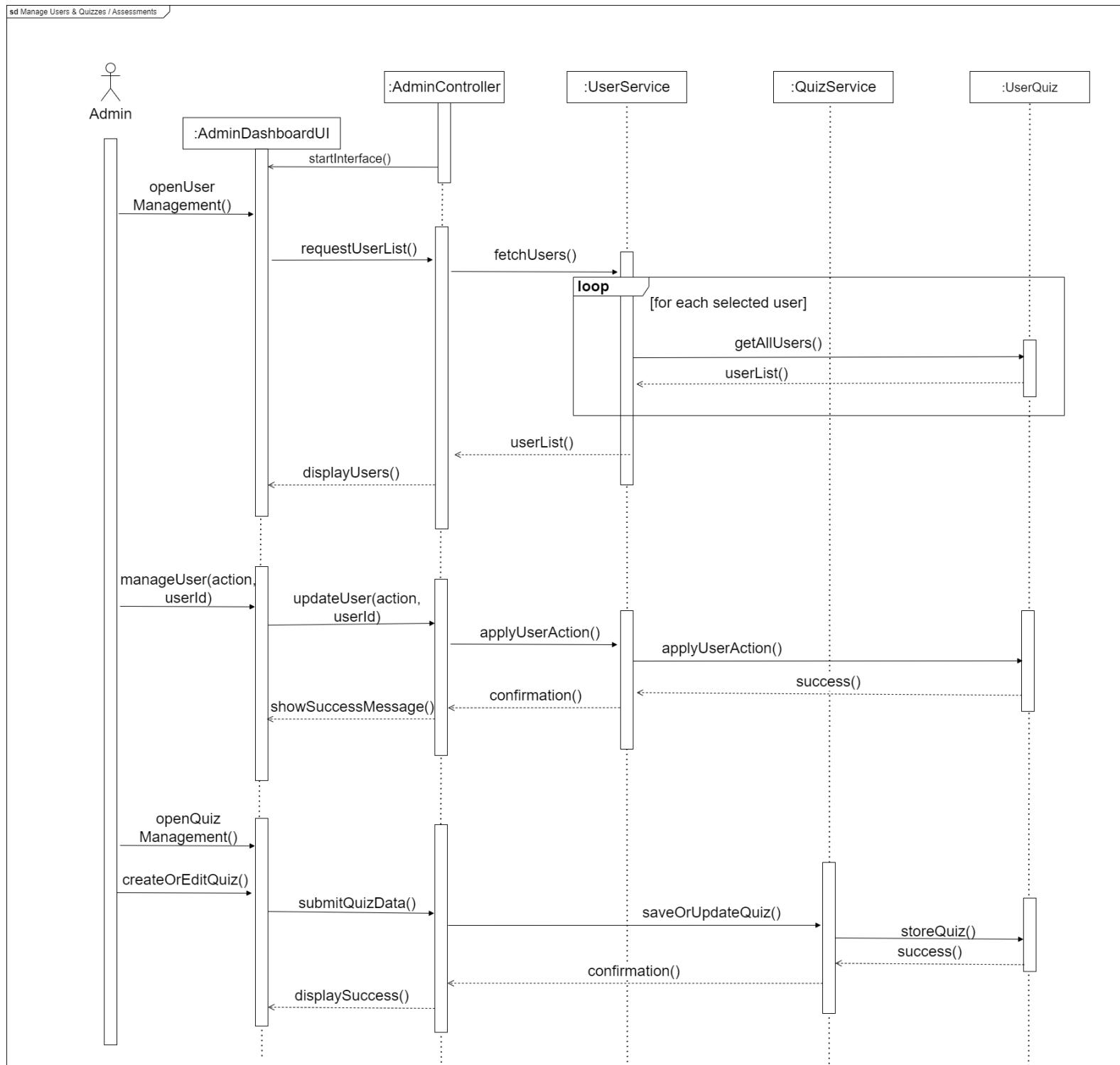


Figure 29: Sequence Diagram for Manage Users and Quizzes / Assessments

## 4.6. Collaboration Diagram

### 4.6.1. Collaboration Diagram for User Authentication (Access User to System)

#### 4.6.1.1. Register

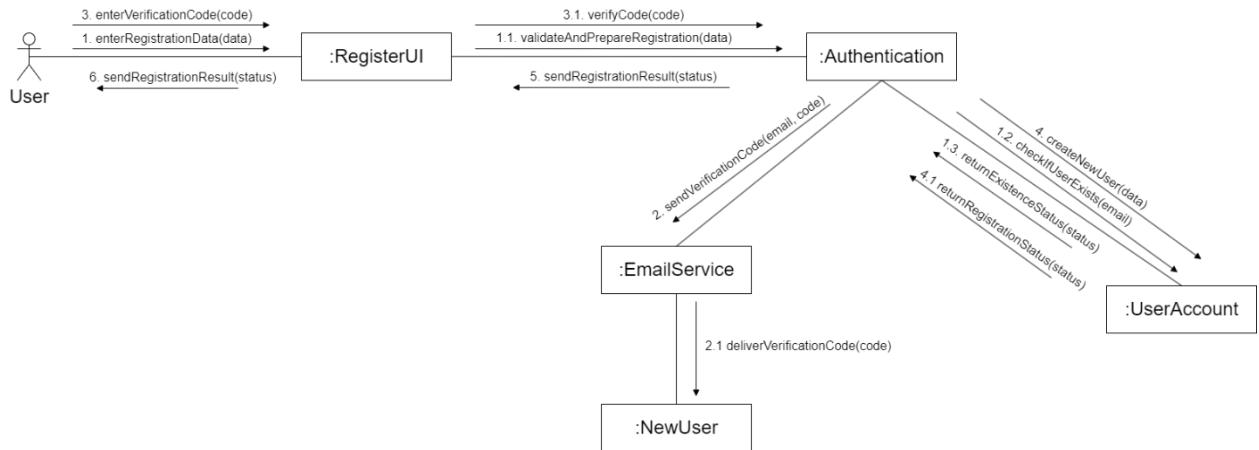


Figure 30: Collaboration Diagram of Register

#### 4.6.1.2. Login

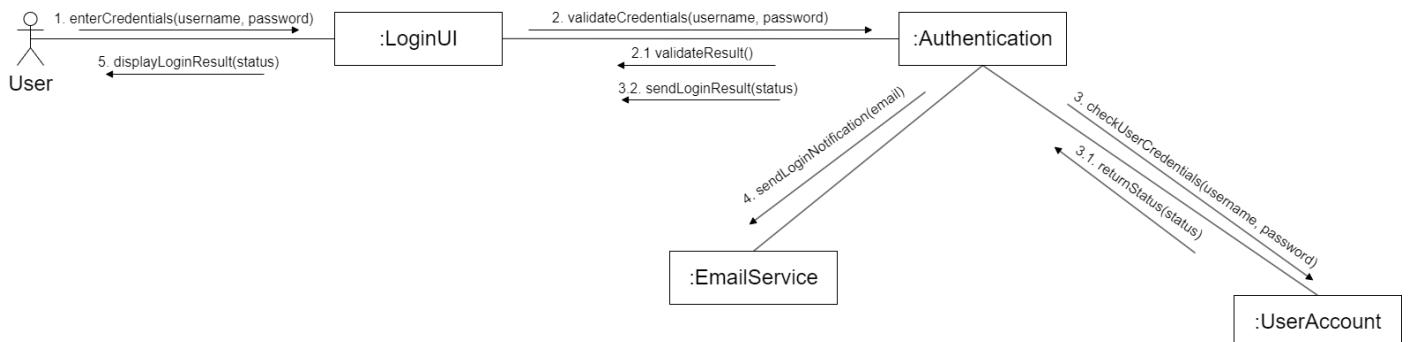


Figure 31: Collaboration Diagram of Login

#### 4.6.2. Collaboration Diagram for Create or Update Portfolio (Candidate)

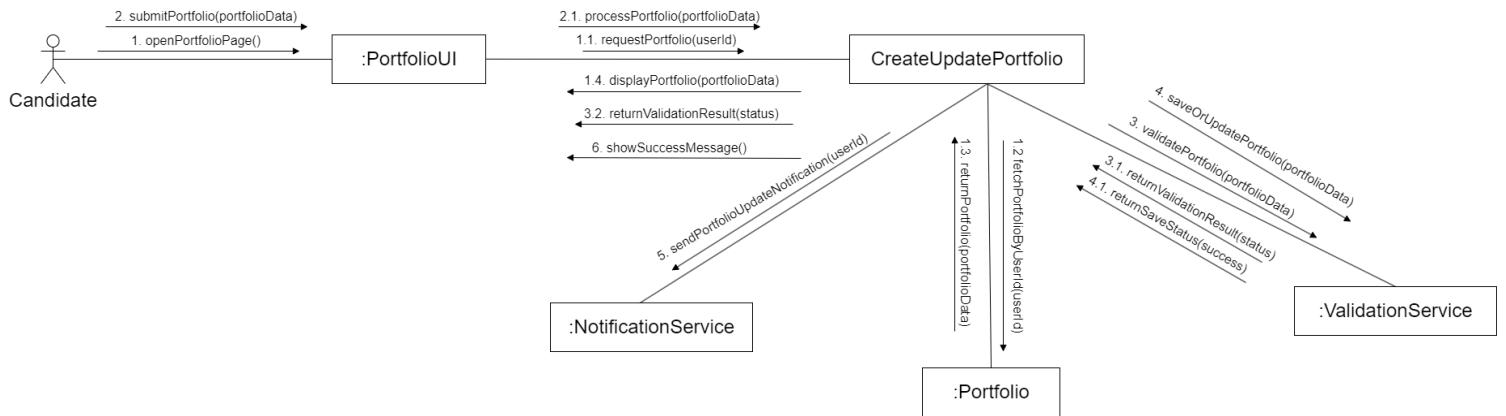


Figure 32: Collaboration Diagram for Create or Update Portfolio (Candidate)

#### 4.6.3. Collaboration Diagram for Job Apply and Application Tracking

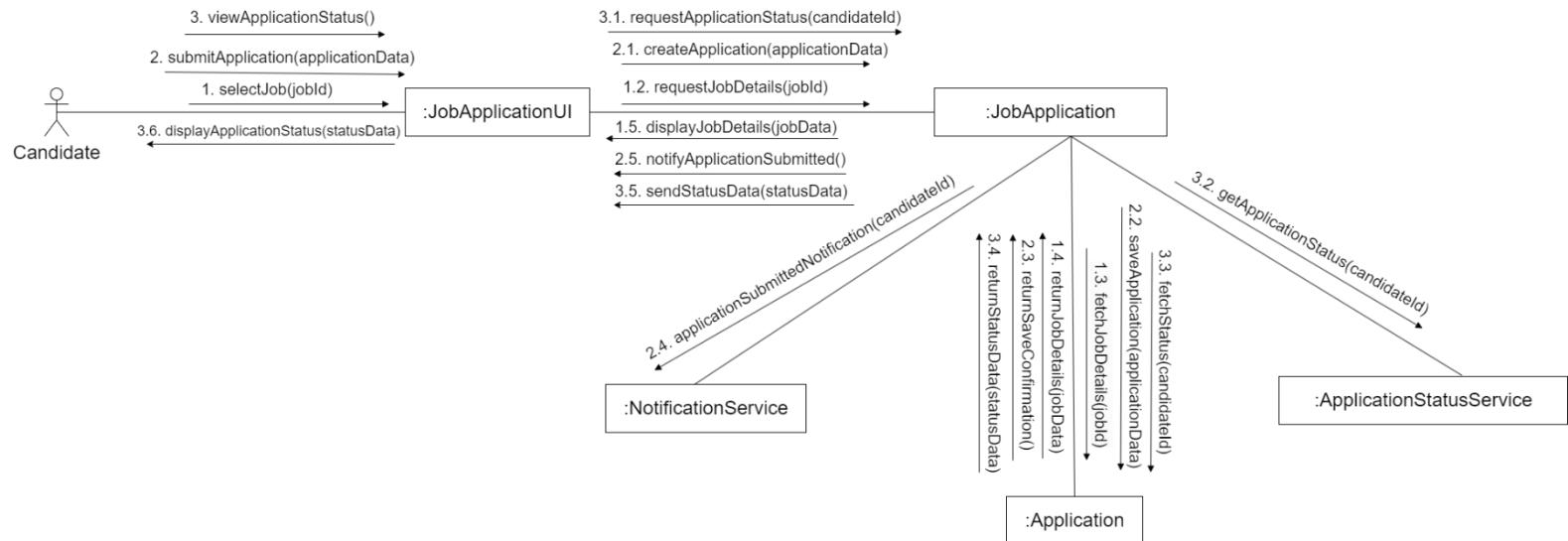


Figure 33: Collaboration Diagram for Job Apply and Application Tracking

#### 4.6.4. Collaboration Diagram for AI Job Recommendations (Candidate)

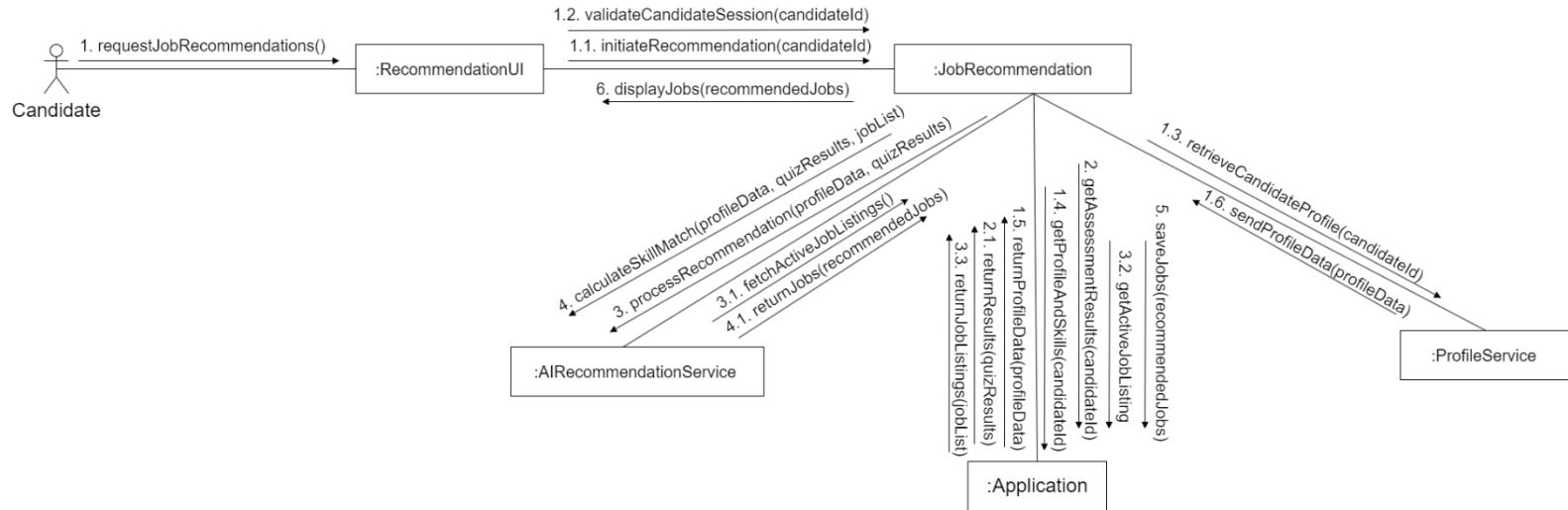


Figure 34: Collaboration Diagram for Job Apply and Application Tracking

#### 4.6.5. Collaboration Diagram for Connection and Messaging

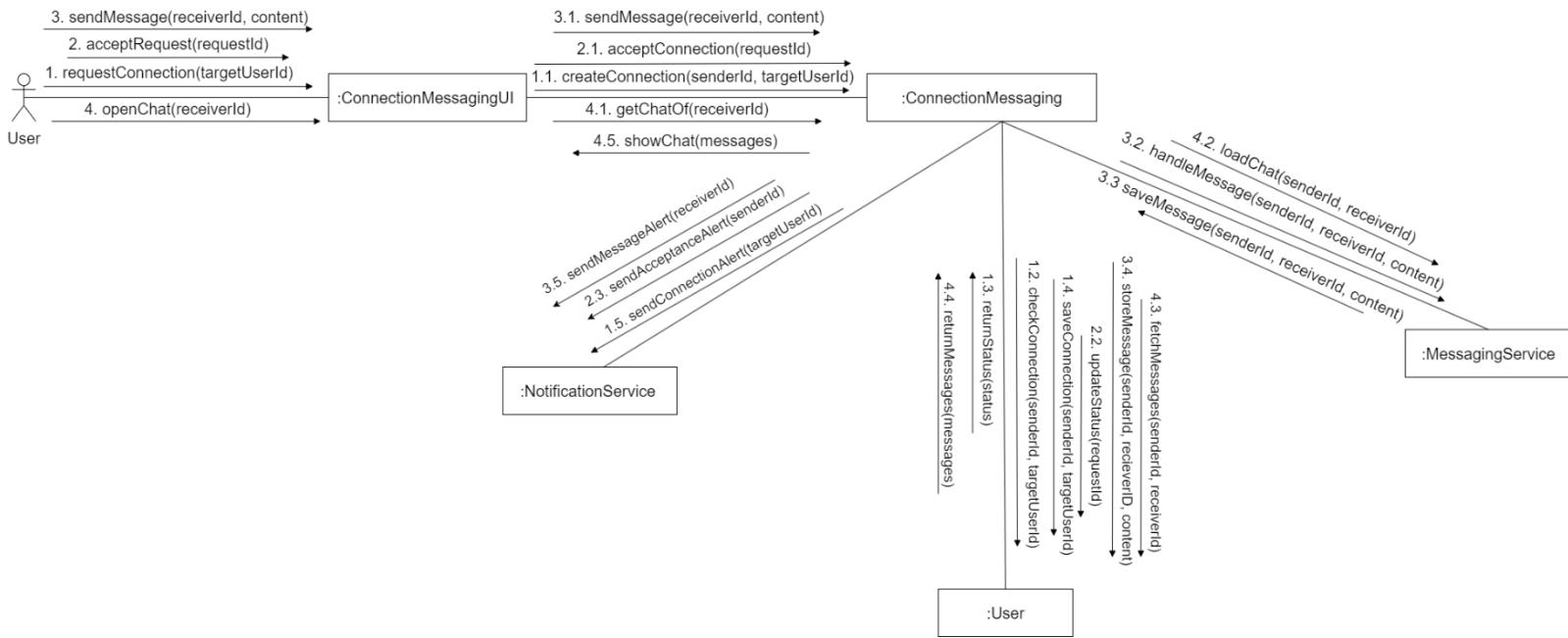


Figure 35: Collaboration Diagram for Connection and Messaging

#### 4.6.6. Collaboration Diagram for Job Posting and Manage Application (ATS Resume scanner)

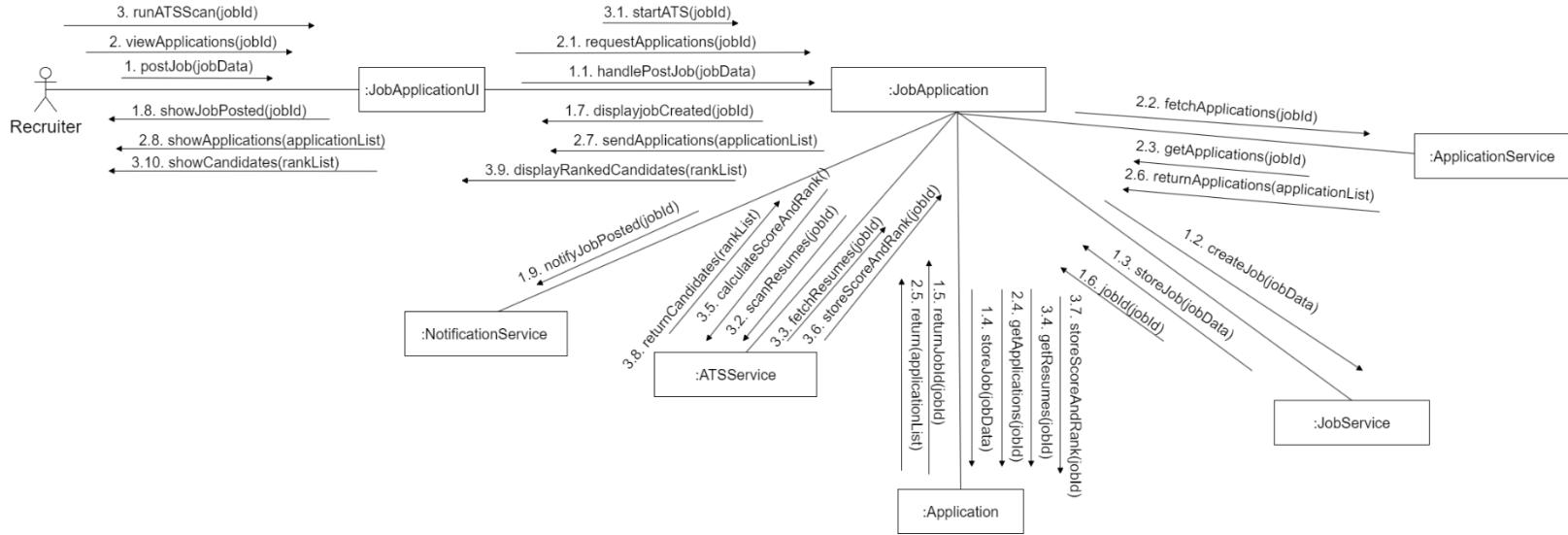


Figure 36: Collaboration Diagram for Job Posting and Manage Application (ATS Resume scanner)

#### 4.6.7. Collaboration Diagram for Search Candidates and View Profile

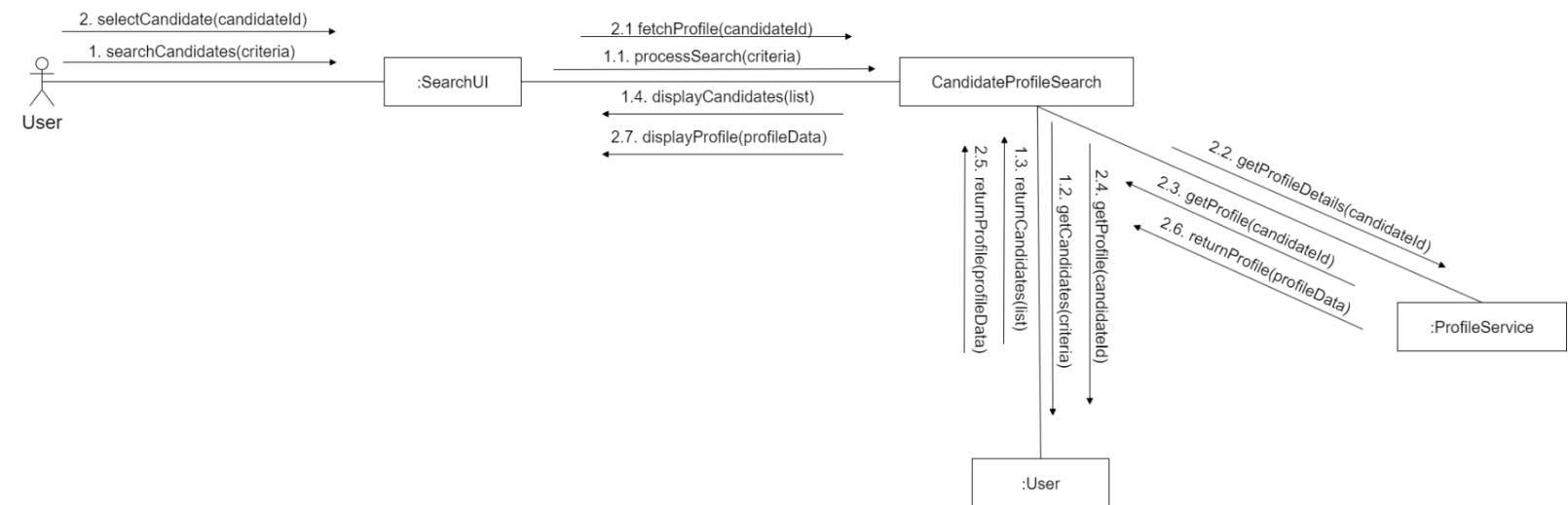


Figure 37: Collaboration Diagram for Search Candidates and View Profile

#### 4.6.8. Collaboration Diagram for Manage Users and Quizzes / Assessments

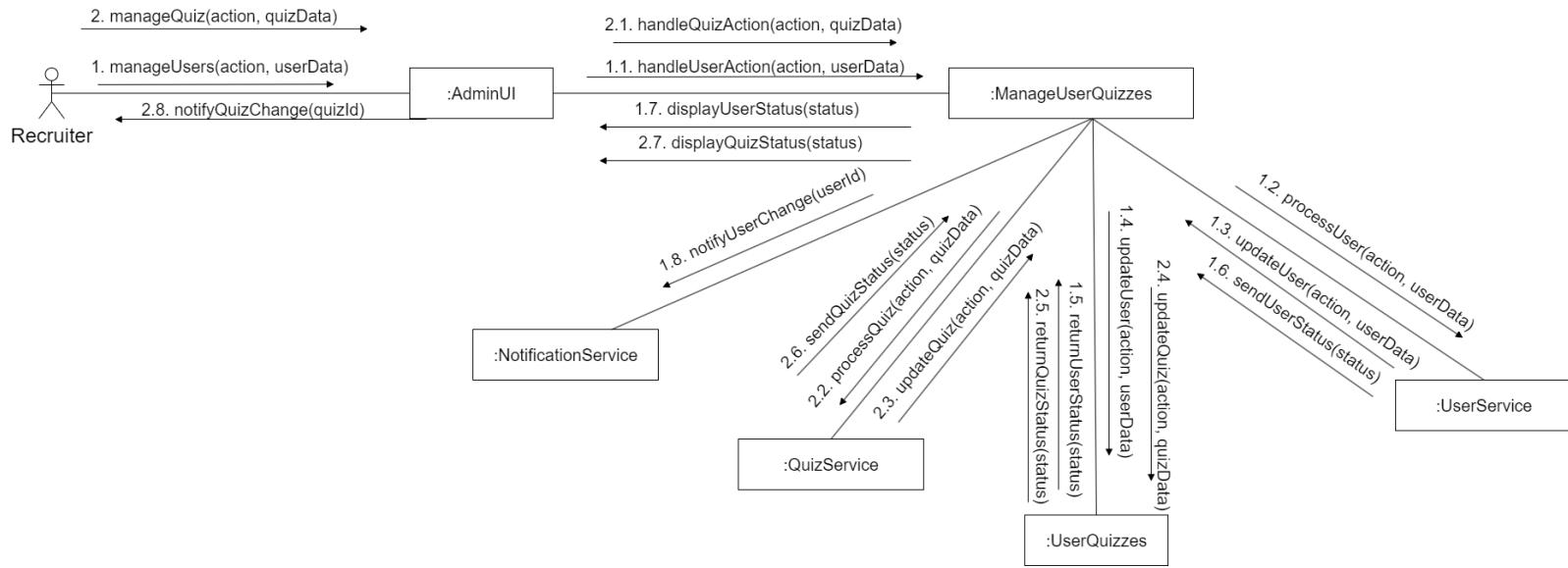


Figure 38: Collaboration Diagram for Manage Users and Quizzes / Assessments

## 4.7. Activity Diagram

### 4.7.1. Activity Diagram for User Authentication (Access User to System)

#### 4.7.1.1. Register

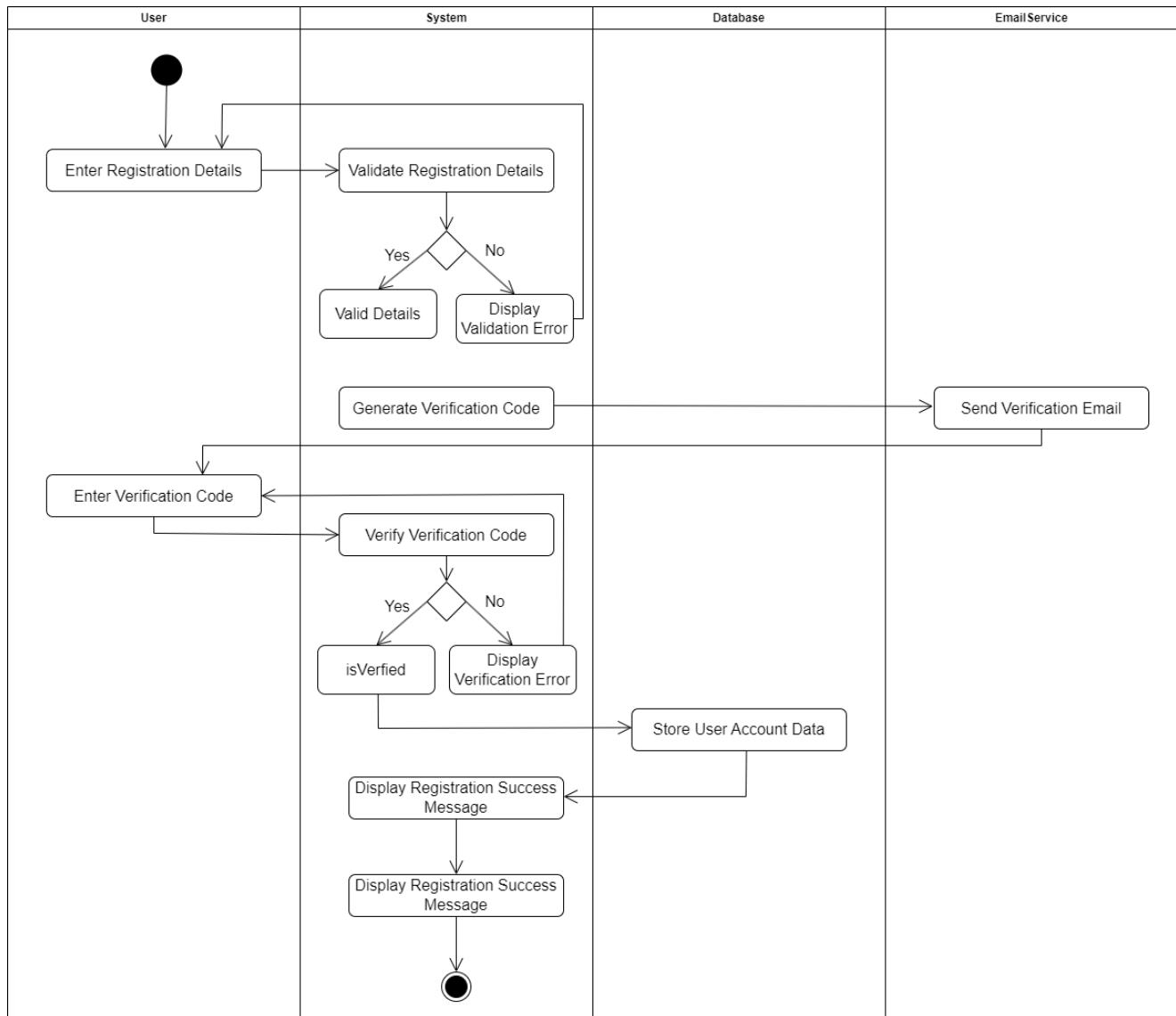


Figure 39: Activity Diagram for Register

#### 4.7.1.2. Login

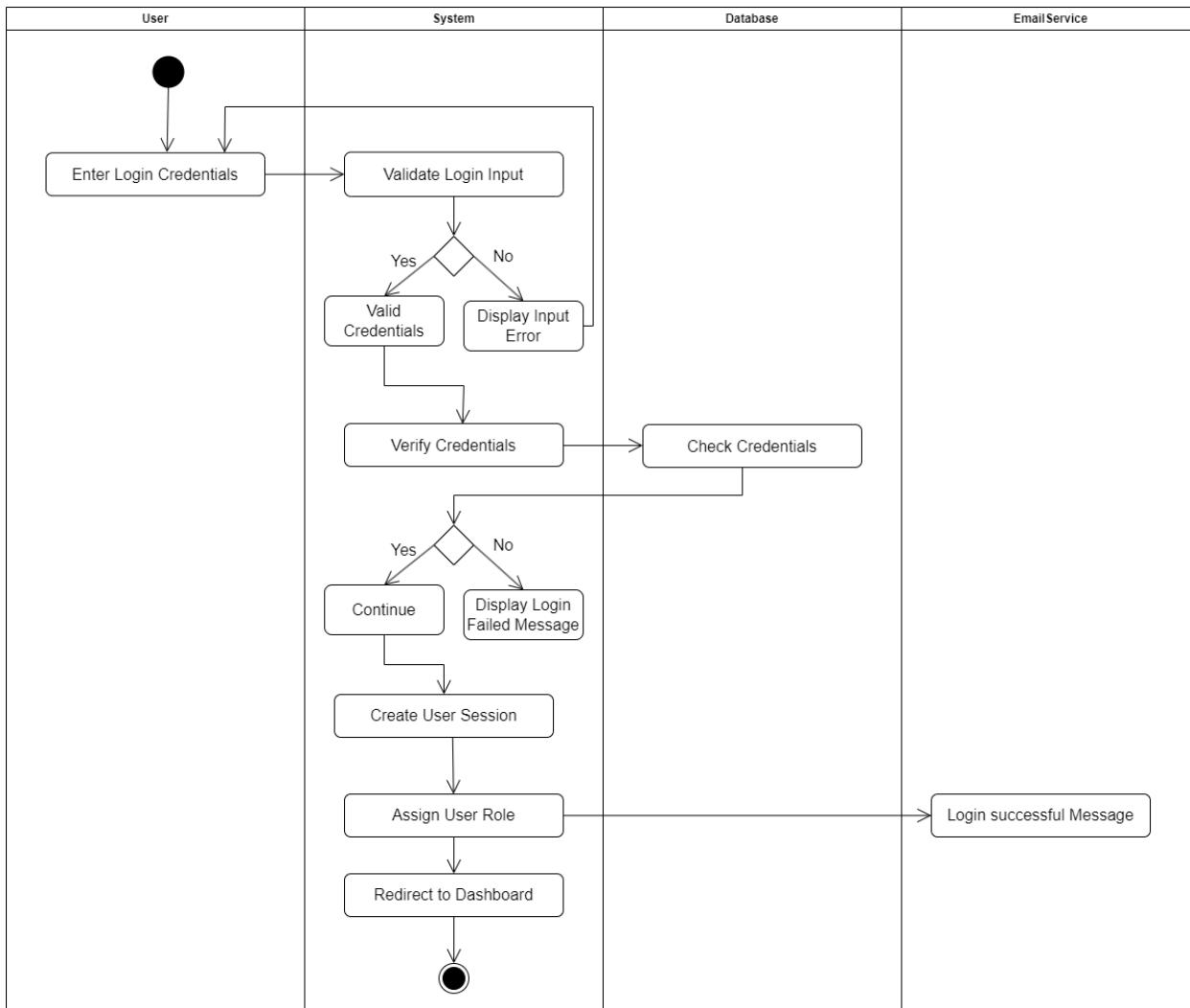


Figure 40: Activity Diagram for Login

#### 4.7.2. Activity Diagram for Create or Update Portfolio (Candidate)

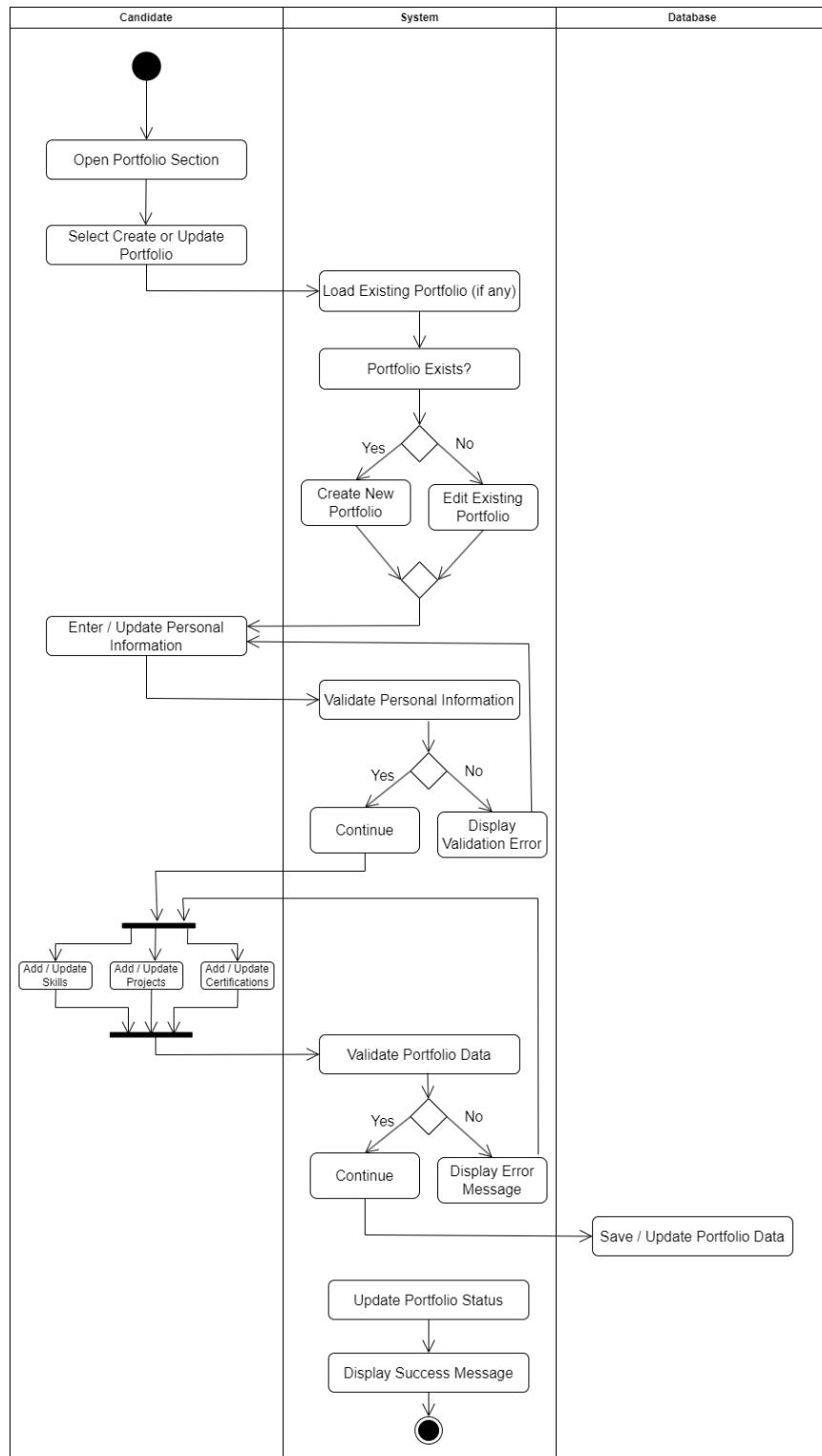


Figure 41: Activity Diagram for Create or Update Portfolio (Candidate)

