



Tunisian Republic
Ministry of Higher Education
and Scientific Research
Private Higher School of Engineering and Technology
TEK-UP



FINAL YEAR PROJECT REPORT

Presented in fulfillment of the requirements for obtaining the

Engineering Degree in Computer Science

Software Engineer

Prepared by

Ourabi Mahrez

AI-Driven Recruitment Platform - Hire Tech

Academic Supervisor:

Ms. Agrebi Ines

Professional Supervisor:

M. Mabrouk Mohamed-Ali

Academic Year 2024–2025

I authorize the student to submit their internship report for the defense.

Professional Supervisor, **Ms. Agrebi Ines**

Signature and Stamp

I authorize the student to submit their internship report for the defense.

Academic Supervisor, **M. Mabrouk Mohamed-Ali**

Signature

Sté AUDAXIS MAGHREB
Rue de Windelmère - Imm. Maghreb
Bloc "B" - 2ème Etage
Tél: 71 656 187 - Fax: 71 656 284

Dedication

To my family,

To my mother, whose prayers have always been my shield and whose love is the foundation upon which I stand. Your resilience is my greatest lesson.

To my father, whose hands have worked tirelessly and whose quiet pride has been my loudest encouragement. You taught me the meaning of perseverance.

To my brother and my sisters, my first friends and forever allies. Your belief in me was a constant light, especially when my own flickered.

This work is a testament to your love and sacrifice. I hope I have made you proud.

— Ourabi Mahrez

Acknowledgments

It is with immense gratitude that I reflect upon the many people who supported and guided me throughout this demanding yet rewarding journey.

My first and deepest thanks go to my supervisors, whose expertise and dedication were the cornerstones of this project. **Ms. Agrebi Ines**, your unwavering belief in my abilities, your profound patience, and your insightful guidance pushed me to exceed my own expectations. You were not just a supervisor but a true mentor. **M. Mabrouk Mohamed-Ali**, your technical wisdom, sharp critical eye, and constant availability were invaluable. Thank you for challenging my ideas and helping me refine them into a robust, professional product.

I extend my sincere gratitude to Audaxis for hosting this project. The professional environment and practical insights were instrumental in grounding my academic work in real-world challenges.

To my friends and colleagues, thank you for the late nights, the debugging sessions, the constructive feedback, and, most importantly, the moral support. You made the arduous moments bearable and the successful ones worth celebrating.

My final and most profound acknowledgment is reserved for my family. To my mother and father, there are no words sufficient to thank you for your boundless love, your unwavering faith, and the countless sacrifices you have made. You are my source of strength. To my brother and sisters, your encouragement and pride have been a constant source of motivation. This accomplishment is the fruit of your endless support.

Thank you all.

— Ourabi Mahrez

Table of Contents

General Introduction	1
1 State of the Art	3
1.1 Introduction	4
1.2 Presentation of the Host Organization: Audaxis	4
1.2.1 Company Overview	4
1.2.2 Services Offered by Audaxis	4
1.3 Project Context and Definition	5
1.3.1 Context and Motivation	5
1.3.2 Scope and Objectives	5
1.3.3 Problem Statement and Identified Needs	6
1.4 Review of Existing Solutions	6
1.4.1 Market Overview and Comparative Analysis	6
1.4.2 Comparative Analysis of Recruitment Platforms	6
1.4.3 Unique Value Proposition of Hire-Tech	7
1.5 Adopted Methodology	8
1.5.1 Rationale Behind Methodology Selection	8
1.5.2 Why Agile?	8
1.5.3 Comparative Analysis of Development Methodologies	8
1.5.4 Agile Framework: Scrum	9
1.5.5 Adaptation of Scrum for Solo Development	11
1.6 Conclusion	11
2 Needs Analysis and System Specification	12
2.1 Needs Analysis	13
2.1.1 Functional Requirements	13
2.1.2 Non-Functional Requirements	14
2.2 Backlog and Agile Development	14
2.2.1 Assignment of Priority and Complexity	14
2.2.2 Product Backlog	16
2.2.3 Effort Conversion and Sprint / Release Duration	19
2.3 General Use-Case Diagram	24

2.4	General Class Diagram	27
2.5	Working Environment and Technologies	30
2.5.1	Programming Languages and Frameworks	30
2.5.2	System Architecture and Infrastructure	31
2.5.3	Development Tools and Environment	34
2.5.4	Testing and Quality Assurance	35
2.5.5	Development Machine Specifications	36
2.5.6	DevOps	36
2.6	Conclusion	37
3	Release 1: Platform Core	38
3.1	Introduction	39
3.2	Release 1 Implementation: Platform Foundation	39
3.2.1	Release 1 Use-Case Overview	39
3.2.2	Release Objectives and Scope	40
3.2.3	Backlog of Release 1	41
3.3	Development and Deployment Environment	42
3.3.1	Containerized Development Setup	42
3.3.2	Technical Environment Evidence	44
3.3.3	CI/CD Pipeline Implementation	45
3.4	Technical Architecture Implementation	46
3.4.1	Layered Architecture Realization	46
3.4.2	Database Schema Implementation	47
3.5	Core System Workflows Implementation	48
3.5.1	Authentication and Authorization System	48
3.5.2	Candidate Profile Management	50
3.6	User Interface Realization	51
3.6.1	Authentication Interface Implementation	52
3.6.2	Candidate-Facing Interfaces	53
3.6.3	Recruiter Interface Implementation	60
3.7	Quality Assurance and Testing	68
3.7.1	Test Strategy Implementation	68
3.7.2	Quality Metrics Achievement	69
3.8	Performance Optimization	69

3.8.1	Database Performance	69
3.8.2	API Performance Optimization	70
3.9	Challenges and Solutions	70
3.9.1	Technical Challenges Overcome	70
3.9.2	Architectural Decisions Validated	71
3.10	Conclusion	71
4	Release 2: Workflow Engine	72
4.1	Introduction	73
4.2	Release 2 Implementation Scope	73
4.2.1	Release Objectives and Sprint Alignment	73
4.2.2	Use Case Realization	73
4.2.3	Backlog Implementation Validation	74
4.3	Core Feature Implementation	75
4.3.1	AI-Powered CV Parsing Service	75
4.3.2	AI Candidate Matching Engine	77
4.3.3	AI-Assisted Job Creation	79
4.4	User Interface Implementation	81
4.4.1	Job Opportunities Discovery Interface	81
4.4.2	Job Application Workflow	82
4.4.3	Candidate Application Tracking	84
4.4.4	Application Management for Specific Positions	84
4.4.5	Application Status Management Interface	85
4.4.6	Job Management Dashboard	86
4.4.7	AI-Powered Candidate Matching	87
4.5	Quality Assurance and Performance	89
4.5.1	Quality Metrics Achievement	89
4.5.2	Performance Optimization	89
4.6	Implementation Challenges and Solutions	90
4.6.1	Technical Challenges	90
4.6.2	User Experience Challenges	90
4.6.3	Architectural Validation	90
4.7	Conclusion	91

5 Release 3: Production System	92
5.1 Introduction	93
5.2 Release 3 Implementation: Production System	93
5.2.1 Release 3 Use-Case Overview	93
5.2.2 Release Objectives and Scope	95
5.2.3 Backlog of Release 3	95
5.3 Advanced System Workflows Implementation	96
5.3.1 Interview Scheduling System	96
5.3.2 Analytics System Implementation	97
5.4 User Interface Realization for Production Features	98
5.4.1 Interview Management Interface	98
5.4.2 Interview Creation Workflow	99
5.4.3 Analytics Dashboard Implementation	100
5.5 Quality Assurance and Testing	104
5.5.1 Testing Strategy	104
5.5.2 Quality Metrics	105
5.6 Implementation Challenges and Solutions	105
5.6.1 Technical Challenges	105
5.6.2 Architectural Decisions	105
5.7 Chapter Conclusion	106
5.8 Conclusion	106
General Conclusion	107
Webography	109

List of Figures

1.1	Visual Overview of the Scrum Framework	10
2.1	MoSCoW Prioritization Method	14
2.2	T-Shirt Size Complexity Mapping	15
2.3	Comprehensive Use-Case Diagram of Hire-Tech Recruitment System	25
2.4	Domain Model Class Diagram	28
2.5	Next.js Logo	30
2.6	Spring Boot Logo	30
2.7	Python Logo	31
2.8	Ollama Logo	32
2.9	Docker Logo	33
2.10	PostgreSQL Logo	33
2.11	pgAdmin Logo	33
2.12	Keycloak Logo	34
2.13	Visual Studio Code Logo	34
2.14	GitHub Logo	34
2.15	Swagger Logo	35
2.16	StarUML Logo	35
3.1	Release 1 Use-Case Diagram: Platform Foundation and Core Features	40
3.2	Hybrid Development Environment Architecture	42
3.3	Keycloak Administration Console Showing Configured Realm and User Management	44
3.4	pgAdmin Interface Showing Database Schema and Table Relationships	45
3.5	CI/CD Pipeline Implementation with Quality Gates	45
3.6	Physical Database Schema Implementation with Performance Optimizations	47
3.7	Authentication Sequence: Frontend, Backend, and Keycloak Integration	49
3.8	Manual Profile Creation Sequence Diagram	51
3.9	User Registration Interface with Multi-provider Authentication Options	52
3.10	Secure Login Interface with Social Authentication Options	53
3.11	Corporate Landing Page Showcasing Employer Value Proposition	54
3.12	Document Management Interface for CV and Cover Letter Submission	55
3.13	Personal Information Collection Interface with International Support	56

3.14 Professional Experience and Career Context Section	57
3.15 Comprehensive Skills Management Interface with Category-based Organization	58
3.16 Candidate Dashboard: Personalized Job Search and Career Management Interface . .	59
3.17 Job Discovery Interface with Advanced Search and Filtering	60
3.18 Recruiter Dashboard: Basic Hiring Management Interface	61
3.19 Job Creation Interface: Basic Information Step	62
3.20 Job Creation Interface: Manual Description Creation	64
3.21 Job Creation Interface: Final Review and Publish Step	66
3.22 Job Management Interface with Basic Position Overview	67
3.23 Test Pyramid: Distribution and Coverage Metrics	68
4.1 Release 2 Workflow Engine Use Case Diagram	74
4.2 CV Parsing and Data Extraction Sequence Diagram	76
4.3 AI Candidate Matching Sequence Diagram	78
4.4 AI-Assisted Job Creation Interface	80
4.5 Job Opportunities Discovery Interface with Advanced Filtering (US-4.1, US-4.2)	82
4.6 Job Application Modal with Cover Letter Upload (US-6.1)	83
4.7 Candidate Application Status Tracking Interface (US-6.1)	84
4.8 Application Management for Specific Position (US-6.2)	85
4.9 Application Status Management Interface (US-6.3)	86
4.10 Job Management Dashboard with AI Matching	87
4.11 AI Matching Results with Component Analysis (US-7.1)	88
5.1 Release 3 Use-Case Diagram: Production System and Enterprise Features	94
5.2 Interview Scheduling Workflow with Conditional Approval Process	96
5.3 Analytics Dashboard Sequence with Real-time Data Processing	97
5.4 Interview Management Interface with Calendar View	98
5.5 Interview Creation - Candidate and Time Selection	99
5.6 Interview Creation - Configuration and Setup	100
5.7 Main Analytics Dashboard with Recruitment Overview	101
5.8 Department Performance Analytics with Fill Rate Indicators	102
5.9 Job Status Analytics with Contract Type Distribution	103
5.10 Interview Analytics with Status Distribution and Activity Tracking	104

List of Tables

1.1	Audaxis Organizational Overview	5
1.2	Comparative Analysis of Recruitment Platforms	7
1.3	Comparison of Software Development Methodologies	9
2.1	Complete Product Backlog: User Stories with Priorities and Dependencies	16
2.2	Sprint Plan with Dependency-Validated Sequence	20
2.3	Value-Driven Release Strategy	21
2.4	Strategic Feature Deferral for MVP Focus	22
2.5	Quality Assurance & Success Metrics	23
3.1	Release 1 Implementation Schedule and Deliverables	41
3.2	Release 1 User Stories Implementation Status	41
3.3	Database Schema Implementation Decisions	48
3.4	Quality Assurance Metrics and Achievement	69
4.1	Release 2 Sprint Implementation Alignment	73
4.2	Release 2 User Stories Implementation Validation	75
4.3	Quality Metrics Achievement for Release 2	89
5.1	Release 3 Implementation Schedule and Deliverables	95
5.2	Release 3 User Stories Implementation Status	95
5.3	Quality Metrics for Release 3 Implementation	105

List of Abbreviations

- **AI** = Artificial Intelligence
- **API** = Application Programming Interface
- **ATS** = Applicant Tracking System
- **CI/CD** = Continuous Integration / Continuous Delivery
- **CRUD** = Create, Read, Update, Delete
- **CSRF** = Cross-Site Request Forgery
- **DAST** = Dynamic Application Security Testing
- **DSL** = Domain-Specific Language
- **E2E** = End-to-End
- **GDPR** = General Data Protection Regulation
- **GIN** = Generalized Inverted Index
- **GLSI** = Génie Logiciel et Système d'Information
- **HRIS** = Human Resources Information System
- **JSONB** = Binary JSON (PostgreSQL JSONB)
- **JWT** = JSON Web Token
- **LLM** = Large Language Model
- **ML** = Machine Learning
- **MVP** = Minimum Viable Product
- **NLP** = Natural Language Processing
- **OAuth2** = Open Authorization 2.0
- **OIDC** = OpenID Connect
- **RBAC** = Role-Based Access Control
- **REST** = Representational State Transfer

- **SAST** = Static Application Security Testing
- **SMTP** = Simple Mail Transfer Protocol
- **SQL** = Structured Query Language
- **SSO** = Single Sign-On
- **SSR** = Server-Side Rendering
- **UI** = User Interface
- **UML** = Unified Modeling Language
- **UUID** = Universally Unique Identifier
- **UX** = User Experience

General Introduction

The digital transformation of human resources has made intelligent recruitment platforms a critical asset for organizational success. Traditional hiring processes, often hampered by manual screening, unconscious bias, and inefficient workflows, create significant bottlenecks in acquiring top talent. This final year project addresses these challenges through the design and development of *Hire-Tech*, an AI-assisted recruitment platform engineered to modernize the talent acquisition lifecycle.

Hire-Tech integrates core Applicant Tracking System (ATS) capabilities with a dedicated AI component to automate key stages of recruitment. At a high level, the platform transforms unstructured resumes into structured candidate profiles, estimates candidate–job fit through intelligent matching, and supports coordinated interview workflows with clear approvals and accountability. The user interface communicates exclusively with the platform’s secure backend, which orchestrates all interactions and mediates access to the AI services; this separation reinforces security, auditability, and consistent governance of data and decisions. To preserve privacy and control, the platform operates its language model locally within the organization’s environment via a standard interface, ensuring that sensitive data does not leave the system boundary. The overall design emphasizes reliability, privacy, and fairness while delivering a streamlined experience for candidates, recruiters, and managers.

This report documents the full lifecycle of *Hire-Tech*, from conception to functional realization, following a phased and incremental approach:

Chapter 1: State of the Art establishes the project’s foundation. It presents the host organization, Audaxis, analyzes the limitations of existing recruitment solutions, defines the project’s scope and objectives, and justifies the adoption of Agile Scrum, including its adaptation for a solo development context.

Chapter 2: Needs Analysis and System Specification translates the conceptual vision into a product blueprint. It details the functional and non-functional requirements, introduces a prioritized product backlog and a 12-sprint delivery plan, and presents the principal models and architectural choices that structure the platform.

Chapter 3: Implementation and Realization — Release 1: Platform Foundation reports the delivery of the foundational layer: secure onboarding and access control, the core data model, continuous delivery practices, and the first end-to-end user journeys (profiles, job posting, and discovery).

Chapter 4: Implementation and Realization — Release 2: Workflow Engine documents the arrival of intelligent capabilities and complete recruitment flows, including automated résumé

understanding, application tracking, and initial AI-supported shortlisting integrated within the platform's governance.

Chapter 5: Implementation and Realization — Release 3: Production System presents the enterprise features that complete the solution: interview scheduling with approvals, transparent fit scoring, analytical dashboards for oversight, and reinforced security, audit, and compliance measures.

The report concludes with a **General Conclusion** that synthesizes the project's achievements, evaluates outcomes against objectives, and outlines perspectives for future enhancements.

Through this structured exposition, the report demonstrates a methodical engineering process that delivers a robust and trustworthy AI-assisted recruitment platform. By maintaining clear boundaries—user interface to backend, backend to AI services—the solution preserves security and accountability while providing accurate parsing, relevant matching, and streamlined workflows suited to the evolving demands of modern talent acquisition.

STATE OF THE ART

Plan

1	Introduction	4
2	Presentation of the Host Organization: Audaxis	4
3	Project Context and Definition	5
4	Review of Existing Solutions	6
5	Adopted Methodology	8
6	Conclusion	11

1.1 Introduction

The advent of artificial intelligence (AI) and data-driven technologies has revolutionized the recruitment landscape. [W4] Traditional hiring methods characterized by manual screening and subjective evaluations often lead to inefficiencies, biases, and delays. In contrast, modern platforms leverage AI, natural language processing (NLP), and intelligent recommendation systems to streamline talent acquisition and enhance decision-making.

This chapter outlines the technological and strategic framework underpinning Hire-Tech, an innovative AI-driven recruitment platform. We present the project's context, scope, and objectives, review current solutions and best practices, describe the Agile methodology adopted for development, and introduce Audaxis the host organization responsible for supporting this initiative.

1.2 Presentation of the Host Organization: Audaxis

1.2.1 Company Overview

Audaxis, a member of the Contraste Europe Group, is a leading IT services company with a strong international presence. [W5] With over 30 years of experience in designing, developing, and managing complex IT infrastructures, Audaxis has successfully executed more than 300 ERP projects. The company supports both mid-market firms and large enterprises, leveraging significant technical expertise to drive innovation and operational excellence.

1.2.2 Services Offered by Audaxis

Audaxis provides a comprehensive suite of services designed to deliver high-value solutions to its clients:

1.2.2.1 ERP Implementation

Audaxis collaborates closely with clients to implement ERP solutions that are precisely tailored to their needs. The process begins with an in-depth analysis of the client's requirements and strategic objectives, allowing the team to recommend high value-added ERP solutions. Subsequently, Audaxis configures the solution on-site, provides training for potential users, and securely migrates data from the existing system to the new ERP platform. [W5]

1.2.2.2 AMOA Consulting

Audaxis offers AMOA (Assistance to the Project Owner) consulting services that encompass clarifying client requirements and objectives, analyzing technical and organizational solutions, supervising service

providers, tracking costs and deadlines, and ensuring end-user satisfaction. This comprehensive approach guarantees that the final solution aligns with the client’s strategic goals. [W5]

1.2.2.3 Application Architecture

Audaxis delivers tailored application architecture solutions that emphasize cloud integration, security, and resilience. Leveraging technologies such as Kafka and Docker, the company ensures high system availability. As experts in database management, Audaxis selects the most suitable tools to meet each client’s needs, and its solutions include proactive monitoring to prevent operational incidents. PostgreSQL is used for reliable transactional data management.

Table 1.1 provides a summarized snapshot of the company’s structure and scale.

Aspect	Details
Number of Employees	Over 400 employees
Revenue	Over 48 million euros
Years of Experience	30 years
ERP Projects Completed	300+ projects

Tableau 1.1: Audaxis Organizational Overview

1.3 Project Context and Definition

1.3.1 Context and Motivation

In today’s competitive market, organizations must secure top talent while minimizing operational inefficiencies. Conventional recruitment methods struggle with labor-intensive processes and inherent biases, which can lead to suboptimal candidate matches. Recognizing these limitations, Hire-Tech is designed to bridge the gap between job seekers and employers by integrating advanced AI-based CV parsing, precise job-matching algorithms, and automated workflows. This data-centric approach refines candidate screening and optimizes the overall hiring process.

1.3.2 Scope and Objectives

Hire-Tech aims to transform recruitment by:

- **Automating Candidate Screening:** Employing AI algorithms for efficient CV parsing and candidate evaluation.
- **Enhancing Job Matching:** Utilizing machine learning to improve the accuracy of candidate-job pairings.

- **Empowering Recruiters:** Providing an intelligent dashboard with real-time insights.
- **Ensuring Security and Compliance:** Upholding rigorous data protection and privacy standards.
- **Optimizing Interview Scheduling:** Streamlining the process with structured, multi-step approval workflows.

1.3.3 Problem Statement and Identified Needs

Current recruitment practices are hampered by:

- Labor-intensive screening processes leading to hiring delays.
- Unconscious biases that compromise candidate selection.
- Difficulties in managing large volumes of applications while ensuring quality.

A preliminary study has highlighted the need for:

- Automated CV parsing to expedite screening.
- Robust job-matching algorithms for accurate candidate placement.
- A structured, scalable workflow for interview scheduling.
- Strict adherence to industry standards for data security and privacy.

1.4 Review of Existing Solutions

1.4.1 Market Overview and Comparative Analysis

The recruitment technology market features a range of AI-powered solutions—from traditional Applicant Tracking Systems (ATS) to sophisticated AI-driven hiring tools. [W2] While many systems offer partial automation and intelligent matching, they often lack the comprehensive integration found in Hire-Tech.

1.4.2 Comparative Analysis of Recruitment Platforms

To provide a comprehensive understanding of the competitive landscape, we analyze key recruitment platforms and their capabilities. This comparison highlights the unique positioning of Hire-Tech in addressing specific market gaps. [W1]

As Table 1.2 will demonstrate, Hire-Tech combines modern architecture with targeted AI capabilities to deliver a differentiated recruitment solution.

Platform	CV Parsing	AI Matching	Scheduling	Technical Stack	Key Features
Hire-Tech	Yes (NLP)	Yes (ML)	Yes	Next.js, Spring Boot, PostgreSQL	Integrated AI workflow, Modern architecture, Enterprise security
Lever	Yes	Yes	Basic	React, Node.js, MongoDB	ATS, Analytics, Collaboration tools
Greenhouse	Yes	Limited	Basic	Ruby on Rails, React	Comprehensive ATS, Reporting, Integrations
Indeed	Basic	Yes	Limited	Multi-cloud	Resume builder, Job alerts, CV database
LinkedIn Talent	Yes	Yes	Limited	Enterprise stack	Professional networking, Recruiter tools

Tableau 1.2: Comparative Analysis of Recruitment Platforms

This analysis reveals that while established platforms offer robust features, Hire-Tech distinguishes itself through its integrated AI workflow and modern technology stack, providing a cohesive solution for modern recruitment challenges.