### 4.7.3. Activity Diagram for Job Apply and Application Tracking

#### 4.7.3.1. Job Apply

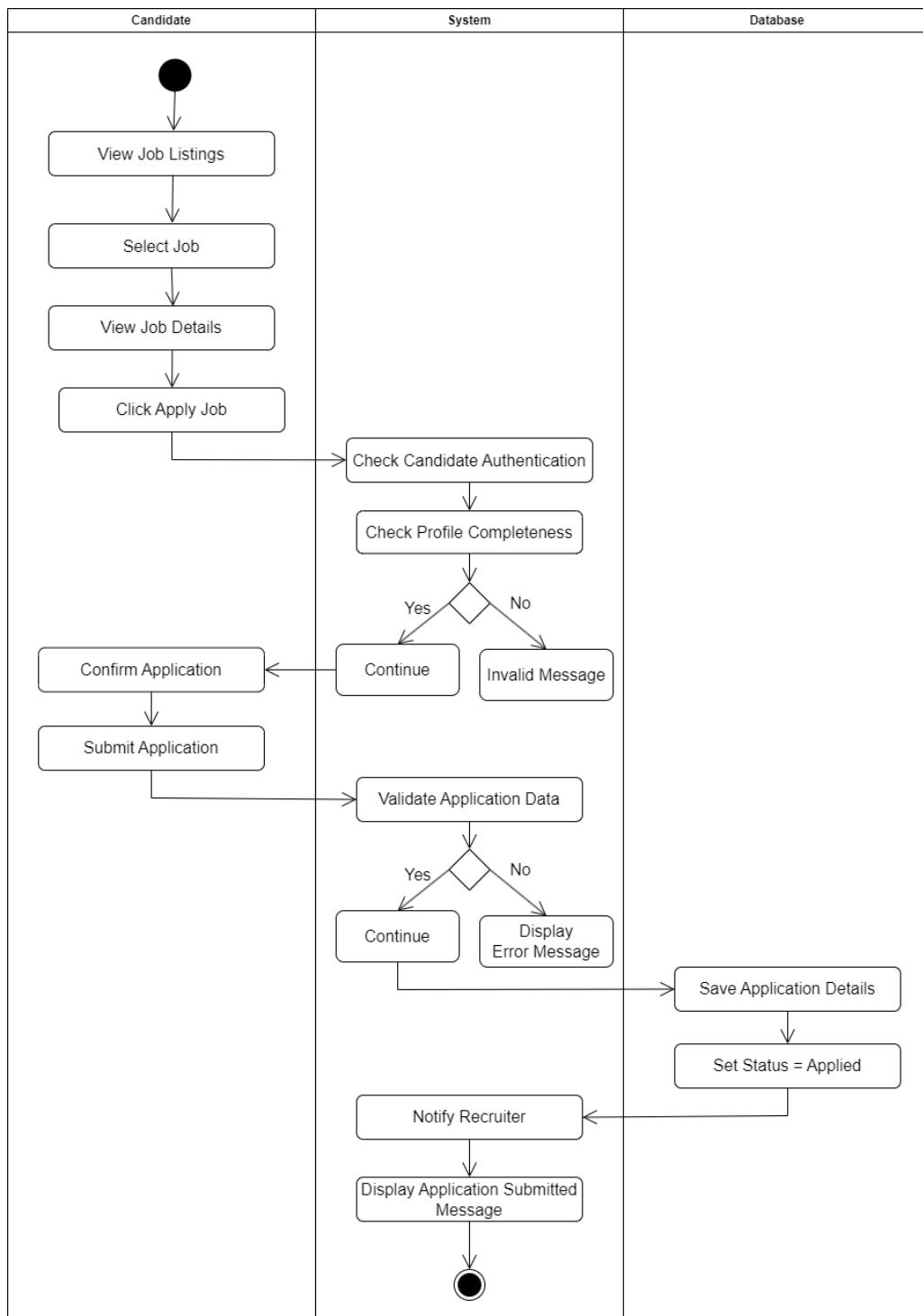


Figure 42:Activity Diagram for Job Apply

#### 4.7.3.2. Track Application Status

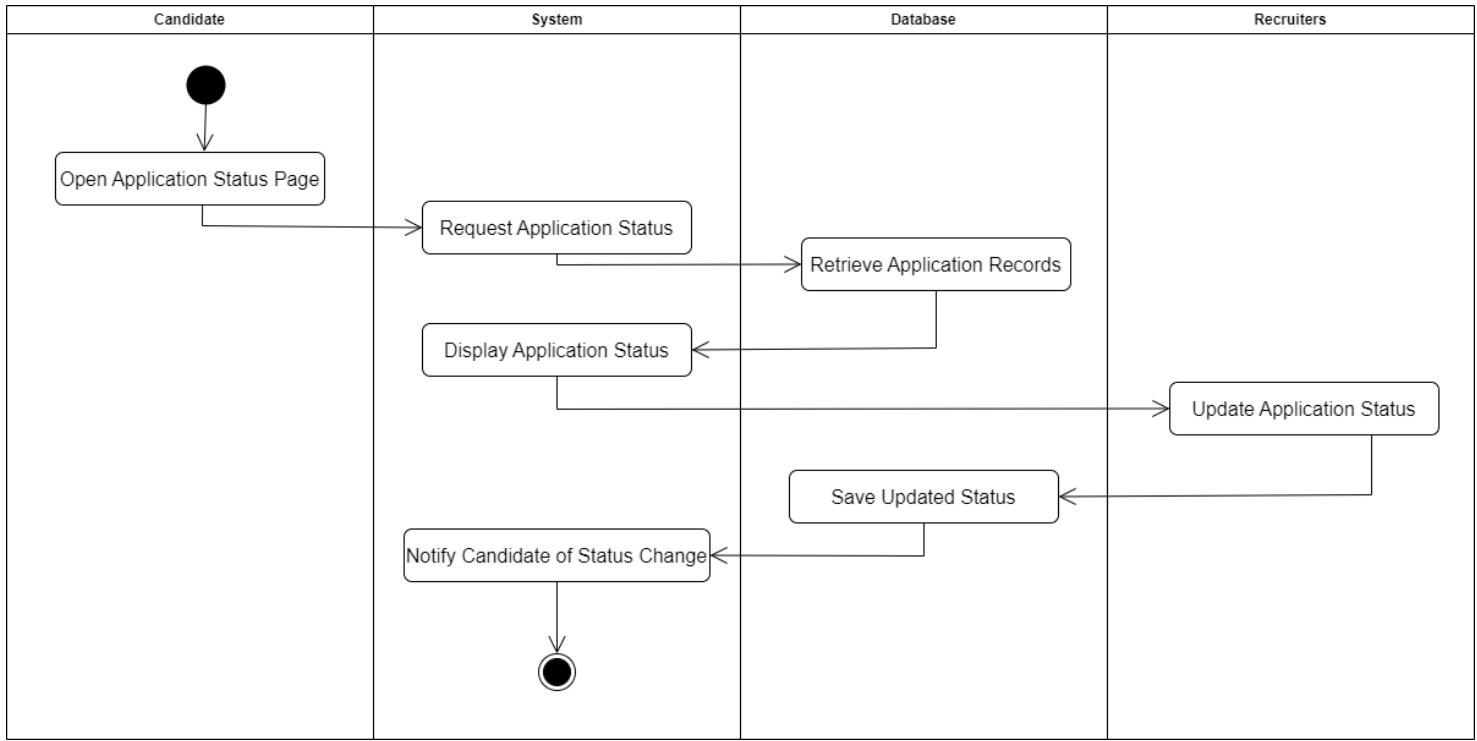


Figure 43: Activity Diagram for Application Tracking

#### 4.7.4. Activity Diagram for AI Job Recommendations (Candidate)

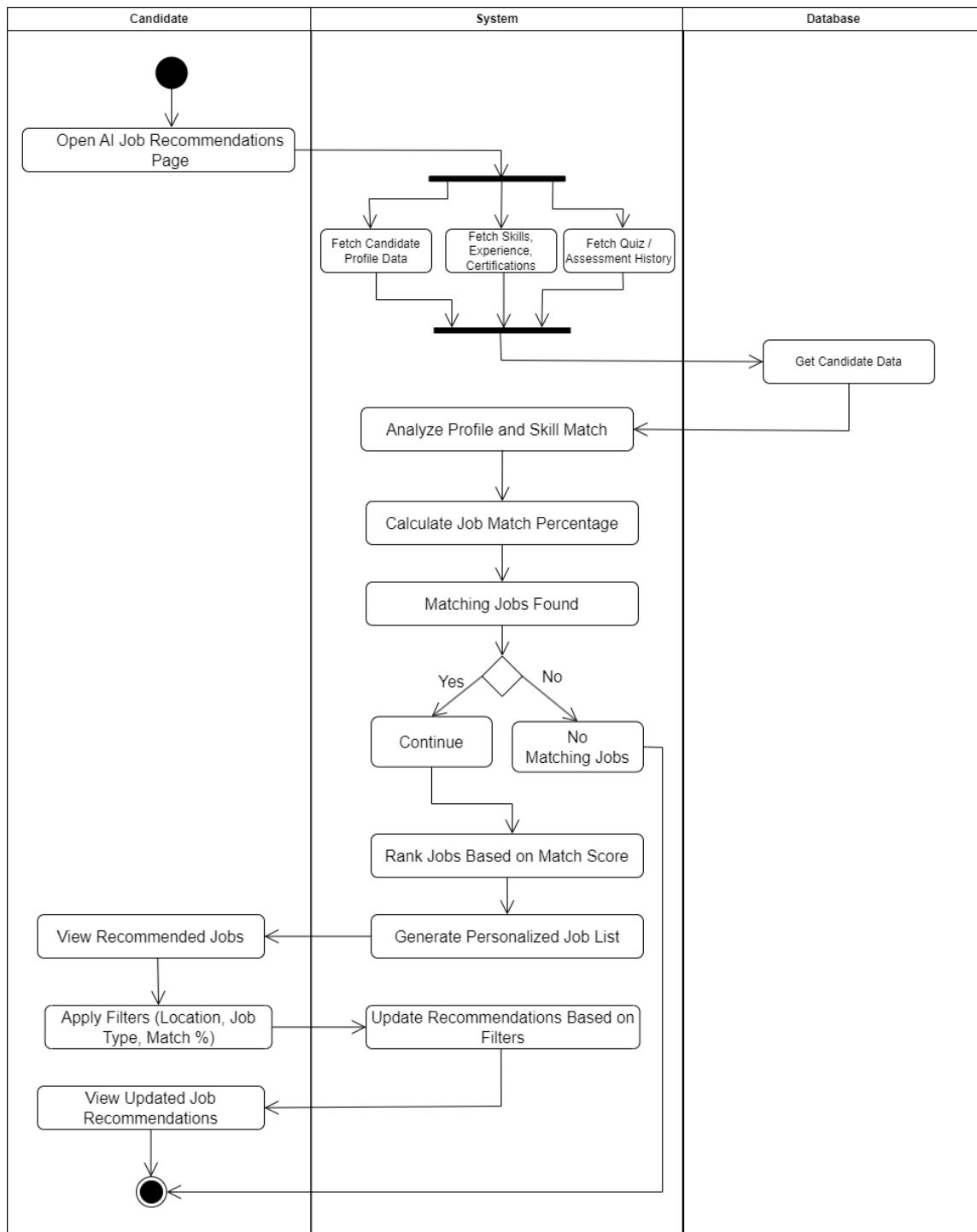


Figure 44:Activity Diagram for AI Job Recommendations (Candidate)

#### 4.7.5. Activity Diagram for Connection and Messaging

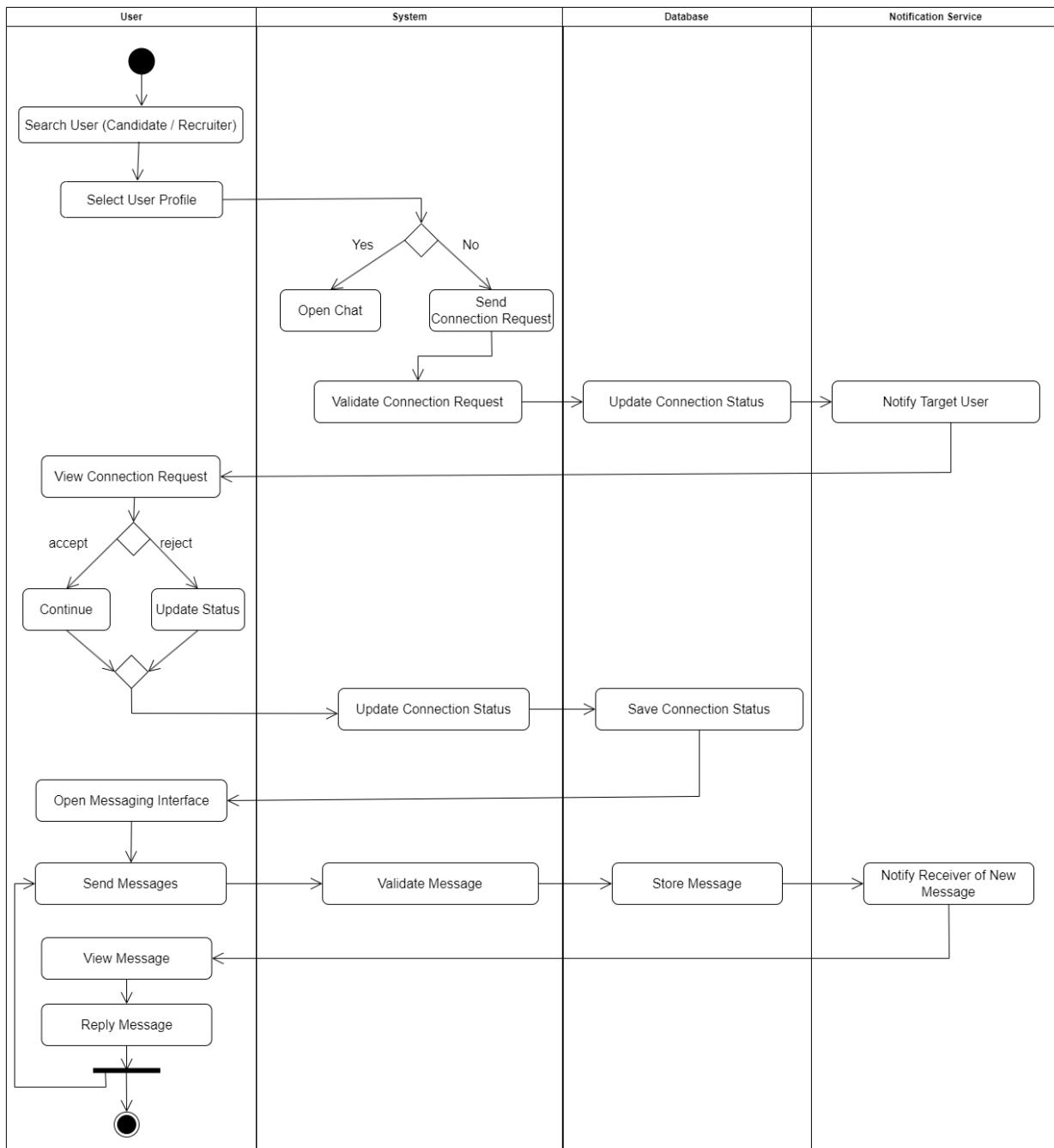


Figure 45: Activity Diagram for Connection and Messaging

#### 4.7.6. Activity Diagram for Job Posting and Manage Application (ATS Resume Scanner)

##### 4.7.6.1. Job Posting

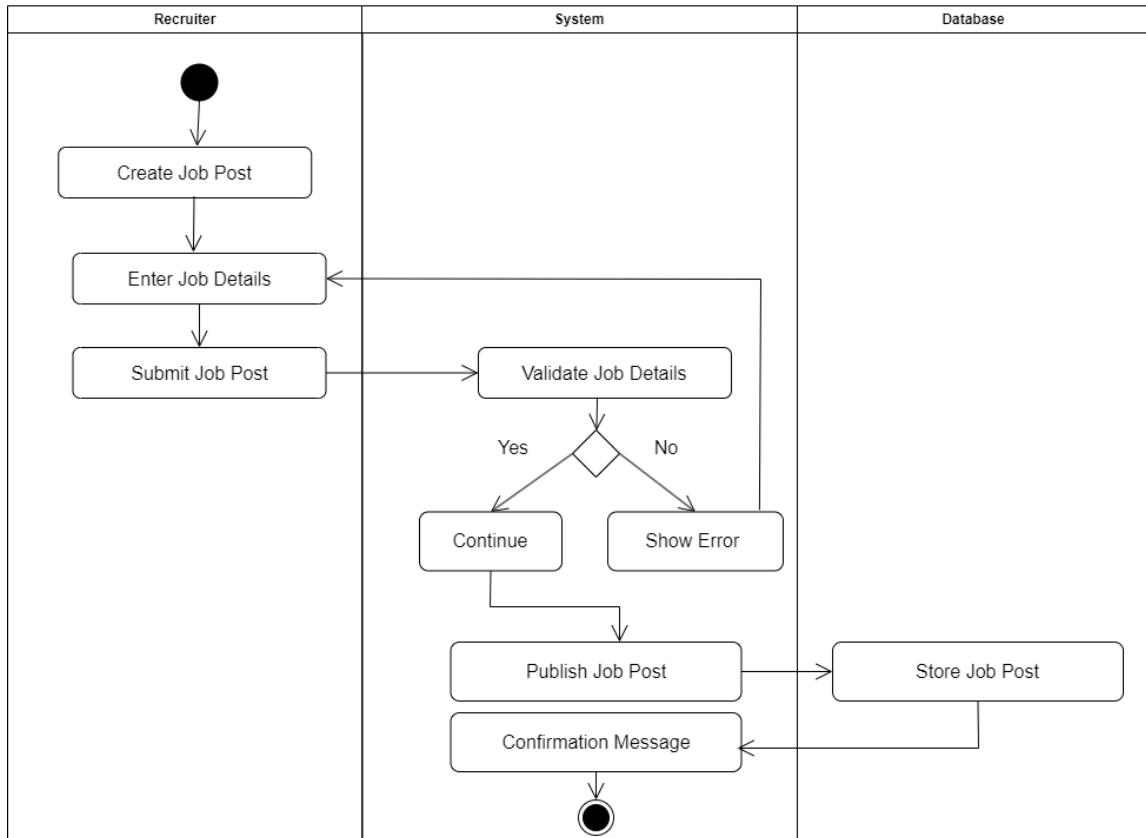


Figure 46: Activity Diagram for Job Posting

#### 4.7.6.2. Manage Application (ATS Resume Scanner)

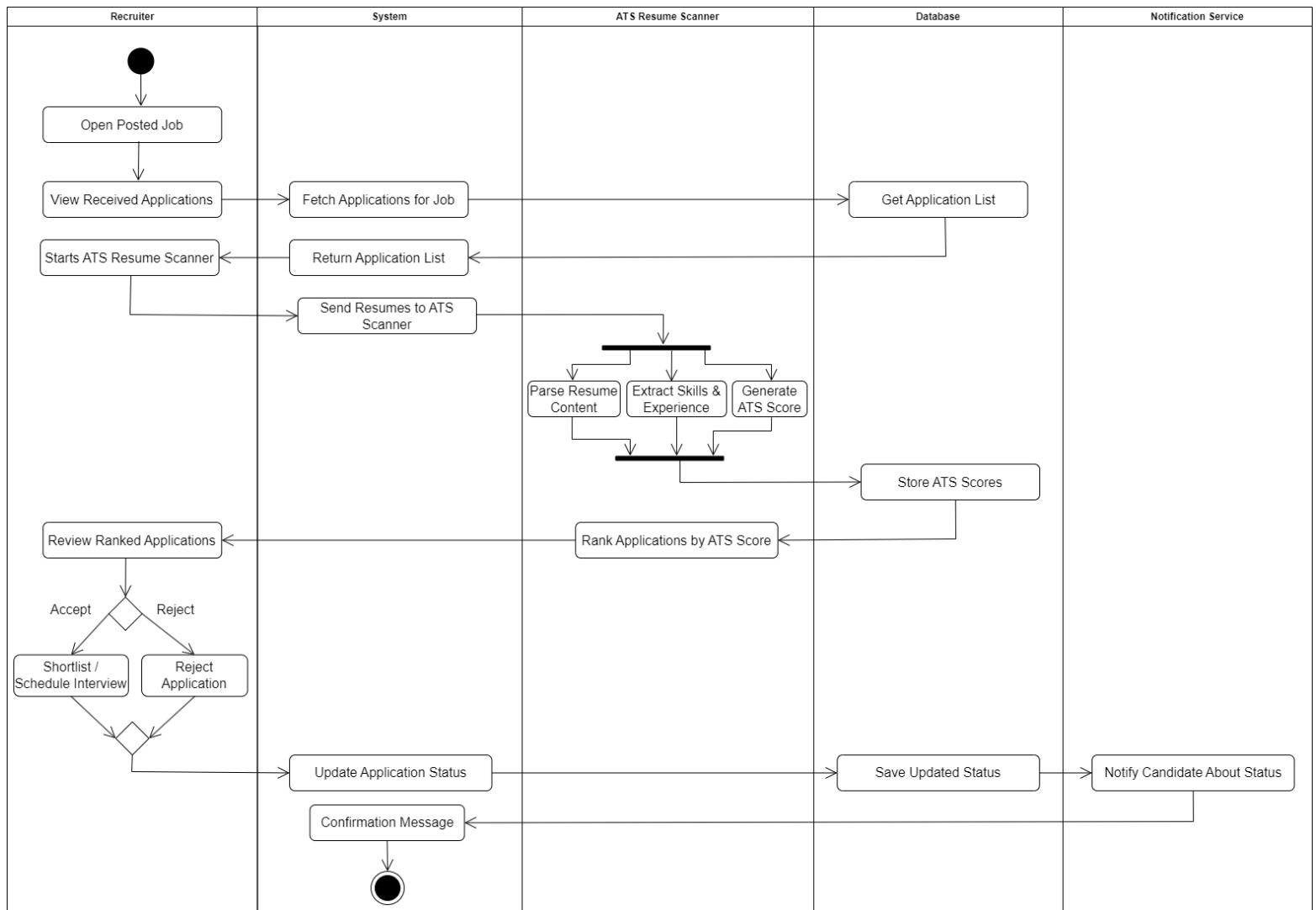


Figure 47: Activity Diagram for Manage Application (ATS Resume Scanner)

#### 4.7.7. Activity Diagram for Search Candidates and View Profile

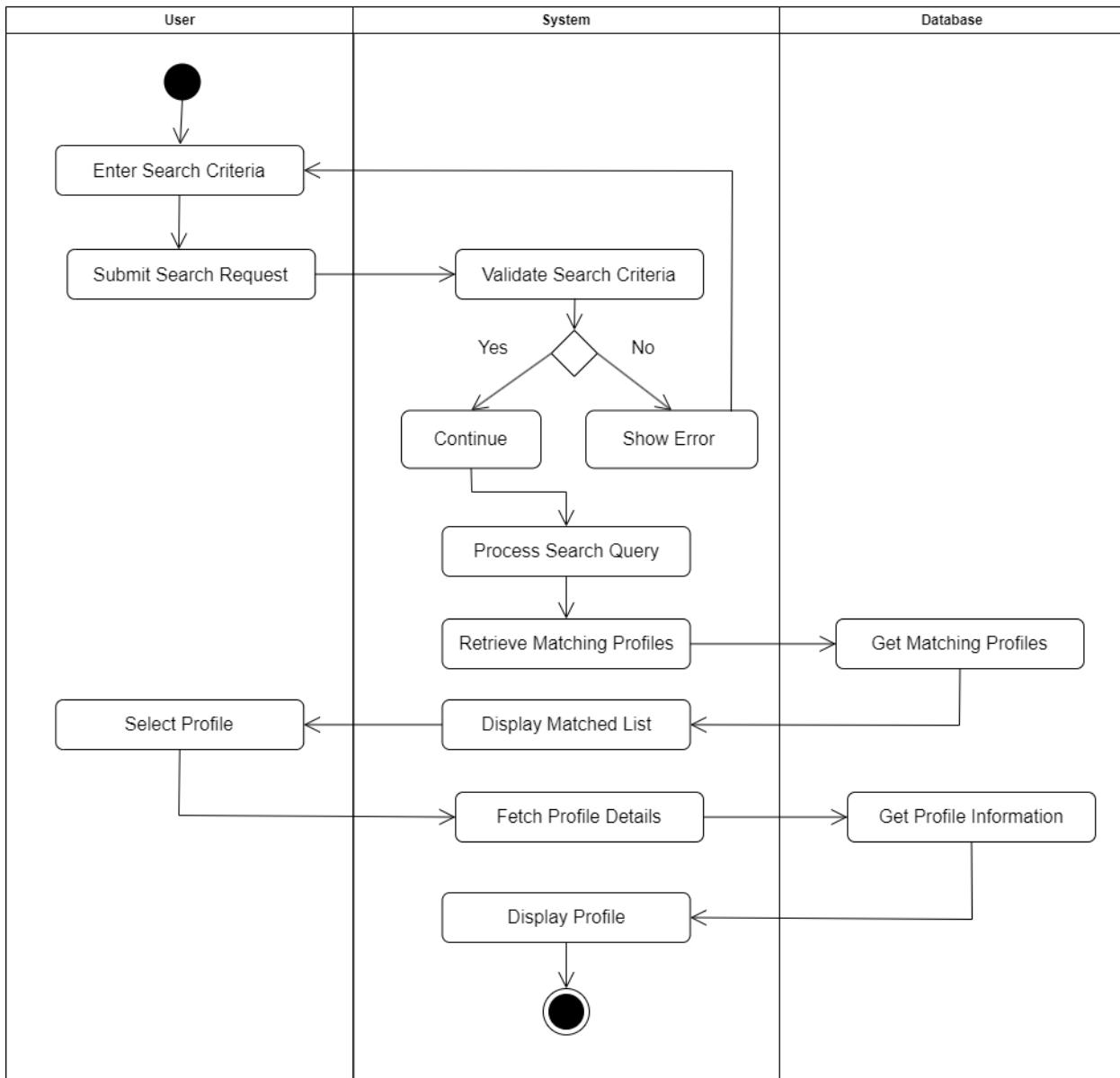


Figure 48: Activity Diagram for Search Candidates and View Profile

#### 4.7.8. Activity Diagram for Manage Users and Quizzes / Assessments

##### 4.7.8.1. Manage Users

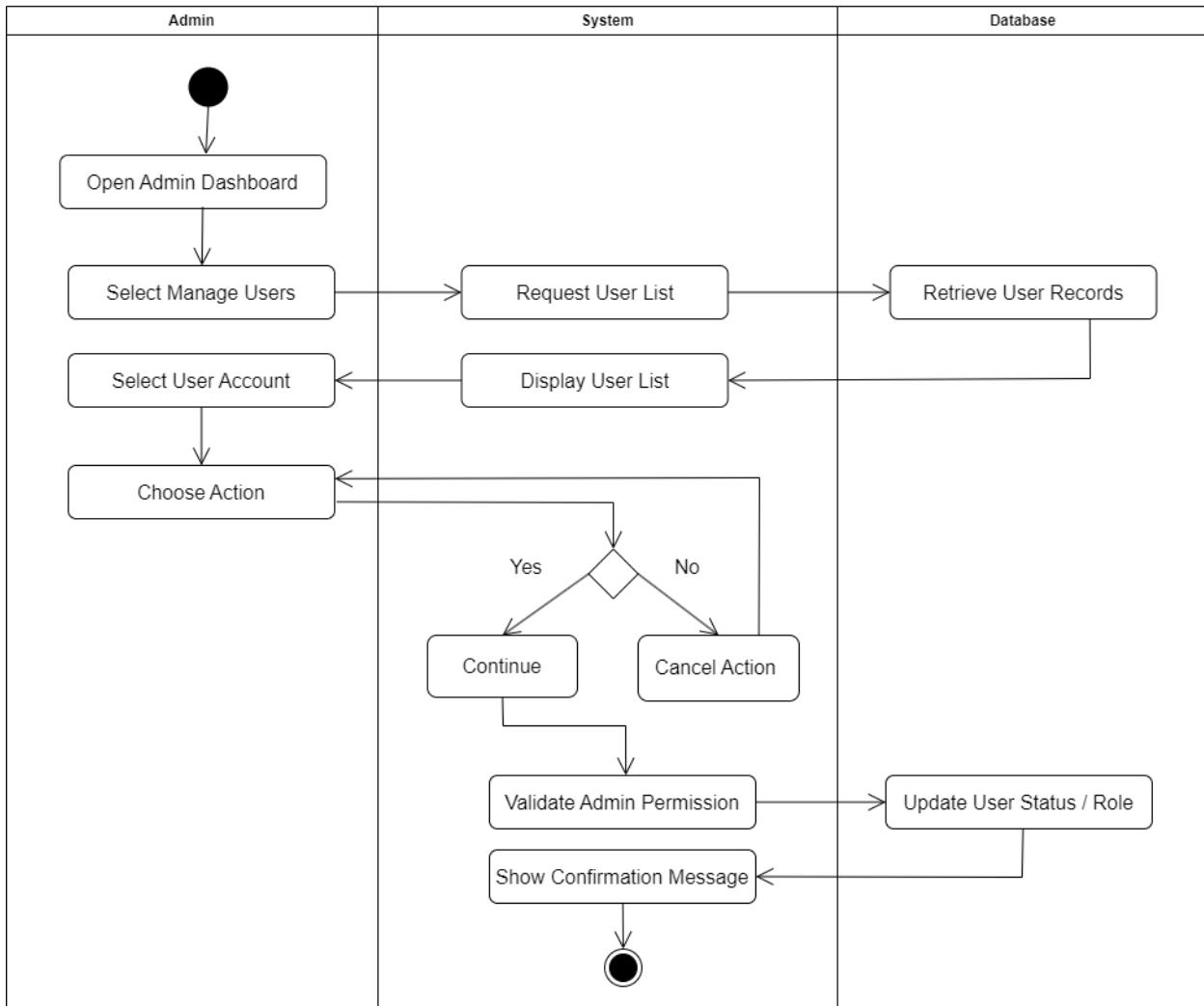


Figure 49: Activity Diagram for Manage Users

#### 4.7.8.2. Manage Quizzes / Assessments

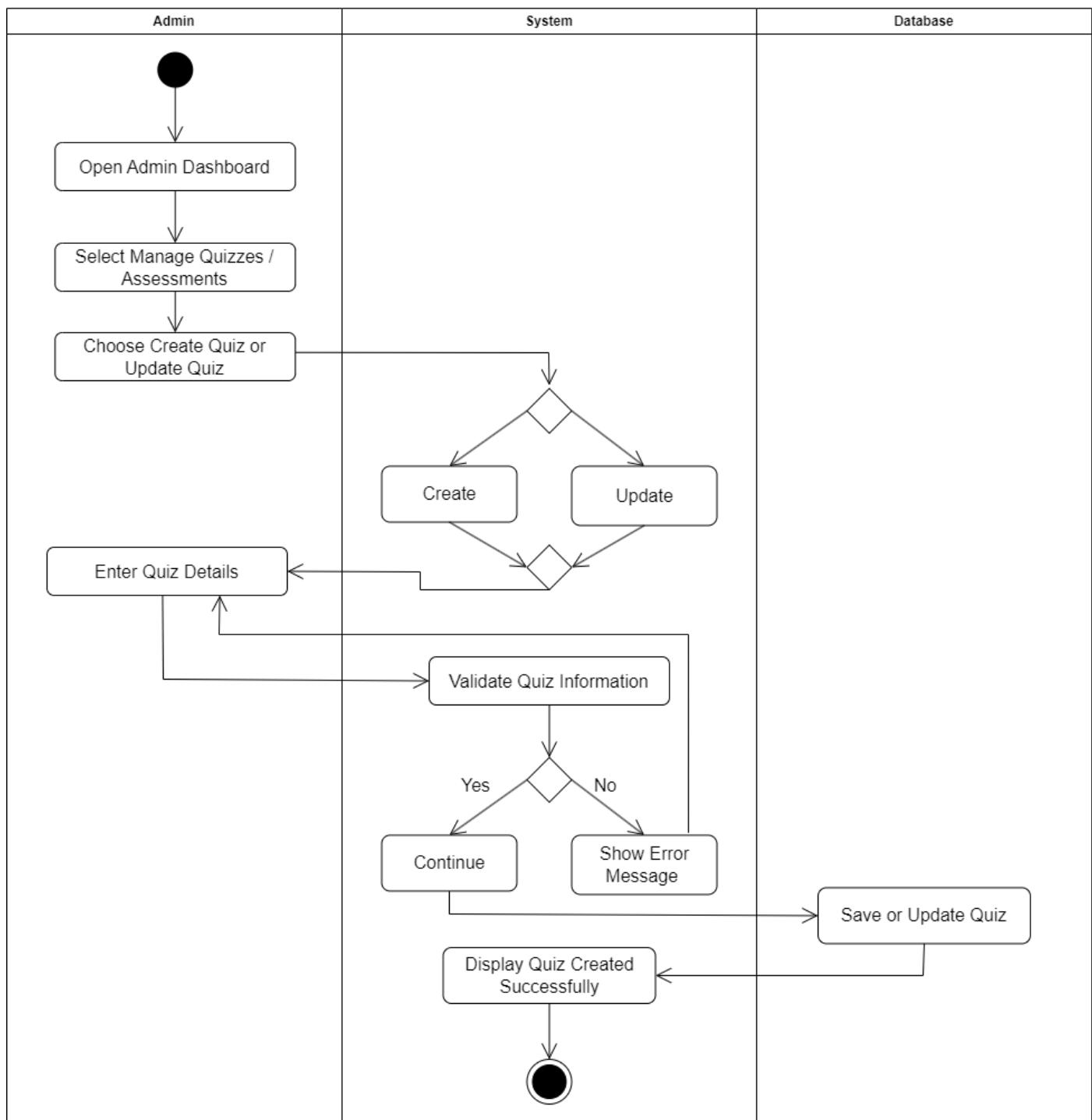


Figure 50: Activity Diagram for Manage Quizzes / Assessments

#### 4.8. DFD Level 0 (Context Level Diagram (Level 0))

Context Level Diagram (Level 0)

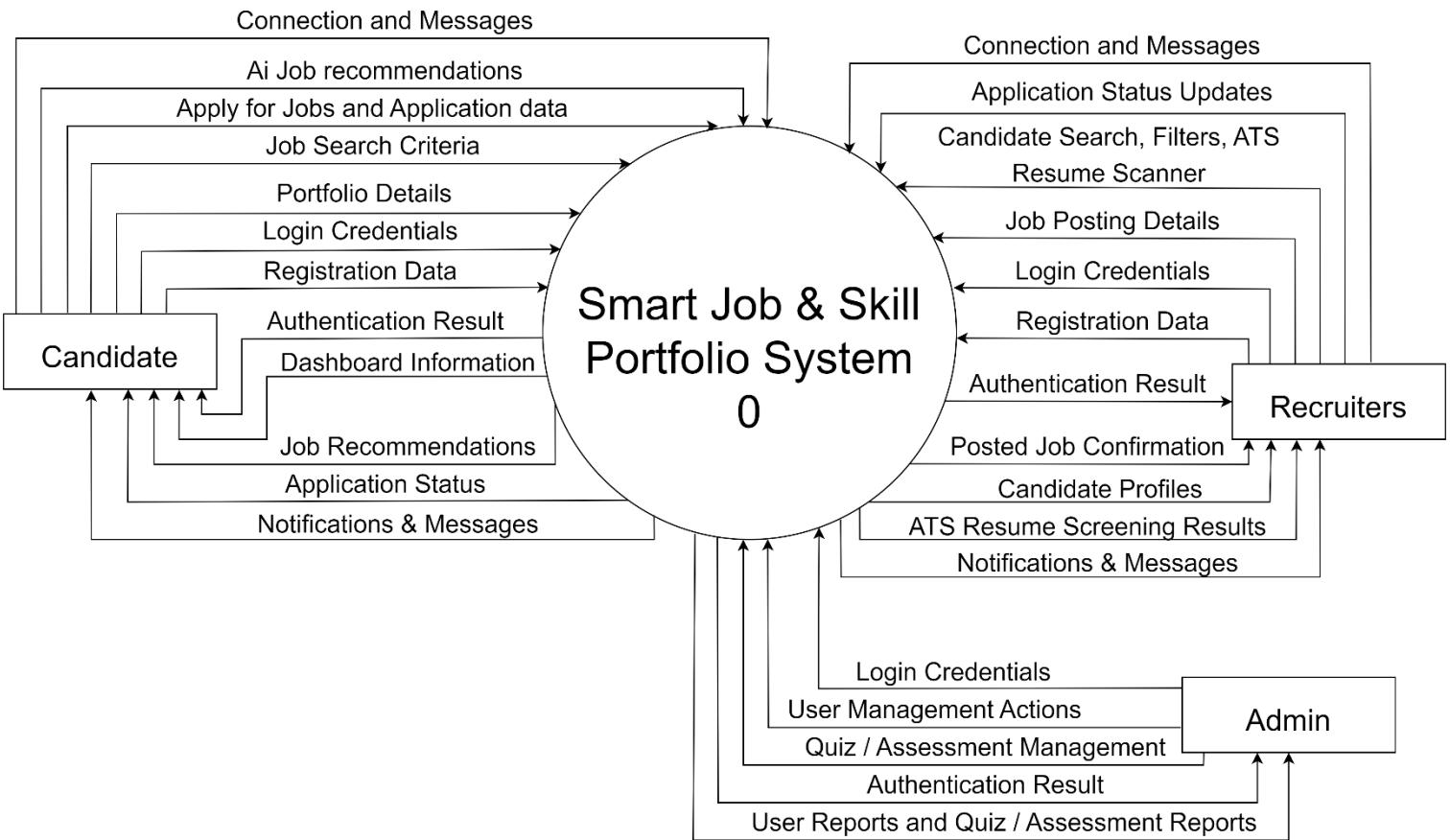


Figure 51: DFD Level 0 (Context Level Diagram (Level 0))

## 4.9. Flow Chart

### 4.9.1. Candidate System Flowchart

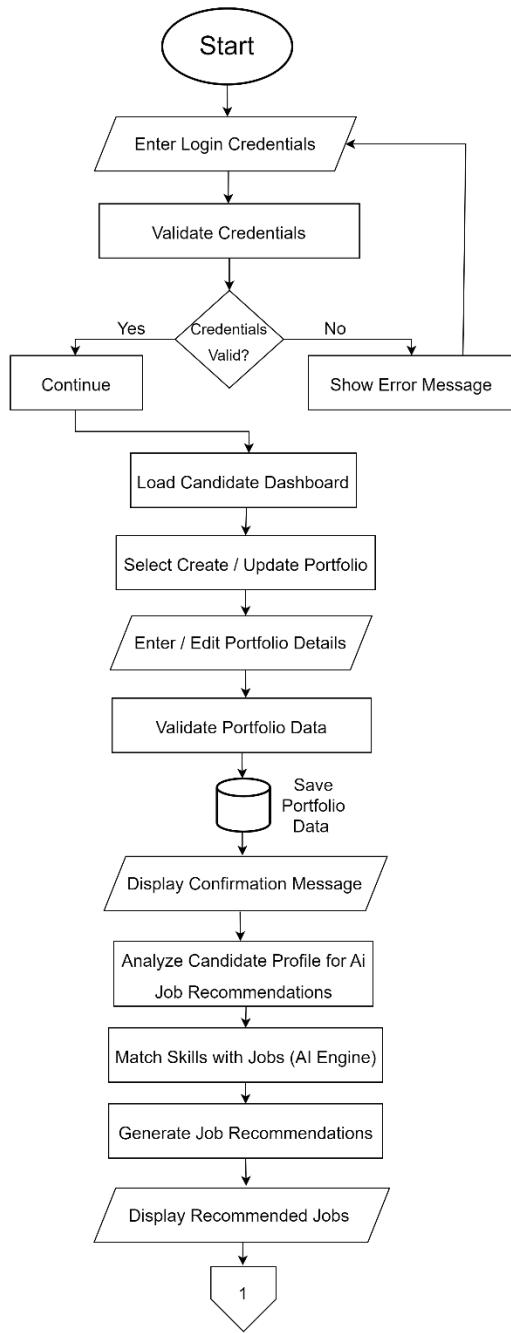


Figure 52: Candidate System Flowchart 1

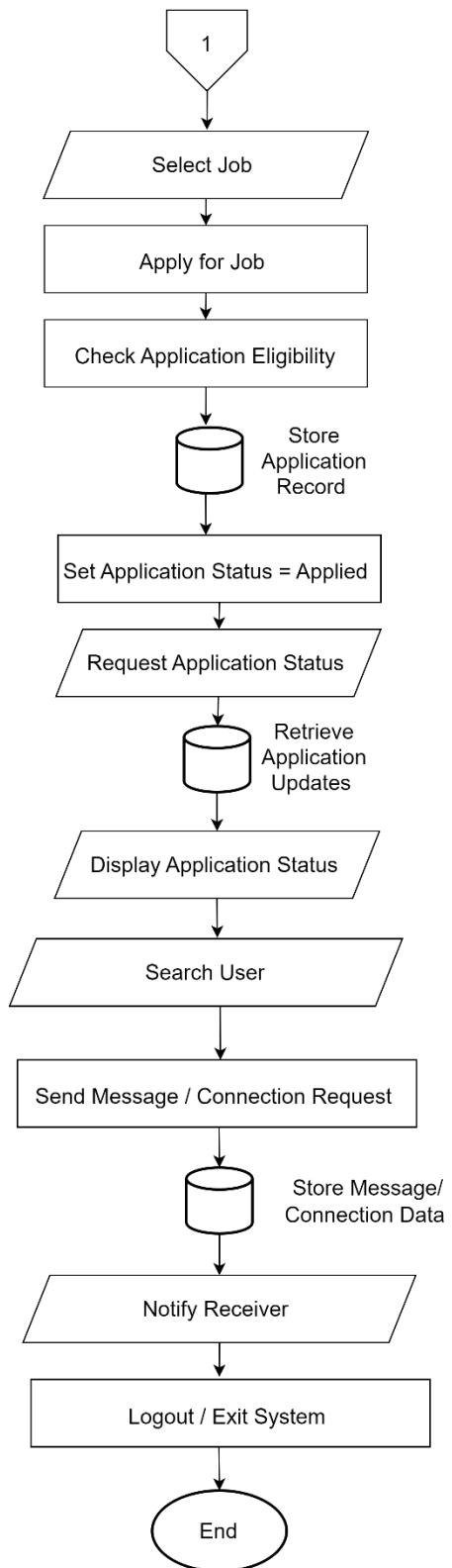


Figure 53: Candidate System Flowchart 2

#### 4.9.2. Recruiter System Flowchart

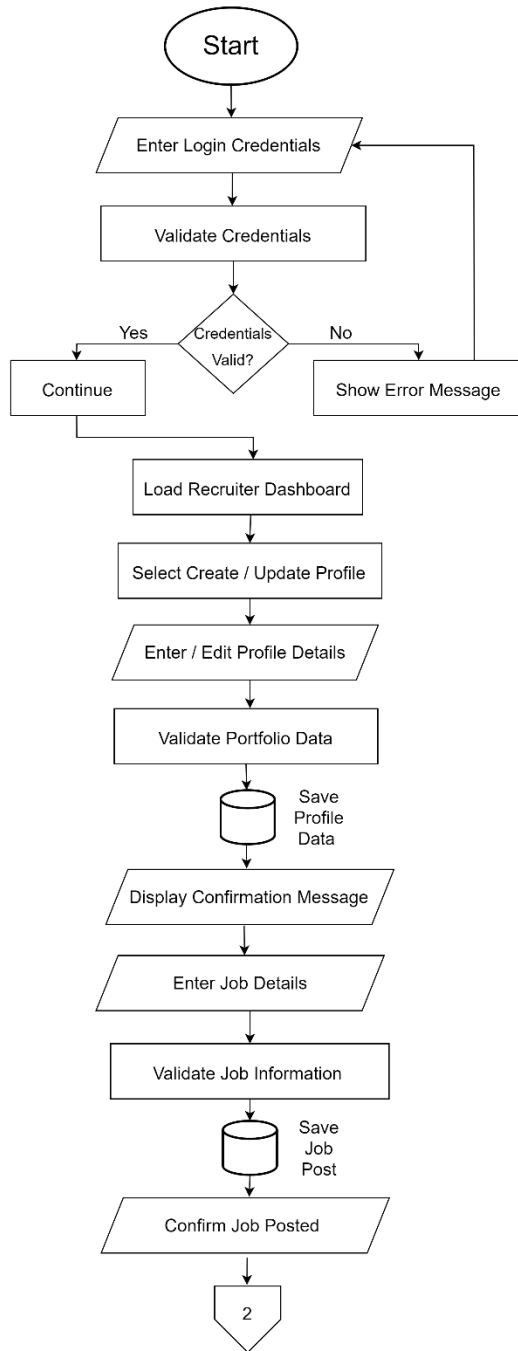


Figure 54: Recruiter System Flowchart 1

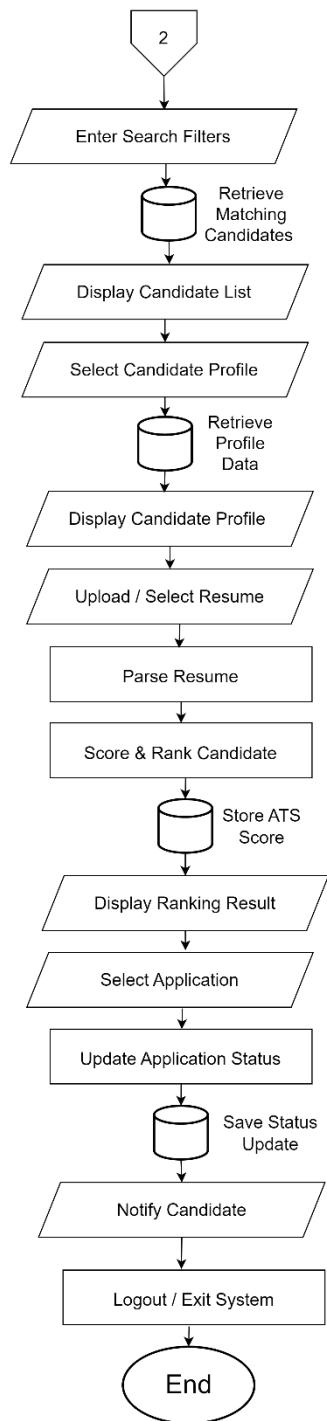


Figure 55: Recruiter System  
Flowchart 2

### 4.9.3. Admin System Flowchart

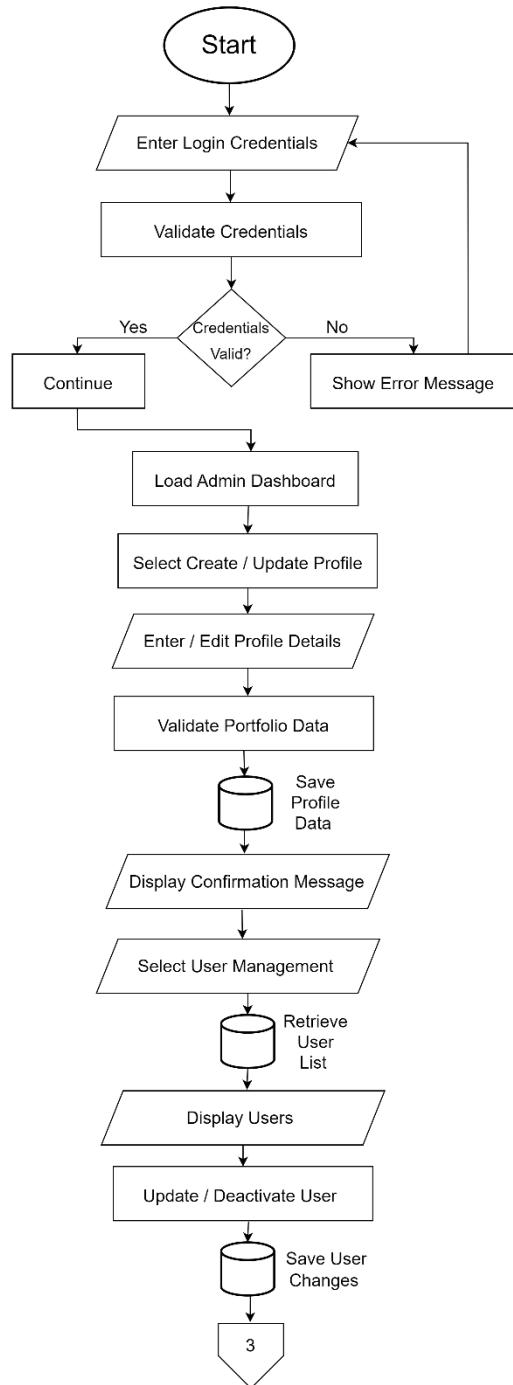


Figure 56: Admin System Flowchart 1

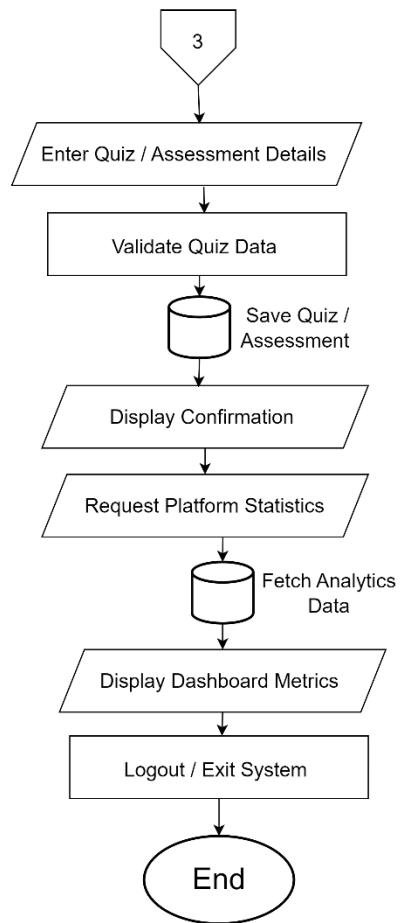


Figure 57: Admin System Flowchart 2

## 4.10. Database Collections and Relationships

Since MongoDB is a NoSQL database, a traditional ERD is not suitable. Instead, a Database Collections and Relationships diagram is used to show collections, their attributes, and how they relate (one-to-one, one-to-many, etc.)

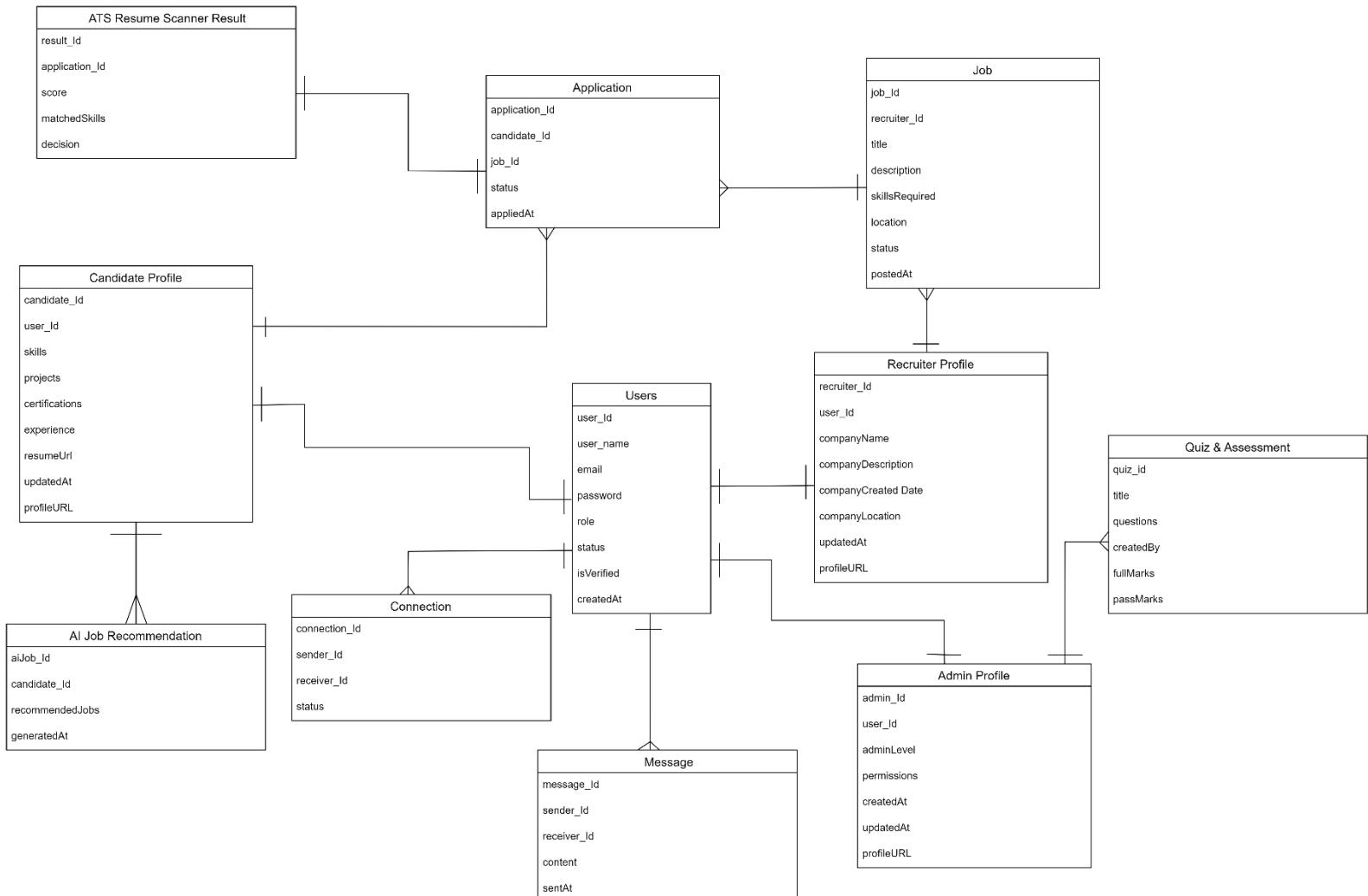


Figure 58: UI Design of Candidate Dashboard Page

## 4.11. Class Diagram

### 4.11.1. Overall, Class Diagram

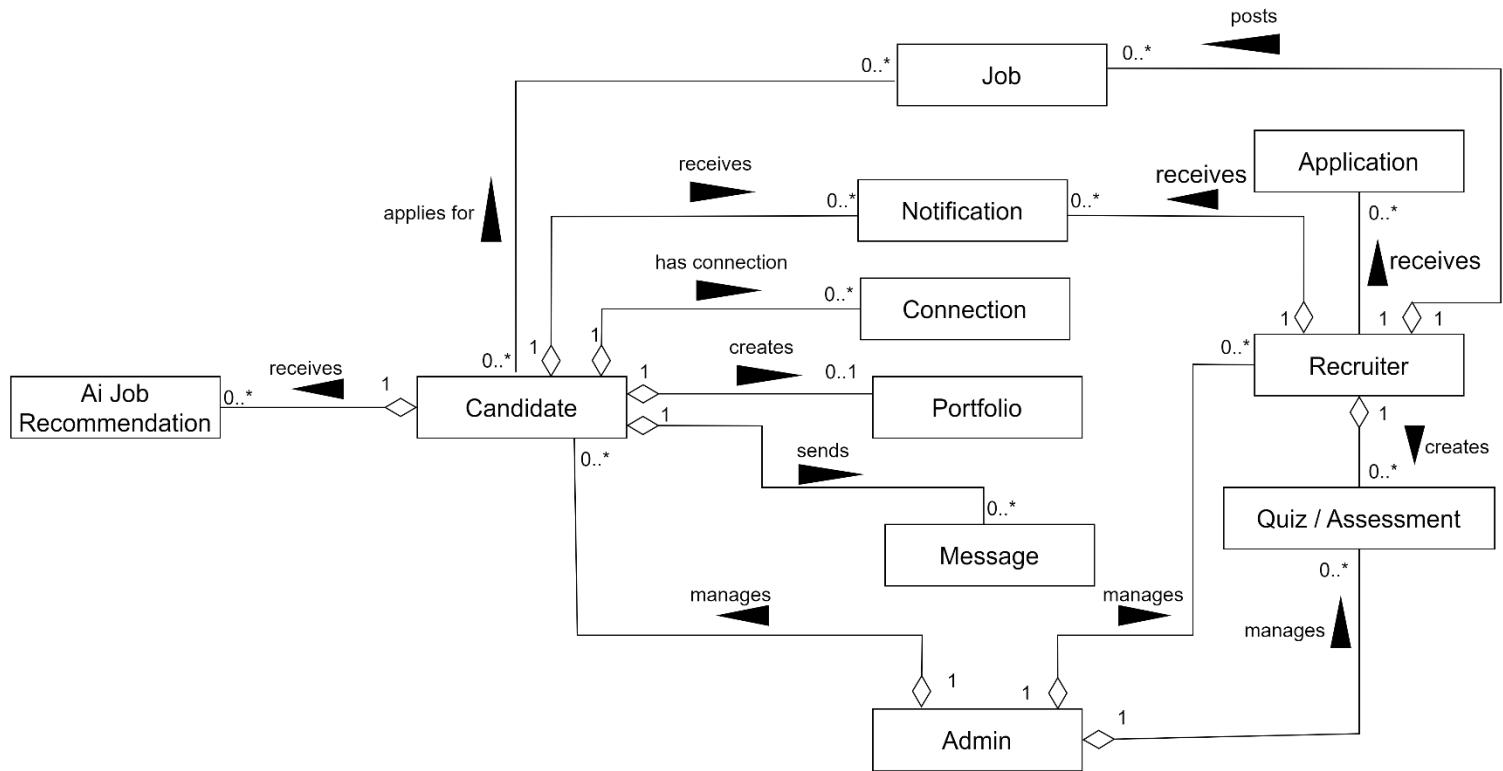


Figure 59: Overall Class Diagram

#### 4.11.2. Individual Class Diagram

##### i. Class Diagram of Candidate

Candidate
<ul style="list-style-type: none"><li>- candidateld: String</li><li>- name: String</li><li>- email: String</li><li>- phone: String</li><li>- password: String</li><li>- skills[]: Array&lt;String&gt;</li><li>- experience[]: Array&lt;String&gt;</li><li>- portfolios[]: Array&lt;String&gt;</li><li>- appliedJobIds[]: Array&lt;String&gt;</li></ul>
<ul style="list-style-type: none"><li>+ register(): Boolean</li><li>+ login(email:String, password:String): Boolean</li><li>+ updateProfile(): Boolean</li><li>+ createPortfolio(portfolio: Portfolio): Portfolio</li><li>+ updatePortfolio(portfolio: Portfolio): Boolean</li><li>+ applyJobs(jobId: String): Boolean</li><li>+ sendConnection(receiverId: String): Boolean</li><li>+ sendMessage(receiverId: String, messageText: String): String</li></ul>

Figure 60: Class Diagram of Candidate

## ii. Class Diagram of Recruiter

Recruiter
<ul style="list-style-type: none"><li>- recruiterId: String</li><li>- name: String</li><li>- email: String</li><li>- phone: String</li><li>- password: String</li><li>- companyName: String</li><li>- postedJobIds[]: Array&lt;String&gt;</li><li>- createdQuizIds[]: Array&lt;String&gt;</li></ul>
<ul style="list-style-type: none"><li>+ register(): Boolean</li><li>+ login(email:String, password:String): Boolean</li><li>+ updateProfile()</li><li>+ createJob(): Job</li><li>+ updateJob(jobId: String): Boolean</li><li>+ createQuiz(): Quiz</li><li>+ assignQuiz(quiz: Quiz): Boolean</li><li>+ viewApplications(jobId: String): Array&lt;Application&gt;</li></ul>

Figure 61: Class Diagram of Recruiter

### iii. Class Diagram of Admin

Admin
<ul style="list-style-type: none"> <li>- adminId: String</li> <li>- name: String</li> <li>- email: String</li> <li>- phone: String</li> <li>- password: String</li> <li>- role: String</li> </ul>
<ul style="list-style-type: none"> <li>+ login(email:String, password:String): Boolean</li> <li>+ manageUsers(): Boolean</li> <li>+ manageJobs(): Boolean</li> <li>+ manageQuizzes(): Boolean</li> <li>+ manageNotifications(): Boolean</li> </ul>

Figure 62: Class Diagram of Admin

### iv. Class Diagram of Portfolio

Portfolio
<ul style="list-style-type: none"> <li>- portfolioId: String</li> <li>- candidateId: String</li> <li>- projects[]: Array&lt;String&gt;</li> <li>- skills[]: Array&lt;String&gt;</li> <li>- certifications[]: Array&lt;String&gt;</li> <li>- educations[]: Array&lt;String&gt;</li> </ul>
<ul style="list-style-type: none"> <li>+ addEducation(): Boolean</li> <li>+ addProject(): Boolean</li> <li>+ addCertification(): Boolean</li> <li>+ updatePortfolio(): Boolean</li> </ul>

Figure 63: Class Diagram of Portfolio

### v. Class Diagram of Job

Job	
- jobId: String	
- recruiterId: String	
- title: String	
- description: String	
- requirements[]: Array<String>	
- postedDate: Date	
+ createJob(): Boolean	
+ updateJob(): Boolean	
+ deleteJob(): Boolean	
+ viewApplications(): Array<Application>	

Figure 64: Class Diagram of Job

### vi. Class Diagram of Application

Application	
- applicationId: String	
- candidateId: String	
- jobId: String	
- status (applied, shortlisted, rejected, hired): String	
- appliedDate: Date	
+ submitApplication(): Boolean	
+ updateStatus(status: String): String	

Figure 65: Class Diagram of Application

### vii. Class Diagram of Message

Message	
- messageId: String	
- senderId: String	
- receiverId: String	
- messageText: String	
- sentDate: Date	
+ sendMessage(): Boolean	
+ deleteMessage(): Boolean	
+ viewMessages(): Array<Message>	

Figure 66: Class Diagram of Message

### viii. Class Diagram of Connection

Connection	
- connectionId: String	
- senderId: String	
- receiverId: String	
- status: String	
- requestedDate: Date	
- acceptedDate: Date	
+ sendRequest(): Boolean	
+ acceptRequest(): Boolean	
+ declineRequest(): Boolean	
+ removeConnection(): Boolean	

Figure 67: Class Diagram of Connection

### ix. Class Diagram of AI Job Recommendation

AI Job Recommendation	
- recommendationId: String	
- candidateId: String	
- recommendedJobIds: Array<String>	
- generatedDate: Date	
+ generateRecommendation(): Boolean	
+ viewRecommendations(): Array<Job>	

Figure 68: Class Diagram of AI Job Recommendation

### x. Class Diagram of Resume / ATS Scan

Resume / ATS Scan	
- resumeId: String	
- candidateId: String	
- fileUrl: String	
- parsedSkills: Array<String>	
- parsedExperience: Array<String>	
+ uploadResume(file: File): Boolean	
+ scanResume(): Boolean	
+ parseDate(): Boolean	

Figure 69: Class Diagram of Resume / ATS Scan

## xi. Class Diagram of Quiz / Assessment

Quiz / Assessment
<ul style="list-style-type: none"> <li>- quizId: String</li> <li>- recruiterId: String</li> <li>- title: String</li> <li>- questions: Array&lt;String&gt;</li> <li>- assignedCandidateIds: Array&lt;String&gt;</li> <li>- createdDate: Date</li> </ul>
<ul style="list-style-type: none"> <li>+ createQuiz(): Boolean</li> <li>+ assignQuiz(candidateIds: Array&lt;String&gt;): Boolean</li> <li>+ gradeQuiz(): Boolean</li> </ul>

Figure 70: Class Diagram of Quiz / Assessment

## xii. Class Diagram of Notification

Notification
<ul style="list-style-type: none"> <li>- notificationId: String</li> <li>- userId: String</li> <li>- type: String</li> <li>- content: &lt;String&gt;</li> <li>- date: Date</li> <li>- status: String</li> </ul>
<ul style="list-style-type: none"> <li>+ createNotification(): Boolean</li> <li>+ markAsRead(): Boolean</li> <li>+ deleteNotification(): Boolean</li> </ul>

Figure 71: Class Diagram of Notification

## 4.12. Wireframes

### 4.12.1. Wireframe of Register Page



The wireframe shows the header with a logo, navigation links (Home, Browse Jobs, Employers, Candidates), a 'Apply Now' button, and a 'Register/Login' button. The main content area has a large title 'Register' and a breadcrumb 'Home > Register'. Below this is the 'Register Now' form.

**Register Now**

---

Accept our terms and conditions and privacy policy

Already have an account? [Login](#)



The footer includes the company logo, a brief description of Jobhub's mission, and a grid of links for Company, Product, Download, and Support categories. It also features social media icons and a copyright notice.

Copyright ©2021 NireLink. All Rights Reserved

Figure 72: Wireframe of Register Page

#### 4.12.2. Wireframe of Login Page

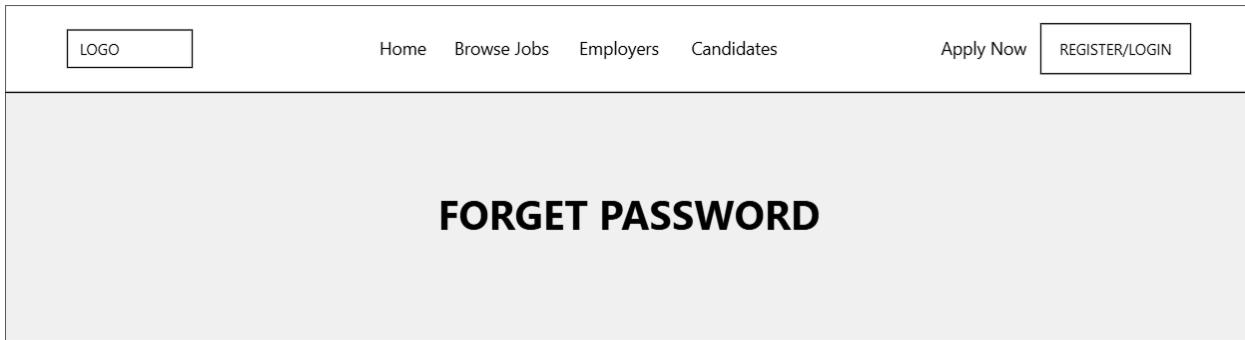
The wireframe illustrates the layout of the login page. At the top, there is a navigation bar with links for Home, Browse Jobs, Employers, Candidates, a '• Apply Now' button, and a 'Register/Login' button. Below the navigation is a large central area containing the word 'Login' and a 'Home > Register' breadcrumb. The main form is titled 'Login Now' and contains fields for 'Email' and 'Password', along with 'Remember me' and 'Forgot Password?' links. A 'Login Now' button is at the bottom of the form. Below the form is a link to 'Already have an account? Login'. The footer features a logo, company links (About us, Our Team, Products, Contact), product links (Feature, Pricing, Credit, FAQ), download links (iOS, Android, Microsoft, Desktop), and support links (Privacy, Help, Terms, FAQ). It also includes a copyright notice and social media icons.

Company	Product	Download	Support
About us Our Team Products Contact	Feature Pricing Credit FAQ	iOS Android Microsoft Desktop	Privacy Help Terms FAQ

Copyright ©2021 HireLink. All Rights Reserved

Figure 73: Wireframe of Login Page

#### 4.12.3. Wireframe of Forget Password Page



## FORGET PASSWORD

This wireframe shows a centered form for password recovery. It features a title 'Forget Your Password', a placeholder for an email address, a 'Verify Account' button, and a link to 'Back to Login Page'.

**Forget Your Password**

Enter your Email to receive verification code

Email

Verify Account

Back to Login Page



Figure 74: Wireframe of Forget Password Page

#### 4.12.4. Wireframe of Verify Account Page

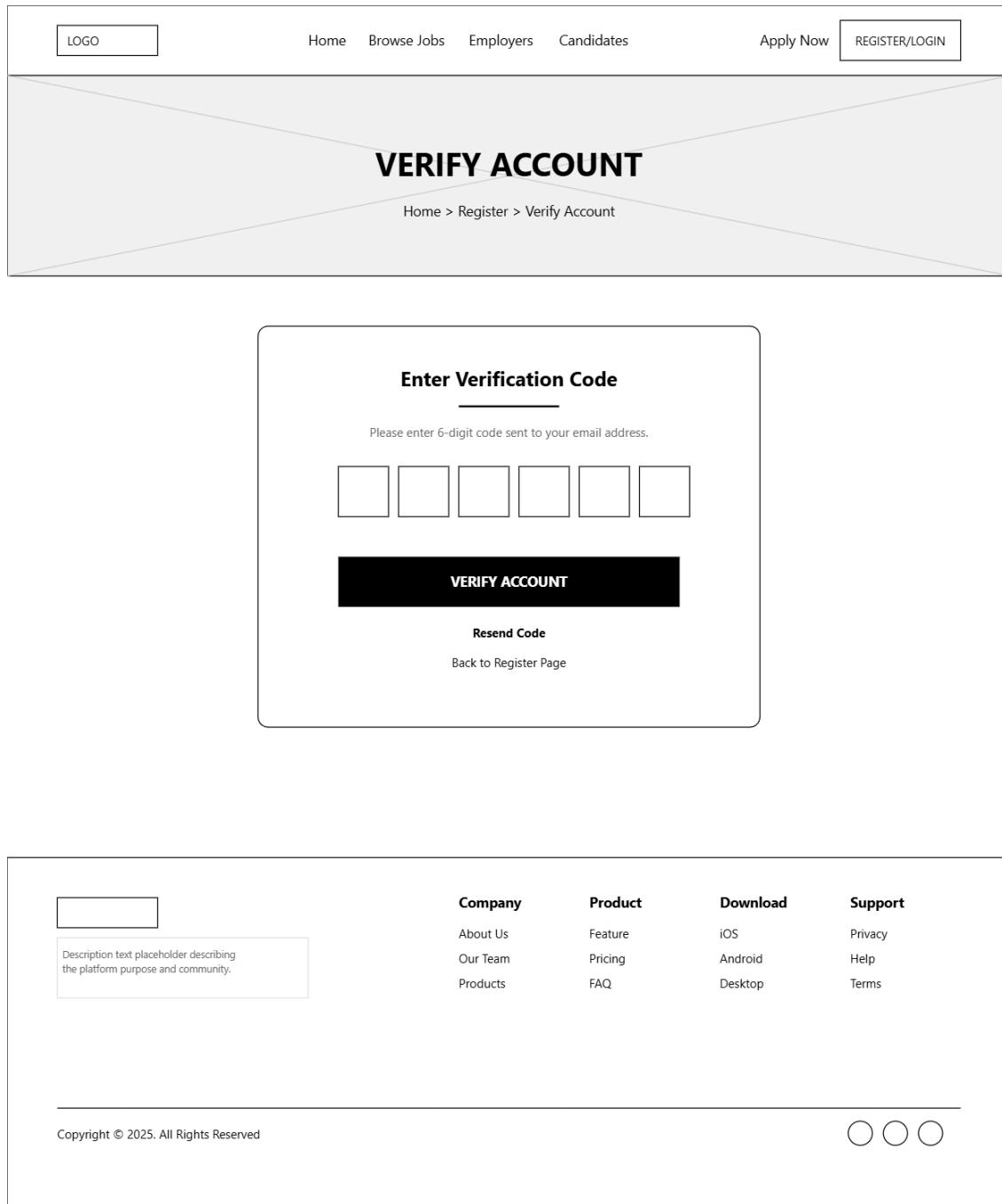


Figure 75: Wireframe of Verify Account Page

#### 4.12.5. Wireframe of New Password Page

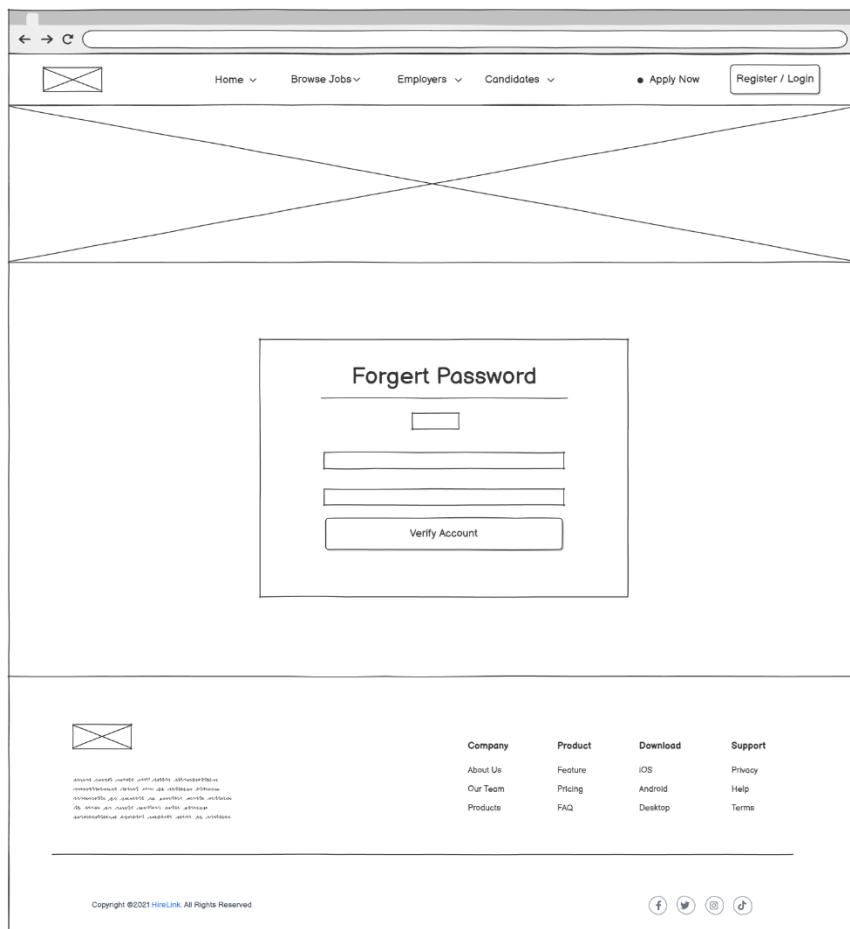


Figure 76: Wireframe of New Password Page

[For more wireframes of different pages, refer to the Appendix.](#)

## 4.13. UI Designs

### 4.13.1. UI Design of Register Page

**Register Now**

**Candidate**    **Recruiter**

Email

Password

Confirm Password

Accept our terms and conditions and privacy policy

**Register Now**

Already have an account? [Login](#)



Jobhub is the heat of the design community and the best resource to discover and connect with designers and jobs worldwide.

#### Company

About us  
Our Team  
Products  
Contact

#### Product

Feature  
Pricing  
Credit  
FAQ

#### Download

iOS  
Android  
Microsoft  
Desktop

#### Support

Privacy  
Help  
Terms  
FAQ

Copyright ©2021 [HireLink](#). All Rights Reserved



Figure 77: UI Design of Register Page

### 4.13.2. UI Design of Login Page

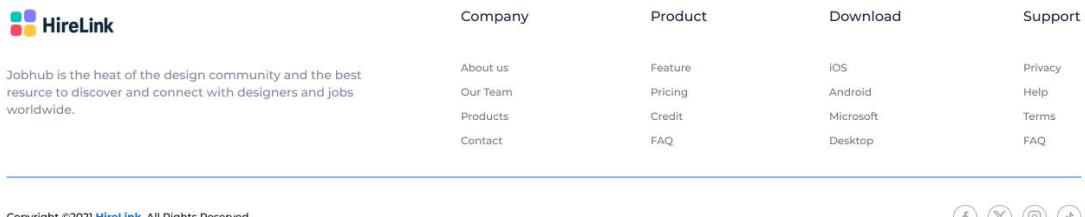
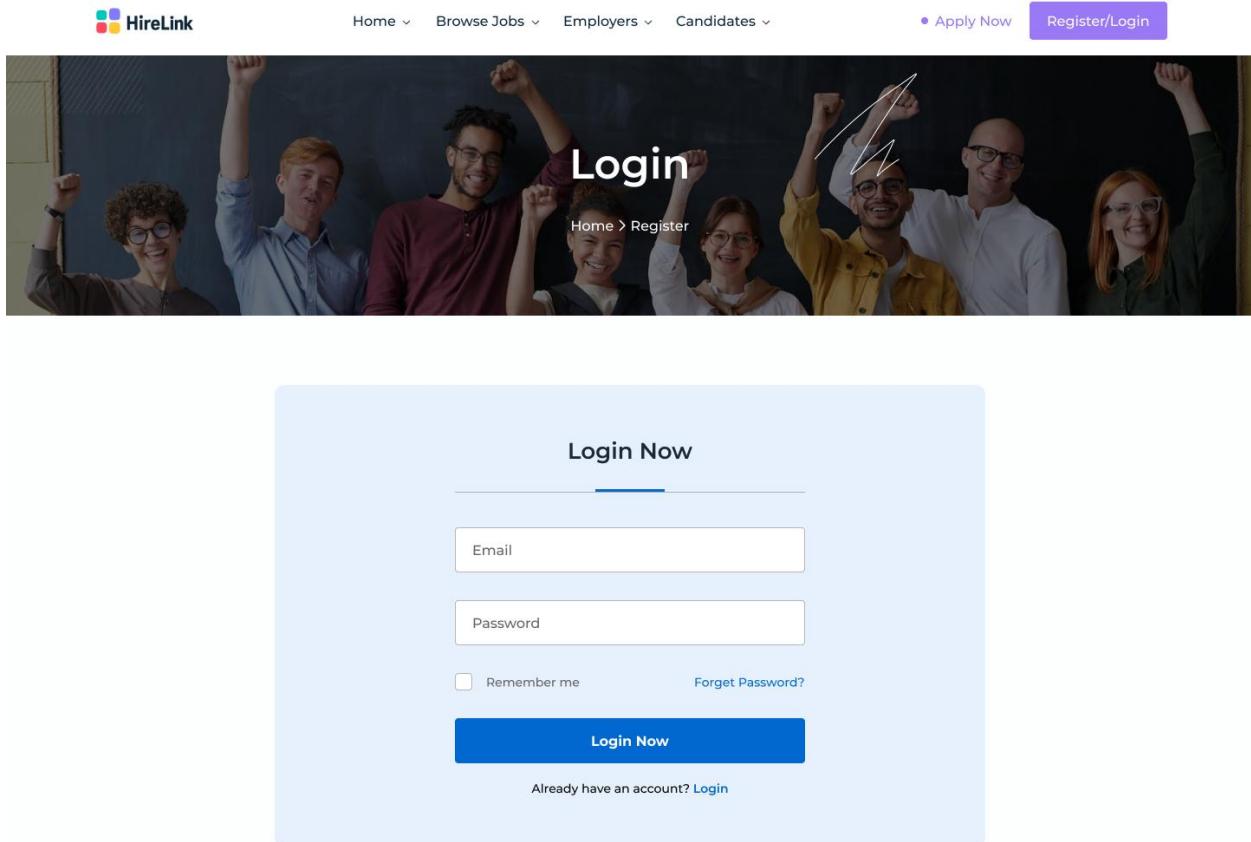
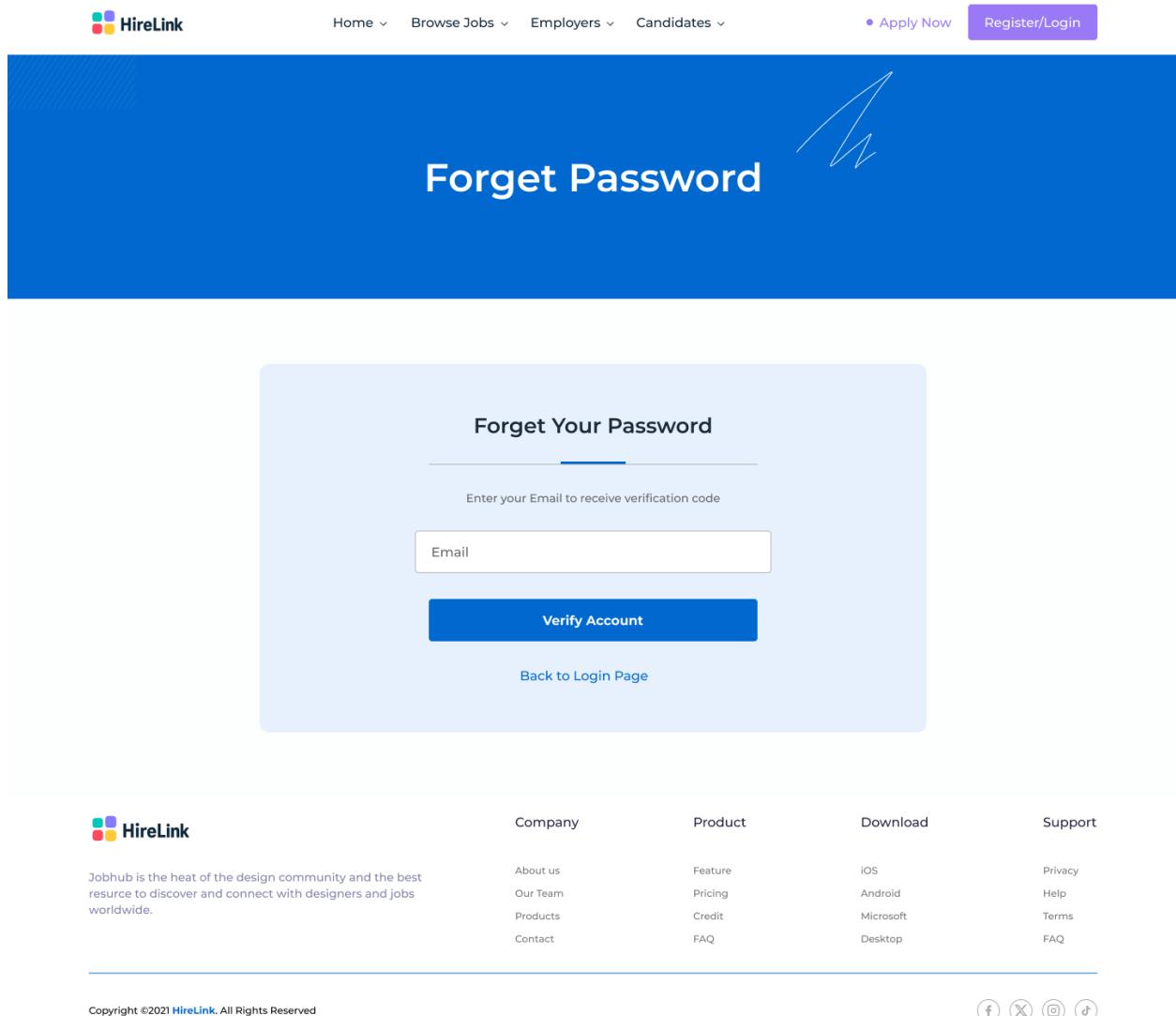


Figure 78: UI Design of Login Page

### 4.13.3. UI Design of Forget Password Page



The image shows the 'Forget Password' page of the HireLink website. At the top, there is a navigation bar with links for Home, Browse Jobs, Employers, Candidates, Apply Now, and Register/Login. The main title 'Forget Password' is centered on a blue background. Below the title is a text input field labeled 'Email' with a placeholder 'Enter your Email to receive verification code'. A large blue button labeled 'Verify Account' is positioned below the input field. At the bottom left, there is a link 'Back to Login Page'. The footer of the website includes the HireLink logo, a brief description of the platform, and links to various sections like Company, Product, Download, and Support. Social media icons for Facebook, Twitter, LinkedIn, and YouTube are also present.

Figure 79: UI Design of Forget Password Page

#### 4.13.4. UI Design of New Password Page

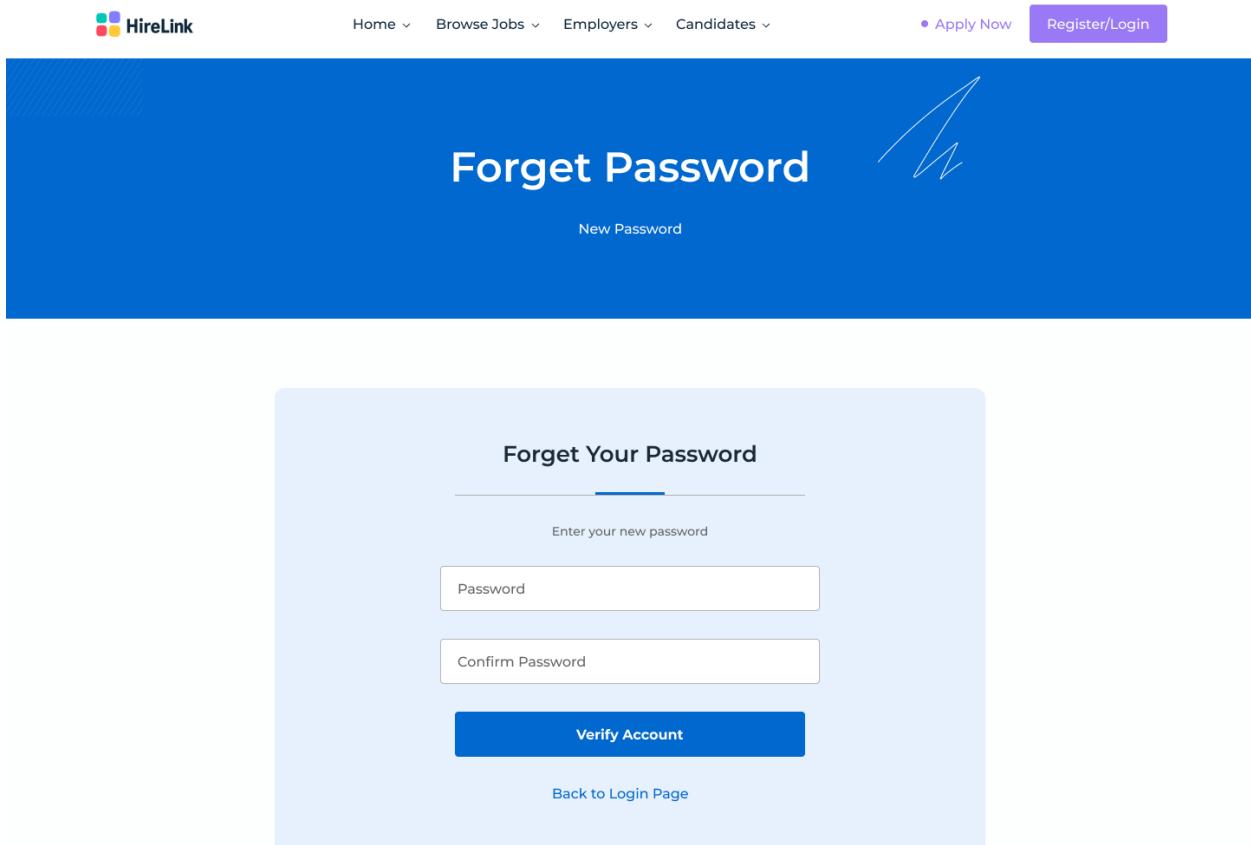


Figure 80: UI Design of New Password Page

#### 4.13.5. UI Design of Verify Account Page

**Enter Verification Code**

Please enter 6-digit code sent to your email address.