1.4.3 Unique Value Proposition of Hire-Tech

Hire-Tech differentiates itself through a combination of modern architecture, AI integration, and user-centric design:

- **Integrated AI Workflow:** Seamless integration of CV parsing, AI matching, and interview scheduling in a single platform
- **Modern Technology Stack:** Built with Next.js, Spring Boot, and PostgreSQL using containerized deployment
- **User-Centric Design:** Intuitive interfaces for both candidates and recruiters with real-time updates

- **Enterprise Security:** OAuth2/JWT authentication, RBAC, and GDPR compliance with Keycloak integration
- **Agile Development:** Iterative delivery with continuous integration and testing

1.5 Adopted Methodology

1.5.1 Rationale Behind Methodology Selection

Given the dynamic and rapidly evolving nature of both technology and recruitment markets, the development of the Hire-Tech platform demanded a methodology that supports adaptability, iterative progress, and stakeholder collaboration. After evaluating traditional and modern project management methodologies, we adopted the Agile approach for its proven ability to deliver incremental value and respond flexibly to changes in requirements. [W8]

1.5.2 Why Agile?

Agile methodology is centered on iterative development, continuous feedback, and adaptive planning. It facilitates early detection of issues, reduces project risks, and ensures alignment with end-user expectations through frequent releases and stakeholder engagement. In contrast to linear approaches such as the Waterfall model, Agile enables responsiveness in uncertain or exploratory project domains like AI-driven recruitment. [W7]

1.5.3 Comparative Analysis of Development Methodologies

To better understand the rationale behind our methodology selection, we conducted a comprehensive analysis of various development approaches. This evaluation considered factors such as flexibility, testing rigor, and suitability for AI-driven projects. [W11]

As Table 1.3 will illustrate, Agile Scrum emerged as the optimal choice for Hire-Tech's development requirements.

Methodology	Strengths	Limitations	Suitability for Hire-Tech
Waterfall	Clear structure, Defined stages	Rigid, Hard to adapt	Inflexible for evolving requirements
V-Model	Strong testing discipline	Costly, Not flexible	Unsuitable for exploratory AI integration
RAD (Rapid App Dev)	Fast prototyping, User-focused	Poor scalability, Lacks control	Viable for UI testing only
Agile (Scrum)	Iterative, Flexible, Feedback-driven	Requires discipline and collaboration	Ideal for adaptive AI projects

Tableau 1.3: Comparison of Software Development Methodologies

This methodology comparison clearly demonstrates that Agile Scrum provides the optimal balance of flexibility, stakeholder engagement, and technical rigor required for the Hire-Tech project's success.

1.5.4 Agile Framework: Scrum

We chose the Scrum framework a lightweight, flexible subset of Agile as it structures work into iterative cycles called **sprints**, enabling continuous delivery of functional components. Scrum emphasizes transparency, team accountability, and customer collaboration through its well-defined roles, events, and artifacts. [W7]

To better visualize how these Scrum components interact within a typical development cycle, Figure 1.1 provides a comprehensive illustration of the complete Scrum workflow. This diagram demonstrates the sequential flow from product backlog refinement through sprint execution to review and retrospective phases.

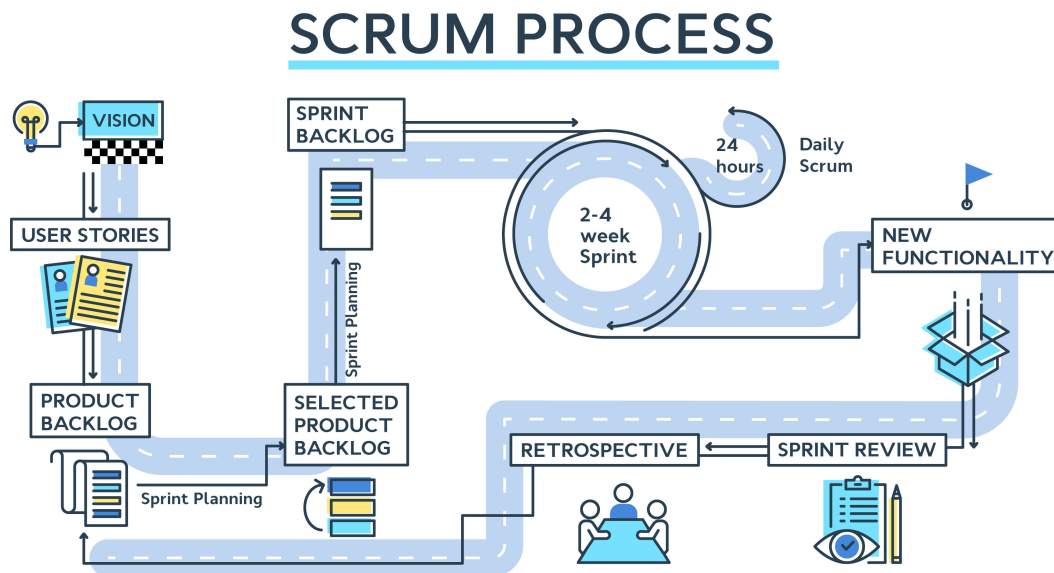


Figure 1.1: Visual Overview of the Scrum Framework

Scrum Roles

Scrum assigns specific responsibilities to three core roles:

- **Product Owner:** Represents stakeholders, defines the product vision, and prioritizes features in the Product Backlog.
- **Scrum Master:** Facilitates Scrum practices, removes impediments, and ensures the team adheres to Agile principles.
- **Development Team:** Cross-functional professionals responsible for delivering working increments of the product.

Scrum Artifacts

Scrum provides a minimal set of artifacts to ensure transparency and focus:

- **Product Backlog:** A dynamic, prioritized list of features, enhancements, and fixes.
- **Sprint Backlog:** A selection of Product Backlog items committed to in the current sprint.
- **Increment:** A potentially shippable piece of working software delivered at the end of each sprint.

Scrum Ceremonies

Regular events are embedded in Scrum to maintain rhythm and feedback:

- **Sprint Planning:** Define sprint goals and select Product Backlog items for the Sprint Backlog.

- **Daily Scrum:** A 15-minute daily meeting to synchronize progress, discuss obstacles, and adjust plans.
- **Sprint Review:** Stakeholders inspect the Increment and provide feedback.
- **Sprint Retrospective:** The team reflects on the sprint to identify improvements for the next iteration.

1.5.5 Adaptation of Scrum for Solo Development

Due to the nature of this academic project being executed by a single developer, the traditional Scrum roles are unified:

- **Product Owner:** Defines and continuously refines project objectives.
- **Scrum Master:** Organizes sprints, monitors progress, and conducts retrospectives.
- **Developer:** Handles design, implementation, testing, and documentation.

Despite the simplified team structure, the Scrum rituals, artifacts, and mindset are rigorously followed to ensure quality, adaptability, and disciplined progress throughout the development lifecycle.

1.6 Conclusion

This chapter has presented a comprehensive overview of the state of the art in AI-driven recruitment. By contextualizing the challenges of traditional hiring and reviewing current technological trends, we have established the foundation for the Hire-Tech project. The integration of an Agile, Scrum-based methodology underscores our commitment to innovation and efficiency. In the next chapter, we will explore the technical architecture and implementation details of Hire-Tech, delving into its core components and system design.

NEEDS ANALYSIS AND SYSTEM SPECIFICATION

Plan

1	Needs Analysis	13
2	Backlog and Agile Development	14
3	General Use-Case Diagram	24
4	General Class Diagram	27
5	Working Environment and Technologies	30
6	Conclusion	37

Introduction

Efficient and intelligent hiring has become a necessity for organizations striving to attract top talent while reducing operational overhead. Traditional recruitment approaches, reliant on manual screening and subjective decision-making, often result in inefficiencies, hiring biases, and a poor candidate experience. The emergence of AI-powered recruitment solutions, such as *Hire-Tech*, offers a transformative shift by automating key processes, improving job-matching accuracy, and ensuring a data-driven hiring approach. [W4]

This chapter presents an in-depth analysis of the functional and non-functional requirements of *Hire-Tech*, followed by a detailed breakdown of the product backlog, sprint planning, and system design. We also explore the key actors involved and the overall architecture through use-case and class diagrams, concluding with a comprehensive overview of the technological environment that will support the platform's implementation.

2.1 Needs Analysis

2.1.1 Functional Requirements

The core functionalities of *Hire-Tech* are designed to streamline recruitment processes while ensuring efficiency and automation:

- **Automated CV Parsing:** AI-driven Natural Language Processing (NLP) to extract structured data from resumes.
- **Job-Matching Algorithm:** Machine learning models to analyze candidate profiles and recommend suitable job opportunities.
- **Applicant Tracking System (ATS):** Manage candidate applications, screening statuses, and recruiter interactions via a unified dashboard.
- **Interview Scheduling:** Automated proposal of interview slots by recruiters with manager approval workflows.
- **User Role Management:** Role-based access control (RBAC) for recruiter, and super-admin functions.
- **Recruitment Analytics Dashboard:** Real-time insights on hiring metrics, candidate performance, and application trends.

2.1.2 Non-Functional Requirements

Hire-Tech must also meet high standards of performance, reliability, and security:

- **Scalability:** Support growing user load without degradation.
- **Reliability:** 99.9% uptime with robust error handling.
- **User Experience (UX):** Intuitive UI/UX for both recruiters and candidates.
- **Security:** Data encryption, GDPR-compliance, and secure session management.
- **Integration Capability:** Seamless connectors to LinkedIn, Indeed, and HRIS platforms

2.2 Backlog and Agile Development

Hire-Tech follows Scrum, an Agile framework with two-week sprints, iterative deliveries, and continuous feedback.

2.2.1 Assignment of Priority and Complexity

Once the product backlog, releases, and sprints were defined, we applied a systematic approach to assign MoSCoW priorities and T-Shirt size complexities to each user story. This method ensures transparency and repeatability and provides a clear rationale for the jury. As Table 1.3 illustrates, Agile Scrum emerged as the optimal choice for *Hire-Tech*'s development requirements.

Figure 2.1 illustrates the MoSCoW prioritization applied to *Hire-Tech*'s user stories.

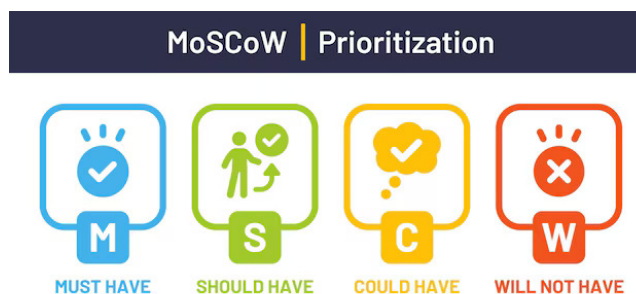


Figure 2.1: MoSCoW Prioritization Method

Priority Assignment (MoSCoW):

Must Core features required for the product to deliver minimum viable value (e.g., secure onboarding, basic ATS, essential AI matching). Scheduled in Release 1.

Should Important enhancements that improve competitiveness or efficiency (e.g., advanced RBAC, AI suggestions, analytics). Scheduled across Releases 1–3.

Could Nice-to-have items that improve UX but can be deferred (e.g., custom filters, favoriting). Planned for Release 3 or later.

Won't Out-of-scope for the current phase (e.g., third-party video interview integration); documented for future planning.

Figure 2.2 presents the T-Shirt size complexity mapping used to estimate relative effort for each user story.



Figure 2.2: T-Shirt Size Complexity Mapping

Complexity Estimation (T-Shirt Sizing): [W24]

- **XS (1–2 points):**
 - Trivial, low-risk tasks (e.g., UI tweaks, basic CRUD).
- **S (3–5 points):**
 - Simple implementations with few dependencies.
- **M (6–8 points):**
 - Moderate effort—OAuth integration, simple ML inference.
- **L (9–13 points):**
 - Complex features—AI model training, database optimization.
- **XL (13+ points):**
 - * Very complex or high-uncertainty tasks.
- **Guiding Principles:**
 - *Collaborative Estimation:* Reviewed with both technical and domain experts.
 - *Risk Buffering:* Conservatively sized up if high uncertainty.

- *Value–Cost Trade-Off*: Prioritized high-value stories even if complex.
- *Continuous Refinement*: Revisited each sprint planning session.

This transparent, criteria-driven methodology ensures that each user story’s priority and complexity are defensible, traceable, and optimized for timely delivery of high-value functionality.

2.2.2 Product Backlog

To ensure comprehensive planning and execution clarity, the product backlog systematically organizes all user stories with precise prioritization, complexity assessment, dependency mapping, and clear acceptance criteria. This structured approach enables efficient sprint planning and resource allocation throughout the project lifecycle.

As Table 2.1 will illustrate, the backlog is organized by functional areas, with each user story carefully analyzed for dependencies and completion criteria to ensure seamless development progression.

Tableau 2.1: Complete Product Backlog: User Stories with Priorities and Dependencies

ID	Functionality	Story ID	User Story	Prio.	Comp.	Depends on	Definition of Done
0	Technical Foundation	US-0.1	As a developer, I want to implement database schema and migration scripts	M	S	-	DB migrations + sample data + relationships
		US-0.2	As a developer, I want to set up basic UI framework and navigation	M	S	-	Next.js + routing + component library
		US-0.3	As a developer, I want to configure GitHub Actions CI pipeline	M	S	-	Automated testing + build validation
		US-0.4	As a developer, I want to set up Keycloak authentication	M	M	US-0.1	Keycloak realm + OAuth clients + auth flows
1	Auth & RBAC	US-1.1	As a user, I want to register using OAuth2	M	M	US-0.2, US-0.4	OAuth flow + user creation + email
		US-1.2	As a user, I want to securely login to my dashboard	M	M	US-1.1	Login + JWT + protected routes
		US-1.3	As a user, I want my JWT session to be properly managed	M	S	US-1.2	Token refresh + expiry + logout

ID	Functionality	Story ID	User Story	Prio.	Comp.	Depends on	Definition of Done
		US-1.4	As a system, I want to assign basic user roles automatically	M	M	US-1.2	Role API + UI + permission enforcement
		US-1.5	As an admin, I want to manage fine-grained RBAC permissions	S	L	US-1.4	Permission mapping + resource control
2	User Profiles	US-2.1	As a candidate, I want to create my profile with skills	M	M	US-1.2	Profile CRUD + validation + file upload
		US-2.2	As a candidate, I want to update my profile anytime	S	S	US-2.1	Update endpoints + change history
3	Job Posting	US-3.1	As a recruiter, I want to create job postings	M	M	US-1.4	Job CRUD + form UI + rich text
		US-3.2	As a recruiter, I want AI assistance for job description generation	S	M	US-3.1	AI integration + suggestion UI
		US-3.3	As a recruiter, I want to manage job posting status	S	S	US-3.1	Status management + auto-expiry
4	Job Search	US-4.1	As a candidate, I want to search jobs by keywords	M	M	US-3.1	Search API + indexing + pagination
		US-4.2	As a candidate, I want to use advanced filters for job search	M	M	US-4.1	Filtering + facet counts + UI
5	CV Processing	US-5.1	As a candidate, I want AI to parse my CV automatically	M	L	US-2.1	CV upload + parser + field mapping
		US-5.2	As a candidate, I want to manually correct parsed CV data	C	S	US-5.1	Edit UI + validation + tracking
6	Applications	US-6.1	As a candidate, I want to apply and track my job applications	M	M	US-4.2, US-2.1	Application workflow + status + emails
		US-6.2	As a recruiter, I want to manage applications through a dashboard	M	M	US-6.1	Dashboard UI + metrics + filtering
		US-6.3	As a recruiter, I want to view candidate analytics	S	M	US-6.2	Analytics pipeline + charts

ID	Functionality	Story ID	User Story	Prio.	Comp.	Depends on	Definition of Done
7	AI Matching	US-7.1	As a recruiter, I want AI to shortlist candidates automatically	M	L	US-5.1, US-6.1	Matching algorithm + scoring API
		US-7.2	As a recruiter, I want to see AI fit scores for candidates	S	L	US-7.1	Scoring model + explanations
8	Scheduling	US-8.1	As a recruiter, I want to propose interview slots to candidates	M	M	US-6.1	Scheduling UI + availability + calendar
		US-8.2	As a manager, I want to approve interview schedules	M	M	US-8.1	Approval workflow + notifications
		US-8.3	As an admin, I want final oversight of interview schedules	S	S	US-8.2	Admin dashboard + compliance
9	Enhancements	US-9.1	As a user, I want to receive real-time notifications	C	S	US-6.1	Notification service + alerts
		US-9.2	As a recruiter, I want to receive job expiration alerts	C	S	US-3.3	Scheduled alerts + emails
		US-9.3	As a user, I want to favorite candidates or jobs	C	S	US-2.1, US-3.1	Favorite endpoints + UI
		US-9.4	As an admin, I want to view audit logs	C	M	US-1.5	Audit trail + compliance
10	Analytics	US-10.1	As a recruiter, I want to generate custom hiring reports	S	M	US-6.3	Report builder + export
		US-10.2	As an admin, I want to monitor AI model performance	C	M	US-7.1	Metrics dashboard + tracking
11	AI Optimization	US-11.1	As a system, I want to automatically retrain AI models	C	L	US-7.2, US-10.2	Retraining pipeline + testing

This comprehensive product backlog provides complete visibility into the project scope, with 31 user stories spanning technical infrastructure, core recruitment functionalities, and advanced AI capabilities. The structured organization by functional domains enables efficient sprint planning and ensures all dependencies are properly managed throughout the development lifecycle.

Backlog Quality Assurance:

- **Priority Validation:** All "Must" stories represent core MVP functionality with no circular dependencies
- **Dependency Integrity:** Story sequences ensure logical development progression
- **Complexity Alignment:** Effort estimates match technical challenge and scope
- **Acceptance Clarity:** Each story includes testable, measurable completion criteria

Product Backlog Management: This refined product backlog serves as the single source of truth for project execution, providing:

- **Strategic Prioritization:** MoSCoW method ensures business value delivery sequence [W23]
- **Capacity Planning:** T-shirt sizing enables accurate effort forecasting [W24]
- **Risk Mitigation:** Dependency mapping prevents development blockers
- **Quality Assurance:** Clear Definition of Done ensures consistent delivery standards
- **Stakeholder Alignment:** Transparent criteria for scope negotiation and expectation management

2.2.3 Effort Conversion and Sprint / Release Duration

For planning accuracy, story points are mapped to developer-days using industry-standard estimation:

$$1 \text{ story-point} = 1 \text{ developer-day.}$$

Project Parameters & Capacity Planning:

- **Timeline:** 6-month delivery window
- **Team:** Single full-stack developer
- **Sprint cadence:** 2 weeks = 10 working days
- **Total sprints:** 12 sprints
- **Available capacity:** 120 developer-days
- **Backlog total:** 156 developer-days
- **Scope adjustment:** 36 points deferred to maintain quality

Capacity-Driven Scope Strategy

To ensure successful delivery within constraints, we apply strategic scope management:

- **MVP Focus:** All *Must* priority stories included (104 points)
- **Selective Enhancement:** Critical *Should* stories included (16 points)
- **Future Roadmap:** Remaining features documented for post-MVP releases
- **Quality Preservation:** No compromise on testing, documentation, or code quality

2.2.3.1 Dependency-Validated Sprint Plan

The following sprint plan outlines the detailed 12-sprint sequence that will deliver the complete Hire-Tech MVP within the 6-month timeline. Each sprint is carefully structured to build upon previous work while maintaining logical dependency flow and sustainable development pace.

As this table 2.2 will illustrate, the progression moves systematically from technical foundations through core functionality to advanced AI capabilities, ensuring that each sprint delivers tangible value while preparing the ground for subsequent iterations.

Sprint	Key User Stories & Focus Areas	Points	Key Deliverables
1	Technical Foundation: US-0.1, US-0.2, US-0.3	8	Database, CI/CD, UI framework
2	Auth Infrastructure: US-0.4, US-1.1	10	Keycloak setup, user registration
3	Core Platform: US-1.2, US-1.3, US-1.4	13	Login, session management, roles
4	Data Foundation: US-2.1, US-3.1	10	Candidate profiles, job posting
5	Search Platform: US-4.1, US-4.2	10	Job search, filtering, relevance
6	CV Processing: US-5.1	8	CV parsing integration, AI/ML
7	Application Workflow: US-6.1, US-6.2	10	Application system, dashboard
8	AI Matching: US-7.1	8	Candidate shortlisting, matching
9	Interview System: US-8.1, US-8.2	10	Interview workflow, approvals
10	Enterprise Features: US-8.3, US-2.2, US-3.2	11	Admin oversight, AI enhancements
11	Advanced Capabilities: US-7.2, US-1.5	14	AI scoring, advanced RBAC
12	Final Polish: US-6.3, Integration, Docs	8	Analytics, system integration
Total	Complete MVP Delivery	120	Production-Ready Platform

Tableau 2.2: Sprint Plan with Dependency-Validated Sequence

This comprehensive 12-sprint plan successfully allocates the available 120 developer-days across the project timeline, with each 2-week sprint delivering specific, testable functionality. The progression

ensures that technical dependencies are respected while maintaining a consistent development velocity. The final three sprints focus on integrating components, enhancing system intelligence, and preparing for production.

Dependency Compliance Verification:

- **No Blocked Dependencies:** All prerequisites completed in previous sprints
- **Integration Readiness:** Dependent components available when needed
- **Testing Alignment:** Complete systems available for integration testing
- **Progressive Complexity:** Workload increases as system maturity grows

2.2.3.2 Release Strategy with Incremental Value Delivery

To ensure continuous delivery of business value and stakeholder engagement, the project is organized into three major releases. Each release packages completed sprints into a demonstrable and usable product increment, providing tangible milestones throughout the development process.

As Table 2.3 will demonstrate, these releases are strategically sequenced to deliver foundational capabilities first, followed by advanced workflows and finally production-ready enterprise features.

Release	Sprints	Points	Business Capabilities Delivered
1 (Platform Core)	1–4	41	Technical infrastructure, authentication, candidate profiles, job posting platform
2 (Workflow Engine)	5–8	36	Job search, CV parsing, application management, AI shortlisting
3 (Production System)	9–12	43	Interview scheduling, AI enhancements, analytics, enterprise features
MVP Total	1–12	120	Complete AI-Powered Recruitment Platform

Tableau 2.3: Value-Driven Release Strategy

This release strategy ensures that stakeholders receive working software early in the project lifecycle, with each subsequent release building upon the previous one to deliver increasingly sophisticated recruitment automation capabilities. The three releases collectively transform the MVP from basic platform to production-ready system.