**Verify Account**

**Resend Code**

[Back to Register Page](#)

Company	Product	Download	Support
About us	Feature	iOS	Privacy
Our Team	Pricing	Android	Help
Products	Credit	Microsoft	Terms
Contact	FAQ	Desktop	FAQ

Copyright ©2021 [HireLink](#). All Rights Reserved

Figure 81: UI Design of Verify Account Page

[For more UI Designs of different pages, refer to the Appendix.](#)

## 4.14. Development Status

### 4.14.1. Setup of Frontend for HireLink Project

```

src > App.tsx > App
3 import Login from "./pages/login";
4 import VerificationPage from "./pages/verificationPage";
5 import ForgotPassword from "./pages/forgotPassword";
6 import NewPassword from "./pages/newPassword";
7 import RecruiterHomePage from "./pages/recruiterHomePage";
8 import CandidateHomePage from "./pages/candidateHomePage";
9 import AdminHomePage from "./pages/adminHomePage";
10 import CandidateProfilePage from "./pages/candidateProfilePage";
11 import "./App.css";
12
13 function App() {
14   return (
15     <Router>
16       <Routes>
17         <Route path="/" element={<Register />} />
18         <Route path="/register" element={<Register />} />
19         <Route path="/login" element={<Login />} />
20         <Route path="/verify-email" element={<VerificationPage />} />
21         <Route path="/forget-password" element={<ForgotPassword />} />
22         <Route path="/reset-password" element={<NewPassword />} />
23         <Route path="/recruiter-home" element={<RecruiterHomePage />} />
24         <Route path="/candidate-home" element={<CandidateHomePage />} />
25         <Route path="/admin-dashboard" element={<AdminHomePage />} />
26         <Route path="/candidate-profile" element={<CandidateProfilePage />} />
27       </Routes>
28     </Router>
29   );
30 }
31
32 export default App;
33

```

Figure 82: Setup of Frontend for HireLink Project

### 4.14.2. Setup of Backend for HireLink Project

```

config > db.js > ...
1 const mongoose = require("mongoose");
2
3 const connectDB = async () => {
4   try {
5     await mongoose.connect(process.env.MONGO_URI);
6     console.log("MongoDB connected successfully");
7   } catch (error) {
8     console.error("Database connection error:", error.message);
9     process.exit(1);
10 }
11
12 module.exports = connectDB;
13
14

```

Figure 83: Setup of Backend for HireLink Project

#### 4.14.3. Collection of Data in MongoDB

The screenshot shows the MongoDB Compass application interface. The left sidebar lists connections: 'cluster0.7ekc5qg.mongodb.net' (selected), 'admin', 'config', 'local', 'myapp' (selected), and 'users'. The main area shows the 'users' collection under 'myapp'. It has 6 documents, 2 validation rules, and 2 indexes. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The results table shows the following data:

_id	full_name	email	password	role	isVerified	verificationCode	verificationCodeExpires	resetCode	resetCodeExpires	createdAt	updatedAt
<code>ObjectID('599338ed09d0ac04522bf108')</code>	"Pratyush Badal"	"pratyushbadal199@gmail.com"	"\$2b\$10\$2A4HrG2iFA1A9XU1/88YHu8A1tqC8D1TsBu12NBV5zorPpHWN1e"	"user"	true	null	null	null	null	2025-12-08T02:32:53.339+00:00	2025-12-08T02:35:12.326+00:00
<code>ObjectID('599338ed09d0ac04522bf109')</code>	"Hirvelink@gmail.com"	"hirvelink@gmail.com"	"\$2b\$10\$uIu0vVfR0MuOKL0n0xV1HEFOqYDuuia4vIHv4Amena7eBaudcUFHLsq"	"user"	false	null	null	null	null	2025-12-09T01:52:14.019+00:00	2025-12-09T01:51:12.653+00:00
<code>ObjectID('599338ed09d0ac04522bf10a')</code>	"Bhishma Pratap Karki"	"bhishmakarki@gmail.com"	"\$2b\$10\$Ku3puJfmcwQZ2cOuvaBmlK\$uak0V7Flh1mH41nh0ClemeWVSC1"	"user"	false	null	null	null	null	2025-12-09T01:52:14.019+00:00	2025-12-09T01:51:12.653+00:00

Figure 84: Collection of Data in MongoDB

#### 4.14.4. Register Functionality

##### i. Backend code of Registration

```

File Edit Selection View Go Run Terminal Help <- > Q backend
EXPLORER JS UserController.js
OPEN EDITORS JS db.js config JS userModel.js controllers
BACKEND assets bin config controllers
JS db.js config JS userModel.js controllers
JS assets JS db.js config JS userModel.js controllers
JS bin JS config JS controllers
JS controllers
JS changePasswordController.js JS userModel.js controllers
JS profileController.js JS userModel.js controllers
JS UserController.js JS userModel.js controllers
JS middleware
JS userModel.js JS userModel.js controllers
JS node_modules
JS public
JS routes
JS passwordRoutes.js
JS profileRoutes.js
JS userRoutes.js
JS verificationRoutes.js
JS utils
JS emailUtils.js
JS passwordResetEmail.js
JS tokenUtils.js
JS WelcomeEmailUtils.js
JS .env
JS .gitignore
JS app.js
JS package-lock.json
JS package.json
JS server.js
OUTLINE
TIMELINE

```

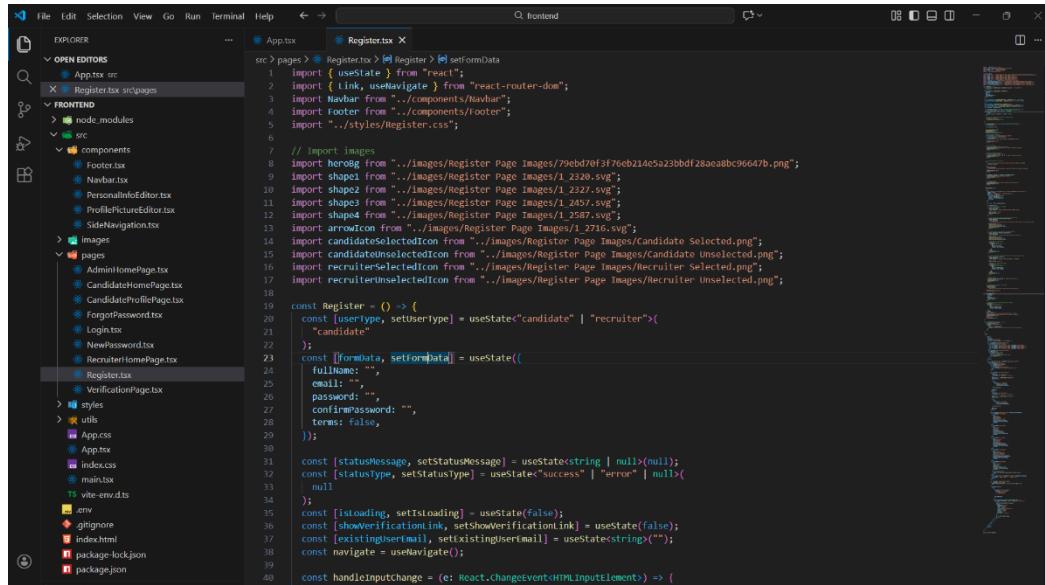
```

File Edit Selection View Go Run Terminal Help <- > Q backend
EXPLORER JS db.js JS UserController.js JS authMiddleware.js JS userModel.js
OPEN EDITORS JS db.js config JS userModel.js controllers
BACKEND assets bin config controllers
JS db.js config JS userModel.js controllers
JS assets JS config JS controllers
JS bin JS config JS controllers
JS controllers
JS changePasswordController.js JS userModel.js controllers
JS profileController.js JS userModel.js controllers
JS UserController.js JS userModel.js controllers
JS middleware
JS authMiddleware.js JS userModel.js controllers
JS models
JS userModel.js JS userModel.js controllers
JS node_modules
JS public
JS routes
JS passwordRoutes.js
JS profileRoutes.js
JS userRoutes.js
JS verificationRoutes.js
JS utils
JS emailUtils.js
JS passwordResetEmail.js
JS tokenUtils.js
JS WelcomeEmailUtils.js
JS .env
JS .gitignore
JS app.js
JS package-lock.json
JS package.json
JS server.js
OUTLINE
TIMELINE

```

Figure 85: Backend codes of Registration

## ii. Frontend code of Registration



The screenshot shows the VS Code interface with two tabs open: 'App.tsx' and 'Register.tsx'. The 'Register.tsx' tab is active, displaying the following code:

```

src > pages > Register > Register.tsx
import { useState } from "react";
import { Link, Navigate } from "react-router-dom";
import Navbar from "../components/Navbar";
import Footer from "../components/Footer";
import "./styles/register.css";

// Import images
import heroImg from "../images/Register Page Images/79ebd70f3f76eb214e5a23bbdf28aea8bc966d7b.png";
import shape1 from "../images/Register Page Images/1_2328.svg";
import shape2 from "../images/Register Page Images/1_2327.svg";
import shape3 from "../images/Register Page Images/1_2457.svg";
import shape4 from "../images/Register Page Images/1_2587.svg";
import arrowIcon from "../images/Register Page Images/1_2714.svg";
import candidateSelectedIcon from "../Images/Register Page Images/Candidate Selected.png";
import recruiterSelectedIcon from "../Images/Register Page Images/Recruiter Selected.png";
import recruiterUnselectedIcon from "../Images/Register Page Images/Recruiter Unselected.png";

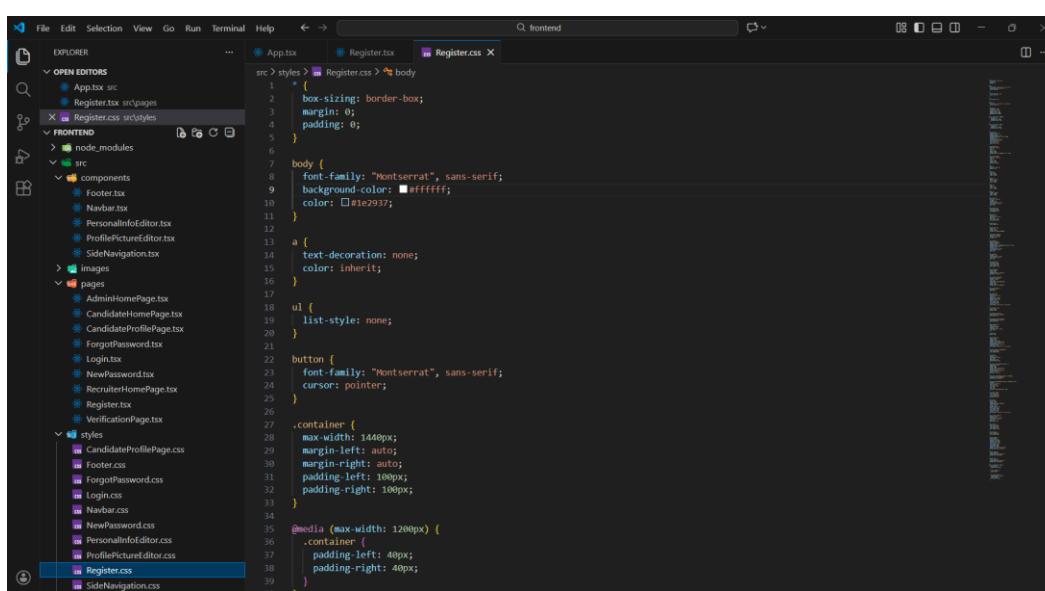
const Register = () => {
  const [userType, setUserType] = useState("candidate" | "recruiter");
  const [formData, setFormData] = useState({
    fullname: "",
    email: "",
    password: "",
    confirmPassword: "",
    terms: false,
  });

  const [statusMessage, setStatusMessage] = useState<string | null>(null);
  const [statusType, setStatusType] = useState<"success" | "error" | null>(null);

  const [isLoading, setIsLoading] = useState(false);
  const [showVerificationLink, setShowVerificationLink] = useState(false);
  const [existingEmail, setExistingEmail] = useState<string>("");
  const navigate = useNavigate();

  const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    ...
  }
}

```



The screenshot shows the VS Code interface with three tabs open: 'App.tsx', 'Register.tsx', and 'Register.css'. The 'Register.css' tab is active, displaying the following CSS code:

```

body {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: "Montserrat", sans-serif;
  background-color: #ffffff;
  color: #1e2937;
}

a {
  text-decoration: none;
  color: inherit;
}

ul {
  list-style: none;
}

button {
  font-family: "Montserrat", sans-serif;
  cursor: pointer;
}

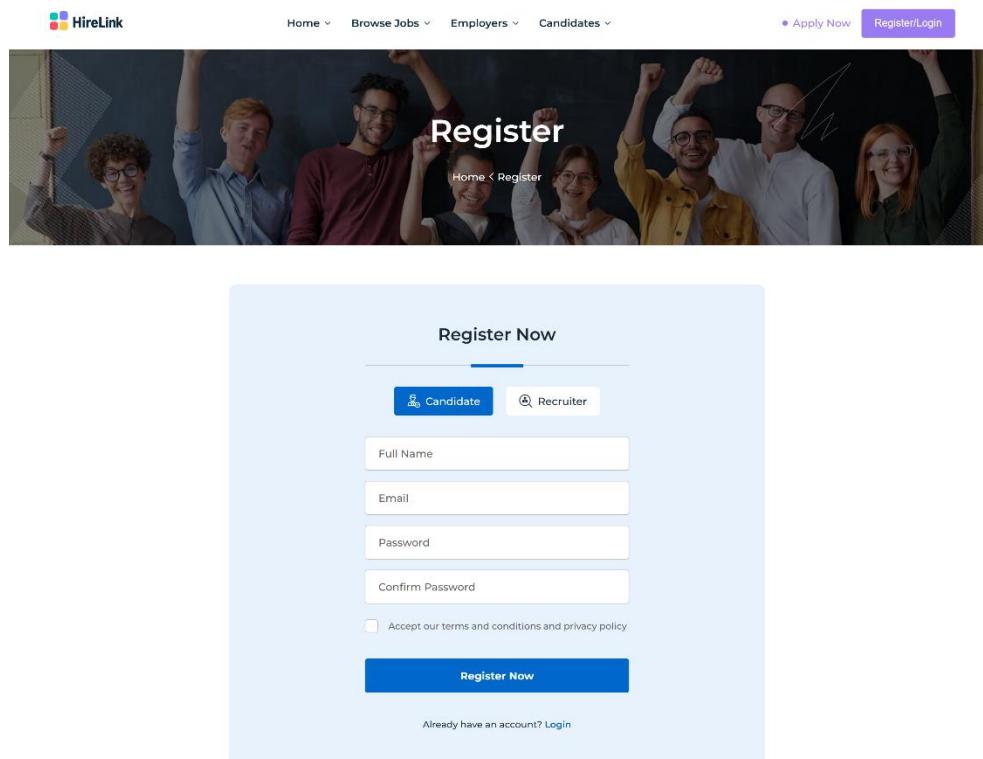
.container {
  max-width: 1400px;
  margin-left: auto;
  margin-right: auto;
  padding-left: 100px;
  padding-right: 100px;
}

@media (max-width: 1200px) {
  .container {
    padding-left: 40px;
    padding-right: 40px;
  }
}

```

Figure 86: Frontend codes of Registration

### iii. UI after Frontend code of Registration



The screenshot shows the 'Register Now' page of the HireLink website. At the top, there is a navigation bar with links for Home, Browse Jobs, Employers, Candidates, Apply Now, and Register/Login. Below the navigation is a banner featuring a group of diverse people with the word 'Register' overlaid. The main form area has a light blue background and contains fields for Full Name, Email, Password, and Confirm Password. There is also a checkbox for accepting terms and conditions and a 'Register Now' button. Below the form, a link to 'Login' is visible. At the bottom of the page, there is a footer with the HireLink logo, a brief description of Jobhub, and links for Company, Product, Download, and Support categories like About us, Feature, iOS, Privacy, etc.

*Figure 87: UI after Frontend code of Registration*

#### **4.14.5. Login Functionality**

#### i. Backend code of Login

The screenshot shows a browser window with a file tree on the left and a code editor on the right. The file tree includes files like `index.js`, `userController.js`, `userModel.js`, `config.js`, `controllers`, `routes`, `utils`, and `server.js`. The code editor displays `userController.js` with logic for user registration and login, including validation of email and password.

```
File Edit Selection View Go Run Terminal Help < > O backend
EXPLORE ... js userController.js x js userModel.js
OPEN EDITORS JS dbjs config
JS userController.js cont...
JS userModel.js models
BACKEND assets bin config JS dbjs controllers
changePasswordCon...
passwordController.js
middleWare
models userModel.js
node_modules
public routes
passwordRoutes.js
profileRoutes.js
usersRoutes.js
verificationsRoutes.js
utils
emailRoutes.js
passwordResetFinal...
tokenRoutes.js
WelcomeEmailRoutes.js
env
gitignore
app.js
package-lock.json
package.json
server.js
> OUTLINE Timeline
```

```
controllers > userController.js @ registerUser @ registerUser
1.0 > ...
1.1 // LOGIN USER
1.2 exports.loginUser = async (req, res) => {
1.3   try {
1.4     const { email, password } = req.body;
1.5
1.6     // Trim and normalize input
1.7     const normalizedEmail = (email || "").trim().toLowerCase();
1.8     const trimmedPassword = (password || "").trim();
1.9
1.10    // Validate required fields
1.11    if (!normalizedEmail || !trimmedPassword) {
1.12      return res.status(400).json({
1.13        success: false,
1.14        message: "Email and password are required",
1.15      });
1.16
1.17    // Email format validation
1.18    if (!emailRegex.test(normalizedEmail)) {
1.19      return res.status(400).json({
1.20        success: false,
1.21        message: "Please provide a valid email address",
1.22      });
1.23
1.24    // Find user by email
1.25    const user = await User.findOne({ email: normalizedEmail });
1.26
1.27    if (!user) {
1.28      return res.status(401).json({
1.29        success: false,
1.30        message: "Invalid email or password",
1.31      });
1.32
1.33    // Check if user is verified
1.34    if (!user.isVerified) {
1.35      // If code is not expired, inform user to verify
1.36      if (user.verificationCode && user.verificationCodeExpires > new Date()) {
1.37        const timestamp = Math.floor((
```

*Figure 88: Backend codes of Login*

## ii. Frontend codes of Login

*Figure 89: Frontend codes of Login*

### iii. UI after frontend code of Login



**Login Now**

---

[Login Now](#)

[Forgot Password?](#)
  
 Don't have an account? [Register](#)



Jobhub is the heart of the design community and the best resource to discover and connect with designers and jobs worldwide.

<a href="#">Company</a>	<a href="#">Product</a>	<a href="#">Download</a>	<a href="#">Support</a>
<a href="#">About us</a>	<a href="#">Feature</a>	<a href="#">iOS</a>	<a href="#">Privacy</a>
<a href="#">Our Team</a>	<a href="#">Pricing</a>	<a href="#">Android</a>	<a href="#">Help</a>
<a href="#">Products</a>	<a href="#">Credit</a>	<a href="#">Microsoft</a>	<a href="#">Terms</a>
<a href="#">Contact</a>	<a href="#">FAQ</a>	<a href="#">Desktop</a>	<a href="#">FAQ</a>

Copyright ©2021 [HireLink](#). All Rights Reserved

(f) (X) (@) (d)

#### **4.14.6. Verification Code Functionality**

#### i. Backend Code of Verification Code

```
File Edit Selection View Go Run Terminal Help < - > Q backend
```

```
EXPLORER OPEN EDITORS controllers > JS UserController.js > registerUser > registerUser
JS db.js config JS userModel.js
JS UserController.js controllers > JS UserController.js > registerUser > registerUser
JS userModel.js models
BACKEND assets bin config JS db.js
controllers
JS changePasswordCon...
JS profileController.js
JS UserController.js
middleware
models
JS UserModel.js
node_modules
public
routes
JS passwordRoutes.js
JS profileRoutes.js
JS userRoutes.js
JS verificationRoutes.js
utils
JS emailUtils.js
JS passwordResetEmail...
JS tokenUtils.js
JS welcomeEmailUtils.js
.env
.gitignore
app.js
package-lock.json
package.json
server.js
```

```
14 exports.registerUser = async (req, res) => {
  ...
}

// Generate verification code (15 minutes expiration)
const verificationCode = generateVerificationCode();
const verificationCodeExpires = new Date(Date.now() + 15 * 60 * 1000);

// Create new user
const newUser = new User({
  fullName,
  email,
  password,
  role,
  isVerified: false,
  verificationCode,
  verificationCodeExpires,
});

await newUser.save();

// Send verification email
await sendVerificationEmail(email, verificationCode);

res.status(201).json({
  success: true,
  message: "Registration successful! Verification code sent to your email.",
  user: {
    id: newUser.id,
    fullName: newUser.fullName,
    email: newUser.email,
    role: newUser.role,
  },
  requiresVerification: true,
});
} catch (error) {
  console.error("Registration Error:", error.message);
  res.status(500).json({
    success: false,
    message: "An error occurred during registration."
  });
}
```

*Figure 90: Backend Code of Verification Code*

## ii. Frontend Code of Verification Code

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows the project structure:
  - OPEN EDITORS:** Login.tsx, Login.css, VerificationPage.tsx
  - FRONTEND:** src
    - pages
      - ForgotPassword.tsx
      - Login.tsx
      - NewPassword.tsx
      - RecruiterHomePage.tsx
      - Register.tsx
      - VerificationPage.tsx
    - styles
      - CandidateProfilePage.css
      - Footer.css
      - ForgotPassword.css
      - LogIn.css
      - Navbar.css
      - NewPassword.css
      - PersonallofficeEditor.css
      - ProfilePictureEditor.css
      - Register.css
      - SideNavigation.css
      - VerificationPage.css
    - utils
      - App.css
      - App.tsx
- Frontend:** A tab bar with "frontend" selected.
- Code Editor:** The VerificationPage.tsx file is open, showing the following TypeScript code:

```
src > pages > VerificationPage.tsx > VerificationPage > handleResendCode
1 import { useState, useEffect } from "react";
2 import { Link, useNavigate, useLocation } from "react-router-dom";
3 import Navbar from "../components/Navbar";
4 import Footer from "../components/Footer";
5 import "./styles/VerificationPage.css";
6
7 // Define typescript interfaces for API responses
8 interface VerificationResponse {
9   errorMessage?: string;
10   success?: boolean;
11   codeExpired?: boolean;
12   rescheduleToken?: string;
13   verified?: boolean;
14   user?: {
15     id: string;
16     fullname: string;
17     email: string;
18     role: string;
19     isVerified: boolean;
20   };
21 }
22
23 interface StatusResponse {
24   isVerified: boolean;
25   hasPendingVerification: boolean;
26   expiresAt?: string;
27   timeout?: number;
28   isExpired?: boolean;
29 }
30
31 const VerificationPage = () => {
32
33   return (
34     <div>
35       <h1>Verification Page</h1>
36       <p>This is the verification page. Please check your email for further instructions.</p>
37       <Link to="/">Go Back</Link>
38     </div>
39   );
40 }
41
42 export default VerificationPage;
```

*Figure 91: Frontend Code of Verification Code*

### iii. UI after frontend code of Verification Code

The screenshot shows the 'Enter Verification Code' page of the HireLink website. At the top, there is a navigation bar with links for Home, Browse Jobs, Employers, Candidates, Apply Now, and Register/Login. The main content area has a light blue background and features a title 'Enter Verification Code'. Below it, a message says 'Please enter 6-digit code sent to skillsyncnep2025@gmail.com'. There are six input fields for entering the verification code. A green success message box contains the text 'Registration successful! Verification code sent to your email.' Below this is a 'Verify Account' button. At the bottom left, it says 'Code expires in: 14:19'. At the bottom right, there are links to 'Back to Register Page' and 'Back to Login Page'. The footer of the page includes the HireLink logo, a brief description of Jobhub, and links to Company, Product, Download, and Support sections. It also features social media icons for Facebook, Twitter, Instagram, and LinkedIn.

Company	Product	Download	Support
About us	Feature	iOS	Privacy
Our Team	Pricing	Android	Help
Products	Credit	Microsoft	Terms
Contact	FAQ	Desktop	FAQ

Copyright ©2021 [HireLink](#). All Rights Reserved

*Figure 92: UI after frontend code of Verification Code*

#### 4.14.7. Forget Password Functionality

##### i. Backend Code of Forget Password

```

    controllers > changePasswordController.js > recordResetCode > recordResetCode
    controllers > changePasswordController.js > recordResetCode -> json (req, res) => {
      ...
    };

    // Generate reset code (15 minutes expiration)
    const resetCode = generateResetCode();
    const resetCodeExpires = new Date(Date.now() + 15 * 60 * 1000); // 15 minutes

    // Update user with reset code
    user.resetCode = resetCode;
    user.resetCodeExpires = resetCodeExpires;
    await user.save();

    // Send password reset email using utility
    const emailSent = await sendPasswordResetEmail(normalizedEmail, resetCode);

    if (!emailSent) {
      return res.status(500).json({
        message: "Failed to send reset email. Please try again.",
        success: false,
      });
    }

    res.status(200).json({
      message: "Password reset code sent to your email.",
      success: true,
      email: normalizedEmail,
      expiration: resetCodeExpires,
    });
  } catch (error) {
    console.error("Forgot password request error:", error);
    res.status(500).json({
      message: "An error occurred during password reset request",
      error: error.message,
    });
  }
}
  
```

Figure 93: Backend Code of Forget Password

##### ii. Frontend Code of Forget Password

```

    src > pages > ForgotPassword.tsx > ForgotPassword > statusType
    1 import { useState } from "react";
    2 import { Link, useNavigate } from "react-router-dom";
    3 import Navbar from "../Components/Navbar";
    4 import Footer from "../Components/Footer";
    5 import "../styles/ForgotPassword.css";
    6
    7 // Import images from Forgot Password Page Images folder
    8 import decoShapeTopRight from "../images/Login Page Images/16_1331.svg";
    9
    10 const ForgotPassword = () => {
    11   const [email, setEmail] = useState("");
    12   const [isSubmitting, setIsSubmitting] = useState(false);
    13   const [statusMessage, setStatusMessage] = useState("");
    14   const [statusType, setStatusType] = useState("success" | "error");
    15   const navigate = useNavigate();
    16
    17   const handleSubmit = async (e: React.FormEvent) => {
    18     e.preventDefault();
    19
    20     // Basic validation
    21     if (!email.trim()) {
    22       setStatusMessage("Please enter your email address");
    23       setStatusType("error");
    24       return;
    25     }
    26
    27     const emailRegex = /^[^@\s]+@[^\s]+\.[^\s]+$/;
    28     if (!emailRegex.test(email)) {
    29       setStatusMessage("Please enter a valid email address");
    30       setStatusType("error");
    31       return;
    32     }
  
```

Figure 94: Frontend Code of Forget Password

### iii. UI after frontend code of Forget Password Page

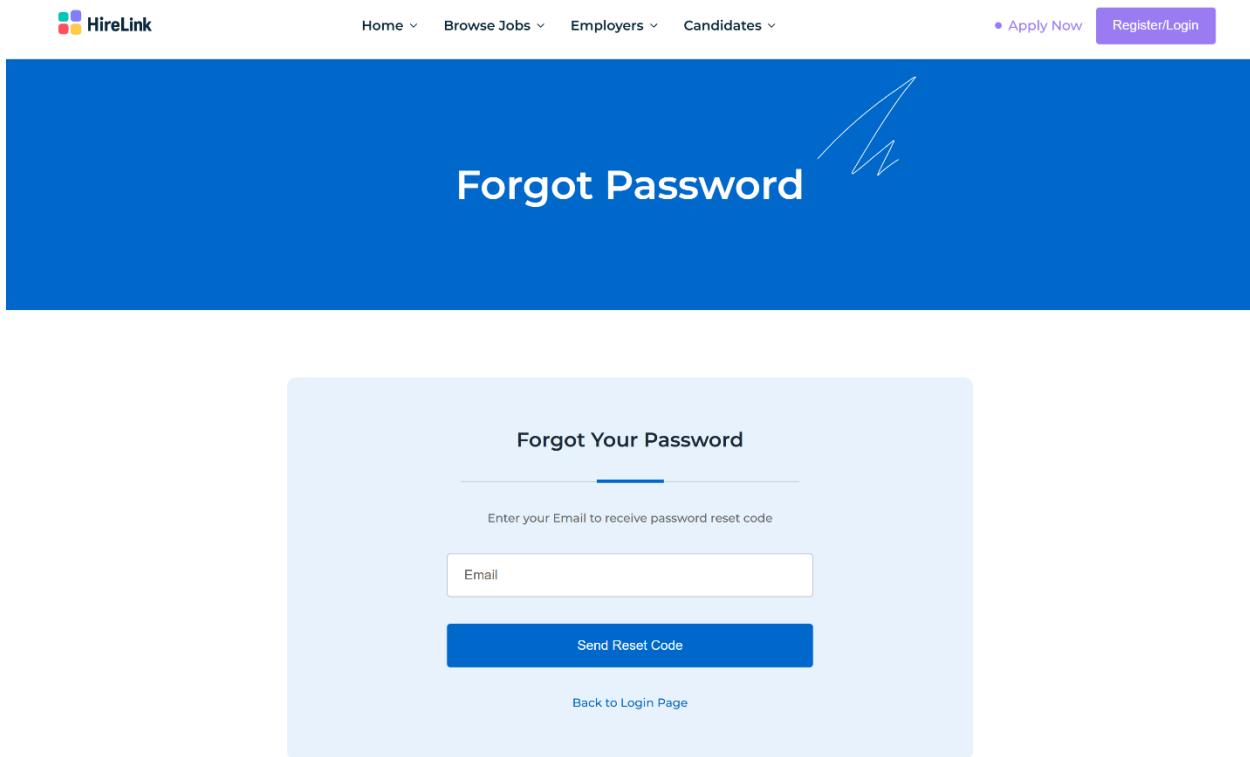


Figure 95: UI after frontend code of Forget Password Page

#### **4.14.8. Reset Password Functionality**

#### i. Backend Code of Reset Password

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it displays the project structure. The `changePasswordController.js` file is open in the editor.
- Editor:** The main pane shows the code for the `changePasswordController.js` file. The code handles a POST request to reset a password using a token. It validates the input, verifies the token, finds the user by ID, and performs password strength validation before updating the user's password.
- Search:** A search bar at the top right contains the text "backend".
- Terminal:** There is no visible terminal output in the bottom right corner.

```
// RESET PASSWORD
exports.resetPassword = async (req, res) => {
  try {
    const { token, newPassword } = req.body;

    // Validate input
    if (!token || !newPassword) {
      return res.status(400).json({
        message: "Token and new password are required",
      });
    }

    // Verify token
    const decoded = verifyToken(token);
    if (!decoded) {
      return res.status(401).json({
        message: "Invalid or expired reset token",
      });
    }

    // Find user by ID
    const user = await User.findById(decoded.id);
    if (!user) {
      return res.status(404).json({
        message: "User not found",
      });
    }

    // Password strength validation (same as registration)
    if (newPassword.length < 8) {
      return res.status(400).json({
        message: "Password must be at least 8 characters long",
      });
    }

    if (!/[A-Z]/.test(newPassword) || !/[a-z]/.test(newPassword)) {
      return res.status(400).json({
        message: "Password must contain upper and lower case letters",
      });
    }
  } catch (err) {
    console.error(err);
    return res.status(500).json({
      message: "Internal server error",
    });
  }
};
```

*Figure 96: Backend Code of Reset Password*

## ii. Frontend Code of Reset Password

```
src > pages > NewPassword.tsx > NewPassword.tsx
1 import { useState } from "react";
2 import { Link, useNavigate, useLocation } from "react-router-dom";
3 import Navbar from "../components/Navbar";
4 import Footer from "../components/Footer";
5 import "./styles/newpassword.css";
6
7 // Import images from Forget Password Page Images folder
8 import decorRight from "../images/Login Page Images/16.1331.svg";
9
10 const NewPassword = () => {
11   const [email, setEmail] = useState("");
12   const [password, setPassword] = useState("");
13   const [confirmPassword, setConfirmPassword] = useState("");
14
15   const [isLoading, setIsLoading] = useState(false);
16   const [statusMessage, setStatusMessage] = useState("");
17   const [statusType, setStatusType] = useState("success" || "error");
18   const navigate = useNavigate();
19   const location = useLocation();
20
21   // Get email and token from location state
22   const email = location.state?.email || "";
23   const resetToken = location.state?.token || "";
24   const initialMessage = location.state?.message || "";
25
26   const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
27     const [name, value] = e.target;
28     if (name === "prev") {
29       setResetToken(value);
30     } else {
31       setEmail(value);
32     }
33   };
34
35   const handleSubmit = async (e: React.FormEvent) => {
36     e.preventDefault();
37
38     // Basic validation
39     if (!FormData.password.trim()) {
```

*Figure 97: Frontend Code of Reset Password*

### iii. UI after Frontend code of Reset Password

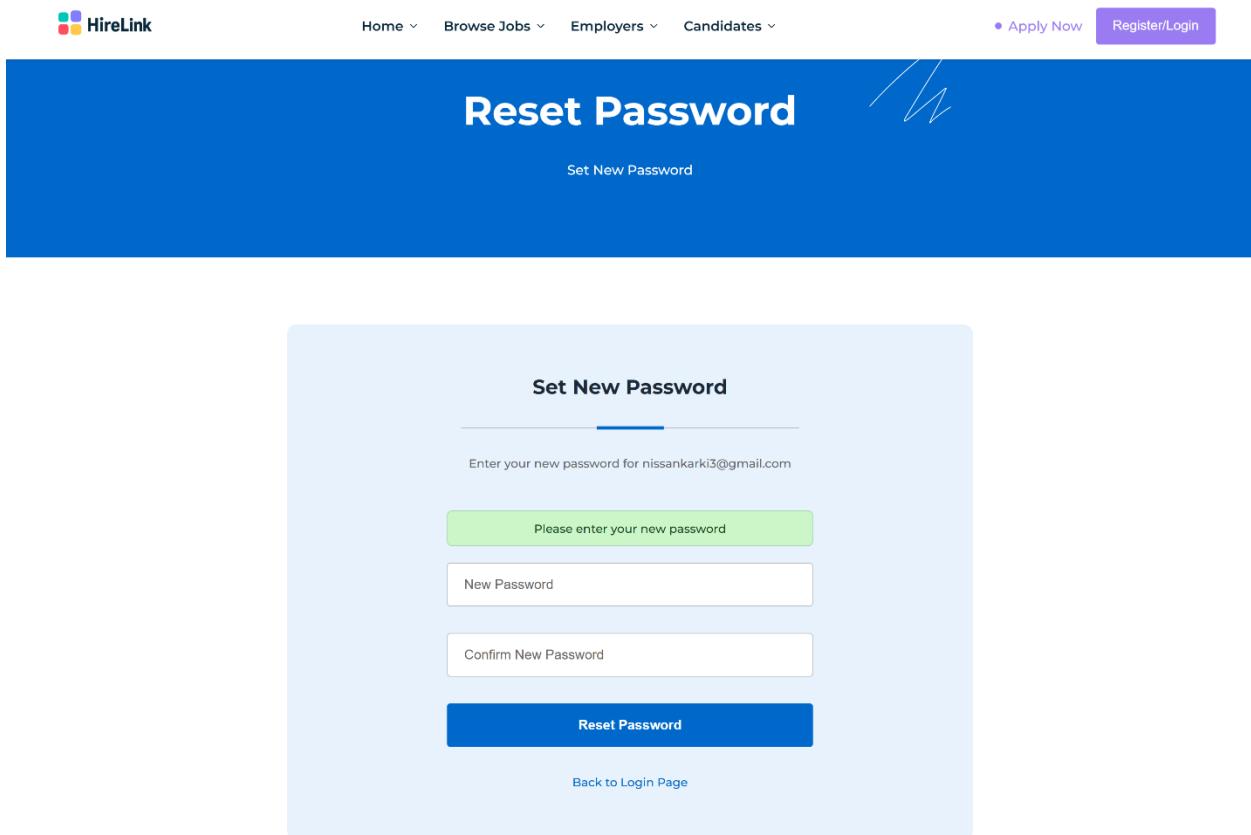
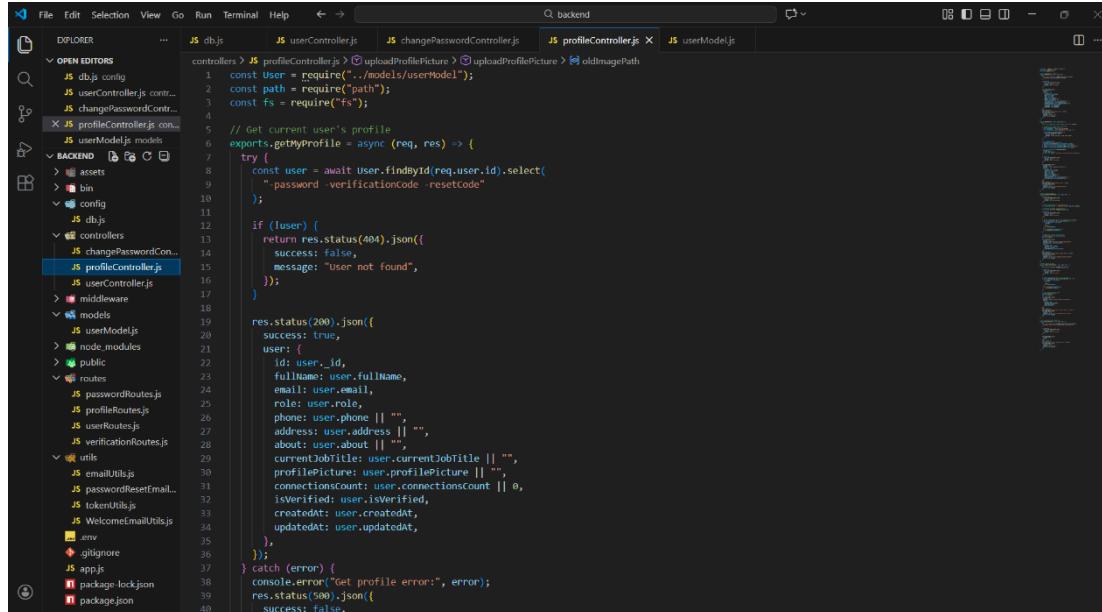


Figure 98: UI after Frontend code of Reset Password

#### 4.14.9. Candidate Portfolio and Profile Page

##### i. Backend Code of Portfolio and Profile Page



```

File Edit Selection View Go Run Terminal Help < - > Q backend
OPEN EDITORS JS db.js JS userController.js JS changePasswordController.js JS profileController.js X JS userModel.js
controllers > JS profileController.js > uploadProfilePicture > uploadProfilePicture > oldImagePath
1 const User = require("../models/userModel");
2 const path = require("path");
3 const fs = require("fs");
4
5 // Get current user's profile
6 exports.getMyProfile = async (req, res) => {
7   try {
8     const user = await User.findById(req.user.id).select(
9       "-password -verificationCode -resetCode"
10    );
11
12     if (!user) {
13       return res.status(404).json({
14         success: false,
15         message: "User not found",
16       });
17   }
18
19   res.status(200).json({
20     success: true,
21     user: {
22       id: user._id,
23       fullName: user.fullName,
24       email: user.email,
25       role: user.role,
26       phone: user.phone || "",
27       address: user.address || "",
28       about: user.about || "",
29       currentJobTitle: user.currentJobTitle || "",
30       profilePicture: user.profilePicture || "",
31       connectionsCount: user.connectionsCount || 0,
32       isVerified: user.isVerified,
33       createdAt: user.createdAt,
34       updatedAt: user.updatedAt,
35     },
36   });
37 } catch (error) {
38   console.error("Get profile error:", error);
39   res.status(500).json({
40     success: false,
41   });
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
635
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1
```

### iii. UI after Frontend Code of Portfolio and Profile Page

The screenshot shows the 'My Profile' section of the HireLink platform. At the top, there's a header with the 'HireLink' logo and a profile picture of Bhishma Pratap Karki. Below the header, the user's name 'Bhishma Pratap Karki' is displayed along with their title 'UI/UX Designer' and the number '+144 Connections'. A sidebar on the left contains links for 'Dashboard', 'My Profile' (which is highlighted in blue), 'Resume', 'Messages', 'Job Alerts', 'Saved Job', and 'Settings'. The main content area is divided into several sections: 'Personal Information' (with fields for Email, Phone Number, and Address), 'About' (with a note that no about information is provided), 'Resume' (with a note that the resume stands as the most crucial document), 'Experience' (with a note to outline employment particulars), 'Education' (with a note to provide information about educational background), 'Skills' (listing 'UI/UX Design'), 'Certifications' (listing 'Design User Experiences with Figma' and 'Full Stack Development with JavaScript, React and Node Certificate'), and 'Project' (listing 'Mobile E-Wallet App Design'). Each section has a '+ Add' button and a edit icon.

Figure 101: UI after Frontend Code of Portfolio and Profile Page

## 5. Analysis of Progress

### 5.1. Progress Review

The project has progressed in line with the planned Work Breakdown Structure and Gantt Chart. The Initiation and Planning phases were completed on schedule, including defining project goals, identifying stakeholders, setting the project scope, conducting feasibility and requirements analysis, creating the product backlog, and planning sprints. The proposal submission milestone was successfully achieved on December 3, 2025, marking the completion of the planning stage.

**Sprint 1, Core Setup and Authentication**, was carried out from December 1 to December 15, 2025, and was completed as planned. Key tasks such as setting up the development environment, creating the project structure, designing the Register and Login UI, building the authentication system, and testing core authentication features were successfully completed. The Sprint 1 milestone was achieved on December 15, 2025, providing a strong base for further development.

The project is currently in **Sprint 2, Portfolio and Profile**, running from December 16, 2025, to January 5, 2026. This sprint focuses on developing portfolio and profile features for Candidates, Recruiters, and Admins. UI design and partial implementation have been completed, and the sprint was extended due to the time required for multiple user profiles and academic workload. Overall, the project remains on track under the revised timeline and is aligned with the final project deadline.

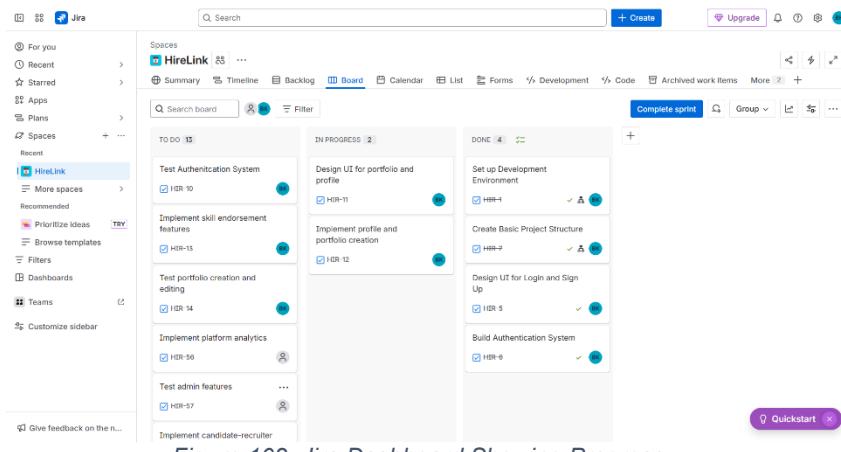


Figure 102: Jira Dashboard Showing Progress

## 5.2. Progress Table

Task	Start Date	End Date	Status	Percent
<b>1. Initiation</b>	<b>November 9, 2025</b>	<b>November 16, 2025</b>	<b>Completed</b>	<b>100%</b>
1.1. Define project goals and expected outcomes	November 9, 2025	November 11, 2025	Completed	100%
1.2. Identify stakeholders	November 9, 2025	November 11, 2025	Completed	100%
1.3. Define project scope	November 11, 2025	November 13, 2025	Completed	100%
1.4. Feasibility and requirement analysis	November 12, 2025	November 15, 2025	Completed	100%
1.5. Get project approval	November 16, 2025	November 16, 2025	Completed	100%
<b>2. Planning</b>	<b>November 17, 2025</b>	<b>November 30, 2025</b>	<b>Completed</b>	<b>100%</b>
2.1. Create product backlog	November 17, 2025	November 20, 2025	Completed	100%
2.2. Prioritize backlog items	November 20, 2025	November 22, 2025	Completed	100%
2.3. Estimate effort for each task	November 22, 2025	November 24, 2025	Completed	100%
2.4. Plan sprints and timeline	November 24, 2025	November 26, 2025	Completed	100%
2.5. Identify risks and mitigation	November 26, 2025	November 28, 2025	Completed	100%
2.6. Plan communication and documentation	November 28, 2025	November 30, 2025	Completed	100%
<b>Milestone: Proposal Submission</b>	<b>December 03, 2025</b>	<b>December 03, 2025</b>	<b>Completed</b>	<b>100%</b>
<b>3. Sprint Execution</b>	<b>December 01, 2025</b>	<b>March 31, 2026</b>		
3.1. Sprint 1: Core Setup and Authentication	December 01, 2025	December 15, 2025	Completed	100%
3.2. Sprint 2: Portfolio and Profile	December 16, 2025	January 05, 2026	Working	40%
3.3. Sprint 3: Job Discovery and Application Tracking	January 05, 2026	January 24, 2026	Not Completed	0%
3.4. Sprint 4: Recruiter Features	January 25, 2026	February 11, 2026	Not Completed	0%
3.5. Sprint 5: Admin Dashboard	February 12, 2026	February 27, 2026	Not Completed	0%
3.6. Sprint 6: Notifications, Messaging and Networking	February 28, 2026	March 15, 2026	Not Completed	0%
3.7. Sprint 7: Testing and Final Documentation	March 16, 2025	March 31, 2026	Not Completed	0%
<b>4. Project Documentation</b>	<b>April 05, 2026</b>	<b>April 11, 2026</b>	<b>Not Completed</b>	<b>0%</b>
<b>5. Collection of Feedback</b>	<b>April 14, 2026</b>	<b>April 17, 2026</b>	<b>Not Completed</b>	<b>0%</b>
<b>6. Submit final report and close project</b>	<b>April 18, 2026</b>	<b>April 18, 2026</b>	<b>Not Completed</b>	<b>0%</b>

Figure 103: Progress Table

### 5.3. Action Plan

[For detailed Action Plan, refer to Appendix.](#)

## 6. Future Work

The upcoming tasks will focus on completing the remaining development and enhancement of the Smart Job and Skill Portfolio platform. This includes finalizing portfolio and profile features for all user roles, ensuring data validation, skill endorsements, and smooth user experience.

Next, the system will implement job discovery, AI-based job recommendations, and application tracking for candidates, along with job posting, candidate search, and ATS resume scanning for recruiters. Admin functionalities such as user management, quiz and assessment management, dashboards, and analytics will also be completed.

Additionally, messaging, notifications, and professional networking features will be integrated and tested thoroughly. The project will then undergo full system testing, optimization, and bug fixing to ensure stability. Finally, documentation, reporting, and presentation preparation will be completed to make the system ready for deployment and use.

## 7. Conclusion

The development of the Smart Job and Skill Portfolio platform has progressed steadily according to the planned sprints and schedule. Foundational work including project initiation, planning, and Sprint 1 tasks such as core setup and authentication has been completed successfully. Sprint 2 is currently in progress, focusing on portfolio and profile features for multiple user roles, which are essential for system functionality.

Analysis of similar platforms helped identify gaps and guided the design of a more integrated and complete solution. The project follows a structured iterative approach to implement, test, and refine features effectively. Regular reviews and action planning have ensured alignment with objectives, and continued development, testing, and documentation will deliver a fully functional, user-friendly, and efficient platform for candidates, recruiters, and admins.

## 8. References

Board Infinity. (2025) *Board Infinity* [Online]. Available from:

<https://www.boardinfinity.com/blog/a-quick-guide-to-prototype-model-in-software-engineering/> [Accessed 27 December 2025].

Boehm, B.W. (1988) *A Spiral Model of Software Development and Enhancement*. New York, NY: IEE.

Bogen, M. & Rieke, A. (2025) *An Examination of Hiring Algorithms, Equity, and Bias* [Online]. Available at: <https://www.upturn.org/static/reports/2018/hiring-algorithms/files/Upturn%20-%20Help%20Wanted%20-%20An%20Exploration%20of%20Hiring%20Algorithms,%20Equity%20and%20Bias.pdf> [Accessed 18 December 2025].

Breezy HR. (2025) *Breezy HR* [Online]. Available from: <https://breezy.hr/> [Accessed 18 December 2025].

Chavan, P.R. et al. (2024) *Enhancing recruitment efficiency: An advanced Applicant Tracking System (ATS)* [Online]. Available at: [https://www.researchgate.net/publication/382099903\\_Enhancing\\_recruitment\\_efficiency\\_An\\_advanced\\_Applicant\\_Tracking\\_System\\_ATS](https://www.researchgate.net/publication/382099903_Enhancing_recruitment_efficiency_An_advanced_Applicant_Tracking_System_ATS) [Accessed 18 December 2025].

CIPD. (2024) *Resourcing and talent planning 2024 | Report* [Online]. Available at: <https://www.cipd.org.uk/knowledge/reports/resourcing-surveys/> [Accessed 18 December 2025].

Expertini. (2025) *Expertini* [Online]. Available from: <https://np.expertini.com/jobs/> [Accessed 29 December 2025].

Hatti, S.K. et al. (2025) *Efficient Resume Screening Tool* [Online]. Available at: [https://www.researchgate.net/publication/387930797\\_Efficient\\_Resume\\_Screening\\_Tool](https://www.researchgate.net/publication/387930797_Efficient_Resume_Screening_Tool) [Accessed 18 December 2025].

Hubstaff tasks. (2025) *What is Scrum methodology?* [Online]. Available from: <https://hubstaff.com/tasks/scrum-project-management> [Accessed 27 December 2025].

International Labour Organization. (2023) *World Employment and Social Outlook: Trends 2023* [Online]. Available at: <https://www.ilo.org/publications/flagship-reports/world-employment-and-social-outlook-trends-2023> [Accessed 18 December 2025].

Köchling, & Wehner , M.C. (2020) *Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development* [Online]. Available at: <https://link.springer.com/article/10.1007/s40685-020-00134-w> [Accessed 18 December 2025].

Magnet.me. (2025) *Magnet.me* [Online]. Available from: <https://magnet.me/> [Accessed 18 December 2025].

merojob. (2025) *merojob* [Online]. Available from: <https://merojob.com/skill/ai> [Accessed 29 December 2025].

Nikolaou, I. (2021) *What is the Role of Technology in Recruitment and Selection?* [Online]. Available at: [https://www.researchgate.net/publication/349031261\\_What\\_is\\_the\\_Role\\_of\\_Technology\\_in\\_Recruitment\\_and\\_Selection](https://www.researchgate.net/publication/349031261_What_is_the_Role_of_Technology_in_Recruitment_and_Selection) [Accessed 18 December 2025].

Peicheva, M. (2022) *DATA ANALYSIS FROM THE APPLICANT TRACKING SYSTEM* [Online]. Available at:  
[https://www.researchgate.net/publication/375517346\\_DATA\\_ANALYSIS\\_FROM\\_THE\\_APPLICANT\\_TRACKING\\_SYSTEM](https://www.researchgate.net/publication/375517346_DATA_ANALYSIS_FROM_THE_APPLICANT_TRACKING_SYSTEM) [Accessed 18 December 2025].

Pressman, R.S. (2010) *Software Engineering: A Practitioner's Approach*. 7th ed. New York: McGraw-Hill.

Radigan, D. (2025) *Kanban* [Online]. Available from:  
<https://www.atlassian.com/agile/kanban> [Accessed 27 December 2025].

Sommerville, I. (2011) *SOFTWARE ENGINEERING*. 9th ed. Boston: Addison-Wesley.

Taya, R., Dutta, D.P. & Goyal, M.S. (2023) *ADOPTION OF TECHNOLOGY ON RECRUITMENT PROCESS A STUDY OF PERCEPTIONS AND ATTITUDES OF HR PROFESSIONALS* [Online]. Available at: <https://ijcrt.org/papers/IJCRT2307426.pdf> [Accessed 18 December 2025].

Torre.ai. (2025) *Torre.ai*. [Online]. Available from: <https://torre.ai/> [Accessed 18 December 2025].

Tuttle, L. & Critchlow, K. (2024) *Digital transformation in talent acquisition: Modern approaches to recruitment and selection* [Online]. Available at:  
[https://www.researchgate.net/publication/391851238\\_Digital\\_transformation\\_in\\_talent\\_acquisition\\_Modern\\_approaches\\_to\\_recruitment\\_and\\_selection](https://www.researchgate.net/publication/391851238_Digital_transformation_in_talent_acquisition_Modern_approaches_to_recruitment_and_selection) [Accessed 18 December 2025].

## 9. Appendix

### 9.1. Review of Similar Projects

#### 9.1.1. Similar Project 2: Breezy HR

Breezy HR is a recruitment platform that helps small and medium-sized businesses manage their hiring process more efficiently. It provides applicant tracking, job posting to multiple boards, drag and drop candidates pipeline management, interview scheduling, and customizable career page. The platform also integrates with tools like Google Calendar, and email system, allowing recruiters to communicate and schedule interviews seamlessly. Real-time dashboards and analytics help monitor recruitment progress and measure performance, making the hiring process more structured and transparent (Breezy HR, 2025).

While Breezy HR provides excellent tools for managing applications and scheduling, it lacks advanced AI-driven job recommendations for candidates. Candidates cannot create detailed portfolios to showcase projects or certifications beyond their resumes. Additionally, the platform does not provide admin tools to manage quizzes, assessments, or system settings, limiting its uses for organizations that need a more integrated and interactive recruitment solution. These gaps show the opportunity for a platform like Smart Job and Skill Portfolio, which combines ATS resume scanner, application tracking, AI-based recommendations, portfolio management, and admin control in one system.

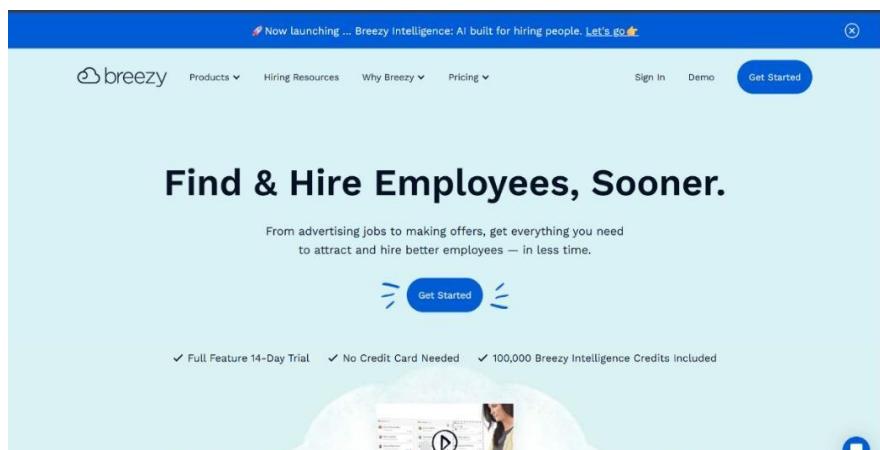


Figure 104: Similar Project 1 (Breezy HR, 2025)

### 9.1.2. Similar Project 3: Magnet.me

Magnet.me is an AI-powered career networking platform that connects students and graduates with employers. Candidates can create professional profiles, add skills, education, projects, and certifications, and then receive AI-based job recommendations. Employers can build a talent community, search for candidates, and maintain ongoing relationships beyond individual job postings. Magnet.me is particularly strong in building engagement and employer branding by allowing recruiters to maintain a pool of potential candidates for future opportunities (Magnet.me, 2025).

However, Magnet.me is limited in its recruitment management capabilities. It does not provide a full application tracking system, automated ATS resume scanner, or integrated dashboards for recruiters and admins. Skill assessments and quizzes for evaluating candidates are also missing. While the platform is excellent for networking and AI-based matching, it lacks the end-to-end functionality needed to manage applications, evaluate skills systematically, and provide detailed analytics. This makes the Smart Job and Skill Portfolio application more comprehensive, as it integrates AI-based recommendations, ATS resume scanner, portfolio management and admin control into a single platform.

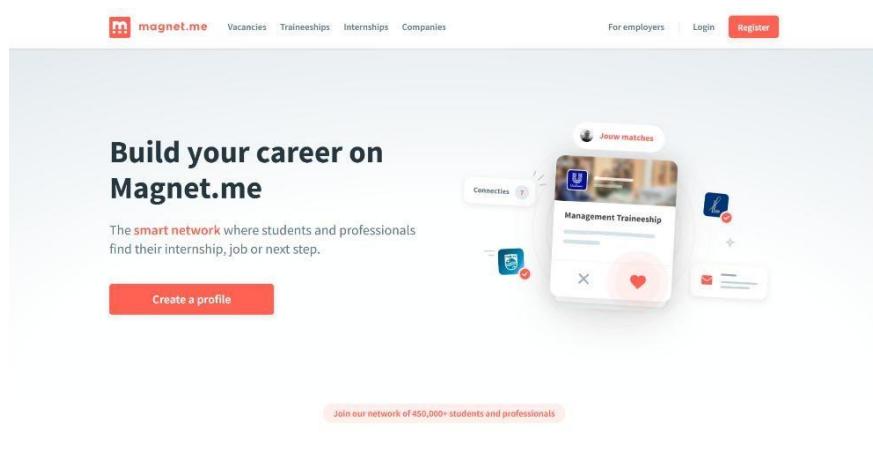


Figure 105: Similar Project 2 Magnet.me

### 9.1.3. Similar Project 4: Torre.ai

Torre.ai is a startup platform that uses artificial intelligence to discover, rank, and match candidates to job openings. Its AI algorithms scan hundreds of sources to evaluate candidate profiles, providing recruiters with a ranked list of potential hires based on skills, experience, and suitability for a role. The system also provides detailed candidate reports, helping recruiters make data-driven decisions and reducing time spent on manual screening. Torre.ai is particularly suitable for high-volume recruitment where automation and speed are critical (Torre.ai, 2025).

Despite its AI strength, Torre.ai lacks several key recruitment management features. It does not support structured portfolio creation for candidates, real-time application tracking, or admin-controlled assessments and quizzes. Communication between candidates and recruiters through the platform is limited, and dashboards for tracking performance metrics are missing. Compared to Torre.ai, the Smart Job and Skill Portfolio application offers a more integrated solution by combining AI job recommendations with full ATS functionality, candidate portfolio management, networking, and admin tools for monitoring and managing recruitment activities.

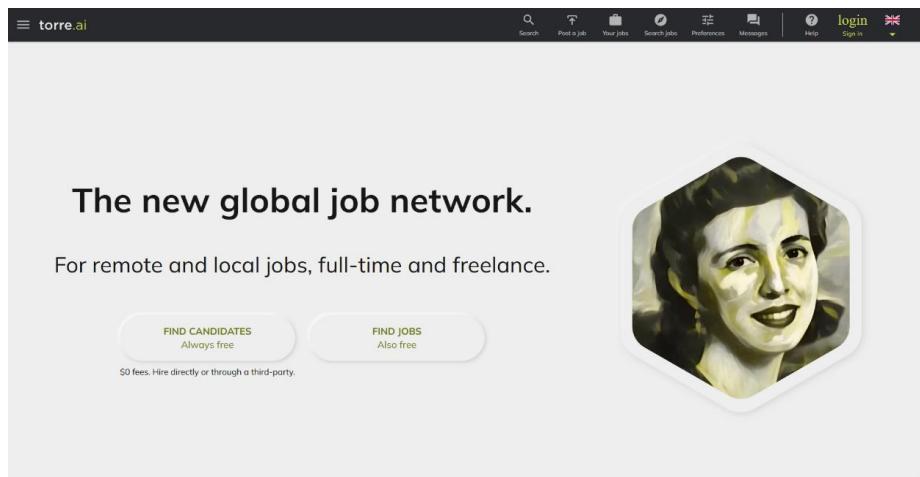


Figure 106: Similar Project Torre.ai

### 9.1.4. Similar Project 5: Merojob AI (Nepal)

Merojob is one of Nepal's most recognized online job portals with a large user base. Recently, it introduced Merojob AI, an AI-enabled application tracking system that assists with job posting, candidate shortlisting, job description assistance, and sorting applicants based on relevance to employer requirements. The platform serves as a central job marketplace where employers can post positions and job seekers can apply or search for opportunities (merojob, 2025).

However, Merojob's AI capabilities focus mainly on recruitment support for employers rather than personalized AI job recommendations for candidates or comprehensive skill assessment modules. It also lacks structured portfolio creation and admin-focused quiz/assessment management that are part of the Smart Job and Skill Portfolio system.

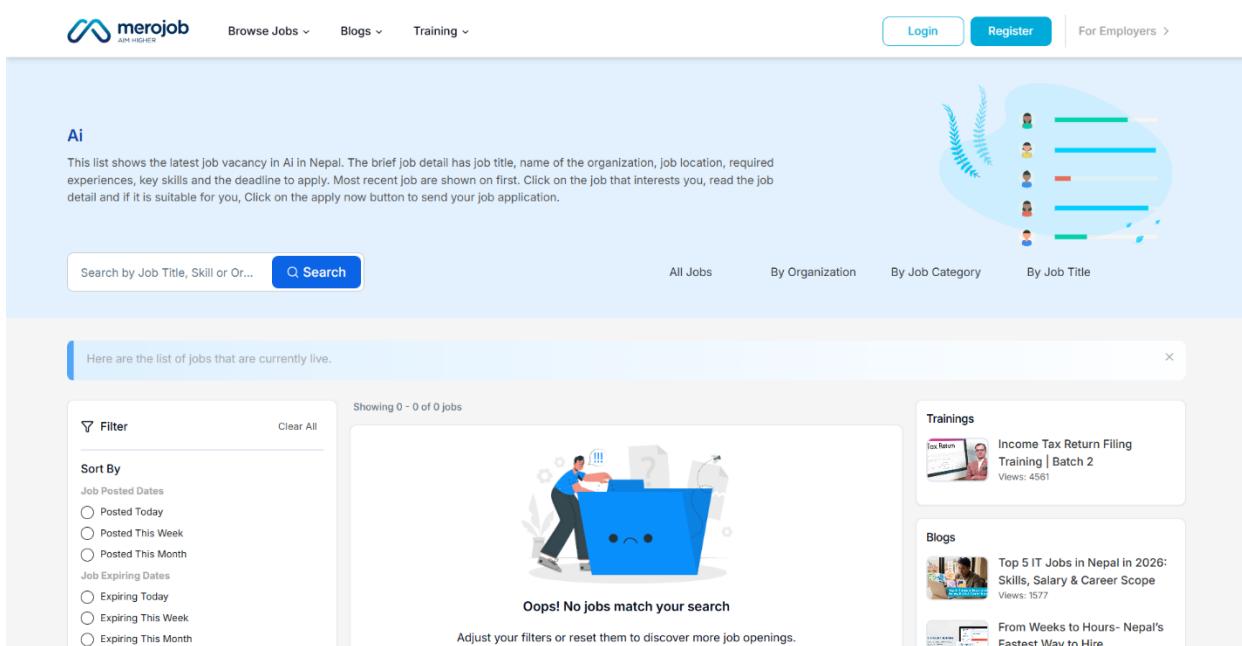


Figure 107: MerojobAI (merojob, 2025)

[Back to Comparison of Similar Projects](#)

## **9.2. Software Development Life Cycle (SDLC)**

The project follows the Software Development Life Cycle (SDLC), which provides a structured roadmap for system development through phases of requirement analysis, design, implementation, testing, and deployment (Sommerville, 2011). SDLC ensures systematic and organized development, minimizing errors, reducing rework, and allowing careful planning of each module.

This approach is particularly suitable for the Smart Job and Skill Portfolio platform because it involves multiple complex modules that must be developed step-by-step while maintaining proper integration, data consistency, and system reliability.

[Back to Considered Methodology](#)

### **9.3. Software Requirements Specification (SRS)**

#### **Introduction**

#### **Purpose**

This Software Requirements Specification (SRS) document describes the functional and non-functional requirements of the Smart Job and Skill Portfolio web application. The purpose of this document is to clearly define system behavior, user interactions, and constraints to guide design, development, and evaluation during the project lifecycle.

#### **Scope**

The Smart Job and Skill Portfolio platform is a web-based system designed to help job seekers create digital portfolios, manage skills, and receive AI-based job recommendations. Recruiters can post job vacancies and analyze candidate resumes using ATS features, while administrators manage users and platform content. The system aims to improve job matching efficiency and skill visibility.

#### **Definitions, Acronyms, and Abbreviations**

SRS: Software Requirements Specification

SDLC: Software Development Life Cycle

ATS: Applicant Tracking System

UI: User Interface

AI: Artificial Intelligence

#### **Overall Description**

#### **Product Perspective**

The system is an independent web application developed using modern web technologies. It follows a client–server architecture where users interact through a web interface and data is managed on a centralized database. The application supports multiple user roles including job seekers, recruiters, and administrators.

## Product Functions

- i. User registration and authentication
- ii. Portfolio and skill management
- iii. Resume upload and ATS analysis
- iv. AI-based job recommendations
- v. Job posting and application management
- vi. User connections and professional networking
- vii. Messaging and communication
- viii. Admin user and content management

## User Classes and Characteristics

**Job Seekers:** Create portfolios, connect with recruiters, send messages, upload resumes, and view job recommendations.

**Recruiters:** Post jobs, connect with candidates, exchange messages, and analyze resumes.

**Administrators:** Manage users, connections, messages, and system content.

## Operating Environment

- i. Web browser (Chrome, Firefox, Edge)

- ii. Server-side application environment
- iii. Database management system

## Constraints

- i. Limited development time
- ii. Individual project scope
- iii. Internet connectivity required

## Assumptions and Dependencies

- i. Users have basic internet access and web literacy
- ii. Messaging features depend on stable network connectivity
- iii. AI recommendations depend on quality of input data

## System Requirements

### Functional Requirements

#### FR1: User Authentication

The system shall allow users to register, log in, and log out securely.

#### FR2: Profile Management

The system shall allow users to create, view, and update profile information.

#### FR3: Portfolio Management

The system shall allow job seekers to add skills, education, experience, and projects.

**FR4: Resume Upload and ATS Scanning**

The system shall allow users to upload resumes and analyze them using ATS techniques.

**FR5: Job Recommendation**

The system shall provide job recommendations based on user skills and profile data.

**FR6: Job Posting**

The system shall allow recruiters to create, update, and delete job postings.

**FR7: User Connections**

The system shall allow users to send, accept, and manage professional connection requests.

**FR8: Messaging System**

The system shall allow connected users to send and receive messages in real time.

**FR9: Admin Management**

The system shall allow administrators to manage users, connections, messages, and platform content.

## **Non-Functional Requirements**

### **NFR1: Usability**

The system shall provide a simple and user-friendly interface.

### **NFR2: Performance**

The system shall respond to user requests within acceptable time limits.

### **NFR3: Security**

The system shall protect user data using authentication and access control mechanisms.

### **NFR4: Scalability**

The system shall support an increasing number of users without performance degradation.

### **NFR5: Reliability**

The system shall ensure data consistency and system availability.

## **External Interface Requirements**

### **User Interface**

The system shall provide a responsive web interface accessible from desktop and mobile devices.

### **Hardware Interface**

No specific hardware requirements beyond standard computing devices.

## Software Interface

The system shall interact with databases and external libraries for AI and ATS processing.

## Future Enhancements

- i. Mobile application support
- ii. Advanced AI matching algorithms
- iii. Interview scheduling and messaging features

## Conclusion

This SRS document defines the core requirements of the Smart Job and Skill Portfolio web application. It provides a clear foundation for system design and development and supports structured and iterative implementation using SDLC and Agile Scrum methodologies.

[Back to Use Case Diagram](#)

## 9.4. Expanded Description

### 9.4.1. Expanded Description of Access User to the System

*Table 2: Expanded Description of Access User to the System*

**Use Case:** Access User to the System

**Actors:** Candidate, Recruiter, Admin

**Description:** Users access the system by registering a new account, logging in with valid credentials, or recovering a forgotten password. The system verifies user details and provides access based on the user role.

**Typical Course of Events:**

Actor Action	System Response
1. The user selects the “Register” option.	2. The user selects the “Register” option.
3. The user enters required registration details such as name, email, password, and user role.	4. The system validates the entered details.
5. The user submits the registration form.	6. The system creates a new user account and shows a registration confirmation message.
7. The user selects the “Login” option.	8. The system displays the login form.
9. The user enters email or username and password.	10. The system validates the login credentials.
11. The user clicks the “Submit” button.	12. The system grants access and redirects the user to the respective dashboard.
13. The user selects the “Forgot Password” option.	14. The system displays the password recovery form.

15. The user enters the registered email address.	16. The system sends a password reset link to the user's email.
17. The user sets a new password using the reset link.	18. The system updates the password and confirms successful reset.

### Alternative Courses

- **Line 4:** If the registration details are invalid or incomplete, the system displays an error message and the use case ends.
- **Line 10:** If the login credentials are incorrect, the system displays an error message and the use case ends.
- **Line 16:** If the entered email address is not registered, the system displays an error message and the use case ends.

### 9.4.2. Expanded Description of Create or Update Portfolio

*Table 3: Expanded Description of Create or Update Portfolio*

**Use Case:** Create or Update Portfolio

**Actors:** Candidate

**Description:** The candidate creates a new portfolio or updates an existing portfolio by adding or modifying personal details, skills, projects, education, certifications, and work experience. The system validates the entered information and saves the portfolio for future viewing by recruiters.

**Typical Course of Events:**

Actor Action	System Response
1. The candidate selects the “Create Portfolio” or “Update Portfolio” option.	2. The system displays the portfolio form with existing details if available.
3. The candidate enters or edits personal information, skills, education, projects, and certifications.	4. The system validates the entered portfolio details.
5. The candidate uploads supporting files such as resume, certificates, or project documents.	6. The system checks file format and upload size.
7. The candidate clicks the “Save” or “Update” button.	8. The system stores the portfolio information in the database.
9. The candidate reviews the saved portfolio.	10. The system displays a confirmation message showing successful portfolio creation or update.

### Alternative Courses

- **Line 4:** If any required portfolio details are missing or invalid, the system displays an error message and the use case ends.
- **Line 6:** If uploaded files are in an unsupported format or exceed the allowed size, the system rejects the files and the use case ends.
- **Line 8:** If the system fails to save the portfolio due to a technical issue, an error message is displayed and the use case ends.

### 9.4.3. Expanded Description of Job Apply and Application Tracking

*Table 4: Expanded Description of Job Apply and Application Tracking*

**Use Case:** Job Apply and Application Tracking

**Actors:** Candidate

**Description:** Candidates search and apply for jobs through the system. They can track the status of their applications in real-time, including updates like applied, shortlisted, rejected, or hired.

**Typical Course of Events:**

Actor Action	System Response
1. The candidate logs in and selects the "Job Search" option.	2. The system displays available jobs with search and filter options (skills, location, job type).
3. The candidate enters search criteria or filters jobs.	4. The system validates input and retrieves matching job listings.
5. The candidate selects a job to view details.	6. The system displays full job description, requirements, company details, and application deadline.
7. The candidate clicks "Apply" for the selected job.	8. The system checks candidate eligibility and portfolio completeness.
9. The candidate confirms application submission.	10. The system stores the application in the database, updates status as "Applied," and sends a confirmation notification.
11. The candidate navigates to "My Applications" to track status.	12. The system retrieves application status from the database and displays updates like "Shortlisted," "Rejected," or "Hired."

### Alternative Courses

- **Line 4:** If no jobs match the search criteria, the system displays “No results found.” Use Case ends.
- **Line 6:** If the candidate does not meet eligibility requirements, the system shows an error message and prevents application. Use Case ends.
- **Line 8:** If the application submission fails due to a system error, the system displays an error. Use Case ends.

#### 9.4.4. Expanded Description of AI Job Recommendations

*Table 5: Expanded Description of AI Job Recommendations*

**Use Case:** AI Job Recommendations

**Actors:** Candidate

**Description:** The candidate receives AI-based job recommendations based on their profile, skills, portfolio, experience, and past activities. The system analyses candidate data and matches it with available job postings to suggest suitable jobs.

**Typical Course of Events:**

Actor Action	System Response
1. The candidate selects the “AI Job Recommendations” option.	2. The system retrieves the candidate profile, portfolio, and skill data.

3. The candidate requests job recommendations.	4. The system analyses candidate data using AI matching logic.
5. The candidate waits for the results.	6. The system compares candidate data with available job postings.
7. The candidate views the recommended job list.	8. The system stores the portfolio information in the database.
9. The candidate selects a job to view details.	10. The system displays full job details and application options.

### Alternative Courses

- **Line 4:** If the candidate profile or portfolio is incomplete, the system displays a message requesting the candidate to update their profile and the use case ends.
- **Line 6:** If no suitable job matches are found, the system displays a message indicating no recommendations are available and the use case ends.
- **Line 8:** If a system error occurs during AI processing, the system displays an error message and the use case ends.

#### 9.4.5. Expanded Description of Connection and Messaging

*Table 6: Expanded Description of Connection and Messaging*

**Use Case:** Connection and Messaging

**Actors:** Candidate, Recruiter

**Description:** Candidates and recruiters can connect with each other and communicate through the system. Users can send connection requests, accept or reject requests, and exchange messages once a connection is established. This feature supports professional networking and direct communication.

**Typical Course of Events:**

Actor Action	System Response
1. The user selects the “Connections” option.	2. The system displays a list of suggested users and existing connections.
3. The user sends a connection request to another user.	4. The system sends the connection request notification to the selected user.
5. The receiving user views the connection request.	6. The system displays options to accept or reject the request.
7. The receiving user accepts the connection request.	8. The system confirms the connection and updates both users’ connection lists.
9. The user selects a connected user to send a message.	10. The system opens the messaging interface.
11. The user types and sends a message.	12. The system delivers the message and displays it in the conversation thread.
13. The receiving user reads the message.	14. The system marks the message as read.

**Alternative Courses**

- **Line 4:** If the connection request cannot be sent due to system error, the system displays an error message and the use case ends.
- **Line 7:** If the receiving user rejects the connection request, the system notifies the sender and the use case ends.
- **Line 12:** If the message fails to send, the system displays a failure notification and the use case ends.

**9.4.6. Expanded Description of Job Posting and Manage Application (ATS Resume Scanner)***Table 7: Expanded Description of Job Posting and Manage Application (ATS Resume Scanner)***Use Case:** Job Posting and Manage Application**Actors:** Recruiter**Description:** Recruiters create and manage job postings and receive applications from candidates. The system uses an ATS resume scanner to automatically analyse, score, and rank submitted resumes based on job requirements. Recruiters can review applications, track status, and make decisions efficiently.**Typical Course of Events:**

Actor Action	System Response
1. The recruiter selects the “Post Job” option.	2. The system displays the job posting form.
3. The recruiter enters job details such as title, description, skills, experience, and deadline.	4. The system validates the entered job information.
5. The recruiter submits the job posting.	6. The system publishes the job and makes it visible to candidates.
7. A candidate applies for the job by submitting a resume.	8. The system stores the application and triggers the ATS resume scanner.
9. The ATS scanner analyses the resume against job requirements.	10. The system generates a resume score and ranking.
11. The recruiter views the list of applications for the job.	12. The system displays candidates with resume scores and application status.
13. The recruiter reviews an application and updates its status (shortlisted, rejected, or pending).	14. The system saves the updated status and notifies the candidate.

### Alternative Courses

- **Line 4:** If the job details are incomplete or invalid, the system prompts the recruiter to correct the information and the use case ends.
- **Line 8:** If the resume upload fails, the system displays an error message and the use case ends.
- **Line 10:** If the ATS scanner cannot process the resume, the system marks the application for manual review, and the use case continues without automated scoring.
- **Line 14:** If the status update fails, the system shows an error message and the use case ends.

#### 9.4.7. Expanded Description of Search Candidate and View Profile

*Table 8: Expanded Description of Search Candidate and View Profile*

**Use Case:** Search Candidate and View Profile

**Actors:** Recruiter, Admin

**Description:** Recruiters or Admins search for candidates based on skills, experience, or other filters. They can view candidate profiles, including portfolio, certifications, and other details, to evaluate suitability for job openings or networking purposes.

**Typical Course of Events:**

Actor Action	System Response
1. The recruiter/admin selects the “Search Candidate” option.	2. The system displays the candidate search form with filter options (skills, experience, location, education).
3. The recruiter/admin enters search criteria.	4. The system validates the input and applies the search filters.
5. The recruiter/admin clicks “Search”.	6. The system retrieves matching candidate profiles from the database.
7. The recruiter/admin selects a candidate from the search results.	8. The system displays the candidate profile, including portfolio, projects, skills, certifications, and resume.
9. The recruiter/admin may send a connection request or message.	10. The system records the request/message and notifies the candidate.
11. The recruiter/admin may shortlist or bookmark the candidate for future reference.	12. The system updates the candidate’s status or adds to the recruiter/admin list.

### Alternative Courses

- **Line 4:** If no candidates match the search criteria, the system displays a “No results found” message. Use Case ends.
- **Line 8:** If the candidate profile is incomplete or private, the system shows limited information. Use Case continues.
- **Line 10:** If the connection or message cannot be sent due to system error, the system shows an error message and the use case ends.

### 9.4.8. Expanded Description of Manage Users and Quizzes / Assessments

*Table 9: Expanded Description of Manage Users and Quizzes / Assessments*

#### Use Case: Manage Users and Quizzes / Assessments

##### Actors: Admin

**Description:** The admin manages all system users (Candidates, Recruiters) by deactivating accounts if necessary and manages quizzes or assessments. This includes assigning quizzes to candidates, reviewing results, and monitoring system activity.

##### Typical Course of Events:

Actor Action	System Response
1. Admin logs in and selects “Manage Users” option.	2. The system displays a list of all registered users with options to deactivate accounts.
3. Admin selects a user to deactivate.	4. The system confirms the selection and shows a message to confirm deactivation.
5. Admin confirms deactivation.	6. The system updates the user status to “deactivated” in the database and shows a confirmation message.
7. Admin selects “Manage Quizzes / Assessments” option.	8. The system displays a list of existing quizzes with options to create, update, assign, or delete.
9. Admin clicks “Create Quiz” and enters quiz title, questions, and options.	10. The system validates quiz details (e.g., question format, required fields).
11. Admin assigns quiz to selected candidates.	12. The system stores assignment data in the database and sends notifications to candidates.

13. Admin reviews completed quiz results.	14. The system retrieves results, calculates scores, and displays performance reports.
---	--

### Alternative Courses

- **Line 6:** If the selected user cannot be deactivated due to system restrictions, the system shows an error message. Use Case ends.
- **Line 10:** If quiz details are incomplete or invalid, the system requests corrections before saving. Use Case ends.
- **Line 12:** If no candidates are selected for assignment, the system shows an error message. Use Case ends.

[Back to Sequence Diagram](#)