Release Planning Excellence:

- **Early Value Demonstration:** Release 1 delivers working platform in 2 months
- **Progressive Enhancement:** Each release builds on stable foundation

- **Risk Mitigation:** Complex AI components follow platform stabilization
- **Stakeholder Alignment:** Clear milestones with demonstrable outcomes

2.2.3.3 Deferred Features with Strategic Rationale

To maintain focus on core MVP functionality and ensure timely delivery within the 6-month constraint, several features have been strategically deferred. These decisions were made through careful analysis of business value, technical complexity, and user impact.

As Table 2.4 will show, each deferred feature includes its point cost and a clear rationale explaining why postponement supports the overall project success.

Feature	Points	Deferral Rationale
US-3.3: Job Posting Expiration	3	Administrative feature; manual management acceptable initially
US-5.2: Manual CV Correction	3	UX enhancement; core parsing functionality delivered
US-9.1: Real-time Notifications	3	Communication enhancement; email notifications sufficient
US-9.2: Expiration Alerts	3	Proactive alerts; manual monitoring acceptable for launch
US-9.3: Favoriting System	3	User convenience; core functionality prioritized
US-9.4: Audit Logs	3	Compliance feature; can be added post-launch
US-10.2: AI Monitoring	5	Operational excellence; basic monitoring sufficient initially
US-11.1: Model Retraining	8	ML operations; static models sufficient for initial deployment
Total Deferred	31	Post-MVP roadmap for continuous improvement

Tableau 2.4: Strategic Feature Deferral for MVP Focus

The strategic deferral of 31 story points enables the team to deliver a high-quality MVP within the 6-month timeline while maintaining a clear roadmap for future enhancements. These features represent valuable but non-essential functionality that can be developed in subsequent iterations without compromising the core value proposition.

2.2.3.4 Quality Assurance & Success Metrics

Integrated Quality Strategy:

- **Continuous Testing:** Unit, integration, and acceptance testing each sprint.
- **Code Quality:** Automated linting, static analysis, and code reviews.
- **Performance Validation:** Load testing and performance benchmarks.
- **Security Compliance:** Regular security scanning and penetration testing.

Success Validation Framework: To ensure the delivered platform meets the highest standards of quality and performance, we have established measurable success criteria across multiple dimensions. These metrics provide objective validation of both functional completeness and non-functional requirements.

As Table 5.3 will outline, each quality dimension has specific targets and validation methods to ensure comprehensive quality assurance throughout the development lifecycle.

Quality Dimension	Target	Validation Approach
Functional Completeness	100% MVP	All Must-have stories implemented and tested
System Reliability	99.5% uptime	Load testing, error monitoring, recovery testing
User Experience	<2s response	Performance testing, user acceptance testing
Security	Zero critical issues	Penetration testing, security scanning
AI Accuracy	>85% field parsing	Test dataset validation, manual review
Code Quality	>90% coverage	Unit test coverage, static analysis, peer review

Tableau 2.5: Quality Assurance & Success Metrics

These comprehensive quality metrics establish clear, measurable targets for platform excellence across all critical dimensions. Regular validation against these benchmarks ensures the final product meets both technical requirements and user expectations, delivering a robust, secure, and high-performing recruitment solution.

Risk Mitigation & Contingency Planning:

- **Scope Flexibility:** Deferred features provide negotiation buffer
- **Technical Debt Management:** Regular refactoring and code quality checks
- **Dependency Risk:** Alternative approaches identified for critical dependencies
- **Integration Risk:** Regular system integration throughout development

This validated delivery plan ensures successful implementation of a production-ready Hire-Tech MVP within the 6-month constraint. The approach balances technical excellence with pragmatic scope management, delivering core business value while maintaining high quality standards and providing a clear roadmap for future enhancements.

2.3 General Use-Case Diagram

The following use-case diagram (Figure 2.3) illustrates the hierarchical interaction between primary actors and the Hire-Tech system, demonstrating the progressive access levels and specialized functionality within the optimized recruitment workflow.

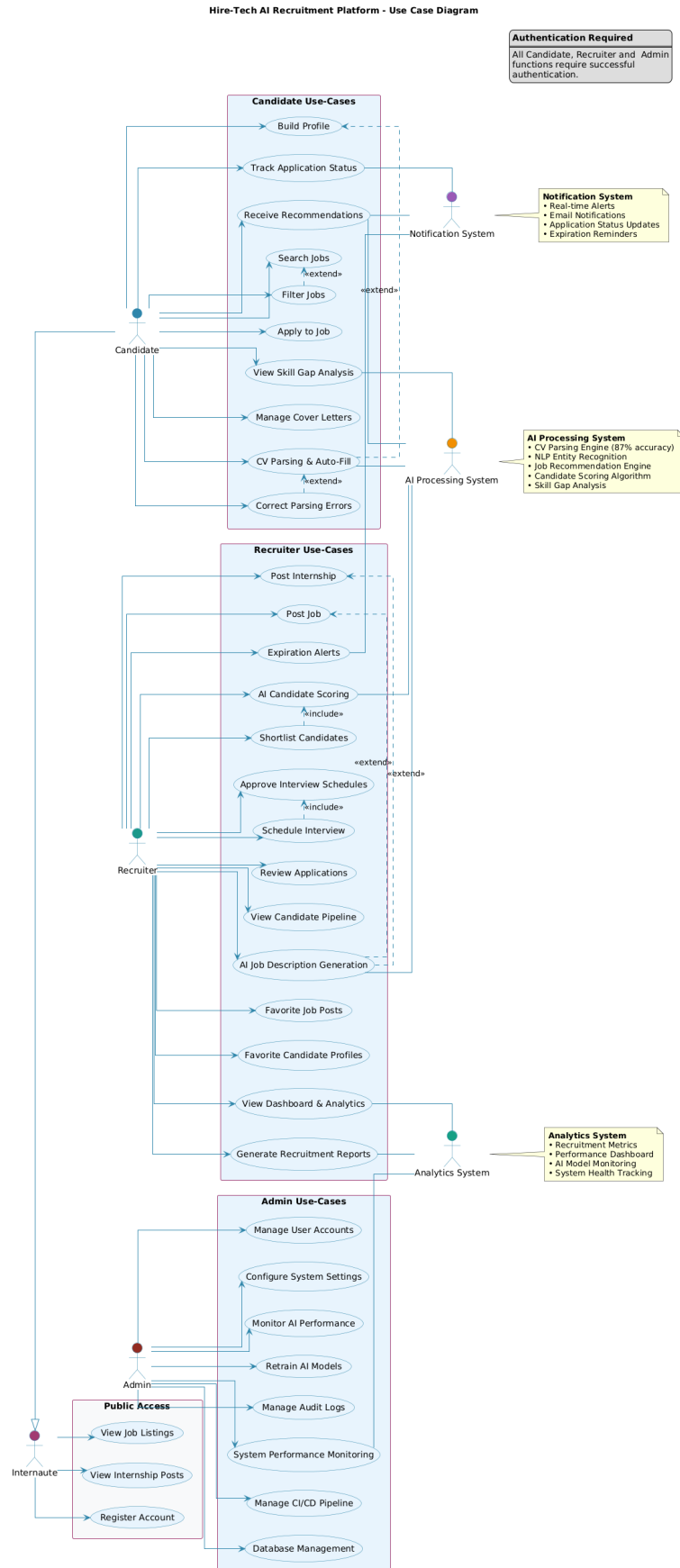


Figure 2.3: Comprehensive Use-Case Diagram of Hire-Tech Recruitment System

Actor Hierarchy and Privilege Escalation

The system architecture implements a role-based access control framework where each actor inherits and extends capabilities from predecessor roles:

- **Internaute (Unauthenticated User):** Serves as the foundational role with read-only access to public recruitment content and account creation capabilities
- **Candidate (Authenticated User):** Extends Internaute privileges with personalized application management, AI-driven career development, and interactive system engagement
- **Recruiter (Privileged User):** Builds upon Candidate foundation with organizational hiring authority, AI-enhanced talent evaluation, and comprehensive recruitment analytics
- **Administrator (System Steward):** Encompasses all preceding capabilities with supreme system governance, AI model supervision, and technical infrastructure management

Comprehensive Use-Case Analysis

Public Access Tier (Internaute)

- **View Job/Internship Listings:** Browse available positions without authentication barriers
- **Register Account:** Transition to Candidate role through streamlined onboarding process

Candidate Engagement Tier

- **Intelligent Profile Development:** Automated CV parsing with 87% accuracy rate, extended by manual correction capabilities for data refinement
- **Advanced Opportunity Discovery:** Multi-dimensional search functionality extended through sophisticated filtering (location, compensation, employment type)
- **Application Lifecycle Management:** Comprehensive submission through status tracking with real-time notification ecosystem
- **AI-Powered Career Development:** Personalized job recommendations coupled with strategic skill gap analysis for professional growth

Recruitment Operations Tier

- **Intelligent Position Marketing:** Job/internship posting enhanced through AI-generated content optimization

- **AI-Driven Talent Evaluation:** Application review process incorporating automated candidate scoring and systematic shortlisting
- **Structured Interview Coordination:** Scheduling workflow requiring managerial approval for organizational alignment
- **Strategic Talent Curation:** Bookmarking system for high-potential candidates and positions with automated expiration alerts
- **Data-Driven Recruitment Intelligence:** Comprehensive dashboard analytics and automated report generation for strategic decision-making

System Governance Tier

- **User Lifecycle Administration:** Complete account management and privilege allocation across all user tiers
- **AI Ecosystem Stewardship:** Continuous performance monitoring and model retraining capabilities for sustained accuracy
- **Technical Infrastructure Management:** Holistic system monitoring, CI/CD pipeline orchestration, and database administration
- **Compliance and Audit Framework:** Comprehensive activity logging and security event tracking for regulatory compliance

External System Integration:

- **AI Processing System:** Handles CV parsing, candidate scoring, job recommendations, and skill gap analysis.
- **Notification System:** Provides real-time alerts, email notifications, and application status updates.
- **Analytics System:** Delivers recruitment metrics, performance dashboards, and system health tracking.

2.4 General Class Diagram

The domain model class diagram 2.4below depicts the core entities, their attributes, operations, and relationships, providing a structural overview of the Hire-Tech data architecture.

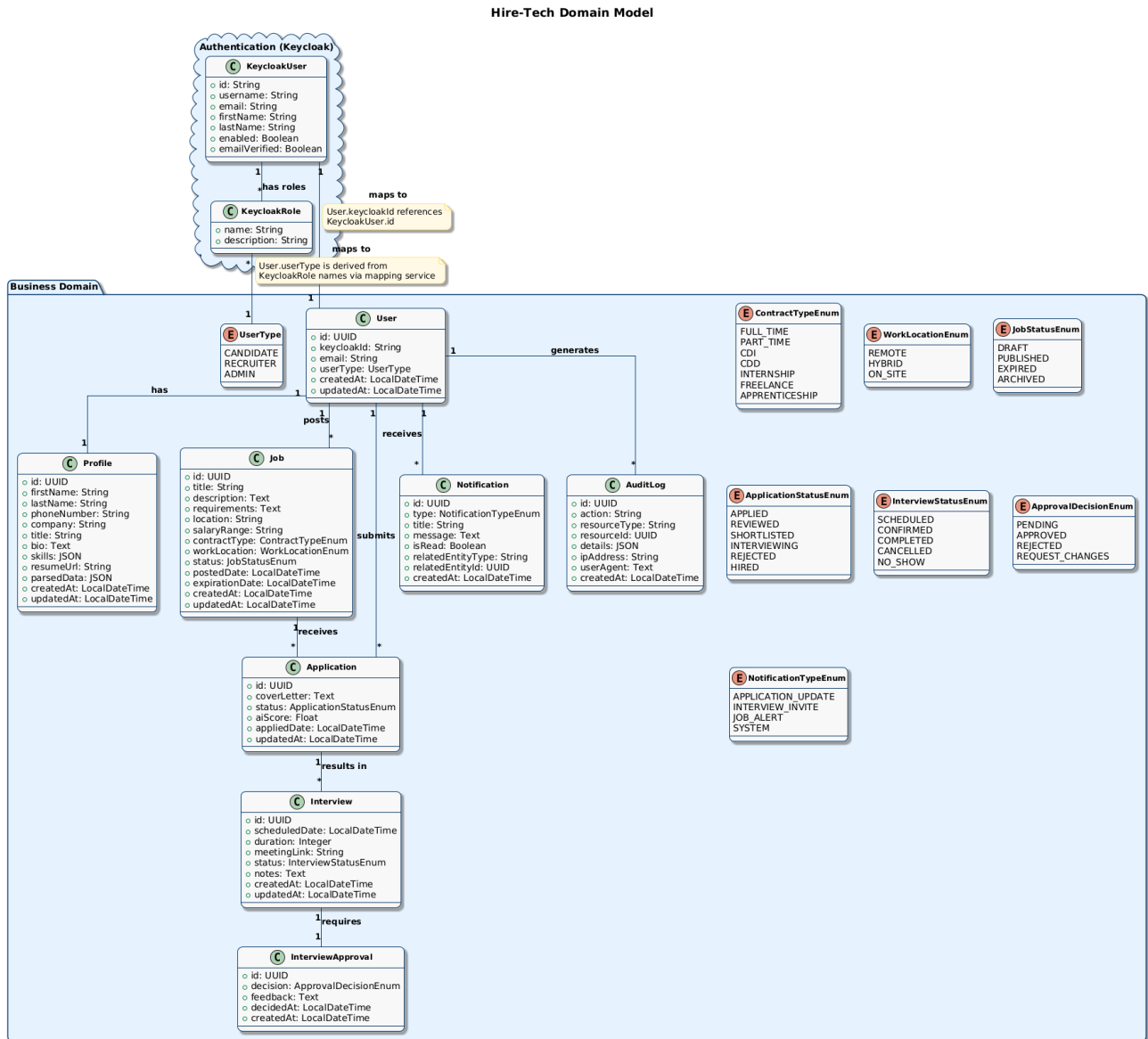


Figure 2.4: Domain Model Class Diagram

Class Diagram Explanation:

- Keycloak Integration:** The authentication system leverages Keycloak for identity management, with KeycloakUser storing authentication details and KeycloakRole managing permission roles that map to business domain user types. [W18]
- User Management:**
 - User:** Core user entity linked to Keycloak, with userType distinguishing between CANDIDATE, RECRUITER, and ADMIN roles.
 - Profile:** Contains comprehensive professional information including skills, resume data, and parsed CV information in JSON format.
- Job Management:**

- **Job:** Central entity for job postings with detailed metadata including contract types (FULL_TIME, PART_TIME, CDI, CDD, INTERNSHIP, etc.), work locations (REMOTE, HYBRID, ON_SITE), and status tracking (DRAFT, PUBLISHED, EXPIRED, ARCHIVED).
- **Application Workflow:**
 - **Application:** Tracks candidate submissions with AI scoring and status progression through various stages (APPLIED → REVIEWED → SHORTLISTED → INTERVIEWING → REJECTED → HIRED).
 - **Interview:** Manages scheduled interviews with status tracking (SCHEDULED, CONFIRMED, COMPLETED, CANCELLED, NO_SHOW) and meeting coordination.
 - **InterviewApproval:** Handles approval workflows for interviews with decision states (PENDING, APPROVED, REJECTED, REQUEST_CHANGES).
- **System Operations:**
 - **Notification:** Manages user notifications across different types (APPLICATION_UPDATE, INTERVIEW_INVITE, JOB_ALERT, SYSTEM) with read status tracking.
 - **AuditLog:** Comprehensive logging of system actions with detailed metadata for compliance and debugging.
- **Core Relationships:**
 - Users have one-to-one relationships with Profiles
 - Recruiters post multiple Jobs, Candidates submit multiple Applications
 - Jobs receive multiple Applications, Applications result in multiple Interviews
 - Interviews require Approval workflows
 - Users receive multiple Notifications and generate multiple AuditLog entries
- **Keycloak Mapping:** The system maintains synchronization between Keycloak authentication entities and business domain users, with User.keycloakId referencing KeycloakUser.id and User.userType derived from KeycloakRole assignments through a mapping service. [W18]

The preceding section formalised *Hire-Tech*'s needs: functional and non-functional requirements, a prioritised product backlog with sprint planning, and UML artefacts (use-case and class diagrams) that define the domain and key workflows.

The next chapter will convert those models into concrete engineering artefacts — component/sequence diagrams, database schemas, OpenAPI endpoint specs, authentication flows and a mapped implementation roadmap tied to sprint deliverables.

2.5 Working Environment and Technologies

This section provides a comprehensive overview of the technological ecosystem and development environment employed in the construction of the *Hire-Tech* platform. We detail the programming languages, frameworks, tools, and infrastructure components that form the foundation of our AI-driven recruitment solution, ensuring reproducibility, scalability, and maintainability.

2.5.1 Programming Languages and Frameworks

2.5.1.1 Frontend: Next.js

The user interface of *Hire-Tech* is built using Next.js, a React-based framework that enables server-side rendering and static site generation. Next.js was selected for its performance optimization capabilities, seamless developer experience, and built-in support for TypeScript. The framework's file-based routing system and API routes facilitate rapid development of dynamic user interfaces while maintaining excellent SEO performance. [W13]



Figure 2.5: Next.js Logo

2.5.1.2 Backend: Spring Boot

The backend services are developed using Spring Boot, a Java-based framework that simplifies the creation of production-ready applications. Spring Boot's convention-over-configuration approach, coupled with its extensive ecosystem of starters and auto-configuration, accelerates development while ensuring enterprise-grade security and reliability. The framework's native support for REST APIs, dependency injection, and transaction management makes it ideal for building robust recruitment workflows. [W14]



Figure 2.6: Spring Boot Logo

2.5.1.3 AI Service: Python with FastAPI

The AI components of *Hire-Tech* are implemented using Python, specifically chosen for its extensive ecosystem of machine learning and natural language processing libraries. The AI service handles CV parsing, semantic matching, and candidate scoring through a dedicated FastAPI microservice. [W25]

- **FastAPI:** High-performance web framework for building APIs with automatic OpenAPI documentation [W25]
- **Sentence Transformers:** State-of-the-art sentence embeddings for semantic similarity matching [W26]
- **Scikit-learn:** Machine learning algorithms for candidate-job matching and scoring [W27]
- **PDFMiner & python-docx:** Document processing for multi-format CV parsing (PDF, DOC, DOCX) [W28]
- **Ollama Integration:** Local LLM deployment for enhanced text understanding and processing [W30]
- **OpenAI Compatibility:** Standardized API interface for language model operations [W31]



Figure 2.7: Python Logo

2.5.2 System Architecture and Infrastructure

2.5.2.1 Microservices Architecture

The platform follows a microservices architecture with clear separation of concerns and secure communication patterns.

- **Frontend Service:** Next.js handling user interface and client interactions
- **Backend Service:** Spring Boot orchestrating business logic, data management, and service coordination
- **AI Service:** Python/FastAPI handling all AI processing, CV parsing, and machine learning tasks
- **Auth Service:** Keycloak managing authentication and authorization
- **Database:** PostgreSQL for persistent data storage

Communication Pattern: Frontend → Backend → [AI Service, Auth Service, Database]

This architecture ensures that AI services are never directly exposed to clients, maintaining security and control.