## 9.5. Wireframes

### 9.5.1. Wireframe of Contact Us Page

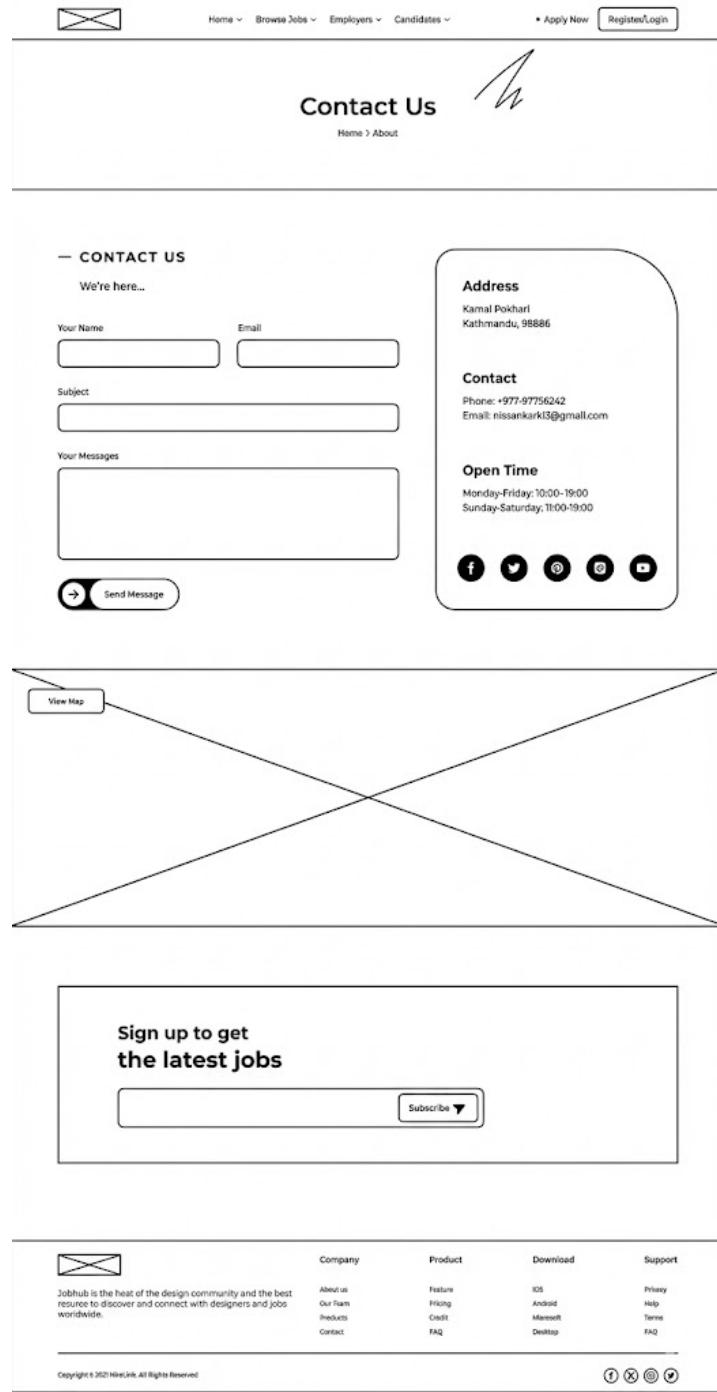


Figure 108: Wireframe of Contact Us Page

### 9.5.2. Wireframe of About Us Page

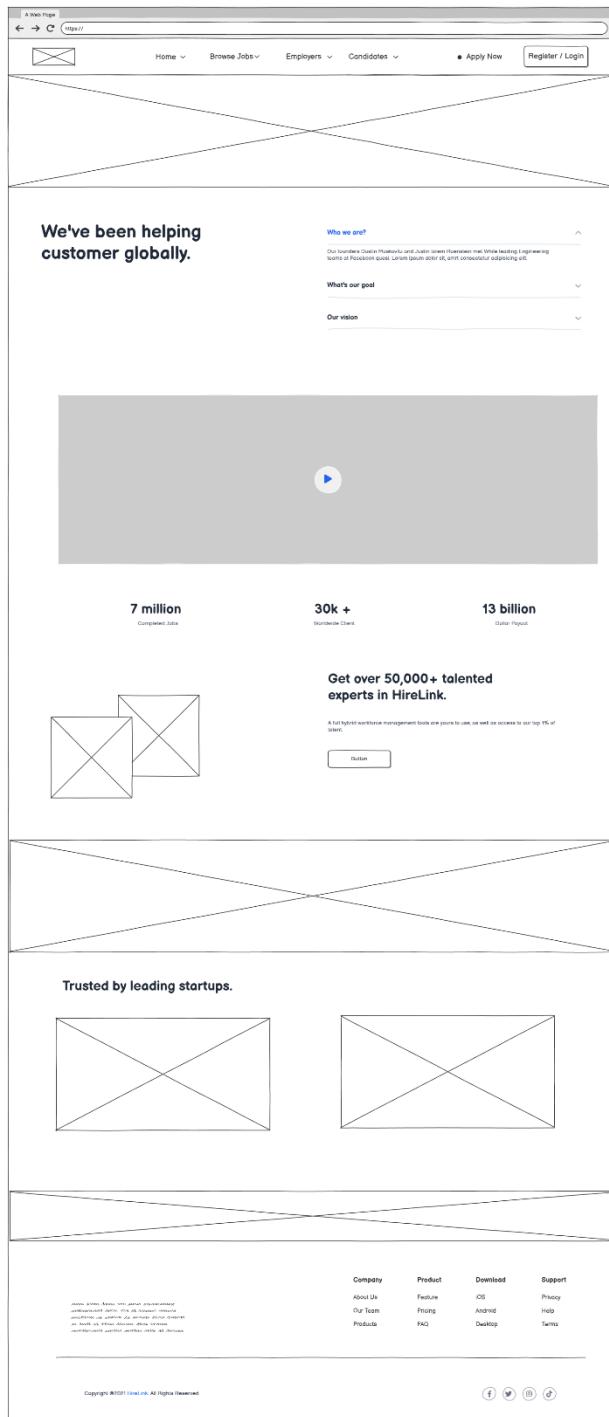


Figure 109: Wireframe of About Us Page

### 9.5.3. Wireframe of Candidate Dashboard Page

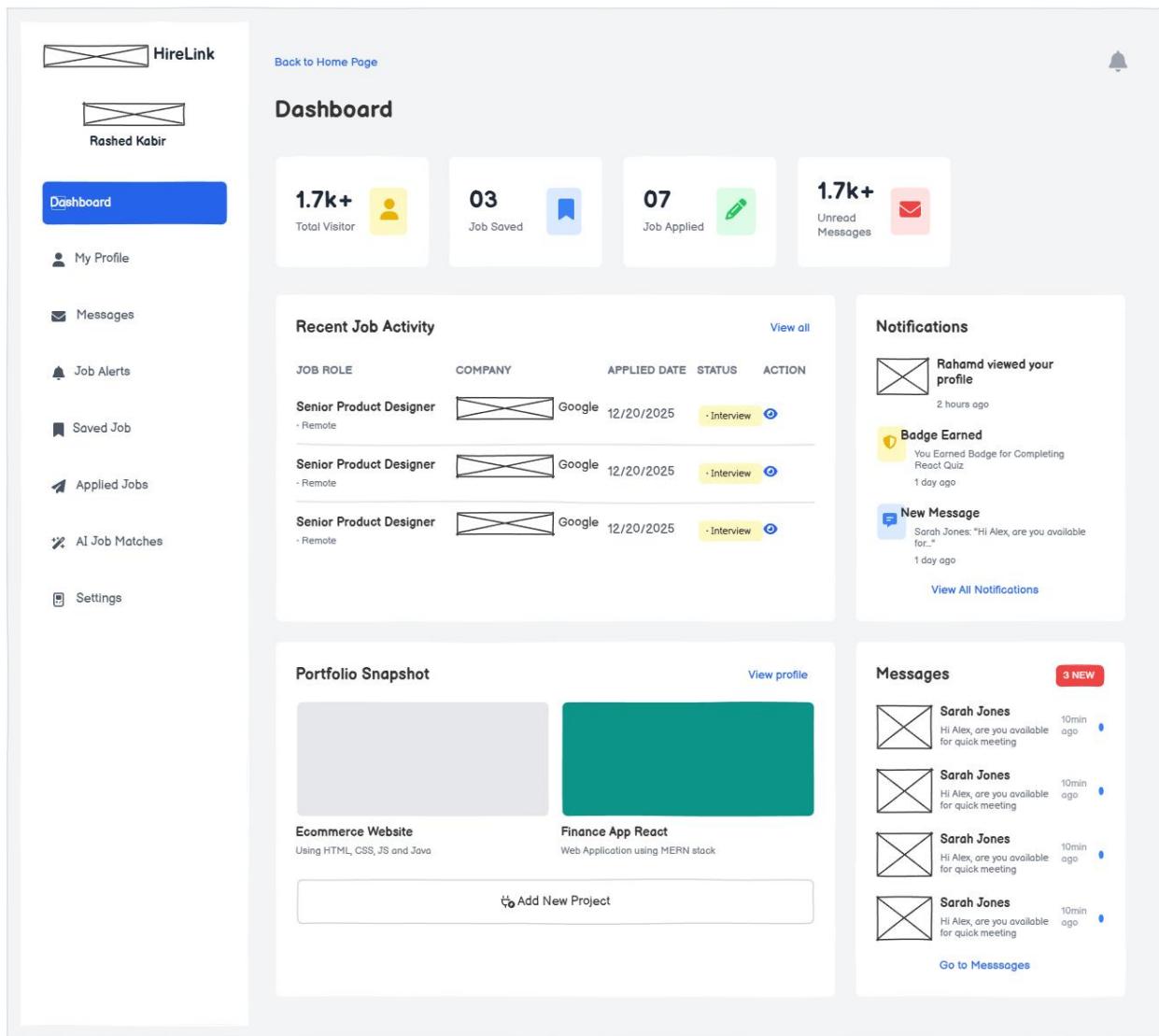


Figure 110: Wireframe of Contact Candidate Dashboard Page

### 9.5.4 Wireframe of Portfolio Page and Profile Page

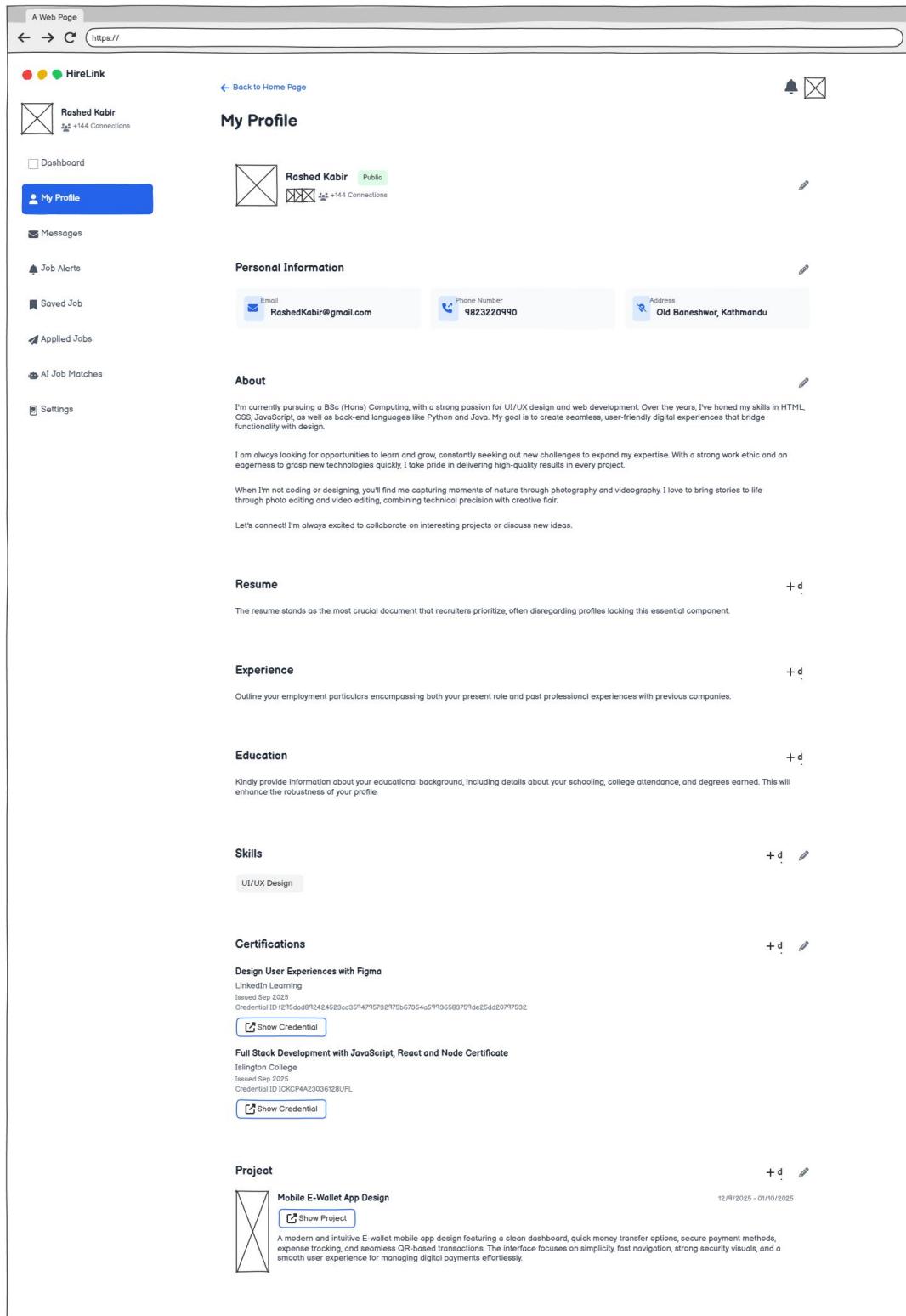


Figure 111: Wireframe of Portfolio Page and Profile Page

### 9.5.5. Wireframe of Message Page

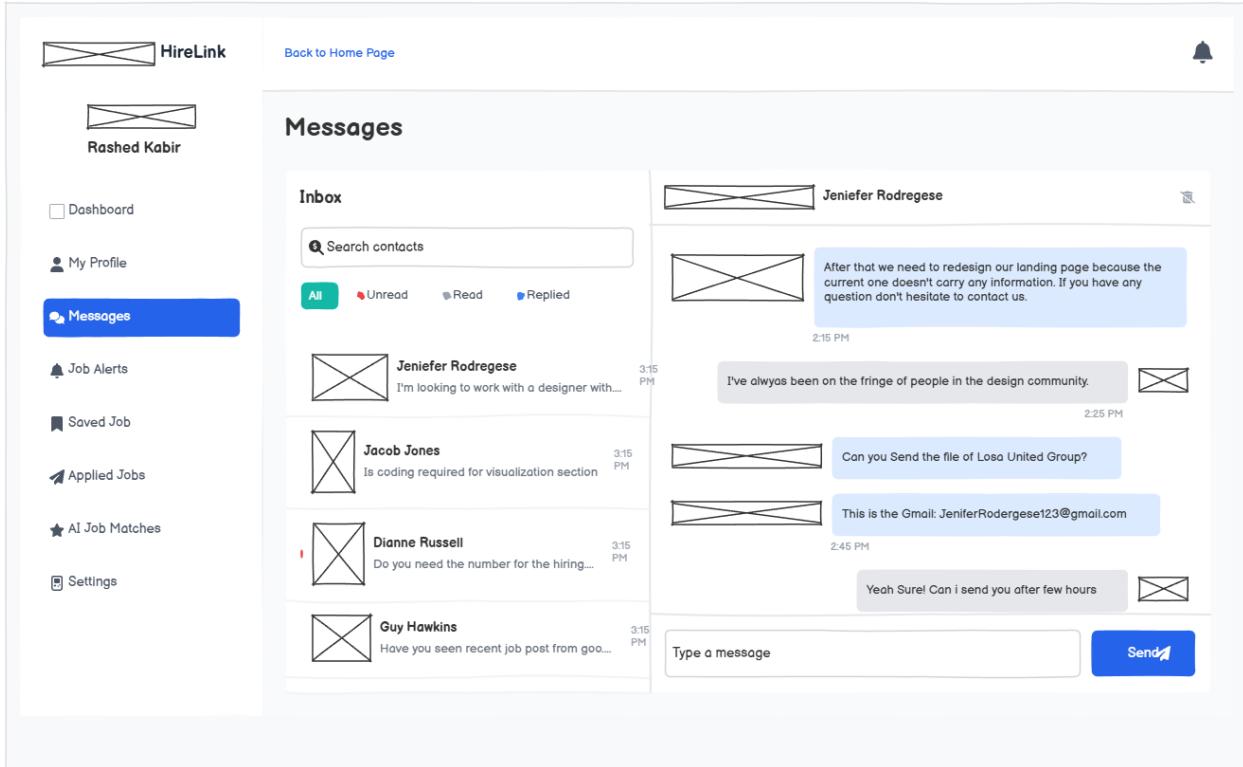


Figure 112: Wireframe of Message Page

### 9.5.6. Wireframe of Skills Assessment and Quizzes Page

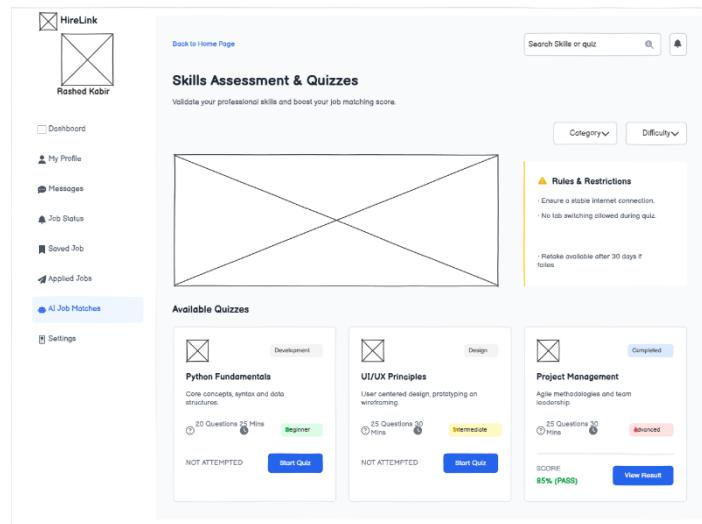


Figure 113: Wireframe of Skills Assessment and Quizzes Page

### 9.5.7. Wireframe of Saved Jobs Page

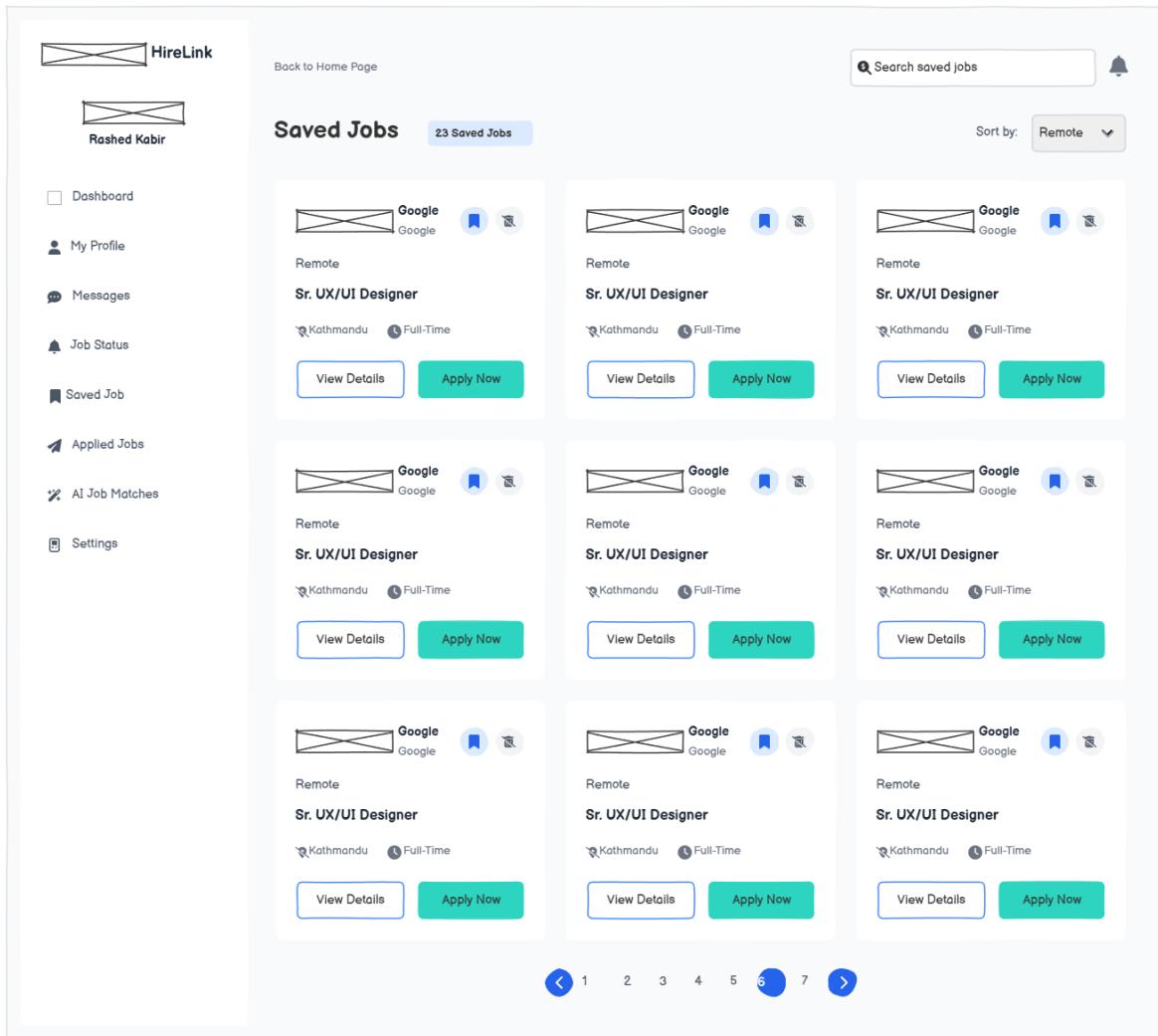


Figure 114: Wireframe of Saved Jobs Page

### 9.5.8. Wireframe of Job Listing Page

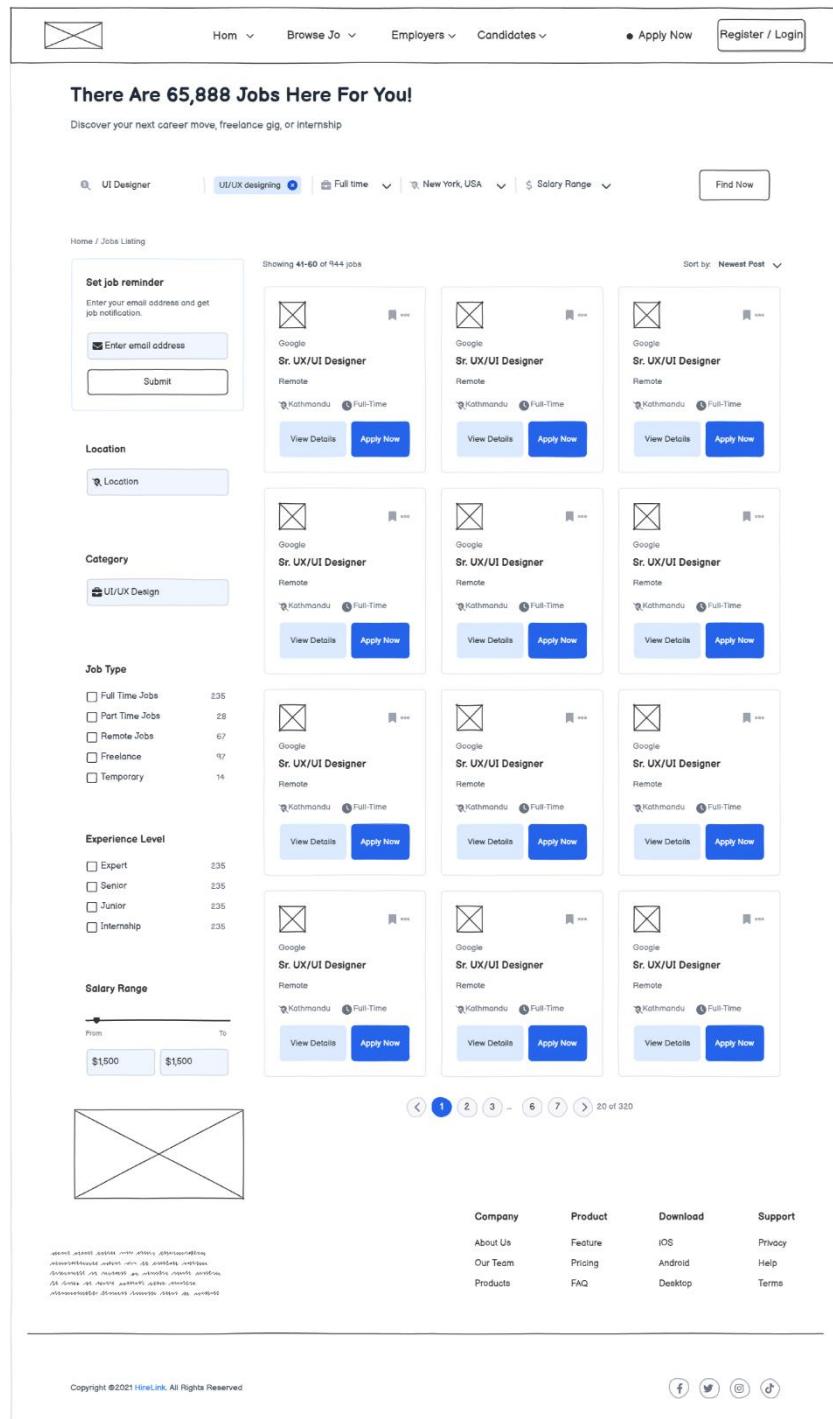


Figure 115: Wireframe of Job Listing Page

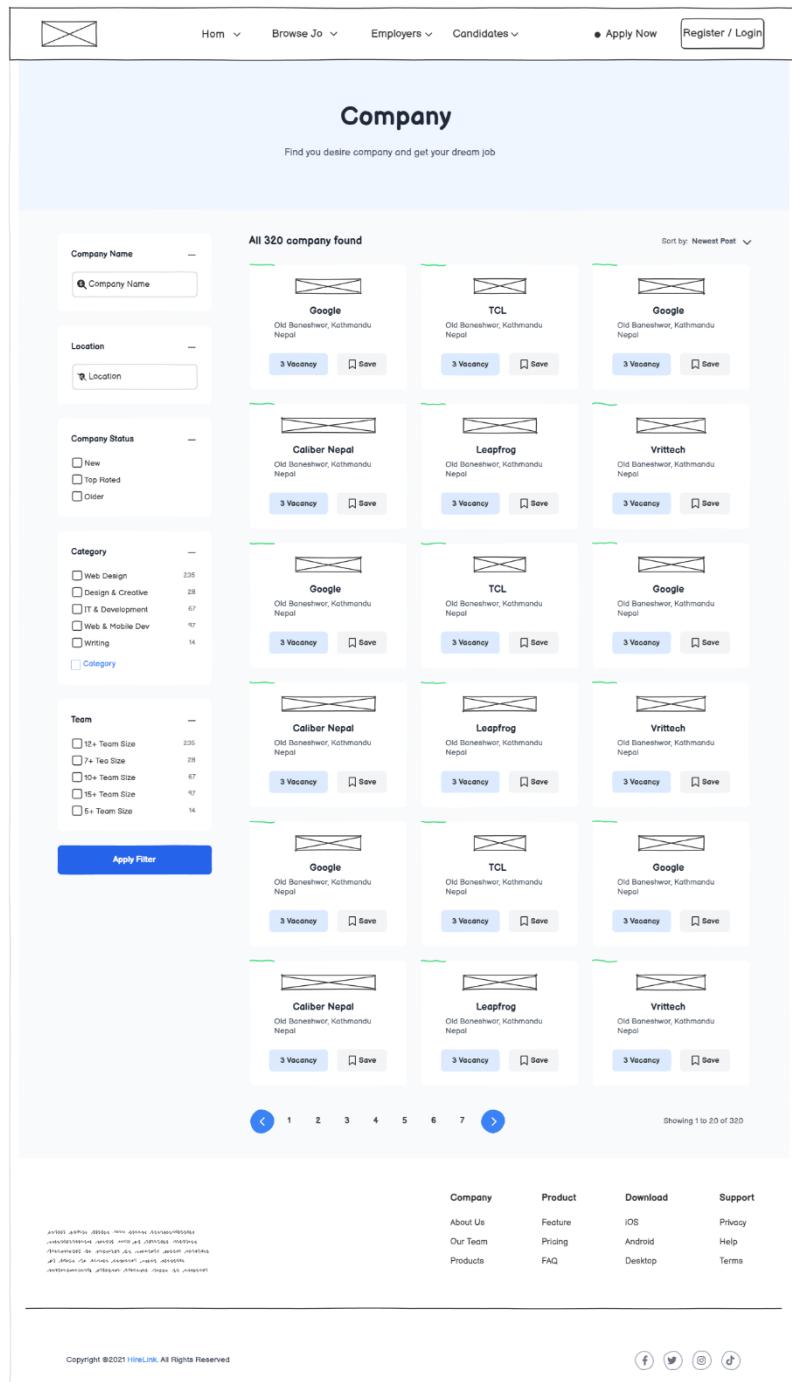
### 9.5.9. Wireframe of Individual Job Details Page

The wireframe for the Individual Job Details Page is organized into several sections:

- Header:** Includes a logo icon, navigation links for Home, Browse Jobs, Employers, Candidates, and a search bar with a magnifying glass icon. Buttons for "Apply Now" and "Register / Login" are also present.
- Title and Breadcrumbs:** The job title "Senior UI / UX Designer" is displayed prominently, along with a breadcrumb trail: Home / Jobs Description.
- Job Overview:** A brief description of the role, mentioning responsibilities like unit testing, TDD, CI, design patterns, and refactoring. It also highlights internal and external training, developer growth, and maintaining existing systems.
- About Company:** A section detailing the company's mission, values, and culture. It includes a short paragraph and a larger block of text describing the company's focus on quality, user-centered design, and continuous improvement.
- Requirements:** A list of qualifications and experience levels, such as a Bachelor's degree in Design, Human-Computer Interaction, Computer Science, or a related field, and 5+ years of professional experience in UI/UX design.
- Responsibilities:** A detailed list of tasks, including leading the end-to-end design process, conducting user research, and staying updated with design trends and tools.
- Overview (Right Panel):** A sidebar containing detailed job specifications:
  - Employee Type:** Full-Time / On-Site
  - Location:** Old Baneshwar, Kathmandu
  - Email:** Google123@gmail.com
  - Salary:** Negotiable
  - Job Category:** Designer
  - Experience:** 3+ Years
  - Qualification:** Bachelor in IT or related fields
  - Job Level:** Senior - Designer
  - Gender:** Male / Female (Both)
  - Date Posted:** April 22, 2023
  - Expiration Date:** April 24, 2023
- Share Openings:** Buttons for sharing the job listing on Facebook and LinkedIn.
- Skills:** A list of required skills: UX Design, UI Design, Prototyping, Wireframing, User Research, Usability Testing, Design Systems, and Visual Design.
- Footer:** Includes a "Company" section with links to About Us, Our Team, and Products; a "Product" section with links to Feature, Pricing, and FAQ; a "Download" section with links to iOS, Android, and Desktop; and a "Support" section with links to Privacy, Help, and Terms. It also features a copyright notice: "Copyright ©2021 HireLink. All Rights Reserved" and social media icons for Facebook, Twitter, LinkedIn, and YouTube.

Figure 116: Wireframe of Individual Job Details Page

### **9.5.10. Wireframe of Company Name Listing Page.**



*Figure 117: Wireframe of Company Name Listing Page.*

### 9.5.11. Wireframe of Company Description Page

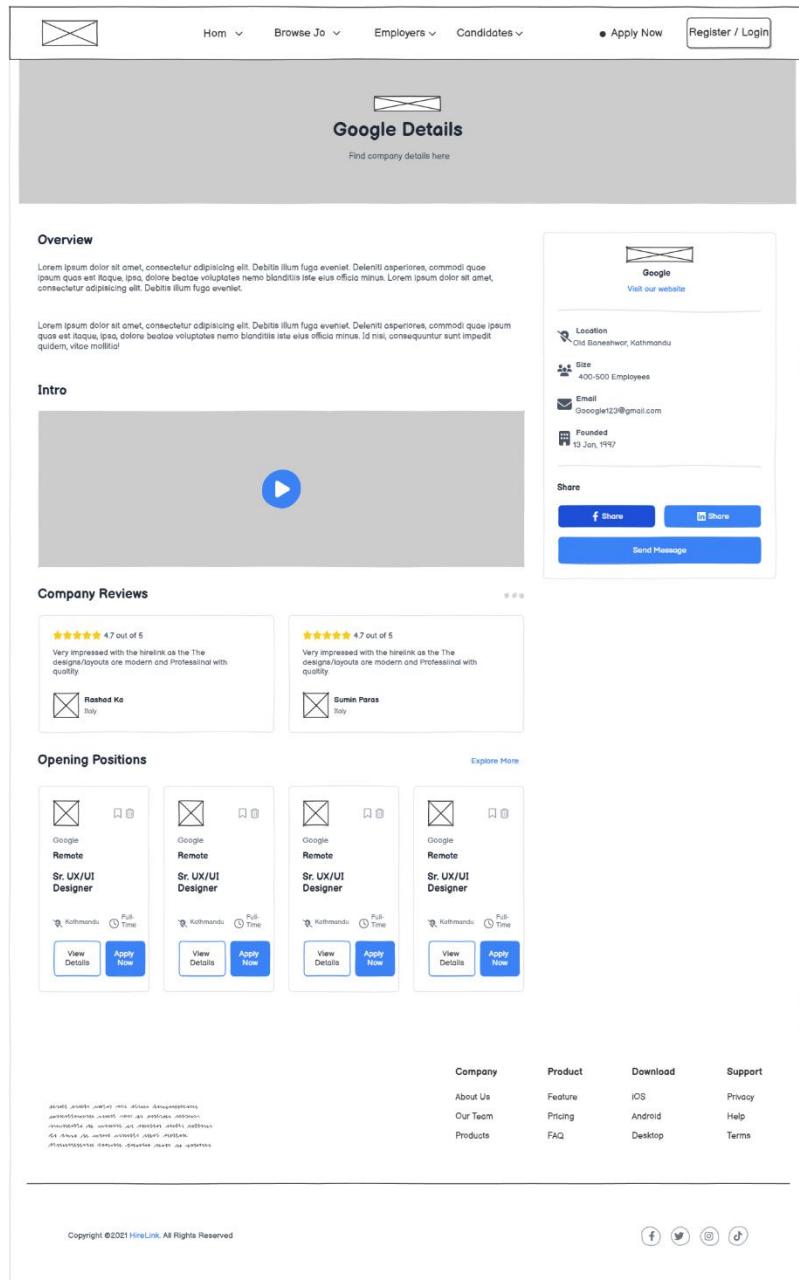


Figure 118: Wireframe of Company Description Page

### 9.5.12. Wireframe of Home page when user is not Authenticated.

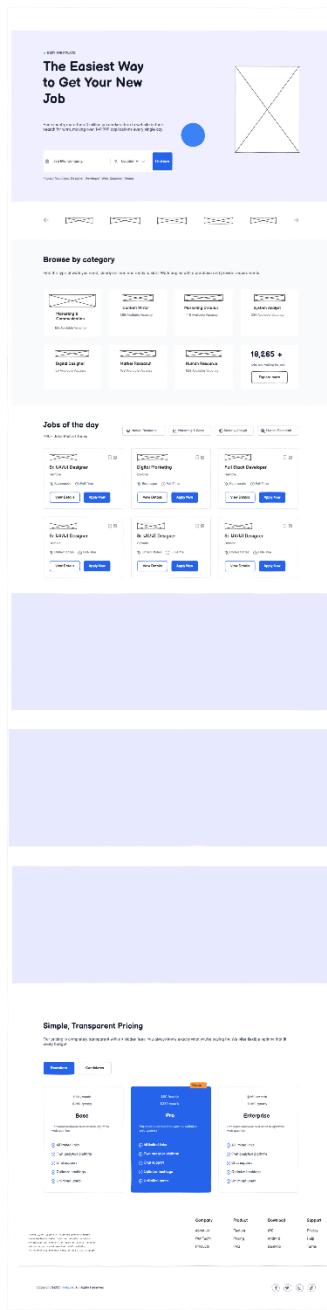


Figure 119: Wireframe of Home page when user is not Authenticated.

### 9.5.13. Wireframe of AI Job Recommendation Listing Page

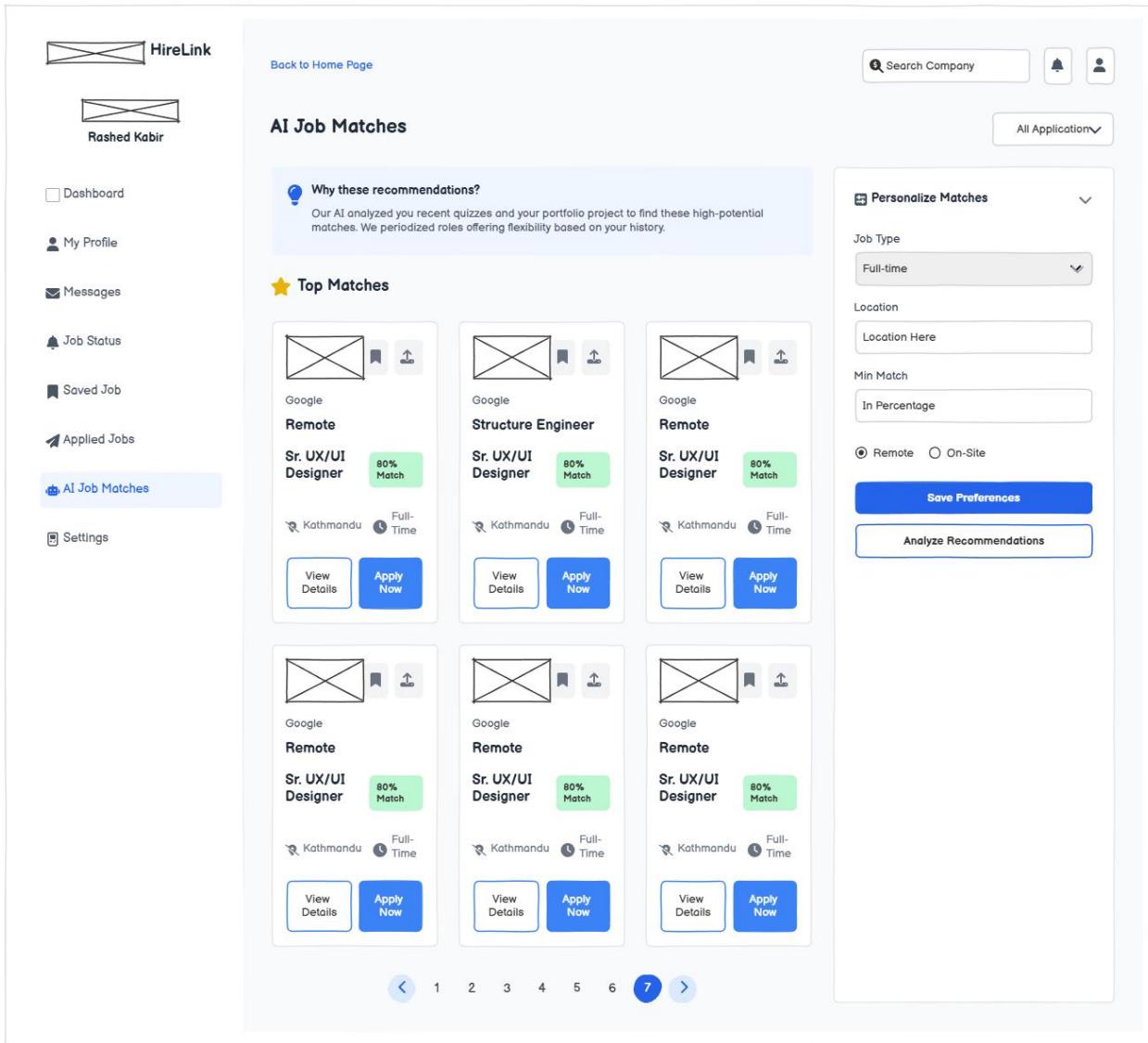


Figure 120: Wireframe of AI Job Recommendation Listing Page

### 9.5.14. Wireframe of Application Status Tracking Page

The wireframe for the Application Status Tracking Page is organized into several sections:

- Header:** Includes the logo "HireLink" with a user icon, a "Back to Home Page" link, a search bar with a magnifying glass icon and placeholder "Search Company", a notification bell icon with a red badge showing "3", and a dropdown menu labeled "All Application".
- User Profile:** Shows a profile picture of "Rashed Kabir" and basic user information.
- Navigation:** A sidebar with links to "Dashboard", "My Profile", "Messages", "Job Status", "Saved Job", "Applied Jobs" (which is highlighted in blue), and "AI Job Matches".
- Summary Metrics:** A row of five boxes showing statistics: "1.7k+" (Applied Jobs), "03" (Interviews), "1.7k+" (Job Offers), "07" (Job Applied), and "07" (Job Applied). Each box contains an icon: a briefcase for applied jobs, a microphone for interviews, a gift box for job offers, and two overlapping circles for job applied.
- Table:** A table listing applied job details. The columns are "Job Role & Company", "Date Applied", "Status", "Last Update", and "Actions". Each row includes a small company logo and a detailed description of the role and location.
- Footer:** A navigation bar with page numbers 1 through 7, where page 6 is highlighted in blue.

Job Role & Company	Date Applied	Status	Last Update	Actions
Sr. UX/UI Designer Google - Remote	Oct 24, 2023	● Interview	2 hours ago	💬 📲 ✎
Data Analyst Williams Racing - On Site	Oct 24, 2023	● Rejected	2 hours ago	💬 📲 ✎
Structure Engineer Mercedes-Benz - On Site	Oct 24, 2023	● Interview	2 hours ago	💬 📲 ✎
Social Media Handler Cadillac F1 - On Site	Oct 24, 2023	● Applied	2 hours ago	💬 📲 ✎
Sr. Account Manager Ferrari F1 Team - On Site	Oct 24, 2023	● Job Offer	2 hours ago	💬 📲 ✎
Account Handler Alpine F1 Team - On Site	Oct 24, 2023	● Interview	2 hours ago	💬 📲 ✎
Sr. Mechanical Eng RedBull Racing - On Site	Oct 24, 2023	● Rejected	2 hours ago	💬 📲 ✎
Radio Maintenance McLaren F1 Team - On Site	Oct 24, 2023	● Interview	2 hours ago	💬 📲 ✎

Figure 121: Wireframe of Application Status Tracking Page

[Back to UI Designs](#)

## 9.6. UI Designs

### 9.6.1. UI Design of Candidate Dashboard Page

The image displays the UI design of the Candidate Dashboard Page for the HireLink platform. The dashboard is organized into several sections:

- Header:** Includes the HireLink logo, a user profile picture of Rashed Kabir, and a "Back to Home Page" link.
- Top Navigation:** A blue button labeled "Dashboard" is highlighted, while other options like "My Profile", "Messages", "Job Alerts", "Saved Job", "Applied Jobs", "AI Job Matches", and "Settings" are shown in a list.
- Recent Job Activity:** A section showing statistics: 1.7k+ Total Visitor, 03 Job Saved, 07 Job Applied, and 1.7k+ Unread Messages.
- Notifications:** A list of notifications:
  - Rahamd viewed your profile (2 hours ago)
  - Badge Earned (You Earned Badge for Completing React Quiz 1 day ago)
  - New Message (Sarah Jones: "Hi alex are you available for..." 1 day ago)
 A "View All Notifications" button is at the bottom.
- Portfolio Snapshot:** Shows two projects: "Ecommerce Website" (Using HTML, CSS, JS and Java) and "Finance App React" (Web Application using MERN stack). A "View profile" link is also present.
- Messages:** A list of messages from Sarah Jones (4 new messages):
  - Hi Alex, are you available for quick meeting (10min ago)
  - Hi Alex, are you available for quick meeting (10min ago)
  - Hi Alex, are you available for quick meeting (10min ago)
  - Hi Alex, are you available for quick meeting (10min ago)
 A "Go to Messages" button is at the bottom.

Figure 122: UI Design of Candidate Dashboard Page

### 9.6.2. UI Design of Portfolio and Profile Page

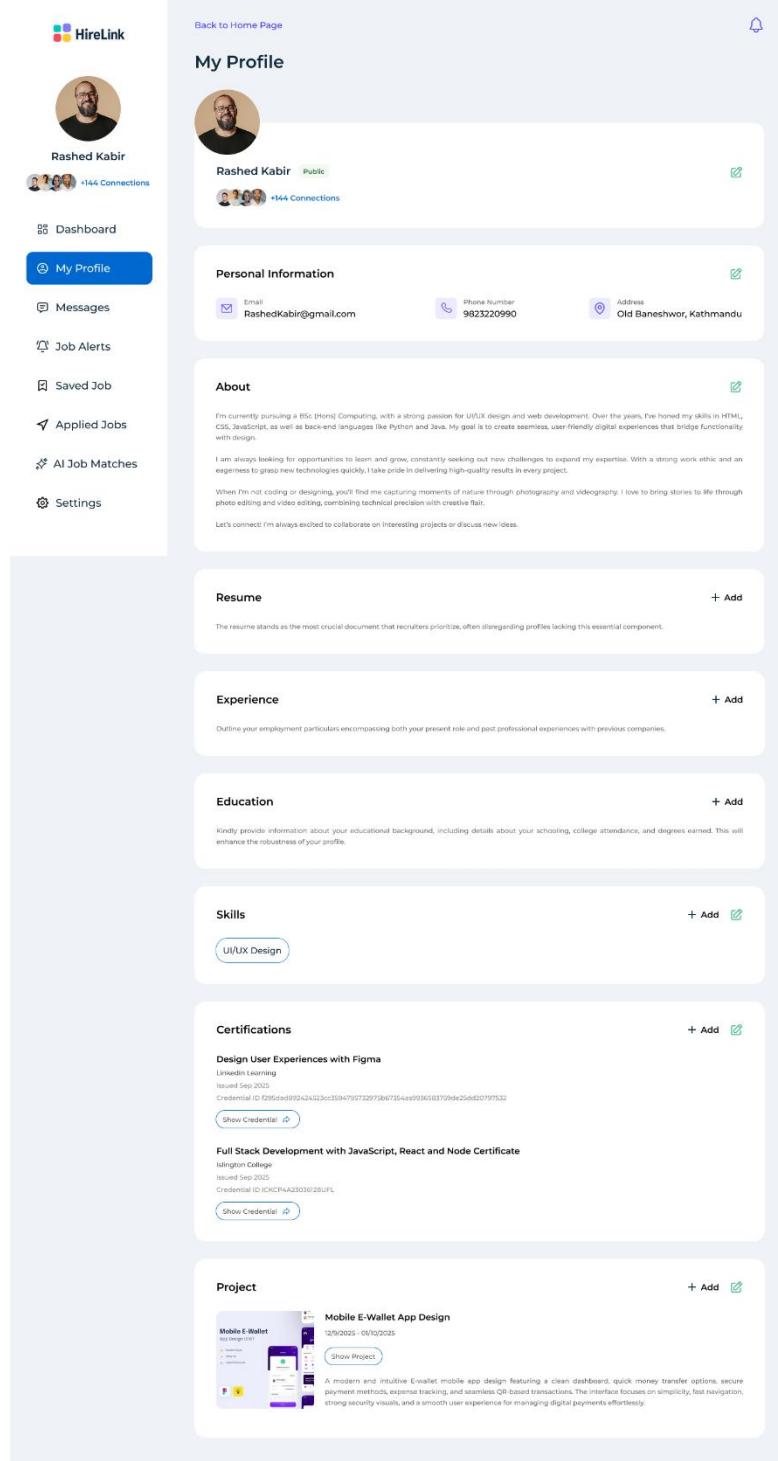


Figure 123: UI Design of Portfolio and Profile Page

### 9.6.3. UI Design of Message Page

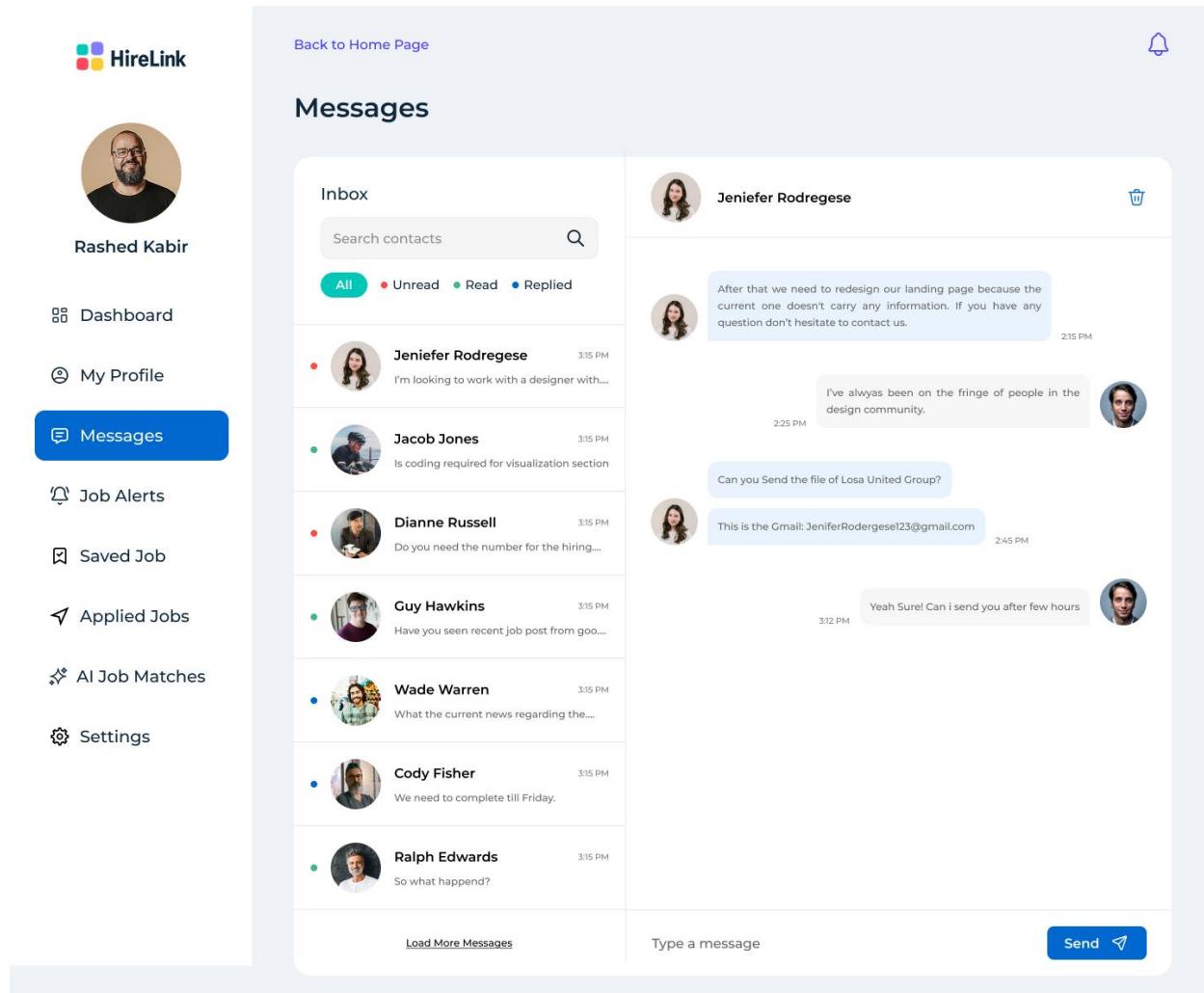


Figure 124: UI Design of Message Page

### 9.6.4. UI Design of Application Status Tracking Page

The screenshot displays the 'Applied Jobs' section of the HireLink application. At the top, there are four summary cards: '1.7k+' Applied Jobs (blue), '03 Interviews' (yellow), '1.7k+' Job Offers (green), and '07 Job Applied' (red). Below this is a table listing eight job applications with details like company, role, date applied, status, last update, and actions. The table has columns: Job Role & Company, Date Applied, Status, Last Update, and Actions. The 'Actions' column contains icons for message, view, and delete. The bottom of the page shows a navigation bar with page numbers 1 through 7.