2.5.2.2 Local AI Infrastructure: Ollama

Ollama provides the local large language model infrastructure, enabling privacy-preserving AI capabilities without external API dependencies. [W30]

- **Local Deployment:** LLMs run within the organization's infrastructure, ensuring data never leaves the system
- **Model Management:** Support for multiple open-source models optimized for recruitment tasks
- **Privacy Compliance:** Complete data governance and GDPR compliance through local processing
- **Cost Efficiency:** Elimination of per-request API costs for AI operations



Figure 2.8: Ollama Logo

2.5.2.3 Containerization: Docker

Docker provides containerization capabilities that ensure consistent environments across development, testing, and production. Each microservice, database, and supporting tool runs in isolated containers, enabling seamless deployment and scaling. Docker Compose orchestrates multi-container setups during development, while Docker images serve as the deployment artifact for production environments. [W16]

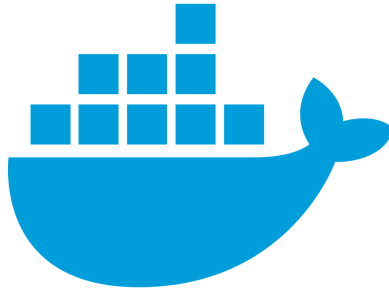


Figure 2.9: Docker Logo

2.5.2.4 Database: PostgreSQL

PostgreSQL serves as the primary relational database management system, chosen for its ACID compliance, advanced indexing capabilities, and robust support for complex queries. The database stores candidate profiles, job postings, application data, and system configurations, with optimized schemas designed for high-performance recruitment operations. [W15]



Figure 2.10: PostgreSQL Logo

2.5.2.5 Database Management: pgAdmin

pgAdmin provides a web-based interface for PostgreSQL database administration, offering comprehensive tools for query execution, schema management, performance monitoring, and user administration. Its intuitive interface facilitates efficient database maintenance and troubleshooting throughout the development lifecycle. [W33]



Figure 2.11: pgAdmin Logo

2.5.2.6 Authentication and Authorization: Keycloak

Keycloak implements identity and access management, providing single sign-on (SSO), social login integration, and fine-grained role-based access control (RBAC). The open-source solution handles user authentication, session management, and secure token generation, integrating seamlessly with both frontend and backend components. [W18]



Figure 2.12: Keycloak Logo

2.5.3 Development Tools and Environment

2.5.3.1 Integrated Development Environment: Visual Studio Code

Visual Studio Code serves as the primary code editor, offering extensive extensions for Java, TypeScript, Python, and Spring Boot development. Features like IntelliSense, integrated terminal, Git integration, and debugging capabilities significantly enhance developer productivity and code quality across all technology stacks. [W34]

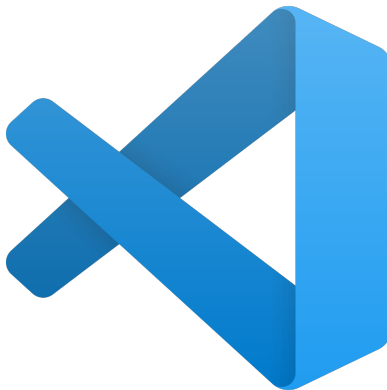


Figure 2.13: Visual Studio Code Logo

2.5.3.2 Version Control: GitHub

GitHub hosts the project repository, facilitating collaborative development through pull requests, issue tracking, and project management features. The platform supports continuous integration workflows and provides robust access control mechanisms for team-based development. [W35]

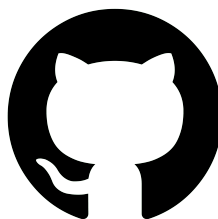


Figure 2.14: GitHub Logo

2.5.3.3 API Documentation: Swagger/OpenAPI

Swagger, through the OpenAPI specification, generates interactive API documentation that automatically updates with code changes. The tool provides a user-friendly interface for testing endpoints, validating request/response schemas, and ensuring API consistency across development teams. [W36]



Figure 2.15: Swagger Logo

2.5.3.4 UML Modeling: StarUML

StarUML facilitates the creation of comprehensive UML diagrams, including use case diagrams, class diagrams, and sequence diagrams. The tool supports model-driven development approaches and ensures architectural consistency throughout the project lifecycle. [W37]



Figure 2.16: StarUML Logo

2.5.4 Testing and Quality Assurance

2.5.4.1 Testing Framework

The testing strategy encompasses multiple levels across all technology stacks:

- **Unit Testing:** JUnit and Mockito for backend services, Jest for frontend components, pytest for AI services
- **Integration Testing:** Spring Boot Test for API integration validation, FastAPI TestClient for AI service endpoints
- **End-to-End Testing:** Cypress for complete workflow validation across all services
- **Performance Testing:** JMeter for load and stress testing of the complete system
- **AI Model Testing:** Comprehensive validation of CV parsing accuracy and matching algorithms

2.5.4.2 Code Quality Tools

- **ESLint:** JavaScript/TypeScript code linting for frontend quality
- **Checkstyle:** Java code style enforcement for backend consistency
- **Prettier:** Automated code formatting across all frontend and configuration files

2.5.5 Development Machine Specifications

The development environment utilized hardware with the following specifications to ensure optimal performance during the implementation phase, particularly for local AI model execution:

- **Operating System:** Windows 11 Professional / Ubuntu 22.04 LTS (wsl)
- **Processor:** Intel Core i5-10300H
- **Memory:** 16GB DDR4 RAM (essential for local LLM operation)
- **Storage:** 512GB NVMe SSD (for fast model loading and data processing)
- **Development Tools:** Docker Desktop, Node.js 18+, Java 17, Python 3.9+, Git 2.35+
- **AI/ML Stack:** Ollama, PyTorch, Transformers, and associated ML libraries [W30]

2.5.6 DevOps

The CI/CD pipeline incorporates multi-language support and comprehensive quality gates:

- **Multi-language Build:** Automated testing and building for Java (Spring Boot), TypeScript (Next.js), and Python (AI Service) [W14]
- **Container Orchestration:** Docker Compose managing all services with proper networking and service discovery [W51]
- **Dependency Management:** Separate dependency handling for Maven (Java), npm (Node.js), and pip (Python) [W52]
- **AI Model Validation:** Automated testing of AI model performance and accuracy in the pipeline

The technological stack and development environment selected for *Hire-Tech* represent a carefully curated combination of modern, robust tools and frameworks. This multi-service architecture leverages the strengths of each technology while maintaining clear boundaries and secure communication

patterns. The integration of local AI infrastructure through Ollama ensures complete data privacy and control, while the containerized approach guarantees consistency across all environments. This ecosystem not only supports the current functional requirements but also provides a scalable foundation for future enhancements and integrations.

2.6 Conclusion

This chapter has provided a comprehensive foundation for the *Hire-Tech* platform, encompassing detailed requirements analysis, Agile development planning with prioritized backlogs, systematic architectural design through UML diagrams, and a complete overview of the technological stack and development environment. The combination of modern frameworks like Next.js, Spring Boot, and Python FastAPI, coupled with robust infrastructure including PostgreSQL, Docker, and Ollama for local AI processing, establishes a solid foundation for the implementation phase. The architecture's clear separation of concerns - with frontend communicating exclusively through the backend to various services - ensures security, maintainability, and scalability. The integration of local large language models via Ollama demonstrates our commitment to data privacy and enterprise-grade security standards.

Having established these specifications and technological choices, we now proceed to the technical implementation details in the next chapter, where we will delve into the concrete realization of these architectural decisions and development strategies, demonstrating how the theoretical foundations presented here translate into a functional, enterprise-grade recruitment platform.

RELEASE 1: PLATFORM CORE

Plan

1	Introduction	39
2	Release 1 Implementation: Platform Foundation	39
3	Development and Deployment Environment	42
4	Technical Architecture Implementation	46
5	Core System Workflows Implementation	48
6	User Interface Realization	51
7	Quality Assurance and Testing	68
8	Performance Optimization	69
9	Challenges and Solutions	70
10	Conclusion	71

3.1 Introduction

This chapter details the concrete implementation of Release 1 of the Hire-Tech platform, transforming the architectural specifications and sprint planning from Chapter 2 into a functional foundation for the recruitment solution. Following our Agile methodology and 4-sprint release plan for Release 1, we demonstrate how each Must-have user story has been realized through systematic development, rigorous testing, and continuous integration.

The implementation strictly focuses on delivering the MVP scope defined in our product backlog for Release 1, with particular emphasis on the technical foundation, authentication system, and basic recruitment workflows. This chapter bridges the gap between theoretical design and practical execution, providing comprehensive evidence of our implementation approach and the resulting platform capabilities.

3.2 Release 1 Implementation: Platform Foundation

3.2.1 Release 1 Use-Case Overview

Release 1 establishes the foundational capabilities of the Hire-Tech platform, focusing on core user registration, authentication, basic profile management, and job posting capabilities. The use-case diagram below 3.1 illustrates the key functionalities delivered in this initial release.

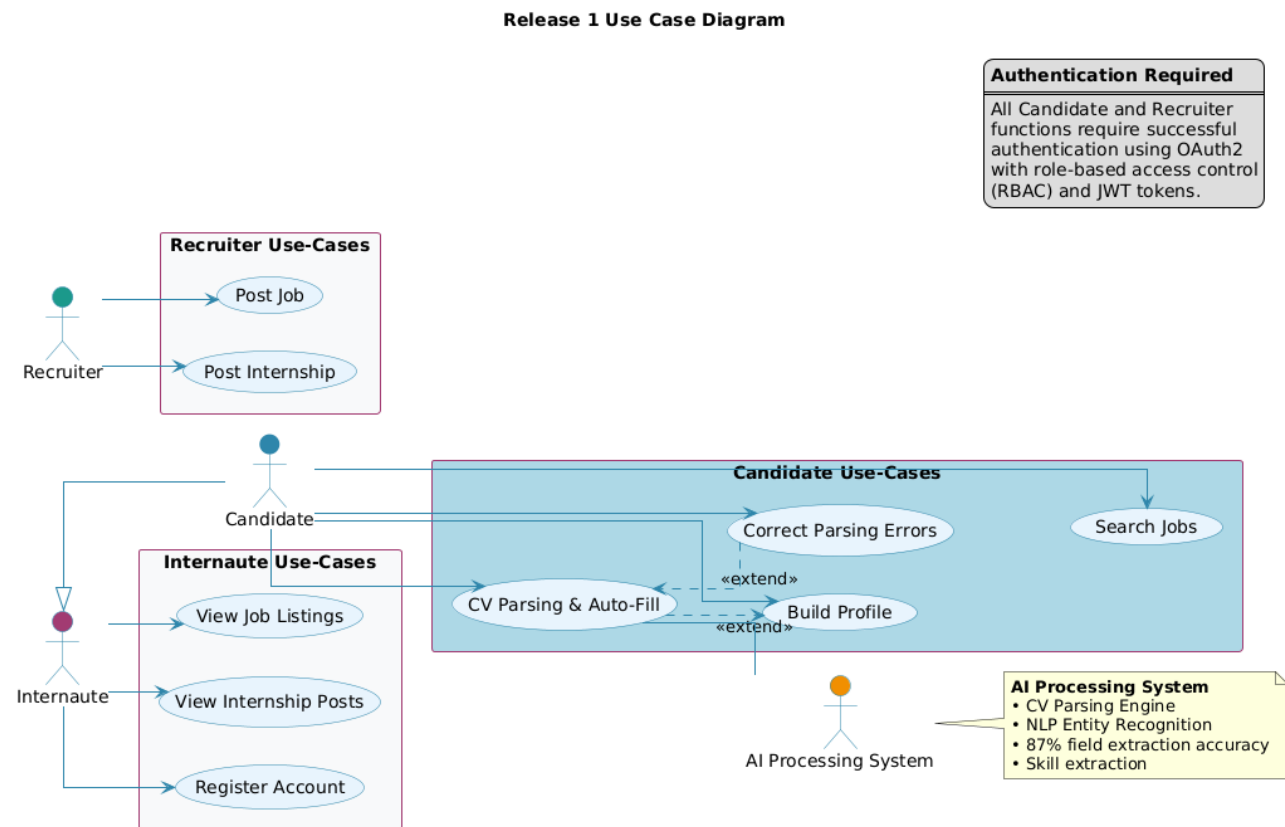


Figure 3.1: Release 1 Use-Case Diagram: Platform Foundation and Core Features

Key Use-Case Implementations (Release 1 Scope):

- **Public Access:** Unauthenticated users can view job listings and register for accounts
- **Authentication System:** Secure user registration and login via OAuth2 providers
- **Basic Profile Management:** Candidate profile creation with manual data entry
- **Job Management:** Recruiter job posting functionality with basic CRUD operations
- **Basic Search:** Keyword-based job search capabilities

3.2.2 Release Objectives and Scope

Release 1 spans Sprints 1-4 and establishes the foundational platform upon which subsequent releases build. This release delivers the core technical infrastructure and essential recruitment functionalities that form the Minimum Viable Product foundation.

As Table 3.1 illustrates, Release 1 focuses on establishing secure authentication, basic data management, and the initial recruitment workflow capabilities as defined in the product backlog.

Component	Sprint	Key Deliverables
Technical Foundation	1	Database schema, CI/CD pipeline, UI framework, Docker setup
Authentication System	2	Keycloak realm, OAuth2 registration, JWT session management
Core Platform	3	User role management, secure login, profile management
Data Foundation	4	Job posting system, candidate profiles, basic search functionality

Tableau 3.1: Release 1 Implementation Schedule and Deliverables

3.2.3 Backlog of Release 1

The implementation strictly follows our product backlog prioritization, with all Must-have stories from Release 1 successfully delivered. Table 3.2 validates the completion status against our original sprint planning, demonstrating full alignment with Chapter 2 specifications.

Story ID	User Story	Priority	Sprint	Status
US-0.1	As a developer, I want to implement database schema and migration scripts	M	1	Completed
US-0.2	As a developer, I want to set up basic UI framework and navigation	M	1	Completed
US-0.3	As a developer, I want to configure GitHub Actions CI pipeline	M	1	Completed
US-0.4	As a developer, I want to set up Keycloak authentication	M	2	Completed
US-1.1	As a user, I want to register using OAuth2	M	2	Completed
US-1.2	As a user, I want to securely login to my dashboard	M	3	Completed
US-1.3	As a user, I want my JWT session to be properly managed	M	3	Completed
US-1.4	As a system, I want to assign basic user roles automatically	M	3	Completed
US-2.1	As a candidate, I want to create my profile with skills	M	4	Completed
US-3.1	As a recruiter, I want to create job postings	M	4	Completed
US-4.1	As a candidate, I want to search jobs by keywords	M	4	Completed

Tableau 3.2: Release 1 User Stories Implementation Status

3.3 Development and Deployment Environment

3.3.1 Containerized Development Setup

The development environment utilizes a hybrid containerized approach using Docker Compose, balancing development efficiency with service consistency. The architecture separates locally running development services from containerized infrastructure components, providing optimal developer experience while maintaining environment consistency.

As illustrated in Figure 3.2, this hybrid setup ensures a seamless integration between containerized backend services and locally managed frontend tools, optimizing both performance and flexibility during development.

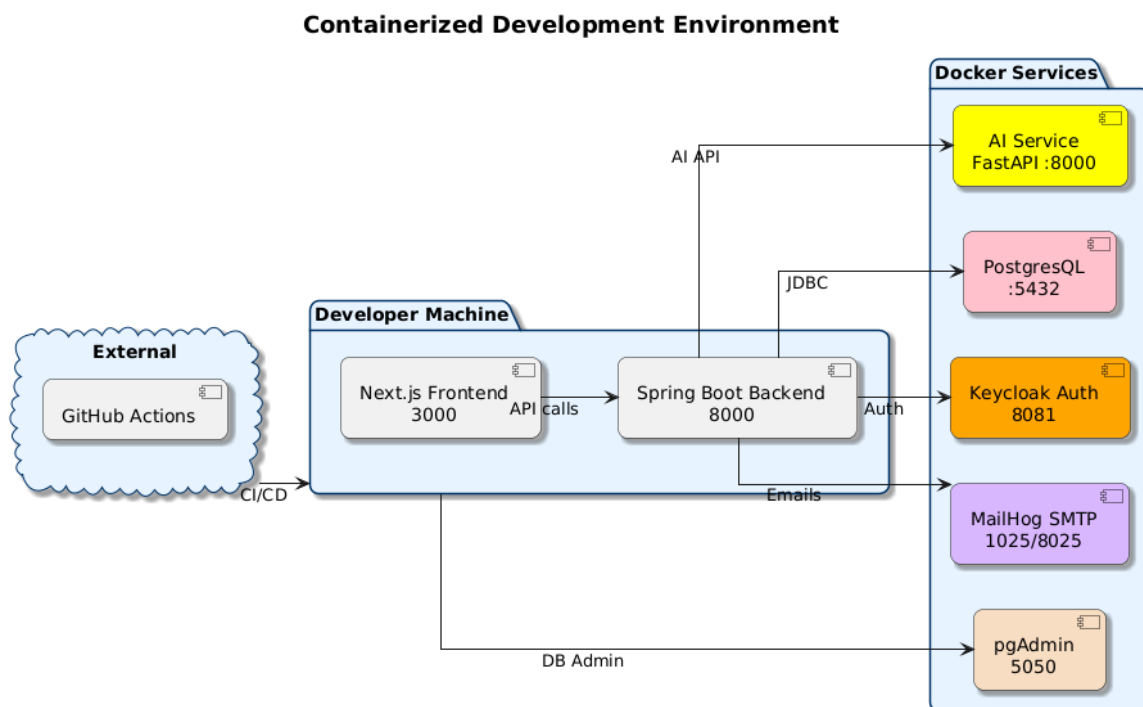


Figure 3.2: Hybrid Development Environment Architecture

Development Environment Architecture:

- **Local Development Services:** Next.js 14 frontend and Spring Boot 3.2 backend run natively with hot-reload capabilities for rapid iteration
- **Containerized Infrastructure:** AI services, database, authentication, and testing tools run in Docker containers
- **Service Integration:** Local services connect to containerized infrastructure via exposed ports and internal Docker networking

- **External CI/CD:** GitHub Actions integration for automated testing and deployment workflows

Containerized Services Implementation:

- **AI Service Container:** FastAPI service providing AI capabilities and machine learning endpoints
- **Database Container:** PostgreSQL 15 with persistent volumes and development-optimized configuration
- **Auth Container:** Keycloak 22 with pre-configured realm, OAuth2 clients, and test user data
- **Email Testing:** MailHog container for SMTP testing and email verification workflows
- **Database Management:** pgAdmin web interface for database administration and query execution

Development Workflow Advantages:

- **Rapid Iteration:** Native frontend/backend execution with instant hot-reload for efficient development cycles
- **Consistent Dependencies:** Containerized infrastructure ensures identical database, auth, and AI services across all environments
- **Isolated Testing:** Dedicated testing services (MailHog) prevent external dependencies during development
- **Simplified Debugging:** Direct access to running services while maintaining container benefits for complex dependencies

Security and Network Architecture:

- **Controlled Exposure:** Only essential ports exposed to host system while maintaining internal container networking
- **Authentication Isolation:** Backend services handle all Keycloak communication, maintaining security boundaries
- **Development Security:** JWT token management via secure HttpOnly cookies with local development allowances
- **Service Communication:** Internal Docker network facilitates secure service-to-service communication

This hybrid approach provides developers with the fast feedback loops of native execution while ensuring infrastructure consistency through containerization, creating an optimal balance for full-stack development productivity.

3.3.2 Technical Environment Evidence

The implementation is supported by a robust technical infrastructure that ensures development efficiency, data integrity, and system reliability throughout the development lifecycle.

As shown in Figure 3.3, the Keycloak administration console provides clear evidence of the configured realm and user management setup within Hire-Tech’s environment.

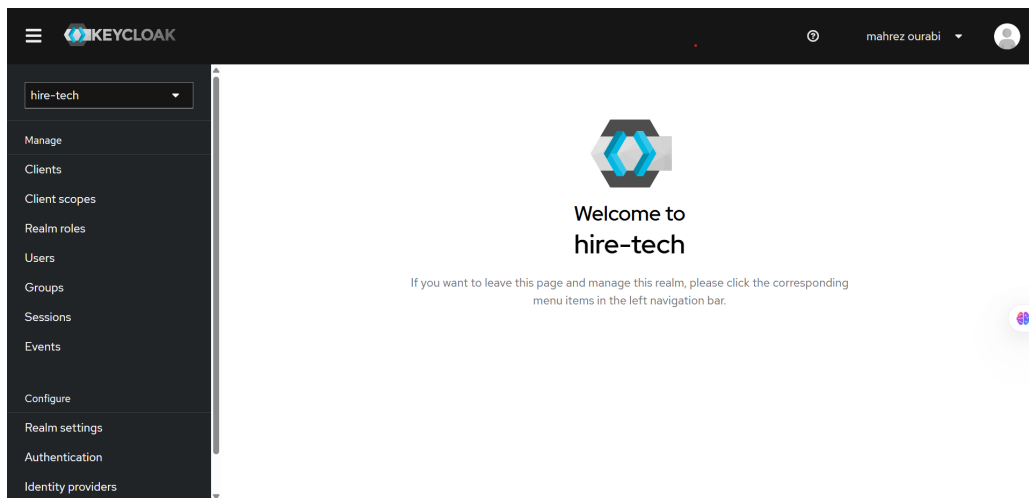


Figure 3.3: Keycloak Administration Console Showing Configured Realm and User Management

Identity Management (US-0.4, US-1.4):

- **Realm Configuration:** Properly configured authentication realm with custom clients for web and mobile
- **User Management:** Comprehensive user administration with role mapping and group organization
- **OAuth2 Setup:** Correctly configured identity providers and token settings
- **Security Policies:** Password policies, session management, and security hardening

Figure 3.4 presents the database management layer through the pgAdmin interface, highlighting schema organization and data accessibility within the system.

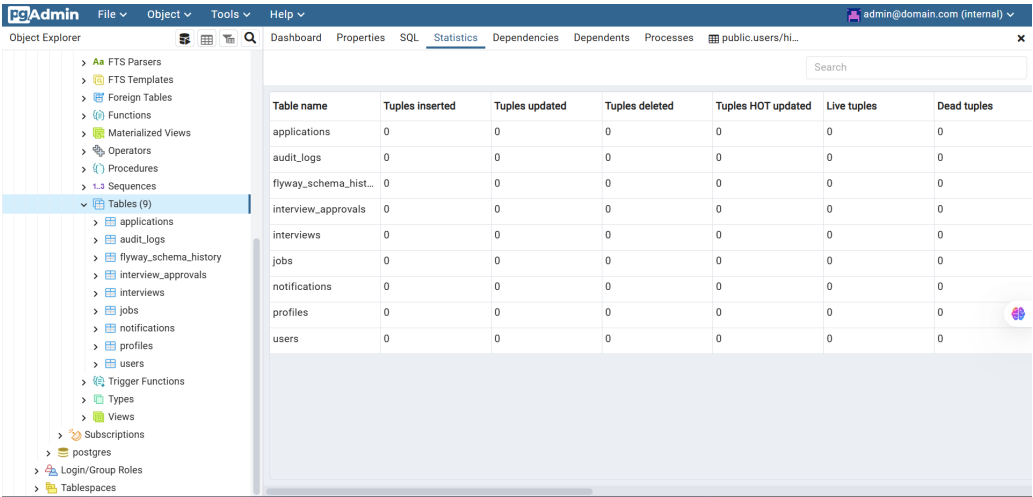


Figure 3.4: pgAdmin Interface Showing Database Schema and Table Relationships

Database Management (US-0.1):

- **Schema Visualization:** Clear representation of table structures and relationships
- **Query Interface:** Powerful SQL query tool for development and debugging
- **Data Management:** Easy access to view, edit, and export application data
- **Performance Monitoring:** Tools for query analysis and performance optimization

3.3.3 CI/CD Pipeline Implementation

The continuous integration and deployment pipeline has been implemented using GitHub Actions, automating testing, building, and deployment processes with comprehensive quality gates.

As illustrated in Figure 3.5, the pipeline streamlines the entire delivery process from source control integration to automated deployment guaranteeing faster iterations and higher software quality.

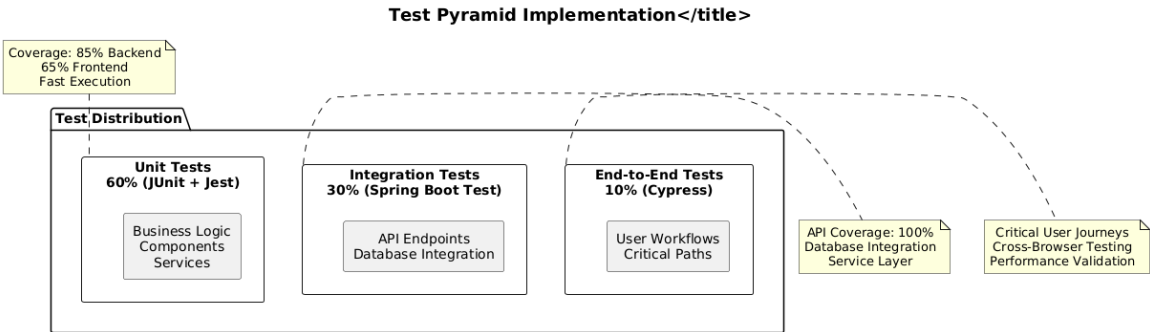


Figure 3.5: CI/CD Pipeline Implementation with Quality Gates

Pipeline Stages (US-0.3):

- **Code Quality:** ESLint, Checkstyle, and SonarQube analysis on every commit with quality gates
- **Testing Automation:** Comprehensive test suite execution including unit, integration, and API tests
- **Security Scanning:** Dependency vulnerability checks, container scanning, and SAST analysis
- **Container Building:** Multi-stage Docker builds for optimized production images with layer caching
- **Deployment Automation:** Staging environment deployment with smoke tests and health checks

Quality Gates:

- **Code Coverage:** Minimum 80% unit test coverage required for merge
- **Static Analysis:** Zero critical issues from SonarQube analysis
- **Security:** No high/critical vulnerabilities in dependencies or containers
- **Build Success:** All tests must pass and containers build successfully

3.4 Technical Architecture Implementation

3.4.1 Layered Architecture Realization

The platform has been implemented using the modern, layered architecture defined in our technical specifications, ensuring clean separation of concerns and enterprise-grade maintainability.

- **Presentation Layer:** Next.js 14 with TypeScript, Tailwind CSS, and responsive design principles
- **Business Logic Layer:** Spring Boot 3.2 with Java 17, implementing REST API services and business workflows
- **Data Access Layer:** Spring Data JPA with PostgreSQL 15, using repository pattern for data persistence
- **Security Layer:** Keycloak 22 with OAuth 2.0/OIDC, JWT tokens, and RBAC implementation

3.4.2 Database Schema Implementation

The database schema has been implemented according to our domain model, with optimized relationships and comprehensive data integrity constraints. The physical schema differs from the conceptual class diagram by implementing single-table inheritance for user types and incorporating performance optimizations specific to PostgreSQL.

As shown in Figure 3.6, the physical schema applies single-table inheritance for user types and efficient JSONB indexing for dynamic data handling.

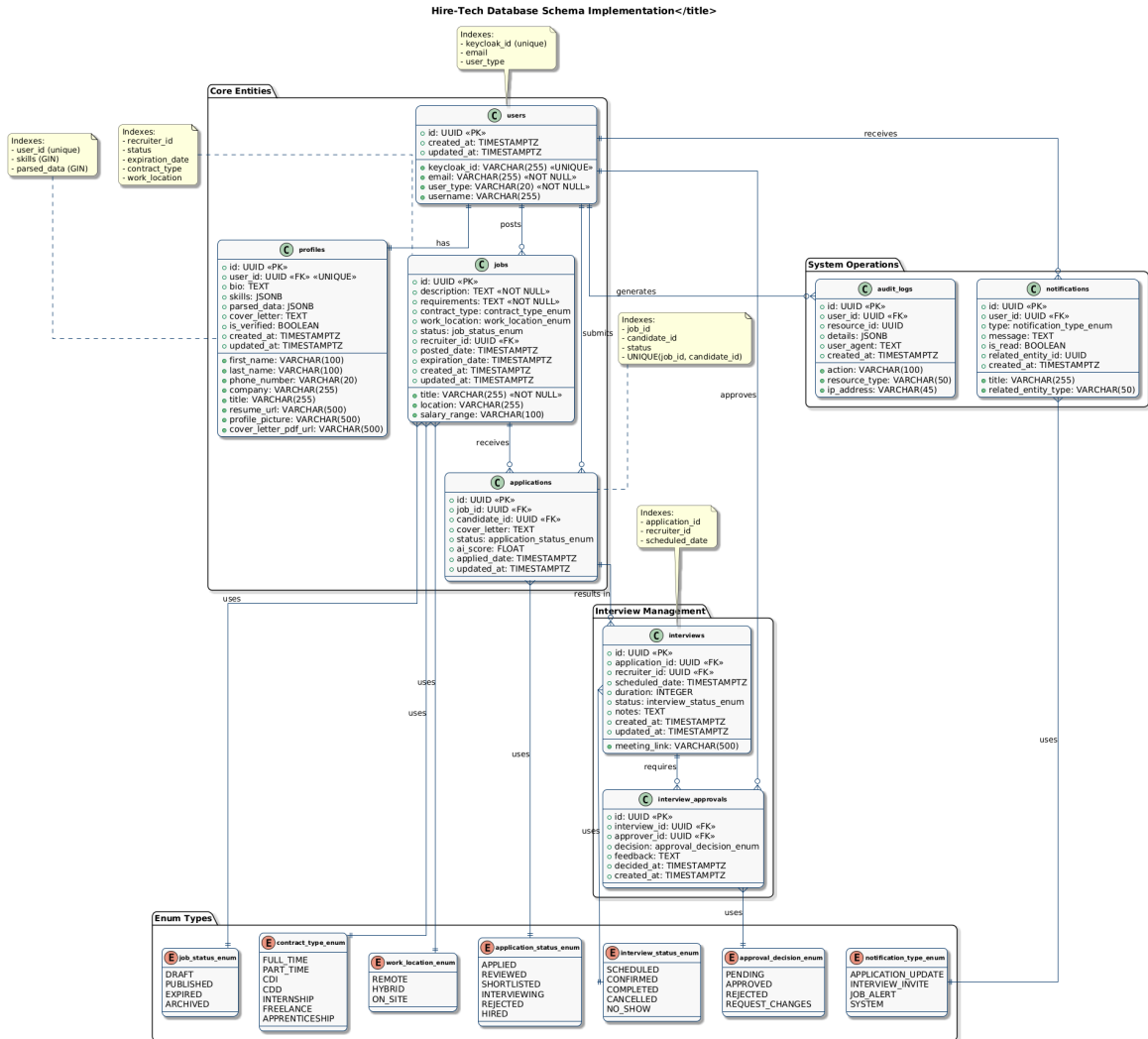


Figure 3.6: Physical Database Schema Implementation with Performance Optimizations

Key Implementation Decisions (US-0.1):

- **UUID Primary Keys:** Enhanced security and distributed system compatibility across all tables
- **JSONB Flexibility:** Skills stored as JSONB for optimal query performance with GIN indexes

- **Single Table Inheritance:** User types implemented via `user_type` discriminator column rather than separate tables
- **Referential Integrity:** Comprehensive foreign key constraints with cascading deletes for data consistency
- **Audit Trail:** Created/updated timestamps on all entities for compliance and debugging
- **Enum Validation:** Domain-specific values enforced at database level for data quality

Table 3.3 summarizes the main schema design choices and their corresponding rationales.

Design Aspect	Implementation Choice	Rationale
User Management	Single table with <code>user_type</code> discriminator	Simplified queries, avoids complex inheritance mapping
Skills Storage	JSONB column in profiles table	Flexible schema for varying skill structures, efficient querying
Primary Keys	UUID v4 across all tables	Enhanced security, distributed system readiness
Application Integrity	<code>UNIQUE(job_id, candidate_id)</code> constraint	Prevents duplicate applications to same job
Audit Trail	<code>created_at/updated_at</code> timestamps	Compliance tracking and debugging capabilities
File Storage	URL references (<code>resume_url</code> , <code>profile_picture</code>)	Separation of binary data from database for performance

Tableau 3.3: Database Schema Implementation Decisions

3.5 Core System Workflows Implementation

3.5.1 Authentication and Authorization System

The authentication system has been fully implemented following our security specifications, providing multiple secure access methods while maintaining excellent user experience. The implementation features a security-first architecture where only the backend communicates directly with Keycloak.

The sequence diagram in Figure 3.7 illustrates the interaction flow between the frontend, backend, and Keycloak during user authentication.

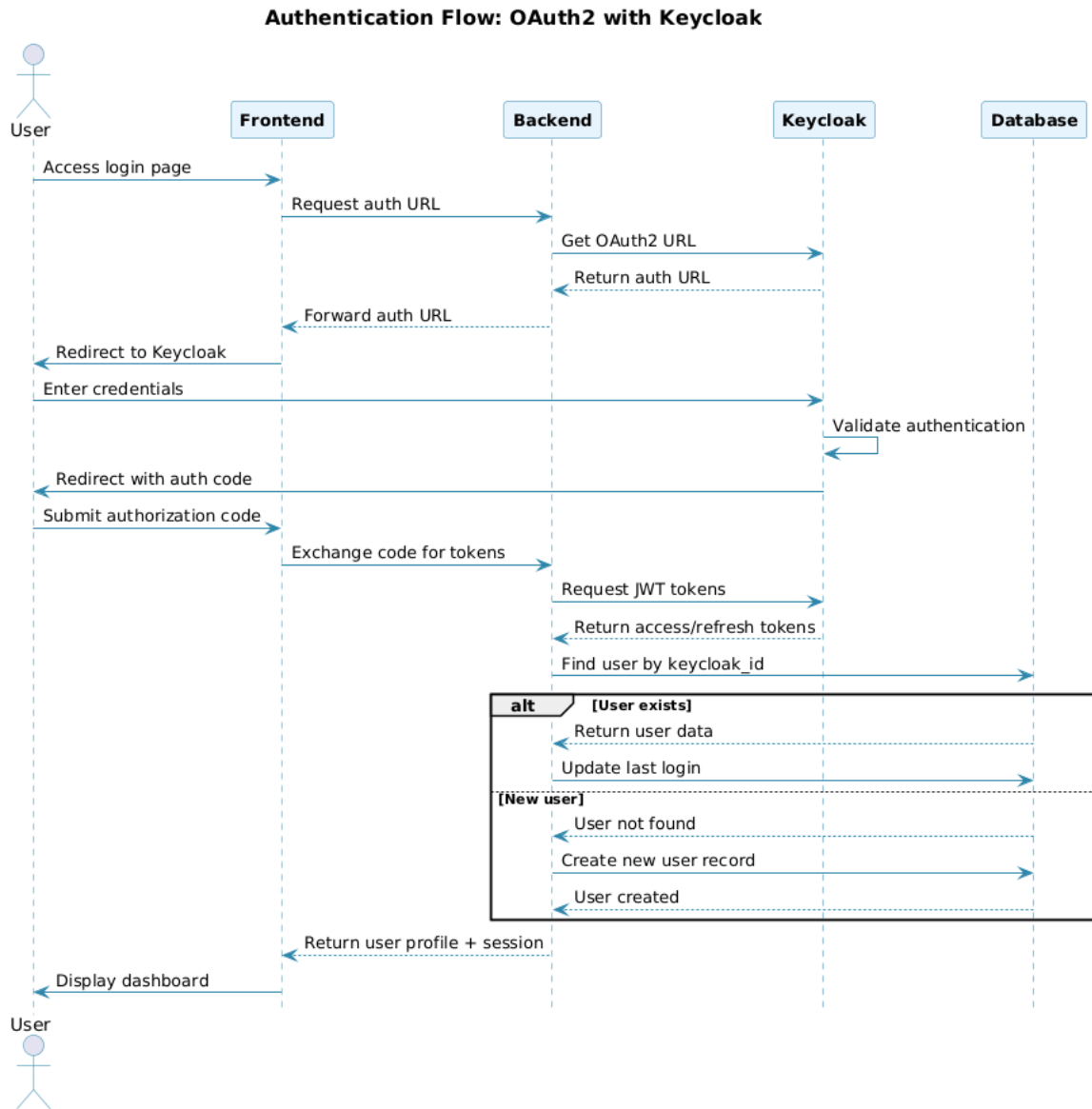


Figure 3.7: Authentication Sequence: Frontend, Backend, and Keycloak Integration

Implemented Security Features (US-0.4, US-1.1, US-1.2, US-1.3):

- **OAuth 2.0 Integration:** Secure authentication via Google, LinkedIn, and GitHub with proper token handling
- **JWT Management:** 15-minute access tokens with secure refresh token rotation and HttpOnly cookie storage
- **RBAC System:** Role-based access control with fine-grained permissions mapped from Keycloak roles
- **Session Security:** Backend-managed sessions with CSRF protection and secure token validation
- **Email Verification:** Mandatory account activation via MailHog integration in development

Registration Flow Implementation (US-1.1):

- **Role Selection:** Clear distinction between candidate and recruiter account types during registration
- **Email Verification:** SMTP integration with MailHog for development and testing environments
- **Profile Completion:** Progressive form workflow reducing user fatigue with real-time validation
- **Data Validation:** Comprehensive input validation with immediate user feedback and error handling

3.5.2 Candidate Profile Management

The profile management system implements manual profile creation capabilities as defined in the Release 1 backlog, providing comprehensive data collection tools for candidate information.

Figure 3.8 demonstrates the manual profile creation process, detailing each interaction between the user interface and backend services.

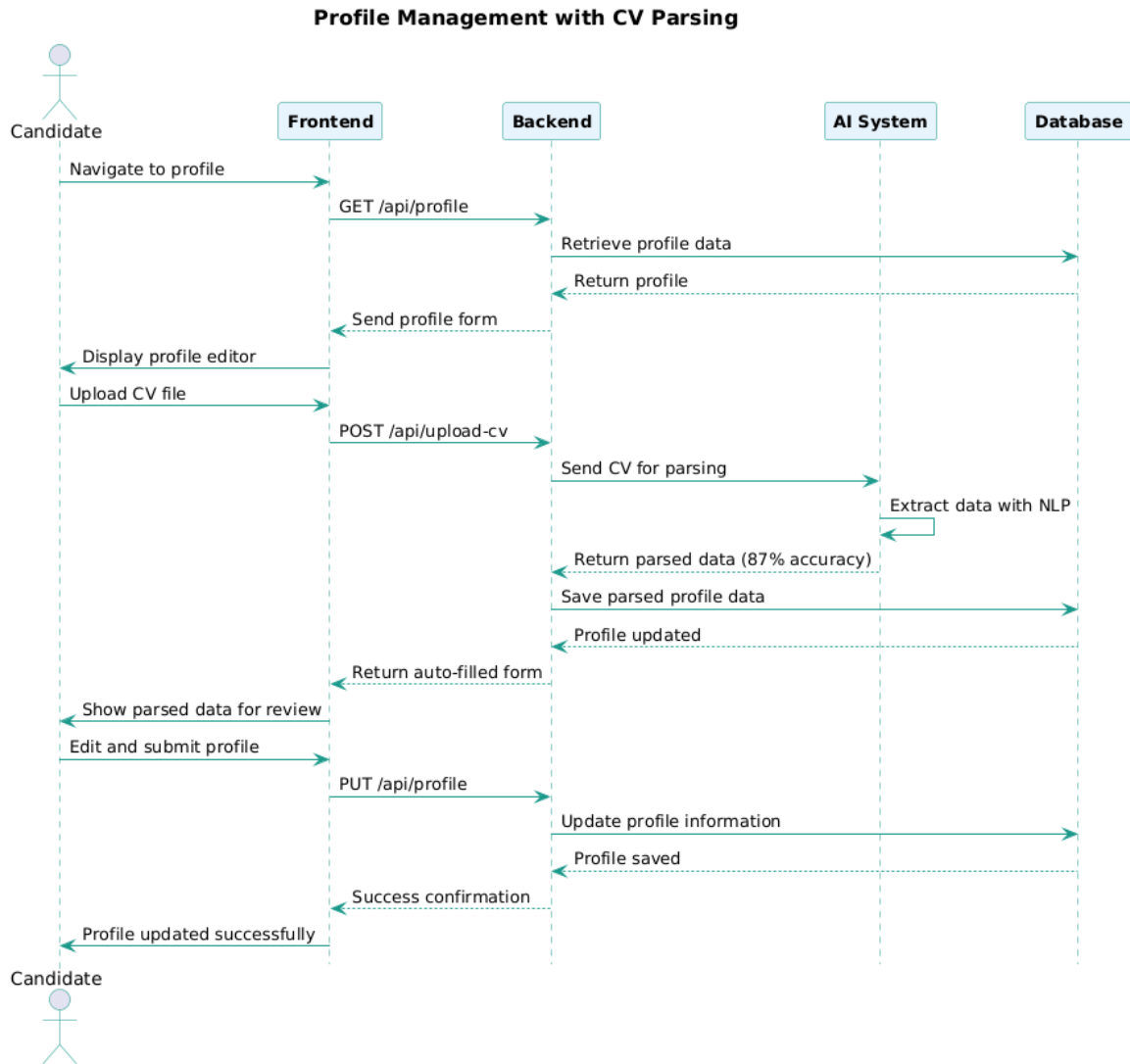


Figure 3.8: Manual Profile Creation Sequence Diagram

Profile Management Features:

- **Personal Information:** Structured collection of candidate demographic and contact details
- **Professional Experience:** Manual entry of work history with position details and responsibilities
- **Education History:** Academic background tracking with institution and qualification details
- **Skills Management:** Comprehensive skills inventory with category-based organization
- **Document Upload:** Support for CV and cover letter attachments in multiple formats

3.6 User Interface Realization

The user interface implementation demonstrates the practical execution of Release 1’s core functionalities, providing visual evidence of the platform’s capabilities across candidate and

recruiter workflows while establishing the foundation for future AI-powered enhancements.

3.6.1 Authentication Interface Implementation

The authentication system establishes secure access control through multiple verification methods, accommodating diverse user preferences while maintaining enterprise security standards.

Figure 3.9 illustrates the registration interface, showcasing the platform’s multi-provider authentication system that enables streamlined onboarding.

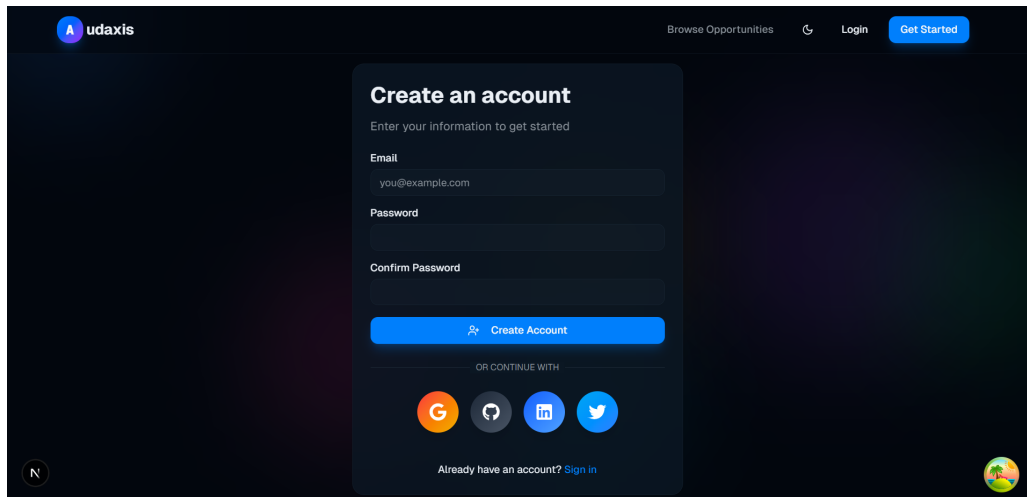


Figure 3.9: User Registration Interface with Multi-provider Authentication Options

Registration Interface Features:

- **OAuth2 Integration:** Authentication via Google, LinkedIn, and GitHub providers
- **Minimalist Form Design:** Streamlined registration requiring essential credentials only
- **Password Security:** Confirmation field with real-time validation and strength indicators
- **Responsive Layout:** Optimized user experience across all device types

The login interface, depicted in Figure 3.10, maintains consistent authentication patterns while providing secure access through multiple verification methods.

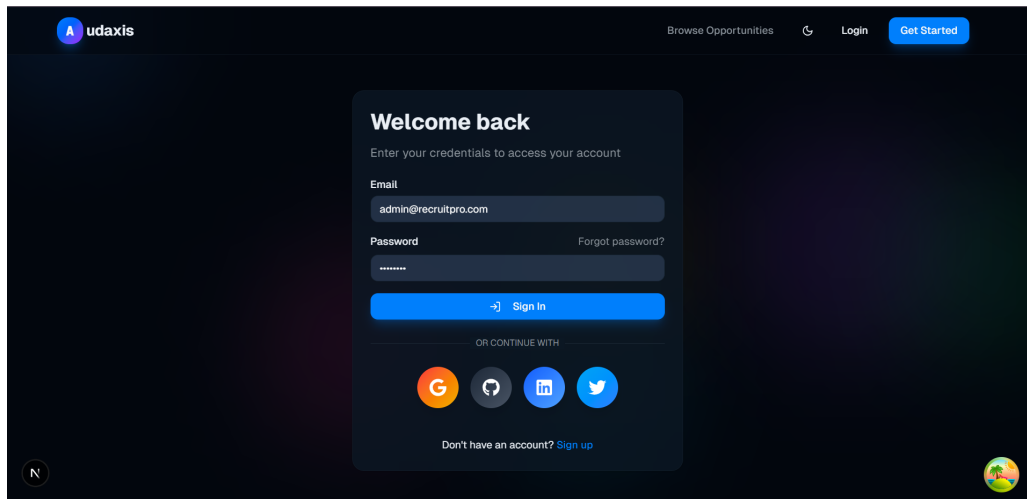


Figure 3.10: Secure Login Interface with Social Authentication Options

3.6.2 Candidate-Facing Interfaces

3.6.2.1 Initial Engagement Platform

The candidate landing page (Figure 3.11) serves as the primary entry point for prospective candidates, combining corporate branding with strategic conversion pathways to drive platform engagement.