Job Role & Company	Date Applied	Status	Last Update	Actions
Google Sr. UX/UI Designer Google • Remote	Oct 24, 2023	Interview	2 hours ago	
WILLIAMS RACING Data Analyst Williams Racing • On Site	Oct 24, 2023	Rejected	2 hours ago	
Mercedes-Benz Structure Engineer Mercedes-Benz • On Site	Oct 24, 2023	Interview	2 hours ago	
Cadillac F1 Social Media Handler Cadillac F1 • On Site	Oct 24, 2023	Applied	2 hours ago	
Ferrari F1 Team Sr. Account Manager Ferrari F1 Team • On Site	Oct 24, 2023	Job Offer	2 hours ago	
Alpine F1 Team Account Handler Alpine F1 Team • On Site	Oct 24, 2023	Interview	2 hours ago	
RedBull Racing Sr. Mechanical Eng RedBull Racing • On Site	Oct 24, 2023	Rejected	2 hours ago	
Mclaren F1 Team Radio Maintenance Mclaren F1 Team • On Site	Oct 24, 2023	Interview	2 hours ago	

Figure 125: UI Design of Application Status Tracking Page

### 9.6.5. UI Design of Saved Job Page

The screenshot displays the 'Saved Jobs' page of the HireLink application. The sidebar on the left includes a profile picture of Rashed Kabir, a 'Dashboard' link, 'My Profile', 'Messages', 'Job Status', a highlighted 'Saved Job' button, 'Applied Jobs', 'AI Job Matches', and 'Settings'. The main content area shows 23 saved job listings for 'Sr. UX/UI Designer' at Google, all listed as 'Remote' positions in Kathmandu, categorized as Full-Time. Each listing includes a 'View Details' button and an 'Apply Now' button.

Job Title	Employer	Location	Type
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time
Sr. UX/UI Designer	Google	Kathmandu	Full-Time

Figure 126: UI Design of Saved Job Page

### 9.6.6. UI Design of Quizzes and Assignments Page

**HireLink**

Rashed Kabir

Dashboard

My Profile

Messages

Job Status

Saved Job

Applied Jobs

AI Job Matches

Settings

Back to Home Page

Search Skills or quiz

Category ▾

Difficulty ▾

**Skills Assessment & Quizzes**

Validate your professional skills and boost your job matching score.

**Profile & AI Job Match Impact**

Completing skill assessments significantly improves your visibility to recruiters. Our AI also provides personalized recommendations based on your skill assessment and quiz results.

High chance of getting hired

**Rules & Restrictions**

- Ensure a stable internet connection.
- No tab switching allowed during quiz.
- Complete the quiz in the given time frame.
- Retake available after 30 days if failed.

**Available Quizzes**

Quiz Title	Category	Attempts	Time	Level	Status	
Python Fundamentals	Development	20 Questions	25 Mins	Beginner	NOT ATTEMPTED	<a href="#">Start Quiz</a>
UI/UX Principles	Design	25 Questions	30 Mins	Intermediate	NOT ATTEMPTED	<a href="#">Start Quiz</a>
Project Management	Completed	25 Questions	30 Mins	Advanced	SCORE 85% (PASS)	<a href="#">View Result</a>

Figure 127: UI Design of Quizzes and Assignments Page

### 9.6.7. UI Design of Job Listing Page

The screenshot displays the HireLink job listing page. At the top, a banner states "There Are 65,888 Jobs Here For you!" with a search bar for "UI Designer" and filters for "Full time", "New York, USA", and "Salary Range". Below this is a sidebar with filtering options: "Set job reminder" (input field for email), "Location" (dropdown), "Category" (dropdown), "Job Type" (list: Full Time Jobs 255, Part Time Jobs 26, Remote Jobs 67, Freelance 97, Temporary 14), "Experience Level" (list: Expert 225, Senior 225, Junior 225, Internship 225), and "Salary Range" (range slider from \$1500 to \$15000). A "Recruiting?" section offers to post a job. The main content shows a grid of 20 job listings for "Sr. UX/UI Designer" at "Google" (Remote, Full-Time) in "Kathmandu", each with "View Details" and "Apply Now" buttons. The grid is labeled "Showing 41-60 of 944 jobs" and "Sort by: Newest Post". At the bottom, there are logos for Samsung, Google, Facebook, Pinterest, AVAYA, Avis, Nielsen, and Doordash, followed by a footer with links for Company (About us, Our Team, Products, Contact), Product (Feature, Driving, Credit, FAQ), Download (iOS, Android, Microsoft, Desktop), and Support (Privacy, Help, Terms, FAQ).

Figure 128: UI Design of Job Listing Page

### 9.6.8. UI Design of Individual Job Details Page

The screenshot shows a job details page for a "Senior UI / UX Designer" position at "Google". The top navigation bar includes links for Home, Browse Jobs, Employers, Candidates, Apply Now, and Sign In. The main content area features a large header image of a colorful abstract graphic. Below the header, the job title "Senior UI / UX Designer" is displayed, along with a "Home / Jobs Description" link, an "Apply Now" button, and a "Save Job" button.

**About Company:**

Google's development team focuses on unit testing, TDD, CI, design patterns and refactoring. Internal and external training is encouraged through mentoring, guided self-learning, conferences, user groups and training courses. We maintain and improve existing codebases, and create new systems, exposing developers to constant variety.

**Job Overview:**

Our team understands the performance implications of serving more than 25,000 page requests per-hour, crafting awesome user experiences. While we leverage existing tech, we also research new technologies to overcome technical and business challenges, to maintain our industry-leading status.

**Requirements:**

- ✓ Bachelor's degree in Design, Human-Computer Interaction, Computer Science, or a related field.
- ✓ 5+ years of professional experience in UI/UX design, product design, or interaction design.
- ✓ Proven experience designing for web and mobile applications.
- ✓ Strong portfolio showcasing modern, user-centered design work.
- ✓ Expertise in Figma, Adobe Creative Suite, Sketch, or similar tools.
- ✓ Solid understanding of UX principles, accessibility, responsive design, and usability heuristics.
- ✓ Experience with design systems, component libraries, and prototyping tools.
- ✓ Strong communication, presentation, and collaboration skills.

**Responsibilities:**

- ✓ Lead the end-to-end design process from research, wireframing, prototyping, and testing to final implementation.
- ✓ Conduct user research, usability testing, and competitive analysis to guide design decisions.
- ✓ Collaborate with developers to ensure design consistency and feasibility during implementation.
- ✓ Develop and maintain a design system and ensure brand consistency across all digital touchpoints.
- ✓ Present design concepts and rationale to stakeholders clearly and confidently.
- ✓ Create user flows, wireframes, mockups, and interactive prototypes using tools like Figma, Adobe XD, or Sketch.
- ✓ Stay updated with design trends, tools, and best practices in UX/UI and product design.

**Overview:**

- Employee type: Full-Time / On-Site
- Location: Old Baneshwor, Kathmandu
- Email: Google123@gmail.com
- Salary: Negotiable
- Job Category: Designer
- Experience: 3+ Years
- Qualification: Bachelor in IT or related fields
- Job Level: Senior - Designer
- Gender: Male / Female (Both)
- Date Posted: April 22, 2023
- Expiration Date: April 26, 2023

**Share Openings:**

Share buttons for LinkedIn and Facebook.

**Skills:**

- UI Design, UX Design, Prototyping
- Wireframing, User Research
- Usability Testing, Design Systems
- Visual Design

**Company:**

About Us, Our Team, Products, Contact.

**Product:**

Feature, Pricing, Credit, FAQ.

**Download:**

iOS, Android, Microsoft, macOS, Descrip.

**Support:**

Privacy, Help, Terms, FAQ.

Copyright ©2023 HireLink. All Rights Reserved.

Figure 129: UI Design of Individual Job Details Page

### 9.6.9. UI Design of AI Job Recommendation Page

The screenshot shows the 'AI Job Matches' section of the HireLink platform. At the top, there's a sidebar with a user profile picture of Rashed Kabir and a list of navigation items: Dashboard, My Profile, Messages, Job Status, Saved Job, Applied Jobs, AI Job Matches (which is currently selected), and Settings. The main content area has a header 'AI Job Matches' with a 'Back to Home Page' link and a search bar. A callout box explains why recommendations are made based on recent quizzes and portfolio projects. On the right, there's a 'Personalize Matches' sidebar with dropdowns for Job Type (Full-time), Location (Location Here), and Min Match (In Percentage), along with radio buttons for Remote and On-Site preferences, and buttons for 'Save Preferences' and 'Analyze Recommendations'. The main content displays seven job card snippets in a grid. Each card includes the company logo, name, role, location, and a green circular badge indicating an 80% match. Buttons for 'View Details' and 'Apply Now' are also present.

Figure 130: UI Design of AI Job Recommendation Page

## 9.6.10. UI Design of Company Listing Page

The screenshot displays the 'Company' listing page on the HireLink platform. At the top, there's a search bar with the placeholder 'Find your desire company and get your dream job'. Below the search bar, a message says 'All 320 company found' and 'Sort by: Newest Post'. On the left, there's a sidebar with filters for 'Company Name', 'Location', 'Company Status' (New, Top Rated, Older), 'Category' (Web Design, Design & Creative, IT & Development, Web & Mobile Dev, Writing), and 'Team' (12+ Team Size, 7+ Team Size, 10+ Team Size, 15+ Team Size, 5+ Team Size). A blue 'Apply Filter' button is at the bottom of the sidebar. The main area shows a grid of 320 company cards, each featuring the company logo, name, location, number of vacancies (3 Vacancy), and a 'Save' button. The companies listed include Google, TCL, Caliber Nepal, Leapfrog, Vrittech, and several instances of Google, TCL, and Caliber Nepal. At the bottom, there are navigation arrows and a page number indicator showing 'Showing 1 to 20 of 320'.

The screenshot shows the footer of the HireLink website. It includes the 'HireLink' logo, a copyright notice ('Jobhub is the heart of the design community and the best resource to discover and connect with designers and jobs worldwide.'), and links for 'Company', 'Product', 'Download', and 'Support'. Under 'Company', there are links for 'About us', 'Our Team', 'Products', and 'Contact'. Under 'Product', there are links for 'Feature', 'Pricing', 'Credit', 'FAQ', 'iOS', 'Android', 'Microsoft', 'Desktop', 'Privacy', 'Help', 'Terms', and 'FAQ'. At the bottom, there are social media icons for Facebook, Twitter, LinkedIn, and YouTube, along with a copyright notice ('Copyright ©2022 HireLink. All Rights Reserved').

Figure 131: UI Design of Company Listing Page

## 9.6.11. UI Design of Company Description Page

The screenshot displays the 'Google Details' page on the HireLink platform. At the top, there's a decorative header with the Google logo and the text 'Google Details'. Below this, a search bar says 'Find company details here'. The main content area is titled 'Overview' and contains two paragraphs of placeholder text (Lorem ipsum). Below the overview is a section titled 'Intro' featuring a video thumbnail of a person speaking at a podium. To the right, there's a sidebar with basic company information: Location (Old Baneshwor, Kathmandu), Size (400-500 Employees), Email (Google123@gmail.com), and Founded (13 Jan, 1997). Below the sidebar are 'Share' and 'Send Message' buttons. The 'Company Reviews' section shows two reviews with 4.7 out of 5 stars each, from users Rashaed Ka and Sumin Paras. The 'Opening Positions' section lists four job listings for Sr. UX/UI Designer at Google, all remote, based in Kathmandu, full-time, with 'View Details' and 'Apply Now' buttons. The footer includes the HireLink logo, a brief description of the platform, and links to Company, Product, Download, and Support sections.

Figure 132: UI Design of Company Description Page

### 9.6.12. UI Design of About Us Page

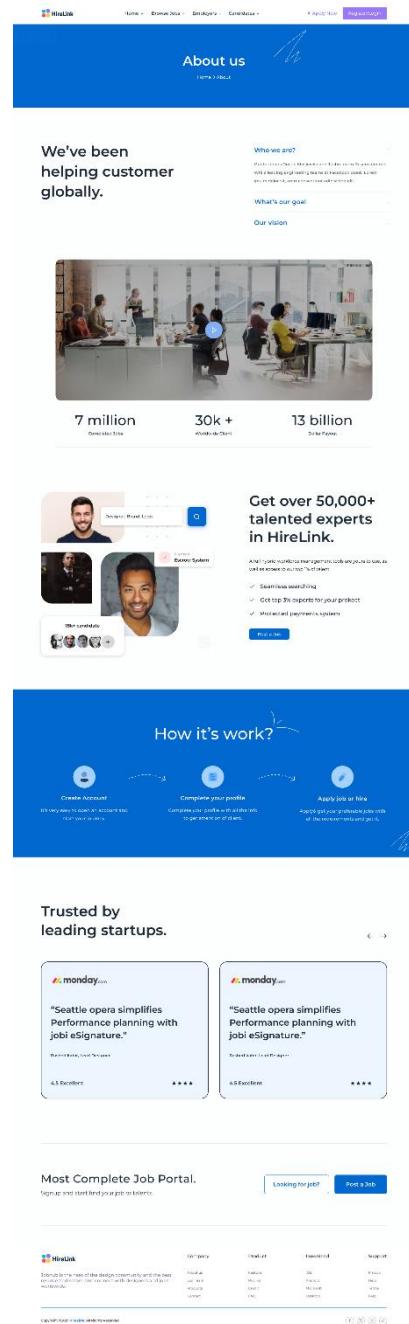


Figure 133: UI Design of About Us Page

### 9.6.13. UI Design of Home Page when user in not authenticated

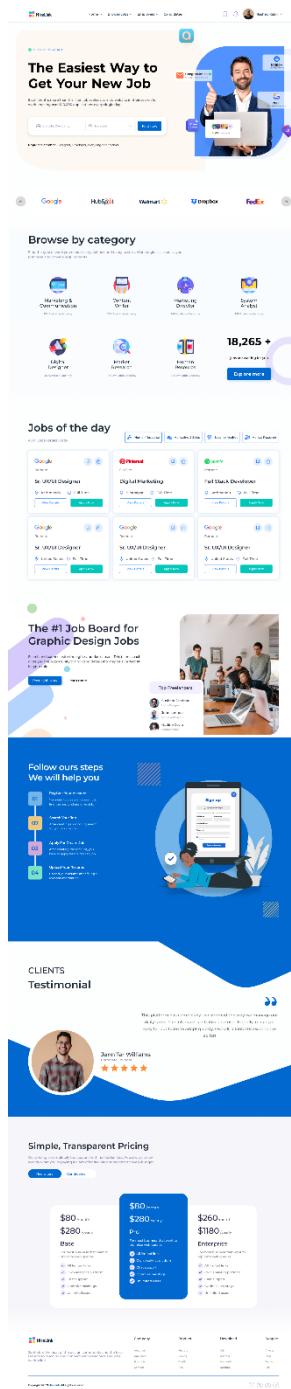


Figure 134: UI Design of Home Page when user in not authenticated

### 9.6.14. UI Design of Contact Us Page

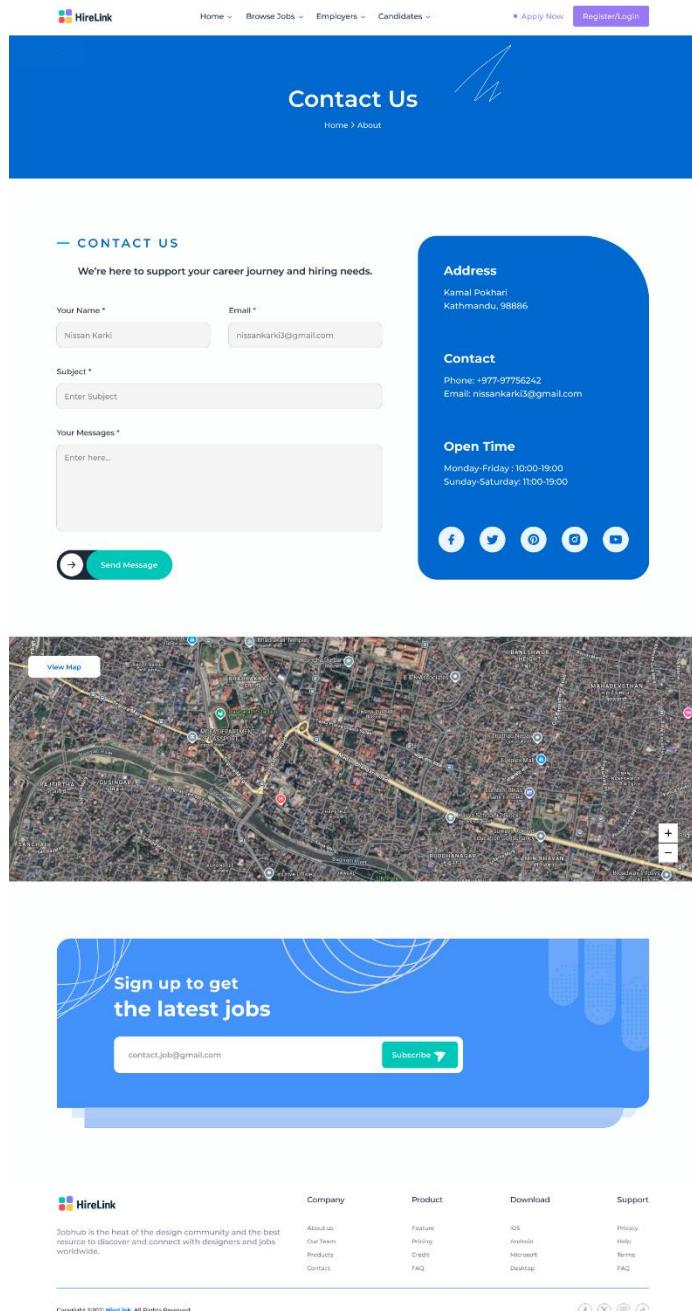


Figure 135: UI Design of Contact Us Page

[Back to Development Status](#)

## 9.7. Action Plan

The next step of the project is to complete the Sprint 2: Portfolio and Profile, which is currently in progress. This sprint includes designing and developing profile and portfolio features for Candidates, Recruiters, and Admin users. Extra time has been allocated until January 05, 2026, as this sprint requires handling multiple user roles and detailed profile information. Testing will be done alongside development to make sure the features work correctly.

After completing Sprint 2, development will continue with Sprint 3 and Sprint 4. These sprints will cover job search, job application, application tracking, AI-based job recommendations, recruiter job posting, candidate search, and ATS resume scanning. Each feature will be developed step by step and tested at the end of the sprint to reduce errors and improve system quality. Any issues found will be fixed before moving to the next sprint.

In the later stages, the project will move to Admin features, messaging, notifications, system testing and documentation. Regular progress reviews will be carried out to ensure the project stays on track. Final testing, documentation updates, and supervisors feedback will be completed before the submitting the final report by April 18, 2026. This approach will help ensure the system is complete, stable, and ready for final submission.

[Back to Future Work](#)

## 9.8. Survey Responses

### Response 1

What is your role?

150 responses

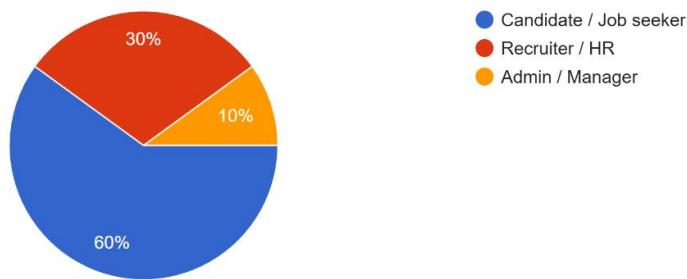


Figure 136: Survey Response 1

### Insights

The majority of users are candidates/job seekers 60%, so the platform should focus on features that meet their needs. Recruiters/HR 30% also form a significant user group, highlighting the importance of tools that connect job seekers and employers. A smaller portion are Admins/Managers 10%, suggesting the platform should include some administrative features, but these are not the main focus.

### Response 2

What is your age group?

150 responses

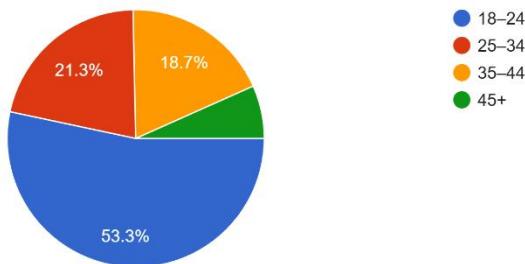


Figure 137: Survey Response 2

## Insights

Most users are young adults aged 18–24, making up over half of the respondents, so the platform mainly appeals to entry-level job seekers and recent graduates. Nearly 75% of users are aged 18–34, emphasizing the need to focus on early-career features. Users 35 and older are very few (under 25%), indicating the platform is less relevant for mid- to late-career professionals.

## Response 3

How often do you apply for jobs or review applications?

150 responses

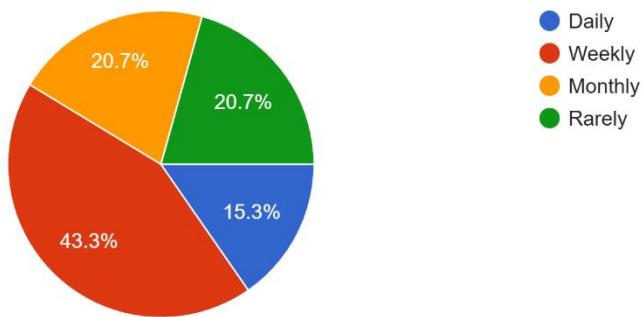


Figure 138: Survey Response 3

## Insights

Most users are highly active, with 64% applying for or reviewing jobs at least weekly (43.3% weekly and 20.7% daily), showing the platform is valuable for active job seekers. Around 36% engage less frequently, either monthly or rarely, indicating the platform should also cater to those in passive or early-stage job searches. Balancing features for both active and occasional users can help maximize overall engagement.

## Response 4

How do you currently showcase your skills and projects?

150 responses

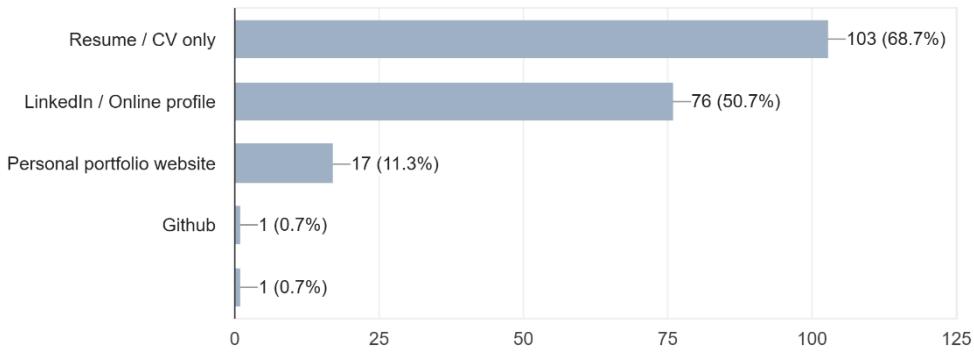


Figure 139: Survey Response 4

## Insights

Most users 68.7% rely only on a traditional Resume/CV, highlighting a need for a more dynamic, centralized portfolio platform. About half 50.7% also use LinkedIn, showing that while social profiles supplement their CV, users still lack a dedicated project showcase. Very few 12% use personal websites or GitHub, indicating low adoption of specialized tools and an opportunity for an all-in-one, user-friendly portfolio solution.

## Response 5

How useful would a platform be that allows you to create a detailed portfolio including projects, skills, and certificates? (Linear scale: 1 = Not useful, 5 = Very useful, required)

150 responses

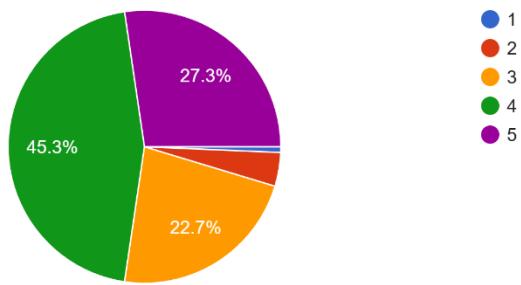


Figure 140: Survey Response 5

## Insights

Most users 72.6% find the platform useful, giving it a rating of 4 or 5, and 45.3% gave the highest rating. However, 27.3% of users feel neutral or see limited value, with ratings of 3 or below. This shows strong interest in the platform but also suggests that its benefits need to be explained more clearly to convince all users.

## Response 6

Would real-time job application tracking help you manage your applications better?  
150 responses

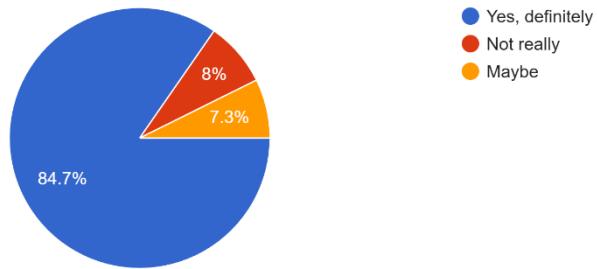


Figure 141: Survey Response 6

## Insights

Over 84% of users see strong value in real-time tracking, showing that it is a highly desired feature. Only 8% feel it would not be helpful, indicating that this feature addresses a clear problem for most job seekers.

## Response 7

How helpful would AI-based job recommendations be for you? (Linear scale: 1 = Not helpful, 5 = Extremely helpful, required)  
150 responses

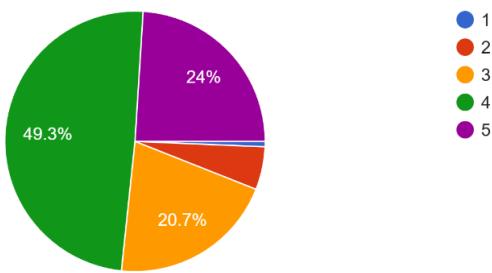


Figure 142: Survey Response 7

## Insights

AI job recommendations are very popular, with 49.3% of users finding them extremely helpful. Overall, nearly 75% of users see them positively, showing that personalized job matching is an important feature. About 20.7% are neutral, meaning the feature needs to provide accurate and relevant suggestions to convince all users. This shows AI recommendations are a key part of the platform's value.

## Response 8

How important is networking and messaging directly with recruiters on a platform?  
150 responses

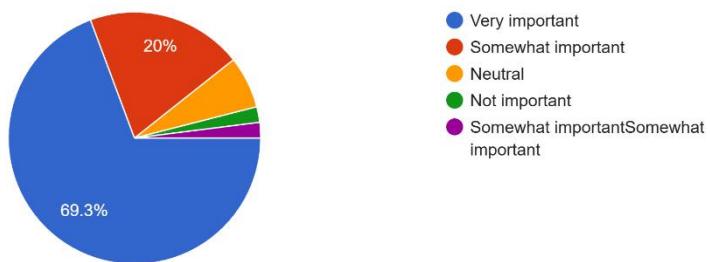


Figure 143: Survey Response 8

## Insights:

Direct networking is an important feature, with 89.3% of users considering it valuable—20% say it's very important and 69.3% somewhat important. This shows that enabling connections between users is a key part of the platform's appeal.

## Response 9

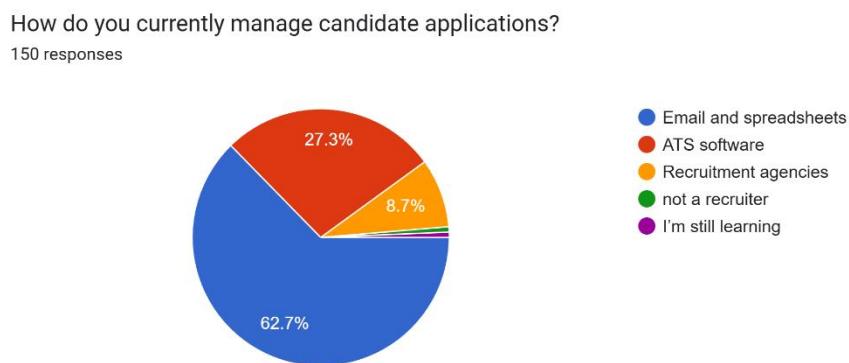


Figure 144: Survey Response 9

## Insights

Most recruiters 62.7% still use manual methods like email and spreadsheets, showing a clear need for a more efficient platform. Only 27.3% use ATS software, meaning professional tools are not yet widely adopted. Some recruiters 8.7% rely on agencies or are still learning, indicating the platform could also help simplify recruitment for beginners. Overall, there's a strong opportunity to provide an accessible, integrated solution.

## Response 10

How useful would a built-in ATS resume scanner be for ranking candidates automatically? (Linear scale: 1 = Not useful, 5 = Very useful, required)

150 responses

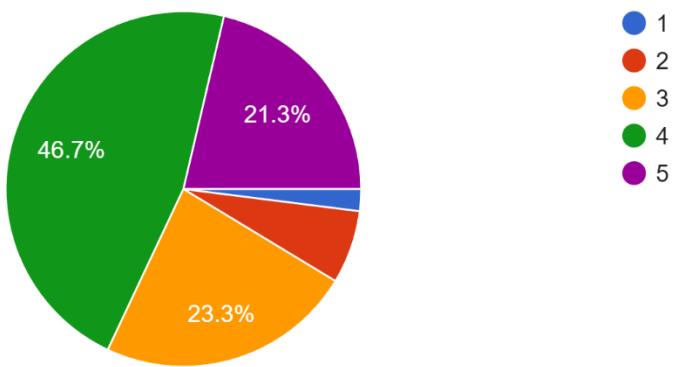


Figure 145: Survey Response 10

## Insights

A built-in ATS scanner is highly valued, with 46.7% of recruiters finding it very useful, showing strong demand for automation in candidate screening. Over 68% see it as useful, confirming that it helps improve efficiency in the recruitment process. About 23.3% are neutral, suggesting that adoption will depend on the tool's accuracy and ease of use. Overall, it is considered an important feature for streamlining recruitment tasks.

## Response 11

Would the ability to assign quizzes or skill assessments to candidates improve your hiring process?

150 responses

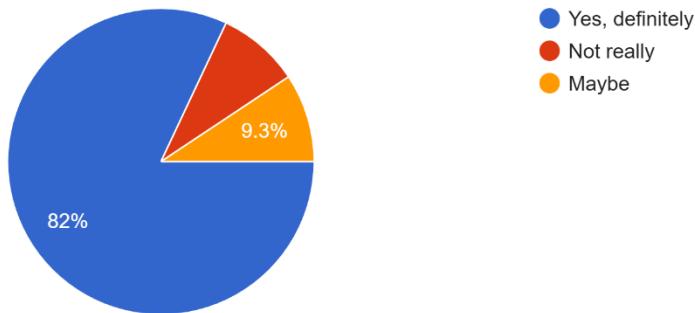


Figure 146: Survey Response 11

## Insights

Skill assessments are considered highly beneficial, with 82% of recruiters saying they are useful, showing strong demand for objective evaluation tools. Only a small group, less than 9%, sees little or no value in this feature. The overwhelmingly positive response indicates that skill assessments are an essential feature for improving efficiency and quality in hiring. Overall, this is a core expectation for a modern recruitment platform.

## Response 12

Which platform features do you consider most important? (Checkboxes, select up to 3, required)

150 responses

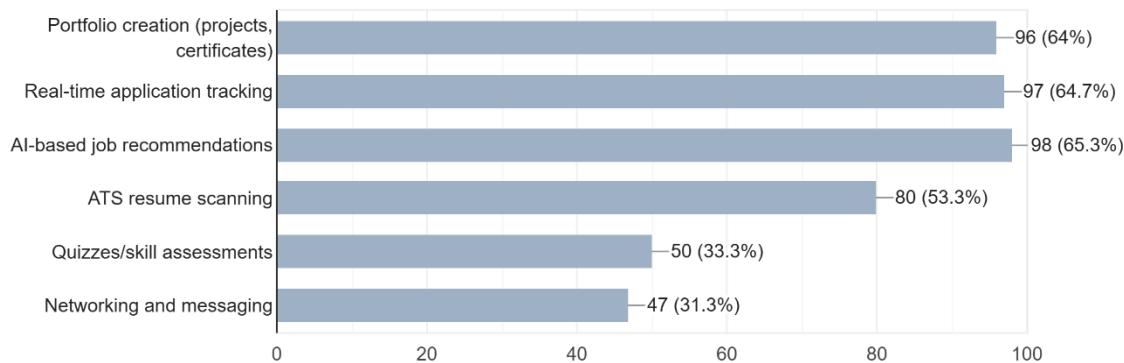


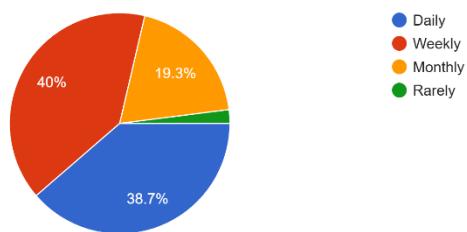
Figure 147: Survey Response 12

## Insights

The top three features, AI recommendations, real-time tracking, and portfolio creation, are each considered essential by about 65% of users, forming the platforms core “must-have” features. ATS scanning is also a high priority for 53.3% of users, making it a key efficient tool for recruiters. Quizzes and networking are secondary priorities, valued by around 30% of users, serving as useful differentiators but not main drivers.

## Response 13

How often would you use such a platform if it met your needs?  
150 responses



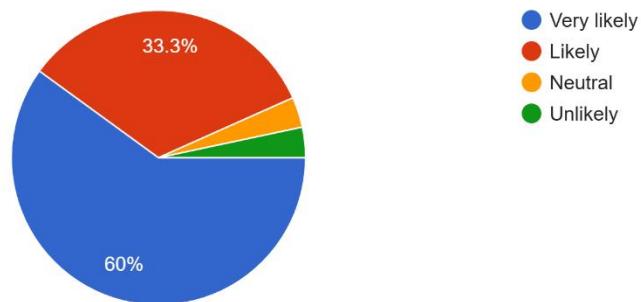
*Figure 148: Survey Response 13*

## Insights

User engagement is expected to be very frequent, with 78.7% of users planning to use the platform daily or weekly, indicating it will become a regular tool for job seeking and recruiting. Monthly users make up 19.3%, showing the platform also supports those with less frequent needs. Very few users rarely engage, confirming the platform meets a consistent, high demand need and is likely to encourage strong repeat usage.

## Response 14

Overall, how likely are you to recommend a Smart Job & Skill Portfolio platform to others?  
150 responses



*Figure 149: Survey Response 14*

There is strong advocacy potential, with 60% of users very likely to recommend the platform and an additional 33.3% likely to recommend it, total 93.3% promoters. This indicates high user satisfaction and a strong fit with the target audience, suggesting good potential for organic growth through word-of-mouth.

## 9.9. Previous Gantt Chart

### i. Initiation Phase

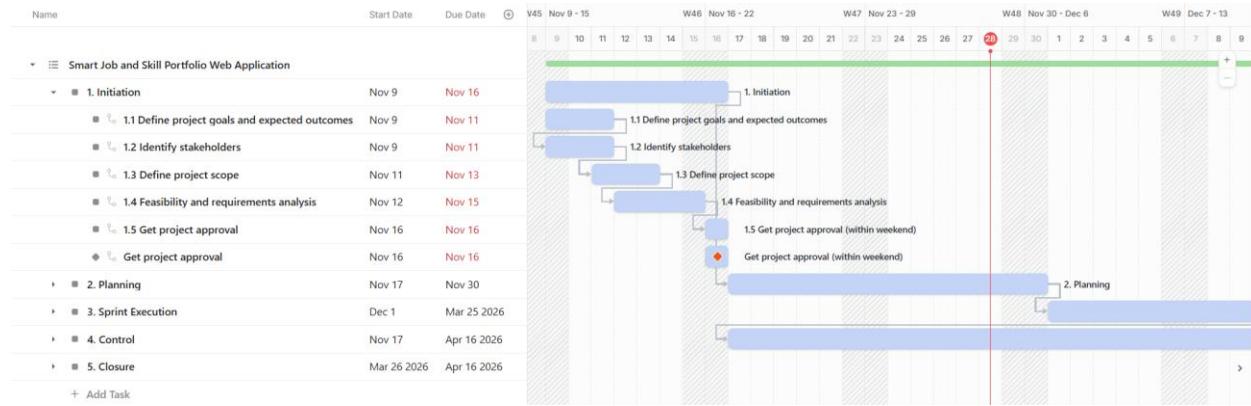


Figure 150: Previous Gantt Chart Initiation Phase

### ii. Planning Phase

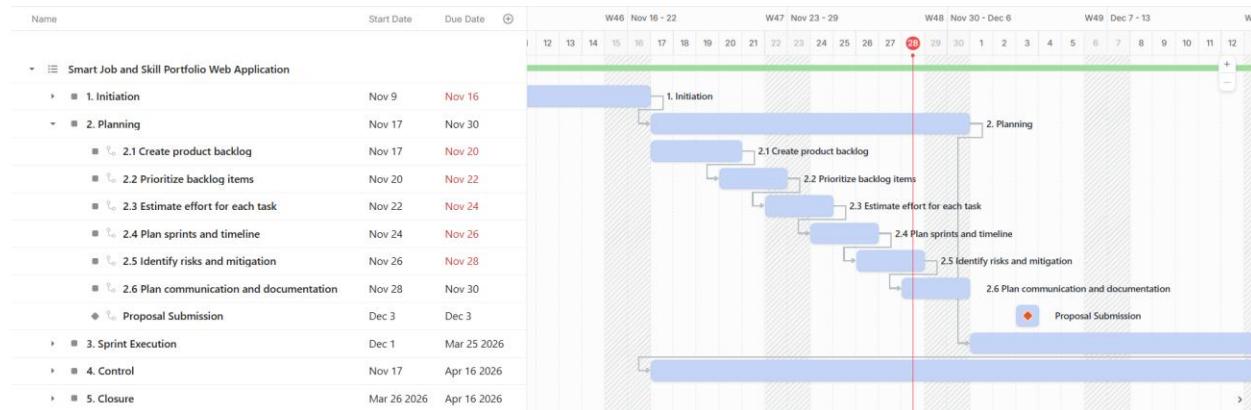


Figure 151: Previous Gantt Chart Planning Phase

### iii. Sprint Execution Phase (Sprint 1: Core Setup and Authentication)

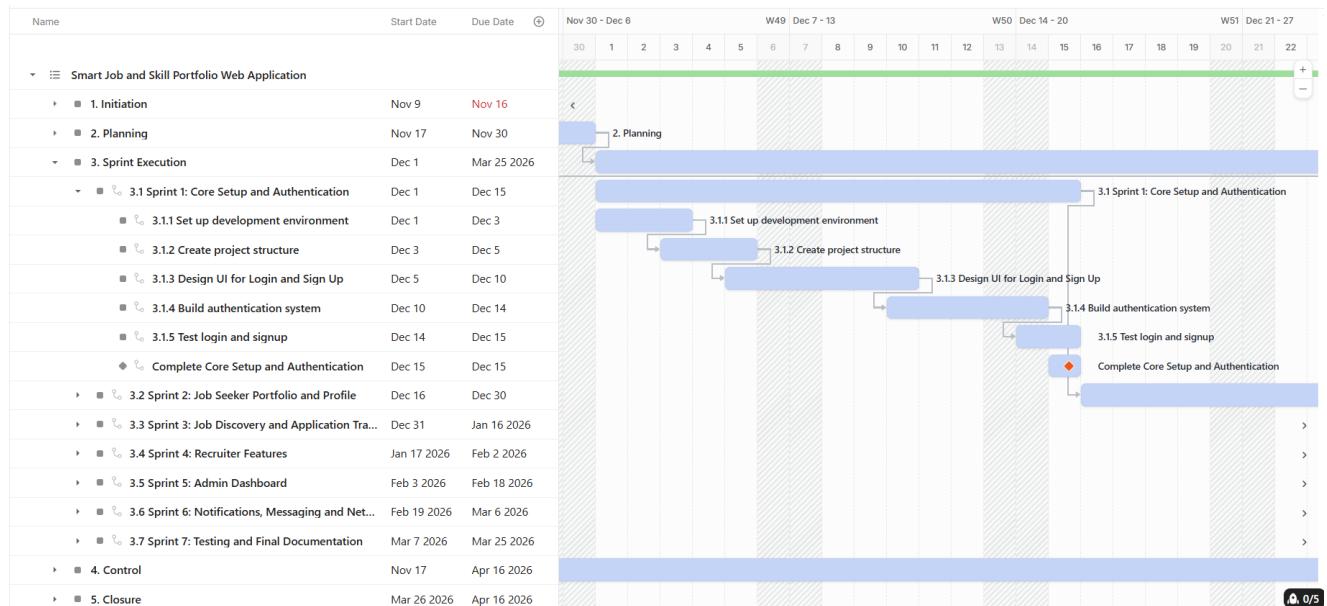


Figure 152: Previous Gantt Chart Sprint 1 Core Setup and Authentication Phase

### iv. Sprint Execution Phase (Sprint 2: Portfolio and Profile)

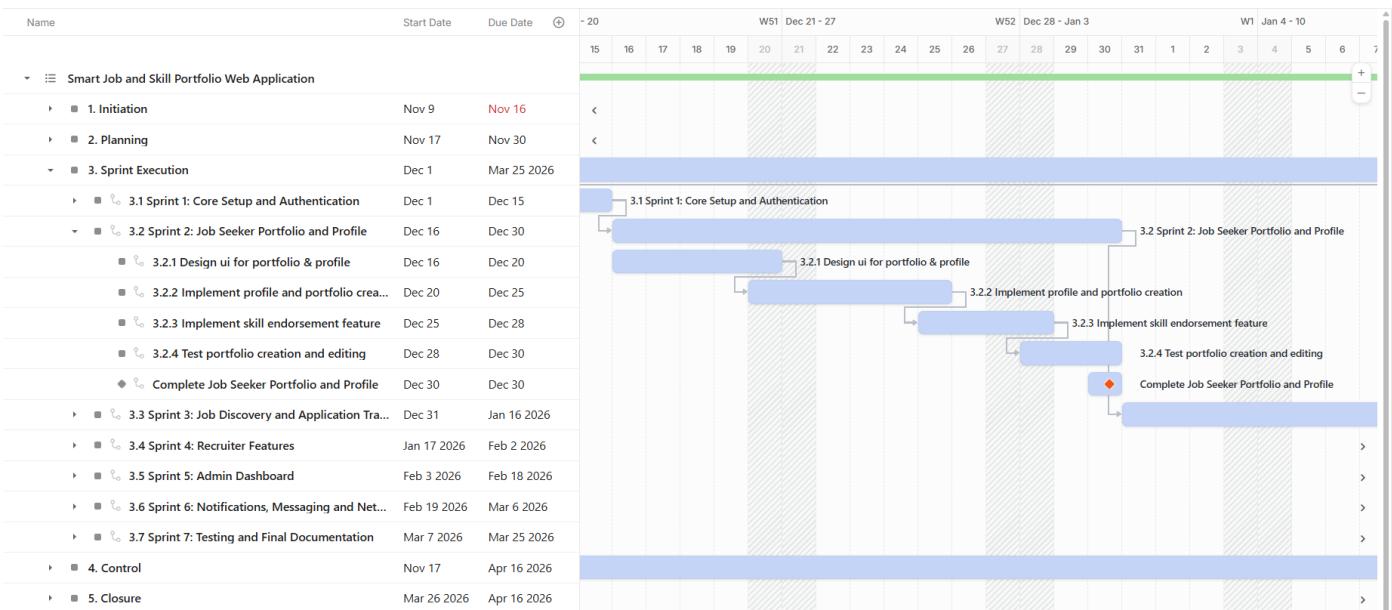


Figure 153: Previous Gantt Chart Sprint 2 Portfolio and Profile Phase

## v. Sprint Execution Phase (Sprint 3: Job Discovery and Application Tracking)

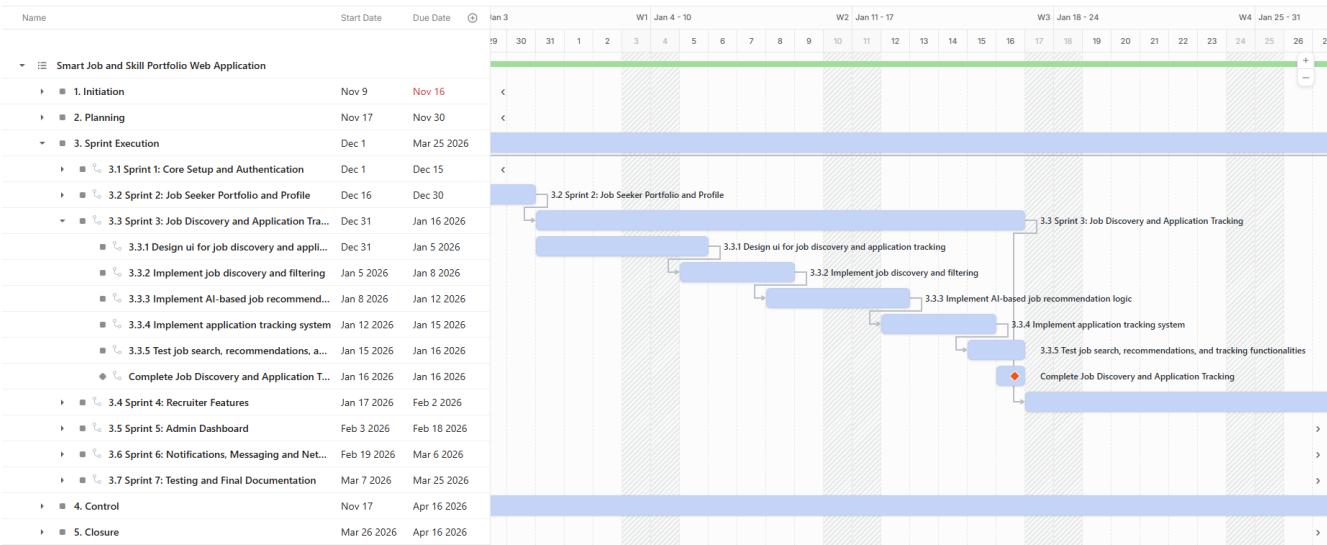


Figure 154: Previous Gantt Chart Sprint 3 Job Discovery and Application Tracking Phase