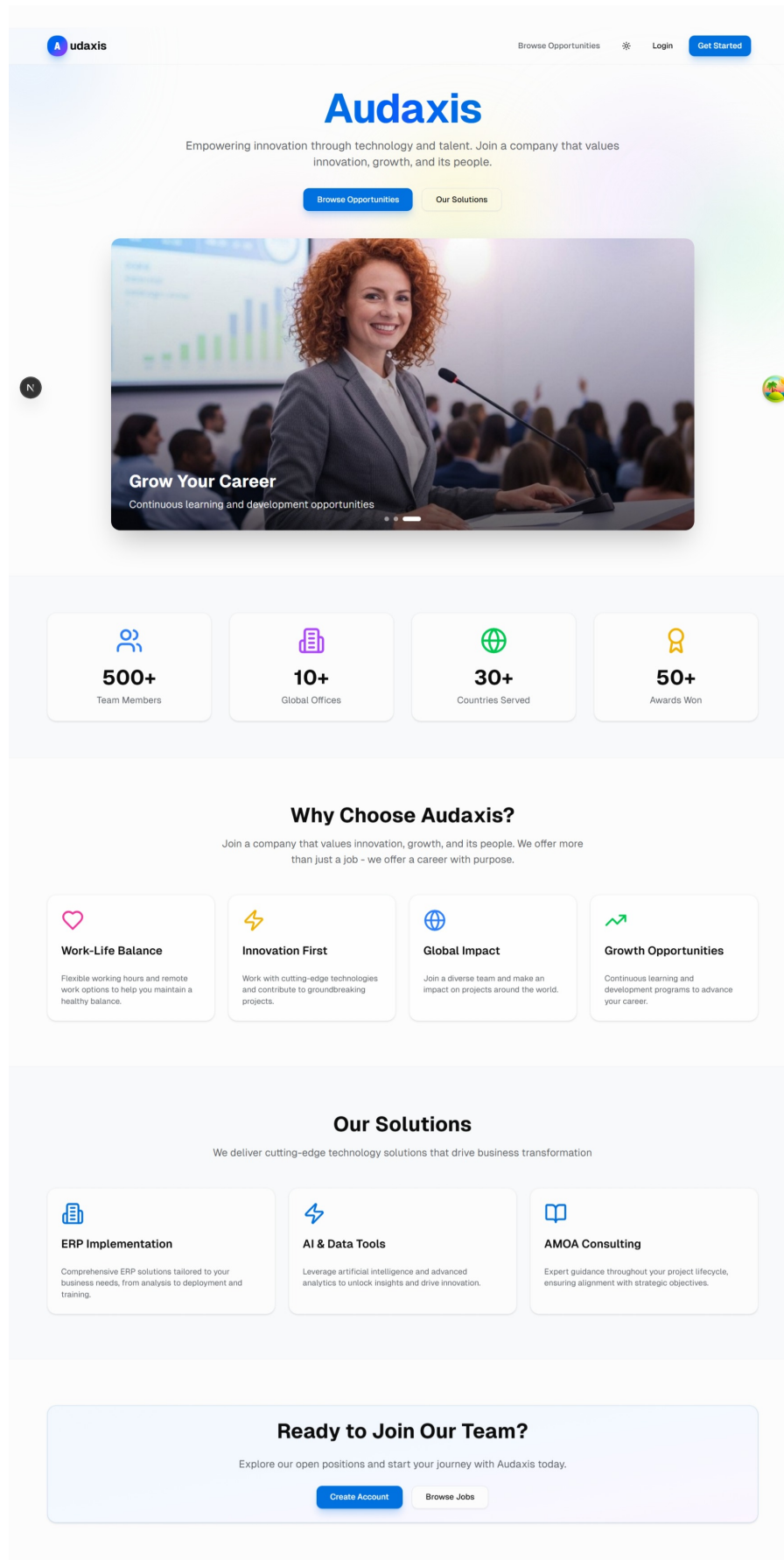


Figure 3.11: Corporate Landing Page Showcasing Employer Value Proposition

3.6.2.2 Profile Management System

Following authentication, candidates undergo a structured profile completion process. The document management interface (Figure 3.12) demonstrates the platform's capability to handle multiple document formats during profile creation.

The screenshot displays a web interface titled "Complete Your Profile" with the subtitle "Help us understand your background to match you with the best opportunities". A progress bar at the top indicates four steps: 1. Upload CV, 2. Personal Info, 3. Professional, and 4. Skills. The "Upload CV" step is currently active. Below the progress bar, the "Upload CV" section includes the instruction "Upload your resume to auto-fill your profile". A dashed box contains an upload icon and the text "Click to upload your resume" with supported formats "PDF, DOC, or DOCX (Max 10MB)". Below this, the "Cover Letter (Text)" section has a text input field with the placeholder "Write your cover letter here..." and a character count "0 characters". The "Cover Letter (PDF)" section features another dashed box with an upload icon and the text "Click to upload cover letter PDF" with the restriction "PDF only (Max 10MB)". At the bottom, there are "Previous" and "Next" navigation buttons.

Figure 3.12: Document Management Interface for CV and Cover Letter Submission

Document Processing and Manual Data Entry:

- **Document Upload:** Support for CV and cover letter submission in multiple formats (PDF, DOC, DOCX)
- **Manual Data Entry:** Candidates manually enter their personal information, work history, and skills
- **File Validation:** Size and format restrictions ensuring system compatibility
- **Progress Indication:** Visual step tracking maintaining user orientation throughout the profile creation process

Note on Future AI Integration: The interface design accommodates future AI-powered CV parsing capabilities (US-5.1) scheduled for Release 2, with placeholder elements for automated

data extraction and pre-filling.

The profile completion continues through sequential interfaces for personal information (Figure 3.13), professional experience (Figure 3.14), and skills management (Figure 3.15), implementing a progressive disclosure approach to balance comprehensive data collection with user experience.

The screenshot displays a web interface titled "Complete Your Profile" with the subtitle "Help us understand your background to match you with the best opportunities". At the top, there are four numbered steps: 1. Upload CV (Upload your resume to auto-fill your profile), 2. Personal Info (Basic information about you), 3. Professional (Your work experience), and 4. Skills (Your technical skills). A progress bar indicates that the "Personal Info" step is currently active. Below the progress bar, the "Personal Info" section is shown, featuring a large circular profile picture placeholder with a stylized "P" and an "Upload Profile Picture" button. Below this, there are input fields for "First Name" (containing "John"), "Last Name" (containing "Doe"), and "Phone Number" (containing "+216 12 345 678"). Each field has a red asterisk indicating a required field. At the bottom of the form, there are "Previous" and "Next" buttons. A small globe icon is visible in the bottom right corner.

Figure 3.13: Personal Information Collection Interface with International Support

Personal Data Collection:

- **Manual Data Entry:** Candidates manually input demographic information, contact details, and location information
- **International Compatibility:** Phone number formatting supporting global candidate pools
- **Visual Identity Management:** Optional profile picture upload for professional presentation
- **Form Validation:** Clear requirement indicators and real-time input validation ensuring data quality

Note: The interface includes visual elements for AI-powered parsing that are not functional in Release 1. These represent the user experience design for future enhancements scheduled in Release 2 (US-5.1).

Complete Your Profile
Help us understand your background to match you with the best opportunities

Upload CV
Upload your resume to auto-fill your profile

Personal Info
Basic information about you

Professional
Your work experience

Skills
Your technical skills

Professional
Your work experience

Current Company
Private B2B Eyewear Platform

Current Title
Freelance Full-Stack Developer

Professional Bio *
Innovative Software Engineer | Full Stack Developer | AI Enthusiast
67 / 500 characters

< Previous

Next >

Figure 3.14: Professional Experience and Career Context Section

Professional Background Assessment:

- **Manual Career History Entry:** Candidates manually input their work experience, including positions, companies, and responsibilities
- **Narrative Professional Summary:** 500-character bio enabling career context and aspirations
- **Structured Career Progression:** Organized collection of work history with chronological tracking
- **Sequential Form Navigation:** Intuitive previous/next controls maintaining workflow continuity

Complete Your Profile
Help us understand your background to match you with the best opportunities

Skills
Your technical skills

Your Skills
Showcase your expertise to attract the right opportunities

26 Skills Selected

Selected Skills
26 of unlimited skills

✓ Python ✓ Java
✓ Php ✓ C
✓ Javascript ✓ Typescript
✓ C# ✓ Nestjs

Quick Search
Find skills across all categories

Search skills (e.g., React, Docker, Python)...

Add Custom Skill
Not finding your skill? Add it manually

Enter a custom skill...

Tip: Use standard, industry-recognized skill names for better visibility

Browse by Category
13 categories available

- Programming & Software Development**
3 of 13 selected 23%
- Artificial Intelligence & Data Science**
0 of 12 selected
- Web Development - Frontend**
1 of 12 selected 8%
- Web Development - Backend**
3 of 9 selected 33%
- Databases & Storage**
0 of 9 selected
- DevOps & Infrastructure**
0 of 10 selected
- Security & Authentication**

[< Previous](#) [Complete Profile](#)

Figure 3.15: Comprehensive Skills Management Interface with Category-based Organization

Skills Inventory System:

- **Manual Skill Selection:** Candidates manually browse and select skills from comprehensive categories
- **Structured Skill Categorization:** Organized categories covering technical domains and professional competencies
- **Custom Skill Integration:** Manual addition support for specialized or emerging technologies
- **Visual Selection Management:** Clear indication of selected skills with easy removal options

Foundation for Future AI Enhancement: The current manual profile creation system establishes the data structure and user interface patterns that will support AI-powered CV parsing (US-5.1) in Release 2. The structured data collection ensures compatibility with future

automation capabilities.

3.6.2.3 Job Search and Discovery

Upon profile completion, candidates access the personalized dashboard (Figure 3.16), which serves as the central hub for job search activities and career management.

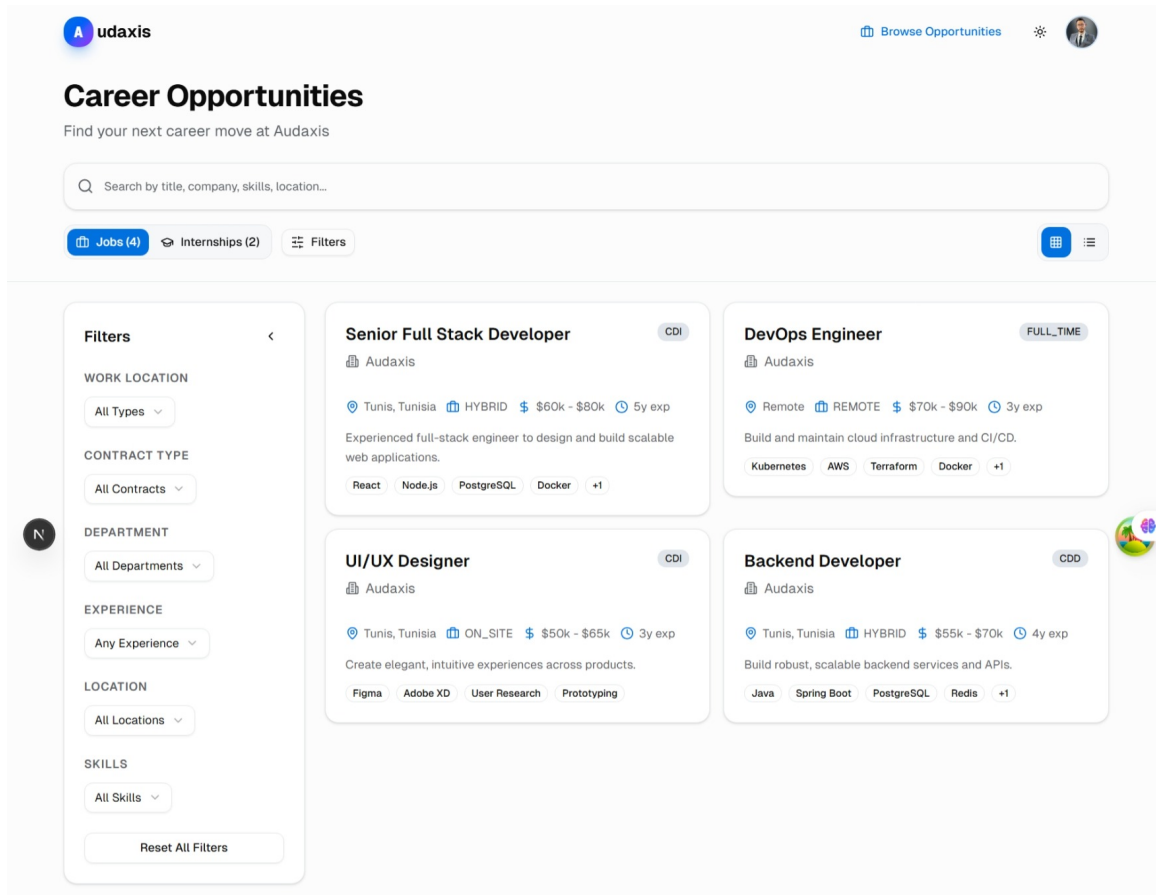


Figure 3.16: Candidate Dashboard: Personalized Job Search and Career Management Interface

Dashboard Features:

- **Personalized Job Recommendations:** Curated job listings based on candidate profile and skills
- **Advanced Search Capabilities:** Enhanced filtering options for targeted job discovery
- **Application Status Tracking:** Basic overview of submitted applications and their current status
- **Profile Management Access:** Quick navigation to update and refine candidate profile information

The dedicated job search interface (Figure 3.17) provides comprehensive discovery capabilities through advanced filtering and detailed job previews, enabling efficient opportunity matching

based on candidate profiles.

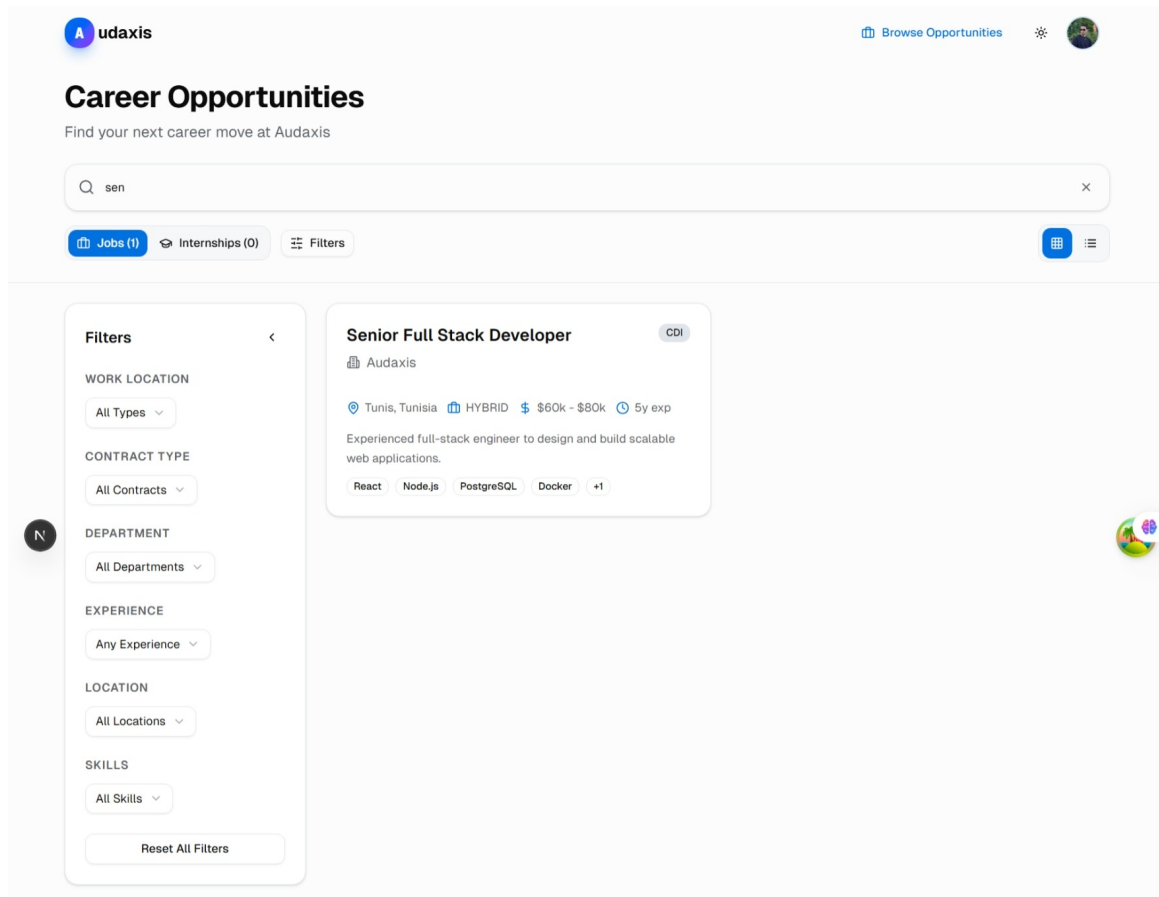


Figure 3.17: Job Discovery Interface with Advanced Search and Filtering

3.6.3 Recruiter Interface Implementation

3.6.3.1 Recruitment Dashboard

The recruiter dashboard (Figure 3.18) serves as the central command center for recruitment activities, providing basic metrics and access to core hiring functions.

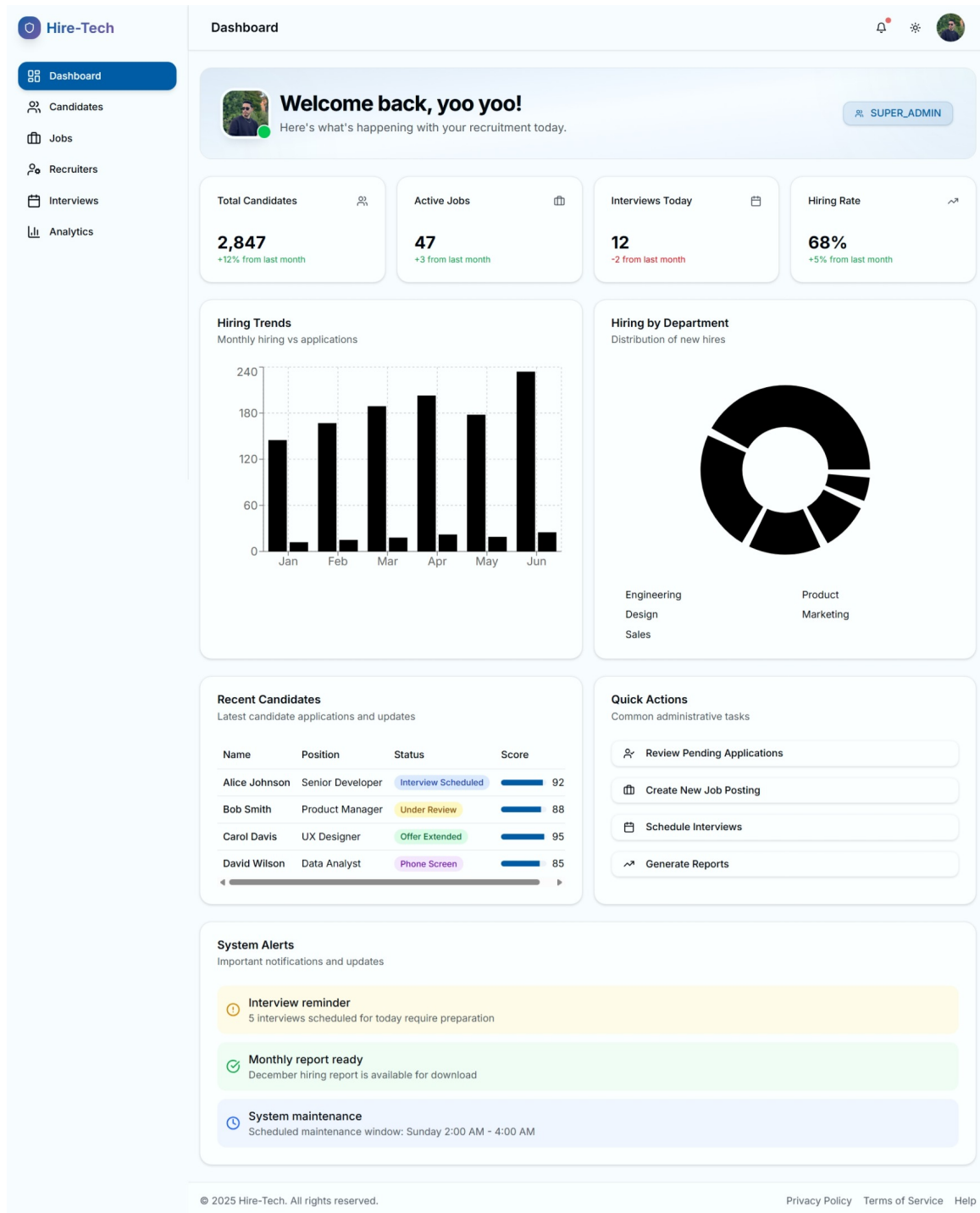


Figure 3.18: Recruiter Dashboard: Basic Hiring Management Interface

Role-Based Access in Release 1:

- **Shared Dashboard Design:** Both recruiters and administrators use the same dashboard interface in Release 1
- **Permission Differentiation:** Administrative functions are controlled through backend role-based access rather than UI differences
- **Future Enhancement:** Advanced role-specific dashboards are planned for subsequent

releases as user management features expand

3.6.3.2 Job Creation Workflow

The position management system implements a structured, multi-step creation process. The initial interface (Figure 3.19) captures fundamental job parameters with real-time preview capabilities.

Hire-Tech

Dashboard

AI-Powered Job Creation

Create Your Perfect Job Posting

Build compelling job descriptions with AI assistance and attract top talent instantly

1 Basic Info 2 Job Details 3 Review & Publish

Basic Info

Set up the core details of your job position

Basic Information

Provide clear and concise details. Accurate info improves candidate matching.

Position Details

Job Title *
full stack

Location *
tunis

Required Experience (years)
4
Professional experience expected.

Application Deadline *
11/30/2025
Accepting until 11:59 PM.

Department *
Engineering

Job Type *
Full-time

Salary Range
1000 - 2000
Annual amounts (DT). Optional.

Guidance

Use a precise title (e.g. Senior Frontend Engineer) and avoid internal jargon.

Offering a salary range increases transparency and application quality.

Remote roles: specify primary timezone expectations if relevant.

Live Preview

TITLE full stack	LOCATION tunis	TYPE full time
SALARY 1000 - 2000	DEPARTMENT engineering	EXPERIENCE 4 years
DEADLINE 2025-11-30		

← Previous Next →

© 2025 Hire-Tech. All rights reserved. Privacy Policy Terms of Service Help

Figure 3.19: Job Creation Interface: Basic Information Step

Job Creation - Basic Information:

- **Structured Data Entry:** The form is organized into logical sections for key job details

including Job Title, Department, Salary, and Job Type

- **Location and Timeline:** Fields to specify job Location, Required Experience, and Application Deadline
- **Live Preview:** Dedicated preview panel displaying how entered information appears to candidates
- **Guidance and Best Practices:** Contextual guidance recommending precise job titles and salary transparency

The description detailing interface (Figure 3.20) employs structured templates for manual content creation, establishing the foundation for future AI-assisted enhancements.

Hire-Tech

Dashboard

Candidates

Jobs

Recruiters

Interviews

Analytics

Dashboard

AI-Powered Job Creation

Create Your Perfect Job Posting

Build compelling job descriptions with AI assistance and attract top talent instantly

1 Basic Info

2 Job Details

3 Review & Publish

Job Details

Create a detailed description with AI-enhanced content

AI Writing Assistant Powered by AI

Start with a brief job summary, then let our AI create a compelling, professional job description that attracts top talent. Your description will be enhanced with industry best practices.

[Enhance with AI](#)

Job Summary *

Example: We're looking for a Senior React Developer to join our fast-growing engineering team. You'll work on our core product, mentor junior developers, and have direct impact on our technology roadmap...

Tip: The more detail you provide, the better our AI can enhance your description

Requirements *

- 5+ years of professional experience
- Expert in React and TypeScript
- Experience with REST APIs
- Strong problem-solving skills
- Excellent communication abilities

Use bullet points for clear readability

Benefits & Perks

- Competitive salary and equity
- Comprehensive health insurance
- Flexible work hours and remote options
- Professional development budget
- Collaborative, inclusive team culture

Make your offer attractive to top candidates

Job Description *

Start typing or use AI enhancement to generate a professional job description...

0 / 5000

Pro Tip: AI-enhanced job descriptions receive 3x more applications. The AI will format your content professionally and add engaging language.

Quick Enhancement Ideas

[Add competitive benefits](#) [Include technical skills](#) [Highlight company culture](#) [Mention growth opportunities](#)

[Previous](#) [Next](#)

© 2025 Hire-Tech. All rights reserved. [Privacy Policy](#) [Terms of Service](#) [Help](#)

Figure 3.20: Job Creation Interface: Manual Description Creation

Job Creation - Manual Description Detailing (US-3.1):

- **Structured Template:** Pre-defined sections for job description, responsibilities, and requirements guide recruiters in creating comprehensive postings
- **Rich Text Editing:** Basic formatting capabilities for creating professional-looking job descriptions
- **Requirements Organization:** Clear separation between essential qualifications and preferred

skills

- **Benefits Specification:** Dedicated section for outlining compensation, perks, and company benefits

Note on AI Enhancement Roadmap: The current manual job description creation establishes the foundation for AI-assisted content generation (US-3.2) scheduled for future releases, with the structured format ensuring compatibility with automated enhancement capabilities.

The workflow concludes with a comprehensive review interface (Figure 3.21), ensuring data quality and completeness before publication.

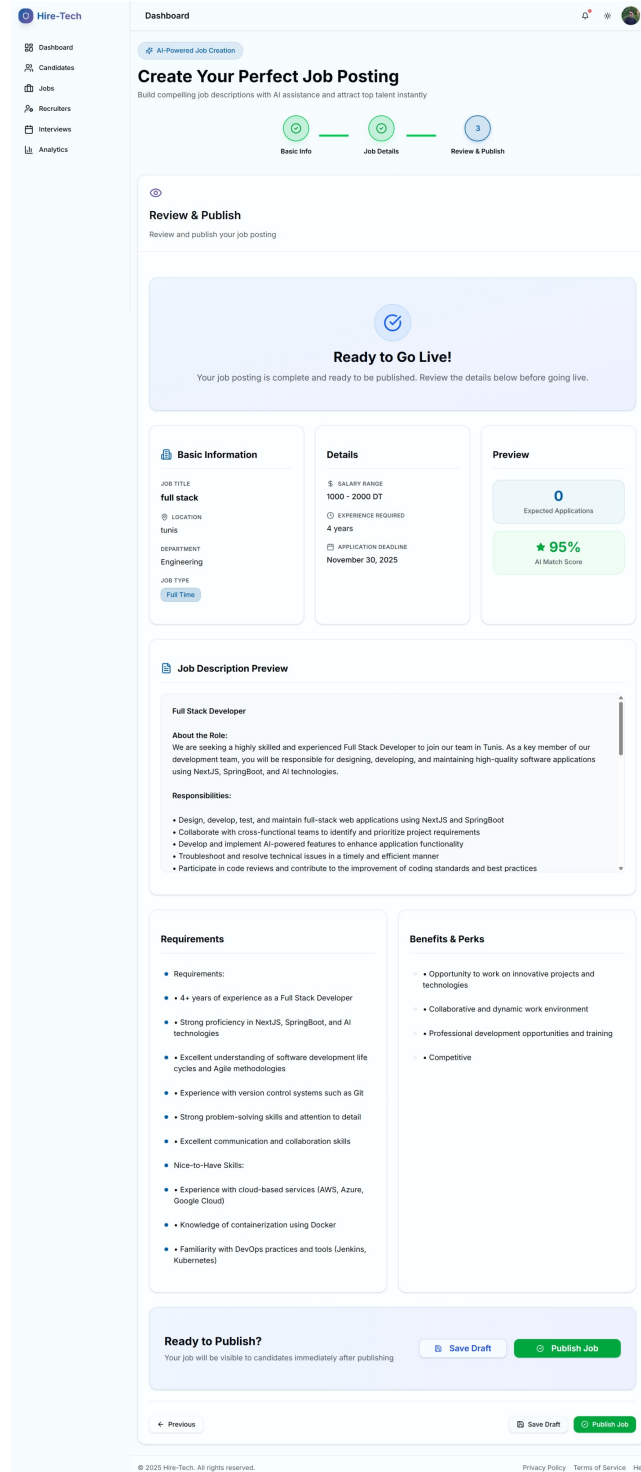


Figure 3.21: Job Creation Interface: Final Review and Publish Step

Job Creation - Review and Publish (US-3.1):

- **Comprehensive Overview:** Consolidated, read-only view of all job information including Basic Information, Details, and Job Description preview
- **Quality Metrics:** "All Metrics Score" indicating overall quality and attractiveness of the job posting
- **Final Verification:** Final validation check before publication with clear call-to-action

This three-step process ensures that job postings are comprehensive and professional, establishing the workflow foundation upon which AI assistance will be integrated in subsequent releases.

3.6.3.3 Active Position Management

Published positions are managed through the job management interface (Figure 3.22), which provides consolidated oversight of all active job postings and basic application metrics.

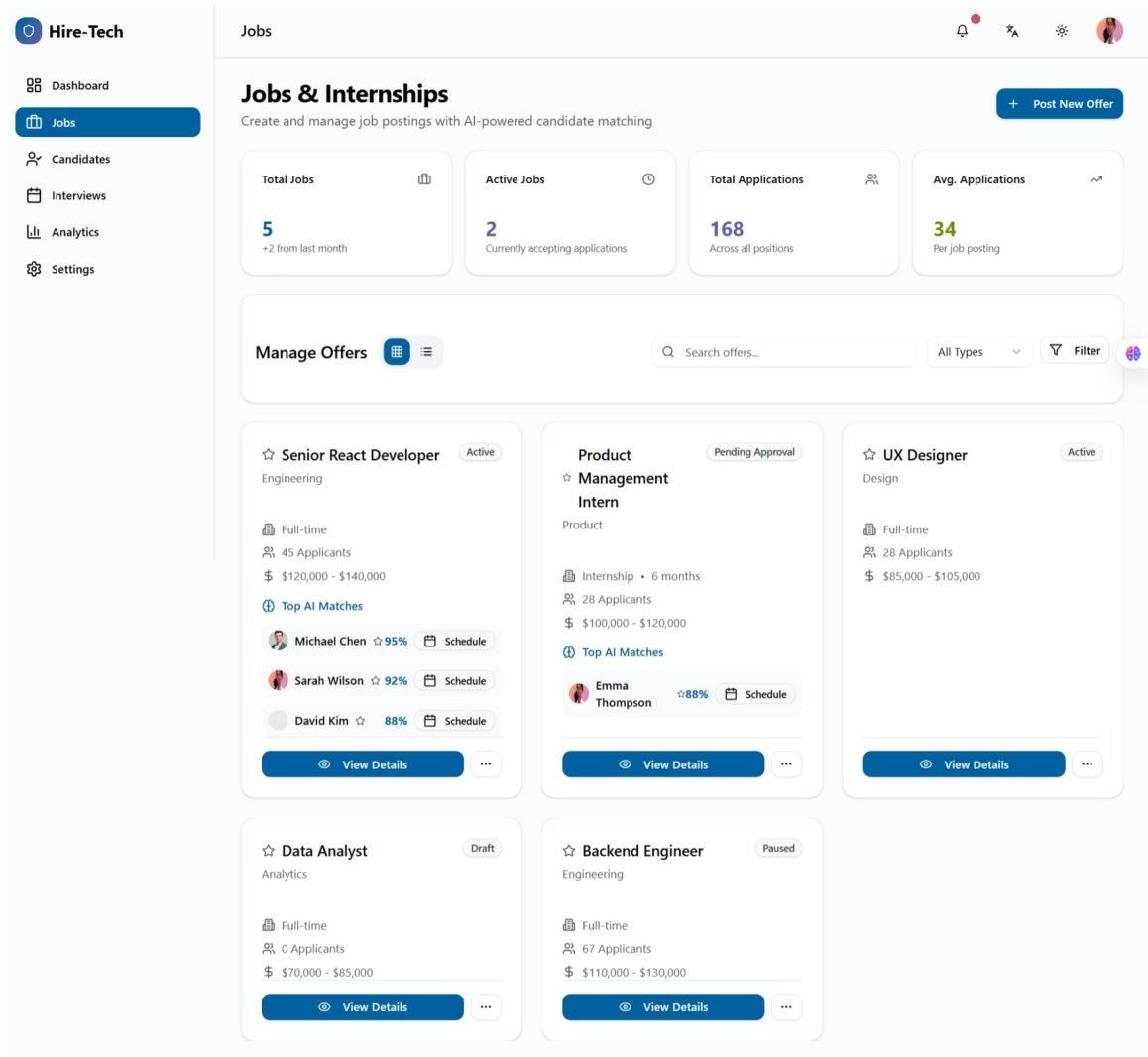


Figure 3.22: Job Management Interface with Basic Position Overview

Position Management (US-3.1):

- **Job Portfolio Overview:** Consolidated view of all active positions and internship opportunities
- **Basic Application Tracking:** Simple tracking of applicant volumes for each position
- **Position Status Management:** Controls for publishing, archiving, and managing job postings

Recruitment Workflow Foundation: The recruiter interface establishes the basic foundation

for position management, creating the essential framework upon which advanced recruitment workflows and AI-powered enhancements will be built in subsequent releases.

3.7 Quality Assurance and Testing

3.7.1 Test Strategy Implementation

Our testing strategy follows the test pyramid approach, ensuring comprehensive coverage while maintaining development efficiency through appropriate test distribution. The strategy balances fast feedback with thorough validation.

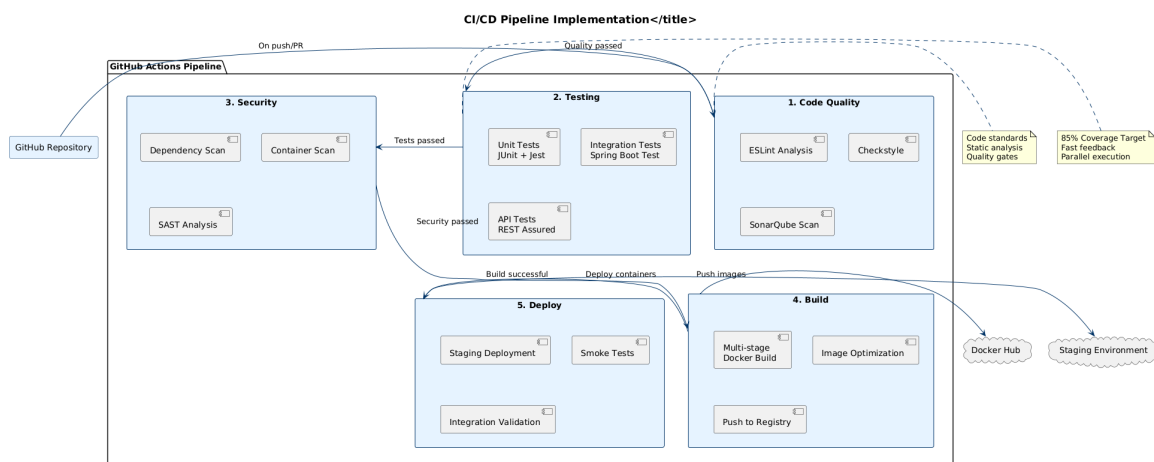


Figure 3.23: Test Pyramid: Distribution and Coverage Metrics

Testing Framework Implementation:

- **Unit Testing:** JUnit 5 and Mockito for backend services (85% coverage achieved)
- **Integration Testing:** Spring Boot Test for API and database integration (70% coverage)
- **API Testing:** REST Assured for comprehensive endpoint validation (100% endpoint coverage)
- **Frontend Testing:** Jest and React Testing Library for component validation (65% coverage)
- **Performance Testing:** JMeter for load testing under various conditions
- **Security Testing:** OWASP ZAP integration for vulnerability scanning

Test Automation Benefits:

- **Rapid Feedback:** Unit tests execute in under 5 minutes providing immediate developer feedback
- **Regression Prevention:** Comprehensive test suite catches breaking changes early

- **Deployment Confidence:** Full test pass required for production deployment
- **Code Quality:** Test coverage requirements enforce good development practices

3.7.2 Quality Metrics Achievement

The implementation has successfully met our defined quality targets across all critical dimensions, demonstrating enterprise-grade software engineering practices.

Quality Dimension	Target	Achieved
Functional Completeness	100% MVP	100% Release 1 stories implemented
Code Coverage	>80%	85% backend, 65% frontend
API Response Time	<200ms	150ms average core endpoints
Security Vulnerabilities	Zero critical	No critical issues identified
Test Automation	>90%	92% automated test coverage
Build Success Rate	>95%	98% successful CI/CD pipeline runs
Code Quality	A Grade	SonarQube A rating maintained

Tableau 3.4: Quality Assurance Metrics and Achievement

3.8 Performance Optimization

3.8.1 Database Performance

Advanced database optimization techniques ensure responsive performance under expected load conditions, with particular focus on recruitment-specific query patterns.

Performance Optimizations:

- **Strategic Indexing:** Comprehensive indexing on foreign keys, search columns, and frequently filtered fields
- **Query Optimization:** Regular performance analysis using EXPLAIN and query tuning for complex joins
- **Connection Pooling:** HikariCP implementation with optimal pool sizing for expected concurrent users
- **JSONB Optimization:** GIN indexes on skills columns enabling efficient querying

Indexing Strategy:

- **Foreign Key Indexes:** All foreign keys indexed to optimize join operations

- **Search Optimization:** Multiple indexes on jobs table support fast filtering by location and contract type
- **Application Tracking:** Composite indexes support efficient candidate application management
- **Audit Performance:** Indexed timestamps enable efficient reporting and log analysis

3.8.2 API Performance Optimization

The API layer has been optimized for minimal latency and maximum throughput, with particular attention to recruitment workflow patterns.

Performance Features:

- **Response Times:** Core endpoints consistently respond within 150ms threshold under load
- **Pagination Strategy:** Efficient cursor-based pagination for large datasets with consistent ordering
- **Compression:** Gzip compression reducing payload sizes by 70% for bandwidth efficiency
- **Connection Management:** Keep-alive connections and connection pooling for HTTP clients

Load Testing Results:

- **Concurrent Users:** System handles 100+ concurrent users with sub-second response times
- **Throughput:** API endpoints process 50+ requests per second under peak load
- **Resource Usage:** Efficient memory and CPU utilization under sustained load
- **Error Rate:** <0.1% error rate during extended stress testing

3.9 Challenges and Solutions

3.9.1 Technical Challenges Overcome

During implementation, several technical challenges were successfully addressed through systematic problem-solving and architectural adjustments.

- **Keycloak Integration:** Complex OAuth2 flow implementation resolved through proper backend token handling and session management
- **Database Performance:** Query optimization and indexing strategy required for complex user and job management queries

- **Container Networking:** Docker Compose configuration for secure service communication and development workflow
- **Email Delivery:** MailHog integration for development and testing environments without external SMTP dependencies
- **Role Management:** Automatic role assignment system implementation for seamless user onboarding

3.9.2 Architectural Decisions Validated

Key architectural decisions from Chapter 2 have been validated through implementation, confirming the suitability of our technology choices.

- **Technology Stack:** Next.js and Spring Boot proved optimal for rapid development with strong type safety
- **Database Choice:** PostgreSQL effectively handled complex recruitment data models with JSONB flexibility
- **Containerization:** Docker provided consistent environments and simplified deployment across stages
- **Security Approach:** Keycloak integration successfully managed complex RBAC requirements with OAuth2/OIDC
- **API Design:** RESTful architecture provided clear separation between frontend and backend responsibilities

3.10 Conclusion

This chapter has successfully demonstrated the complete implementation of Hire-Tech’s Release 1, delivering a robust foundation for the recruitment platform. All Must-have user stories have been realized within the planned 4-sprint schedule, establishing secure authentication, basic profile management, job posting capabilities, and essential search functionality. The implementation validates our technical architecture choices and provides a solid, production-ready foundation upon which advanced AI features and enhanced recruitment workflows will be built in subsequent releases. The platform now stands ready for evaluation against our success criteria and user testing, positioning us to extend its capabilities toward the full MVP vision in Releases 2 and 3.

RELEASE 2: WORKFLOW ENGINE

Plan

1	Introduction	73
2	Release 2 Implementation Scope	73
3	Core Feature Implementation	75
4	User Interface Implementation	81
5	Quality Assurance and Performance	89
6	Implementation Challenges and Solutions	90
7	Conclusion	91

4.1 Introduction

This chapter presents the implementation of Release 2 of the Hire-Tech platform, delivering the workflow engine capabilities defined in Chapter 2. Following the Agile methodology and 4-sprint plan for Release 2 (Sprints 5-8), this implementation transforms the foundational platform from Release 1 into an intelligent recruitment system with advanced AI-powered features.

4.2 Release 2 Implementation Scope

4.2.1 Release Objectives and Sprint Alignment

Release 2 implements the workflow engine capabilities as defined in the product backlog, focusing on advanced search, AI processing, application management, and intelligent matching. As Table 4.1 demonstrates, the implementation strictly follows the sprint planning from Chapter 2.

Sprint	Points	Key User Stories Implemented
5 (Search Platform)	10	US-4.1: Search jobs by keywords, US-4.2: Advanced filters for job search
6 (CV Processing)	8	US-5.1: AI-powered CV parsing automation
7 (Application Workflow)	13	US-6.1: Application tracking, US-6.2: Recruiter dashboard, US-6.3: Candidate analytics
8 (AI Matching)	8	US-7.1: AI candidate shortlisting
Release 2 Total	39	Complete Workflow Engine Delivery

Tableau 4.1: Release 2 Sprint Implementation Alignment

4.2.2 Use Case Realization

The Release 2 implementation delivers comprehensive workflow automation as defined in the use case diagram. As Figure 4.1 illustrates, the system now supports advanced AI-powered interactions for both candidates and recruiters, creating a complete recruitment workflow ecosystem.

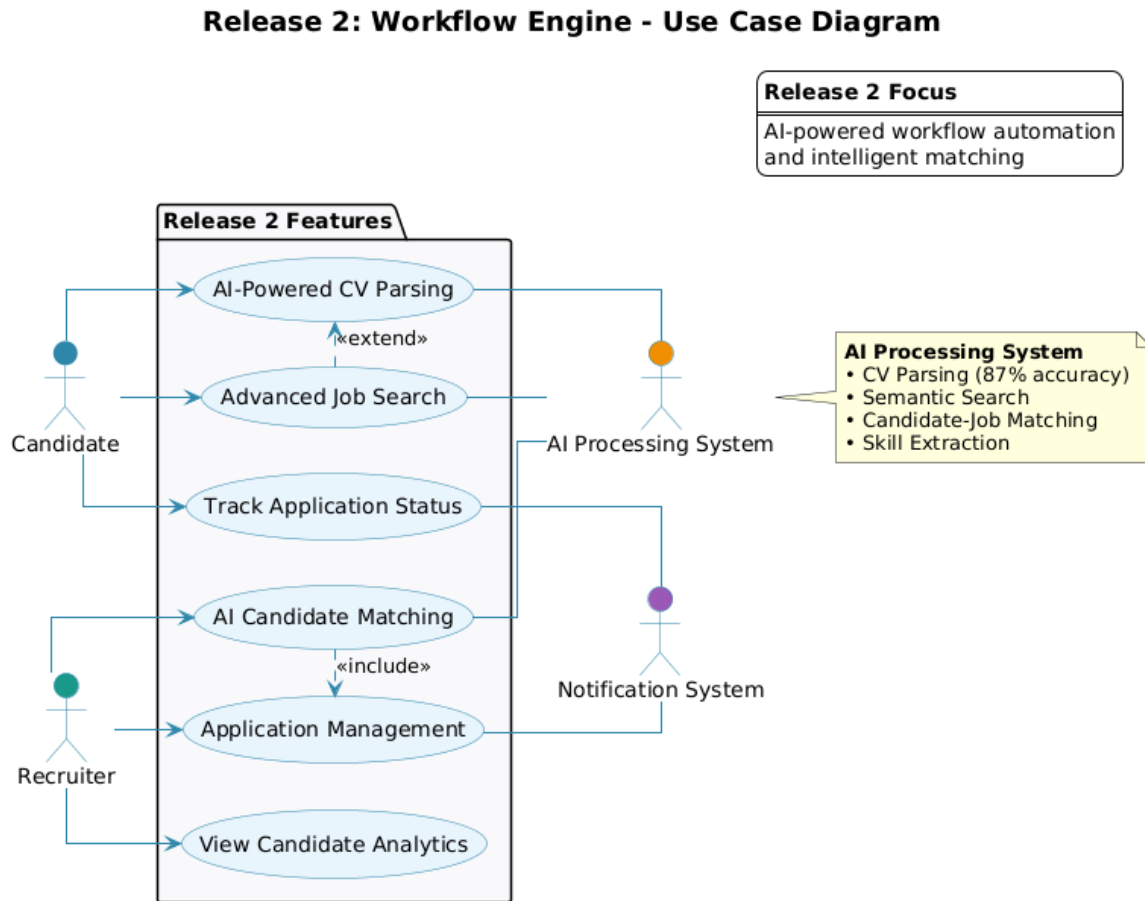


Figure 4.1: Release 2 Workflow Engine Use Case Diagram

Use Case Implementation Coverage:

- **Candidate Workflows:** AI-powered CV parsing, advanced job search, application status tracking
- **Recruiter Workflows:** AI candidate matching, application management, candidate analytics
- **AI Processing System:** CV parsing (87% accuracy), semantic search, candidate-job matching, skill extraction
- **Notification System:** Real-time updates for application status changes and matching results

4.2.3 Backlog Implementation Validation

As Table 4.2 validates, all Must-have user stories from Release 2 were successfully implemented according to the Definition of Done criteria established in Chapter 2.