## vi. Sprint Execution Phase (Sprint 4: Recruiter Features)

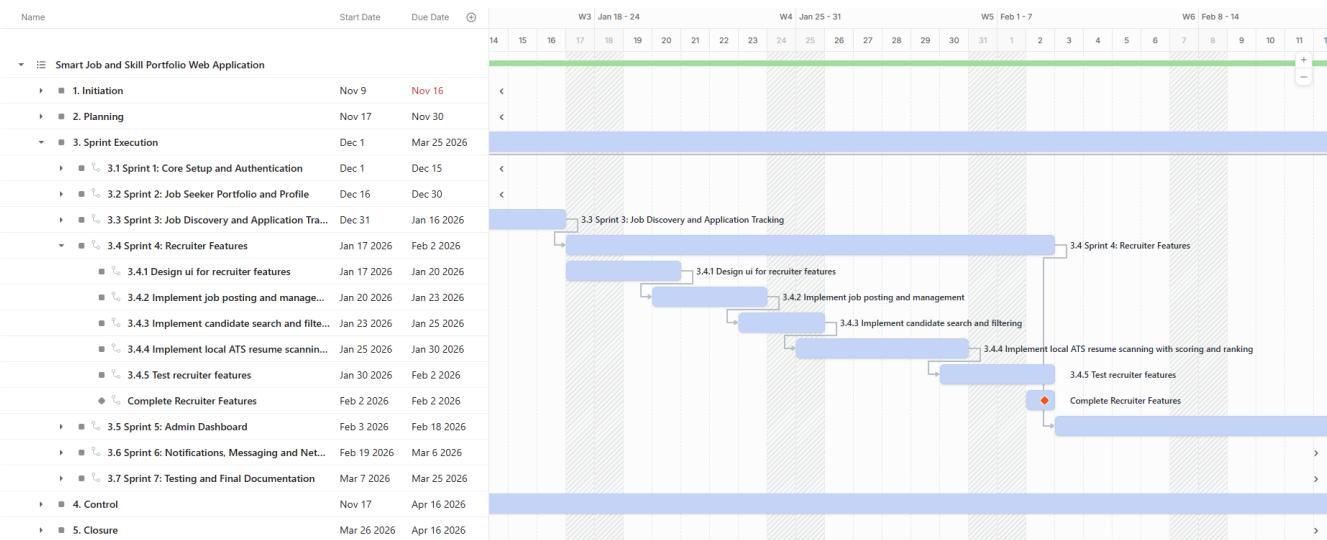


Figure 155: Previous Gantt Chart Sprint 4 Recruiter Features Phase

## vii. Sprint Execution Phase (Sprint 5: Admin Dashboard)

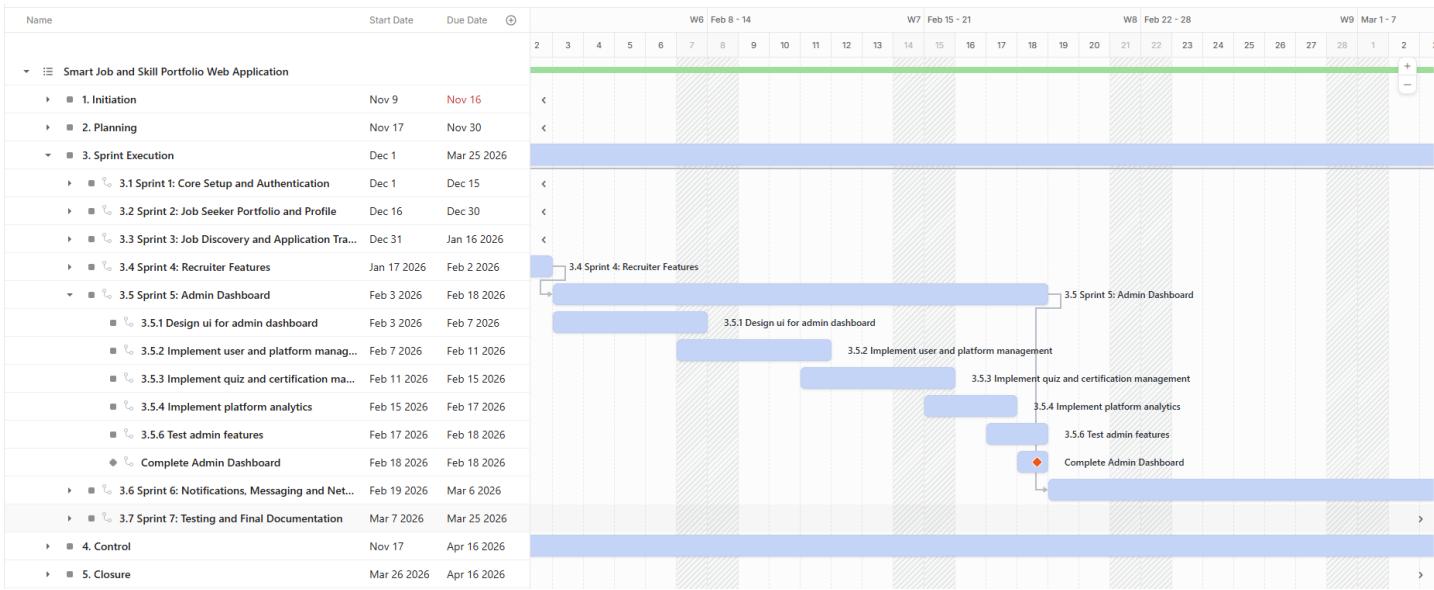


Figure 156: Previous Gantt Chart Sprint 5 Admin Dashboard Phase

## viii. Sprint Execution Phase (Sprint 6: Notifications, Messaging, and Networking)

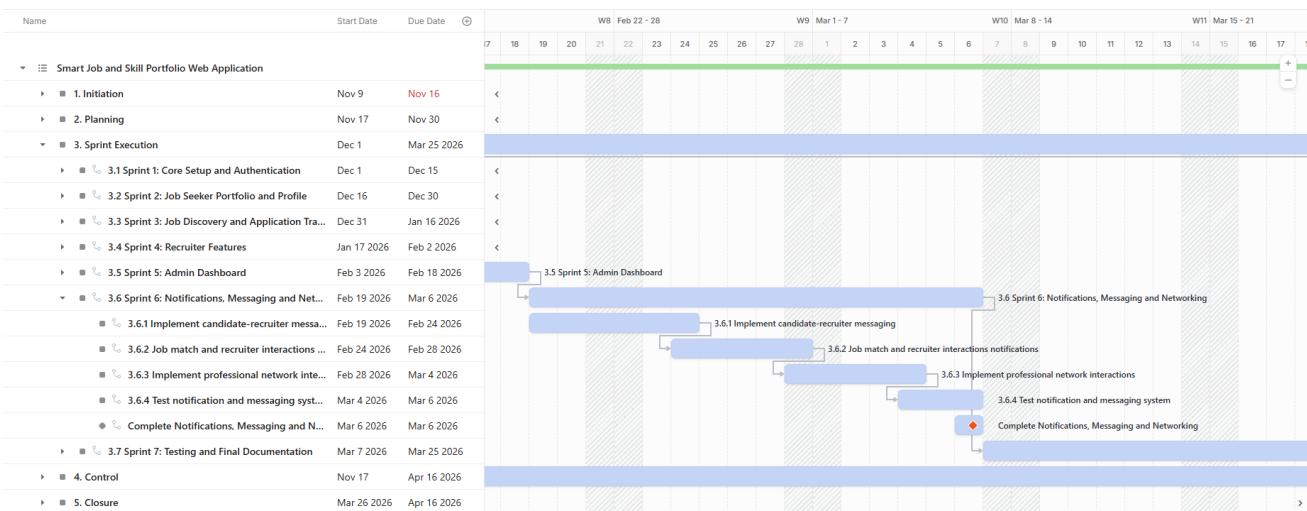


Figure 157: Previous Gantt Chart Sprint 6 Notifications, Messaging, and Networking Phase

## ix. Sprint Execution Phase (Sprint 7: Testing and Final Documentation)

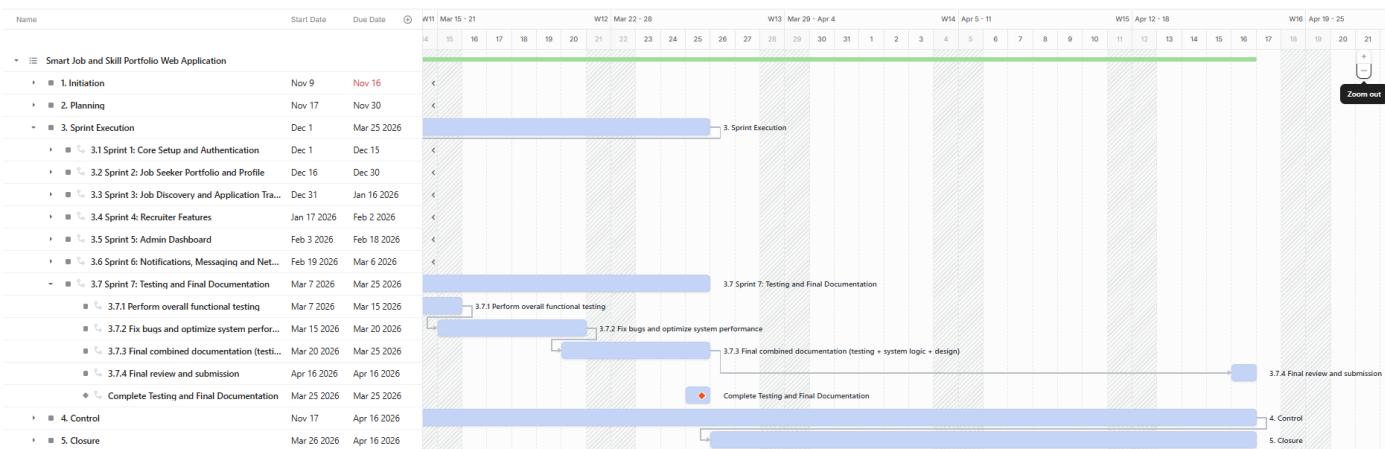


Figure 158: Previous Gantt Chart Sprint 7 Testing and Final Documentation Phase

## x. Control Phase

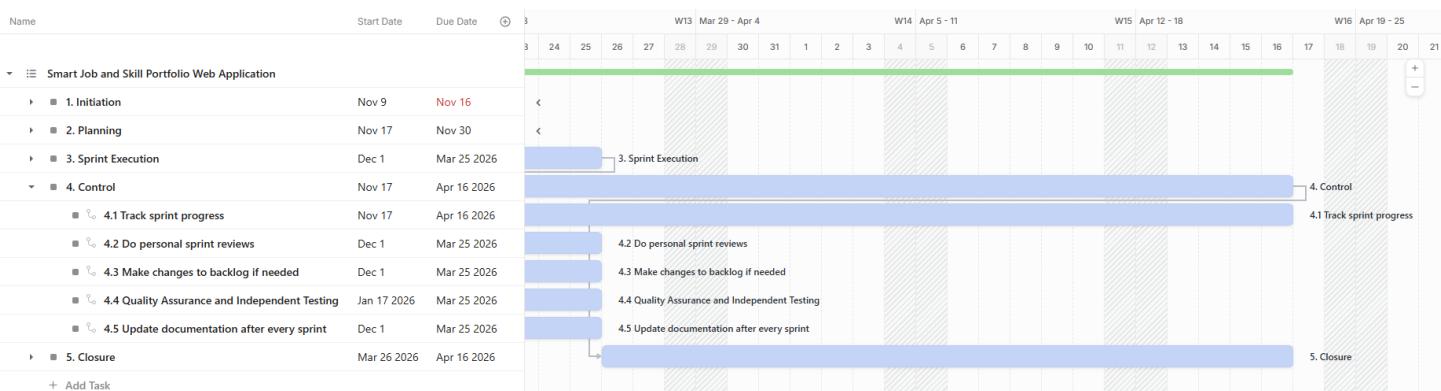


Figure 159: Previous Gantt Chart Control Phase

## xi. Closure Phase

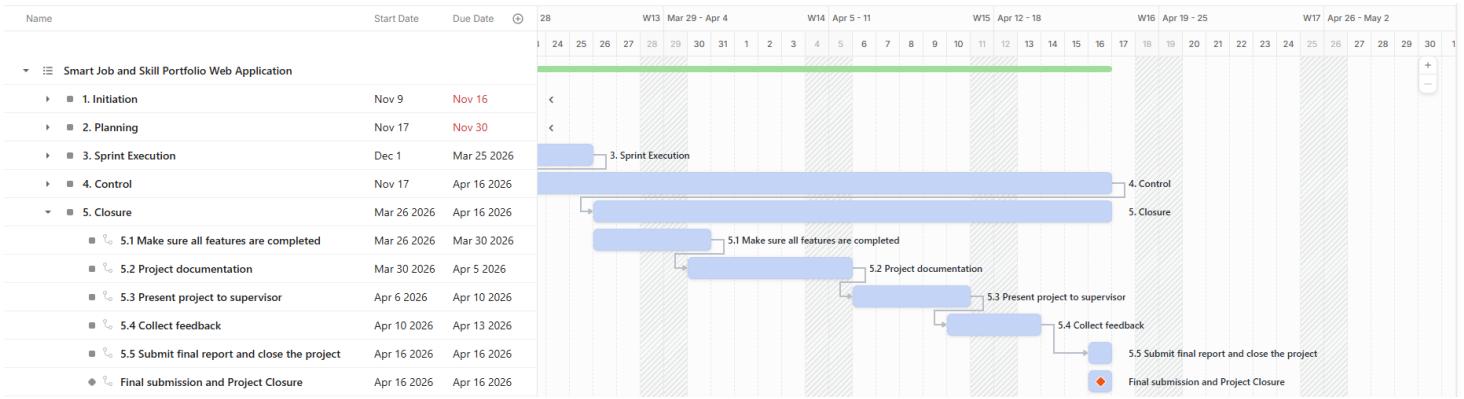
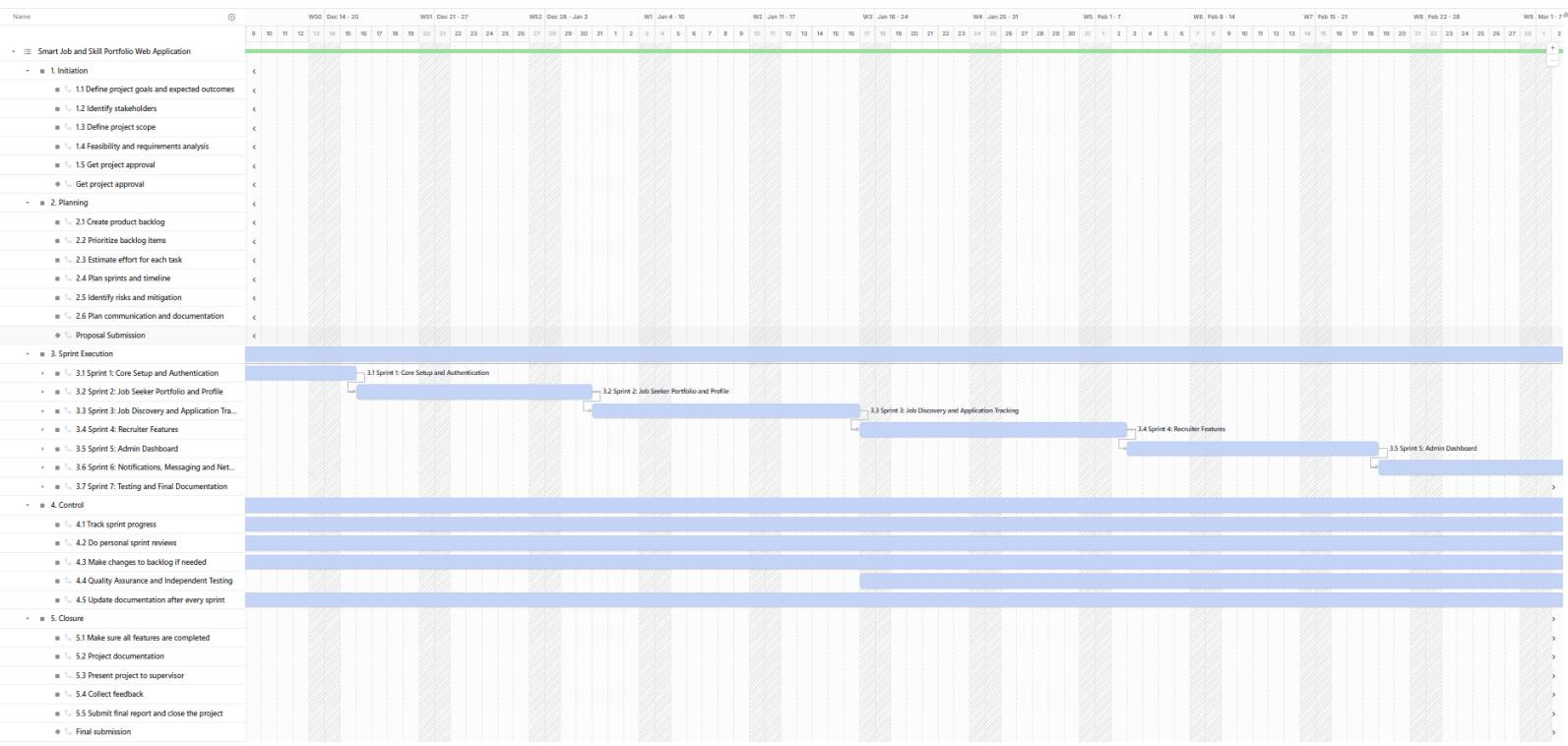
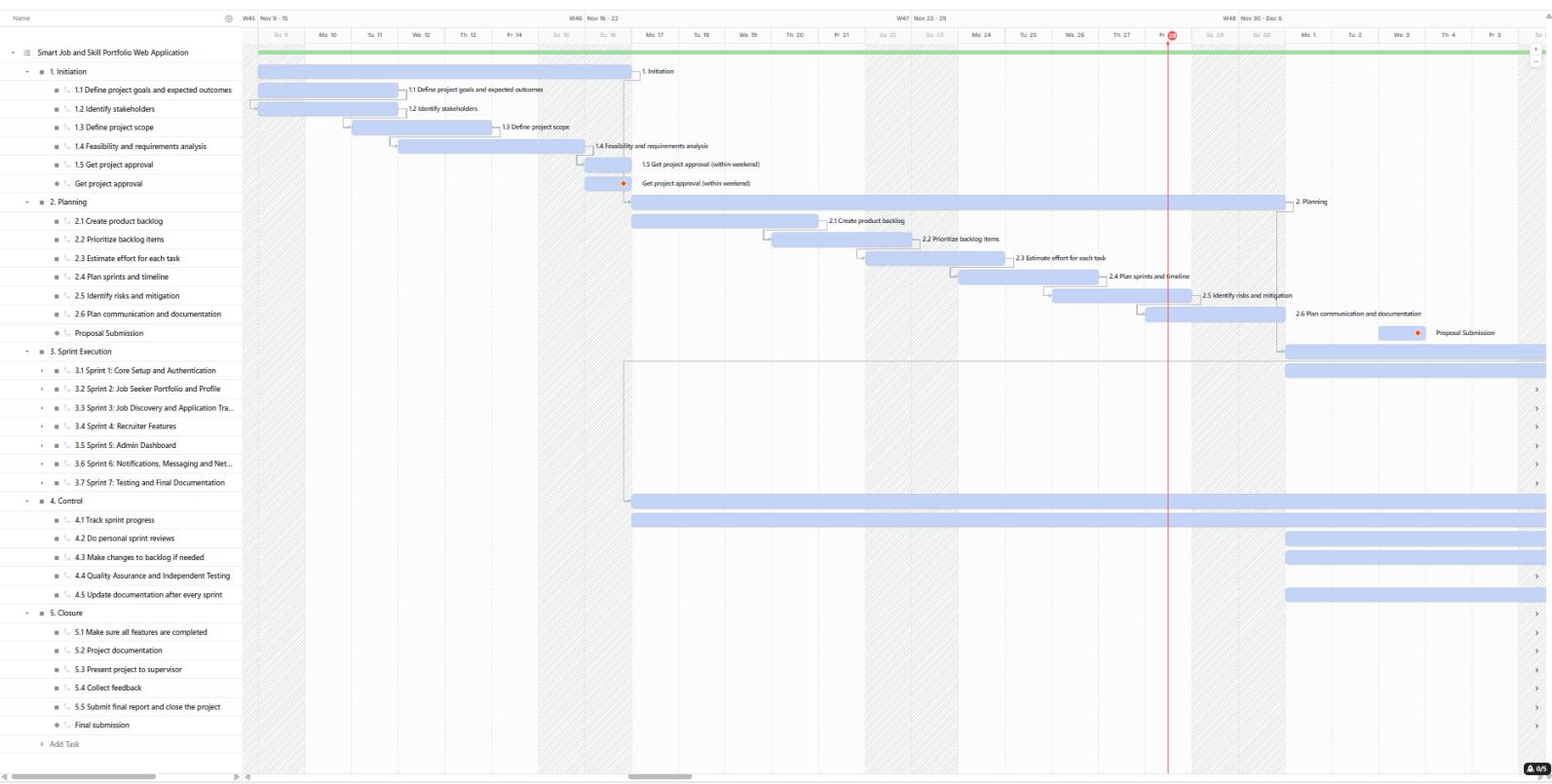
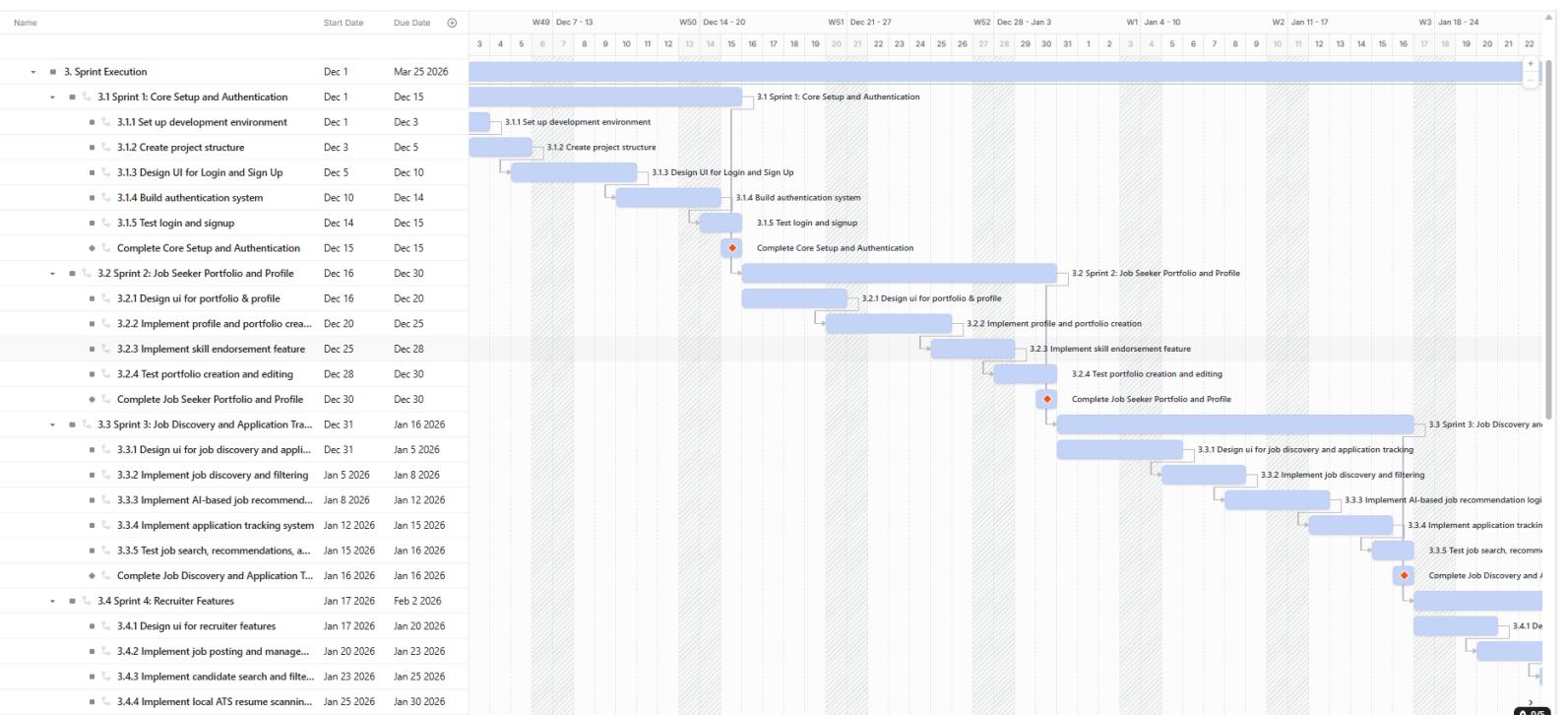
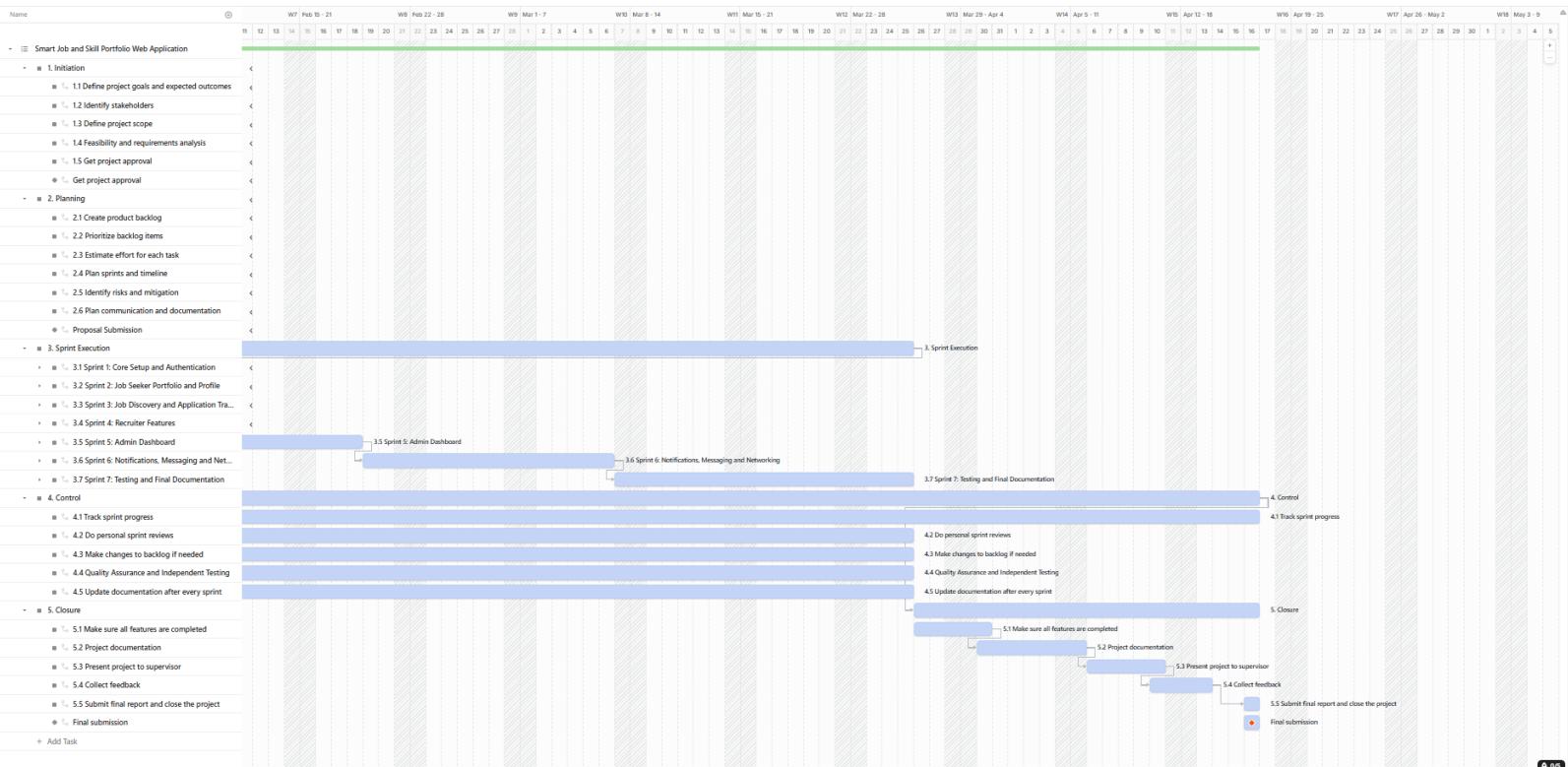
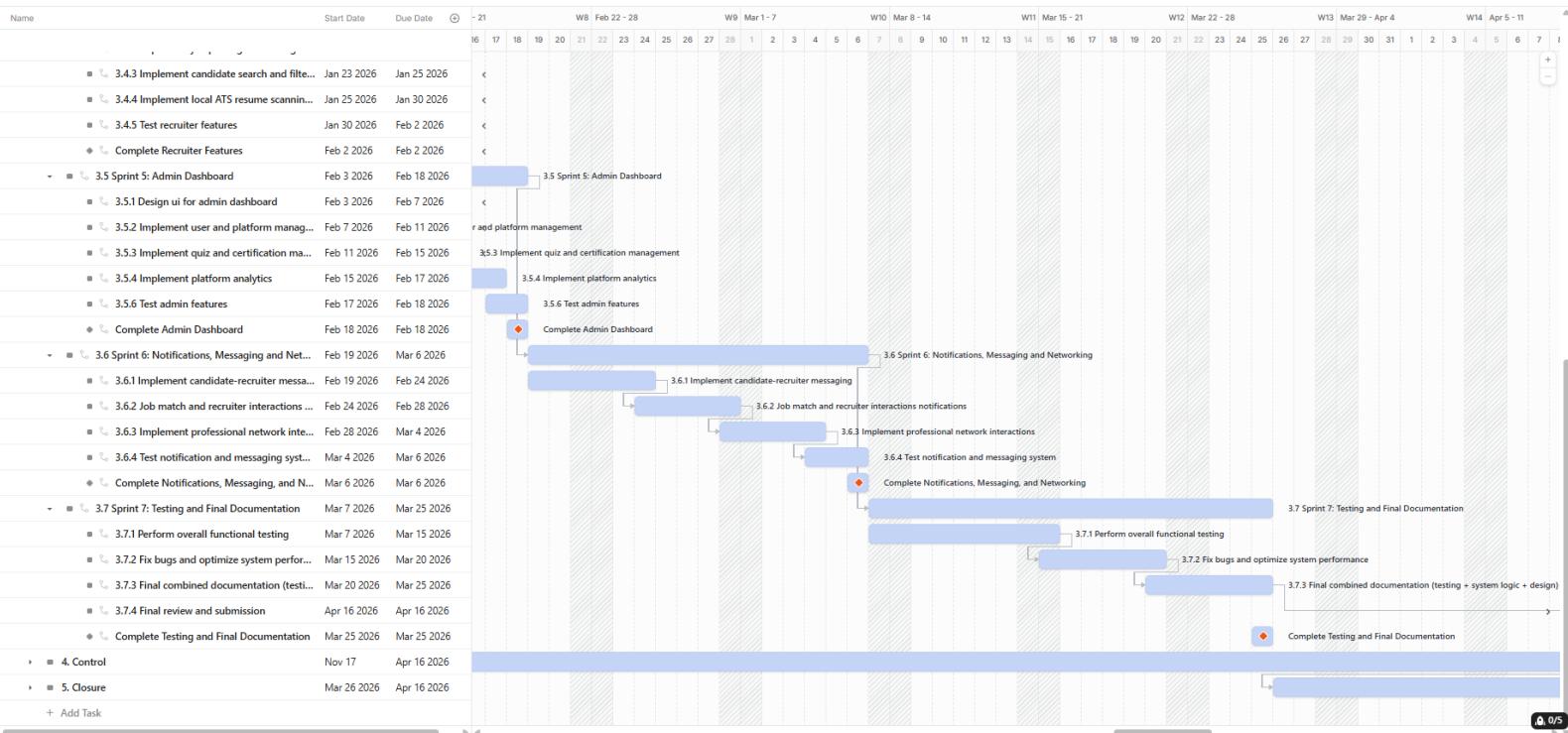
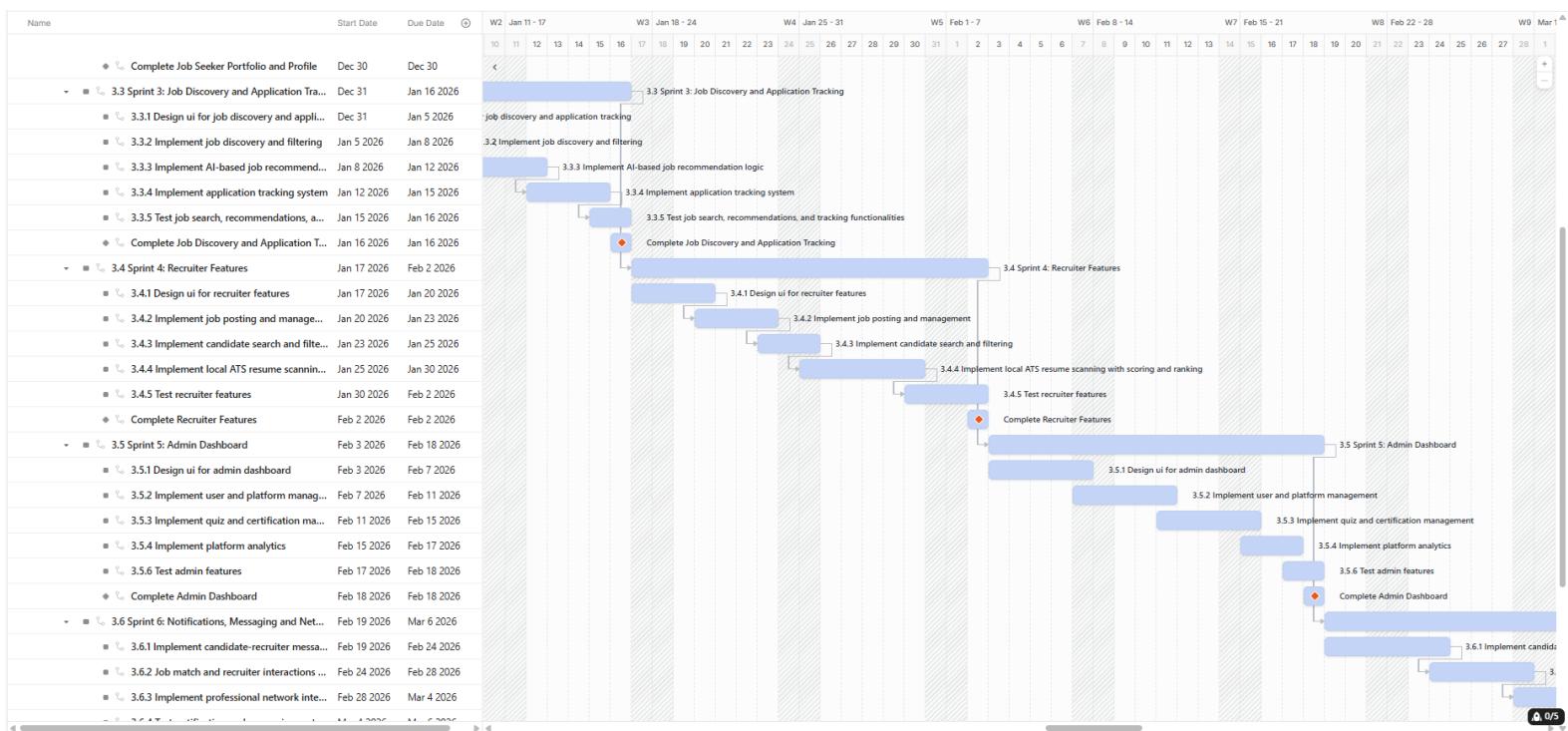


Figure 160: Previous Gantt Chart Closure Phase

## **xii. Full View of Gantt Chart**







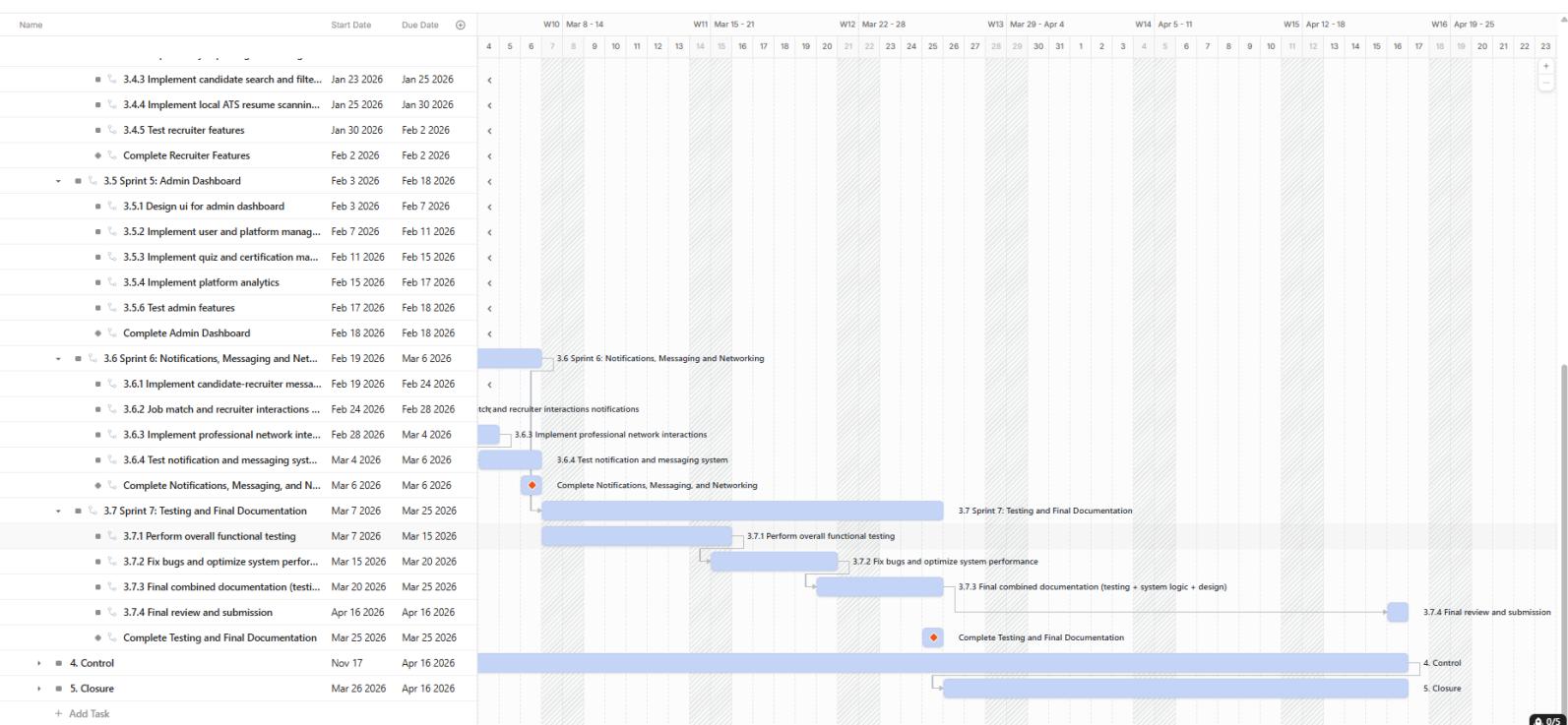


Figure 161: Previous Gantt Chart Full View

[Back to Revised Gantt Chart](#)

[For detailed and larger of Previous Gantt Chart. Click here.](#)