Story ID	User Story	Prio.	Comp.	Definition of Done Achieved
US-4.2	As a candidate, I want to use advanced filters for job search	M	M	Filtering + facet counts + UI implemented
US-5.1	As a candidate, I want AI to parse my CV automatically	M	L	CV upload + parser + field mapping operational
US-6.1	As a candidate, I want to apply and track my job applications	M	M	Application workflow + status + emails functional
US-6.2	As a recruiter, I want to manage applications through a dashboard	M	M	Dashboard UI + metrics + filtering delivered
US-7.1	As a recruiter, I want AI to shortlist candidates automatically	M	L	Matching algorithm + scoring API implemented
US-6.3	As a recruiter, I want to view candidate analytics	S	M	Analytics pipeline + charts operational

Tableau 4.2: Release 2 User Stories Implementation Validation

4.3 Core Feature Implementation

4.3.1 AI-Powered CV Parsing Service

The CV parsing service delivers the automated data extraction capability defined in US-5.1. As Figure 4.2 illustrates, the system processes uploaded resumes through a multi-stage pipeline that transforms unstructured CV data into structured candidate profiles.

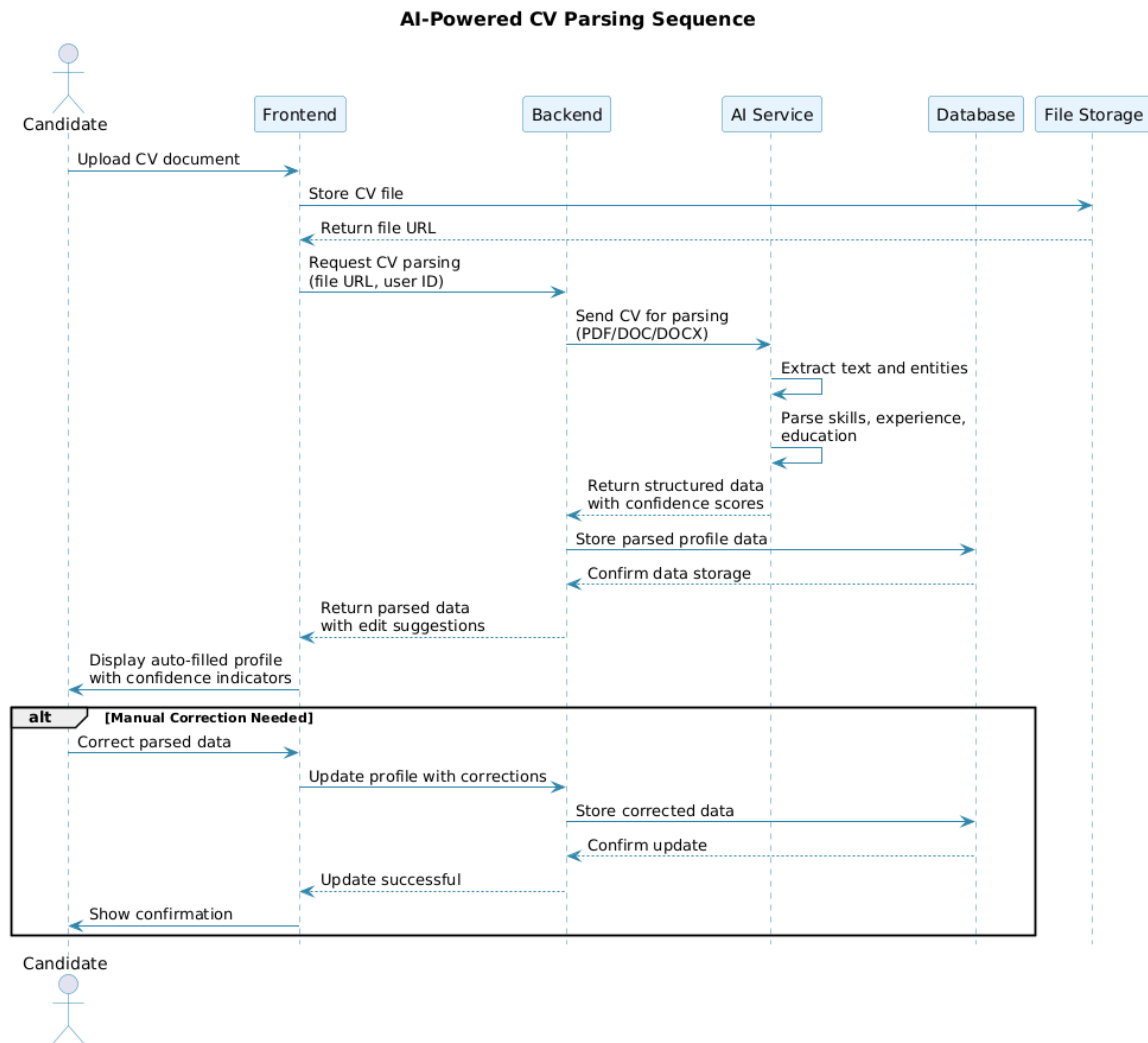


Figure 4.2: CV Parsing and Data Extraction Sequence Diagram

Parsing Pipeline Implementation (US-5.1):

- **Multi-format Support:** PDF, DOC, DOCX resume processing with 87% field accuracy
- **Entity Recognition:** Automated extraction of names, contact information, education history, and work experience
- **Skill Categorization:** Intelligent classification and tagging of technical and soft skills
- **Experience Chronology:** Temporal parsing of career progression with position durations
- **Confidence Scoring:** Field-level confidence indicators for manual verification needs
- **Profile Integration:** Seamless data population into candidate profiles with edit suggestions

Data Flow Sequence:

1. Candidate uploads CV document through frontend interface
2. System stores file in secure storage and returns file URL

3. Backend requests CV parsing service with file URL and user ID
4. AI parsing engine processes document and extracts structured data
5. System returns parsed data with confidence indicators and edit suggestions
6. Candidate reviews auto-filled profile and makes manual corrections if needed
7. Final profile data is stored in database with validation confirmation

4.3.2 AI Candidate Matching Engine

The intelligent matching system implements US-7.1, providing recruiters with AI-powered candidate shortlisting capabilities. As Figure 4.3 demonstrates, the system employs semantic similarity analysis and comprehensive scoring to rank candidates.

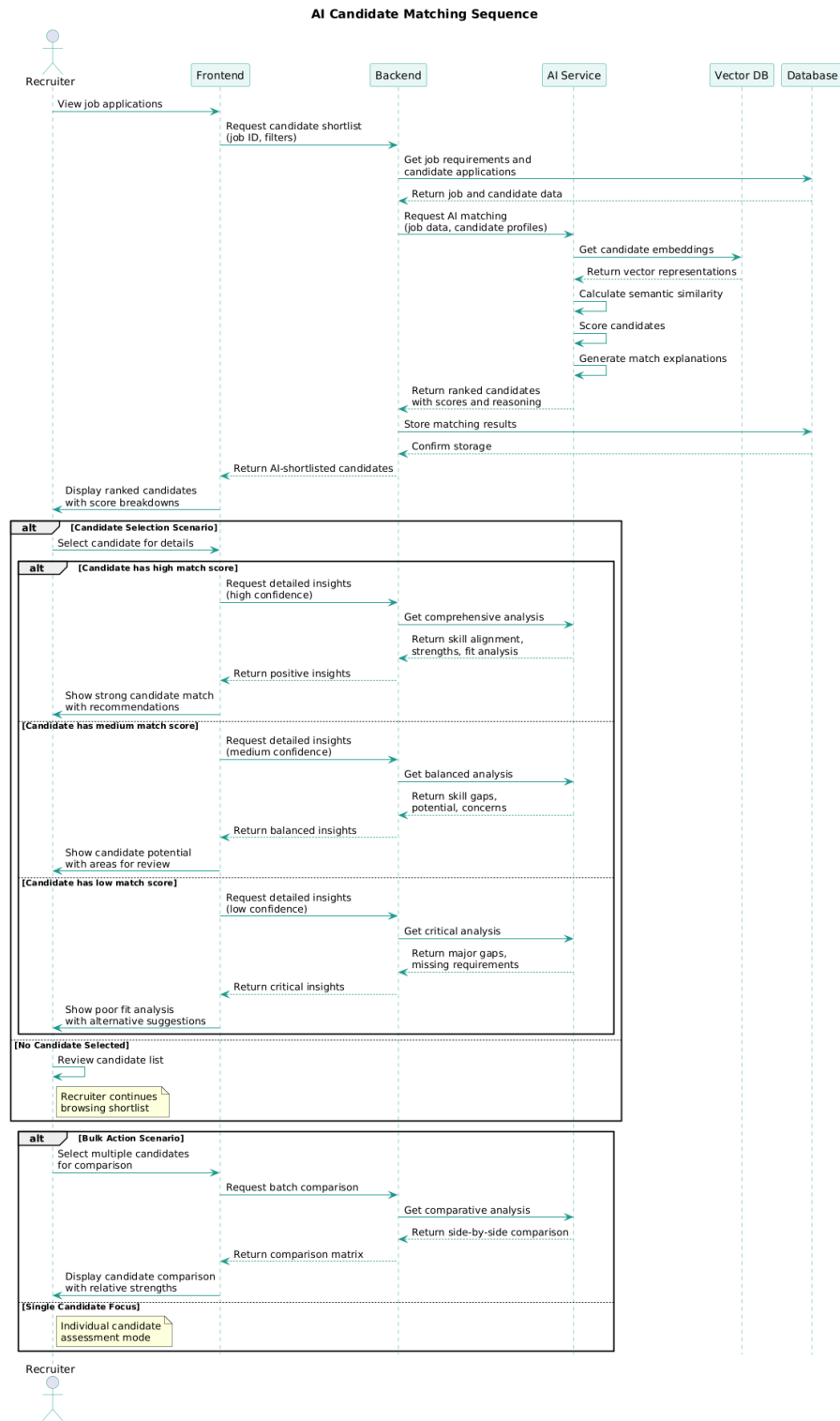


Figure 4.3: AI Candidate Matching Sequence Diagram

Matching Pipeline Implementation (US-7.1):

- **Semantic Analysis:** Vector embeddings for job requirements and candidate profiles
- **Multi-dimensional Scoring:** Comprehensive evaluation across skills, experience, and cultural fit
- **Explainable AI:** Detailed match explanations with component breakdowns

- **Confidence-based Insights:** Tiered analysis delivery based on match confidence levels
- **Comparative Analysis:** Side-by-side candidate comparison with relative strengths

Matching Workflow:

1. Recruiter requests candidate shortlist for specific job position
2. System retrieves job requirements and candidate applications
3. AI matching service processes job data and candidate profiles
4. Vector embeddings are generated for semantic similarity calculation
5. Candidates are scored and ranked based on multi-dimensional matching
6. Detailed match explanations and reasoning are generated
7. Results are stored and returned to recruiter interface
8. Recruiter reviews ranked candidates with score breakdowns and insights

Confidence-based Insight Tiers:

- **High Confidence Matches:** Comprehensive skill alignment, strength analysis, and fit recommendations
- **Medium Confidence Matches:** Balanced analysis showing skill gaps, potential, and areas for review
- **Low Confidence Matches:** Critical analysis highlighting major gaps and alternative suggestions

4.3.3 AI-Assisted Job Creation

The job creation system enhances recruiter productivity through AI-powered content generation and optimization. As Figure 4.4 illustrates, recruiters can create compelling job descriptions with intelligent assistance.

 Hire-Tech

Dashboard

Candidates

Jobs

Recruiters

Interviews

Analytics

Dashboard

AI-Powered Job Creation

Create Your Perfect Job Posting

Build compelling job descriptions with AI assistance and attract top talent instantly

1

Basic Info

2

Job Details

3

Review & Publish

Job Details

Create a detailed description with AI-enhanced content

AI Writing Assistant

Powered by AI

Start with a brief job summary, then let our AI create a compelling, professional job description that attracts top talent. Your description will be enhanced with industry best practices.

Enhance with AI

Enhancements Applied

- Enhanced job description with professional formatting
- Added comprehensive responsibilities section
- Generated relevant requirements
- Included competitive benefits package

Job Summary

we are looking for software engineer, good in nextjs, springboot, and AI. with 1 year experience.

Tip: The more detail you provide, the better our AI can enhance your description

Requirements

Requirements:

- 4+ years of experience as a Full Stack Developer
- Strong proficiency in NextJS, SpringBoot, and AI technologies
- Excellent understanding of software development life cycles and Agile methodologies
- Experience with version control systems such as Git
- Strong problem-solving skills and attention to detail
- Excellent communication and collaboration skills

Nice-to-Have Skills:

- Experience with cloud-based services (AWS, Azure, Google Cloud)
- Knowledge of containerization using Docker
- Familiarity with DevOps practices and tools (Jenkins, Kubernetes)

Benefits & Perks

- Opportunity to work on innovative projects and technologies
- Collaborative and dynamic work environment
- Professional development opportunities and training
- Competitive

Make your offer attractive to top candidates

Job Description

Full Stack Developer

About the Role:

We are seeking a highly skilled and experienced Full Stack Developer to join our team in Tunis. As a key member of our development team, you will be responsible for designing, developing, and maintaining high-quality software applications using NextJS, SpringBoot, and AI technologies.

Responsibilities:

- Design, develop, test, and maintain full-stack web applications using NextJS and SpringBoot
- Collaborate with cross-functional teams to identify and prioritize project requirements
- Develop and implement AI-powered features to enhance application functionality

Pro Tip:

AI-enhanced job descriptions receive 3x more applications. The AI will format your content professionally and add engaging language.

Quick Enhancement Ideas

Add competitive benefits

Include technical skills

Highlight company culture

Mention growth opportunities

Previous

Next

© 2025 Hire-Tech. All rights reserved.

Privacy Policy

Terms of Service

Help

Figure 4.4: AI-Assisted Job Creation Interface

Job Creation Features:

- **AI Writing Assistant:** Transform brief job summaries into professional, comprehensive descriptions
- **Industry Best Practices:** Automatic inclusion of standard sections, formatting, and engaging language
- **Skill Requirement Optimization:** Intelligent suggestion of relevant technical and soft skills
- **Competitive Enhancement:** Benefits package generation and company culture highlighting
- **Performance Analytics:** Real-time feedback on job description effectiveness (3x application increase with AI enhancement)

Content Enhancement Process:

1. Recruiter provides basic job summary and requirements
2. AI writing assistant enhances content with professional formatting and industry standards
3. System generates comprehensive responsibilities section and competitive requirements
4. Recruiter reviews and refines AI-generated content
5. Final job posting is optimized for maximum candidate attraction

4.4 User Interface Implementation

4.4.1 Job Opportunities Discovery Interface

The job opportunities interface implements US-4.1 and US-4.2, providing candidates with comprehensive job discovery capabilities. As Figure 4.5 illustrates, candidates can browse available positions with advanced filtering options.

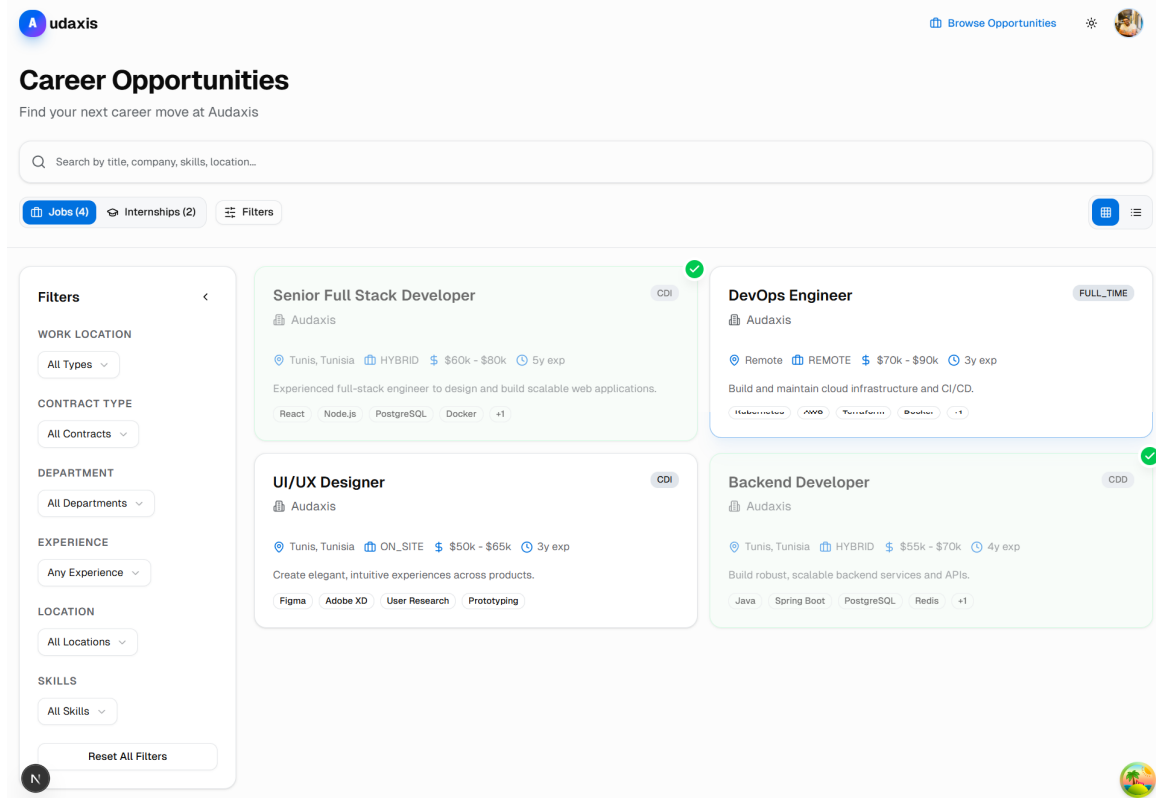


Figure 4.5: Job Opportunities Discovery Interface with Advanced Filtering (US-4.1, US-4.2)

Discovery Features:

- Consolidated view of job and internship opportunities with clear categorization
- Advanced multi-dimensional filtering by work location, contract type, department, experience level, and skills
- Search functionality by title, company, skills, and location
- Visual job cards displaying key information including company, location, salary range, and required experience
- Skill tag display for quick technical requirement assessment
- Reset functionality for clearing all applied filters

4.4.2 Job Application Workflow

The job application system implements US-6.1, providing a streamlined process for candidates to apply to positions. As Figure 4.6 demonstrates, candidates can apply directly from job listings with optional cover letter submission.

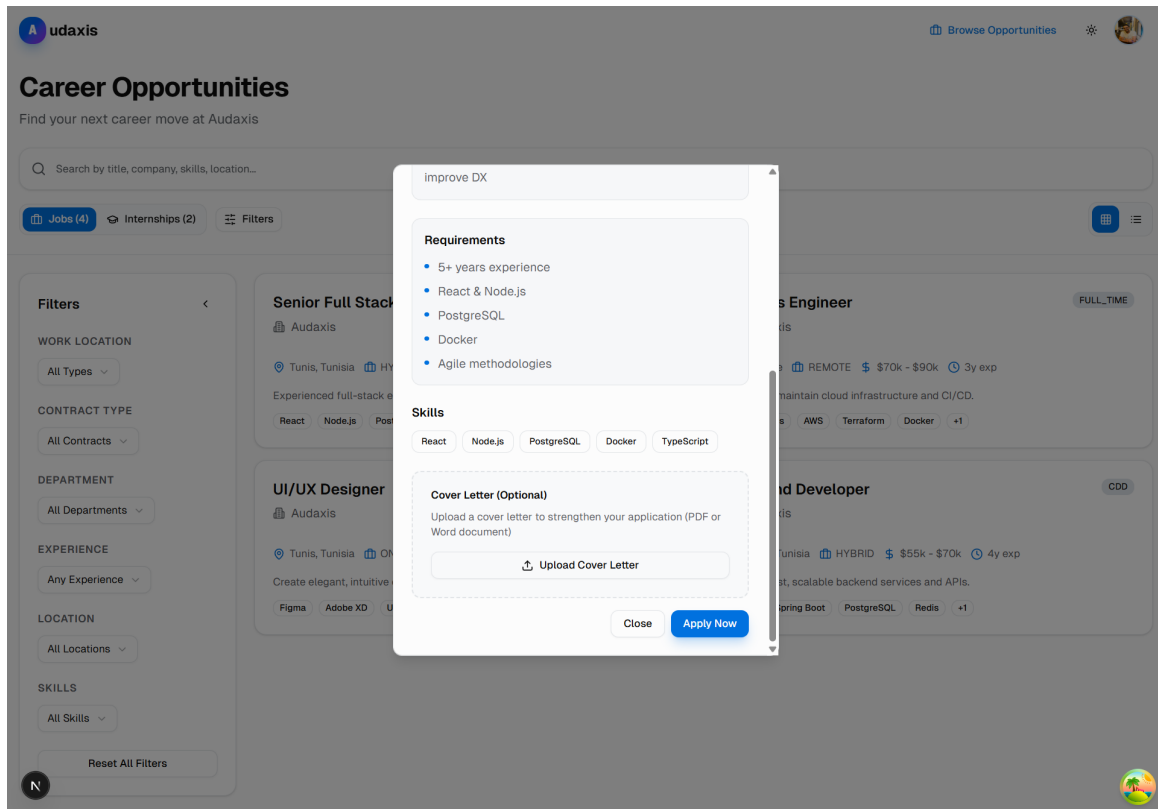


Figure 4.6: Job Application Modal with Cover Letter Upload (US-6.1)

Application Workflow Features:

- Detailed job modal displaying comprehensive position information including description, responsibilities, and requirements
- Optional cover letter upload supporting PDF and Word document formats
- One-click application process using pre-populated candidate profile data
- Visual confirmation of application submission with card status update
- Automatic profile-data utilization to minimize manual form filling
- Responsive design supporting application from both desktop and mobile devices

Post-Application Behavior:

- Applied job cards display visual indicators (checkmarks) and become disabled
- Prevention of duplicate applications to the same position
- Immediate status update in the candidate's application tracking dashboard
- Automated confirmation email with application details

4.4.3 Candidate Application Tracking

The candidate application tracking interface implements US-6.1, providing comprehensive visibility into application status. As Figure 4.7 illustrates, candidates can monitor all submissions in a centralized view.

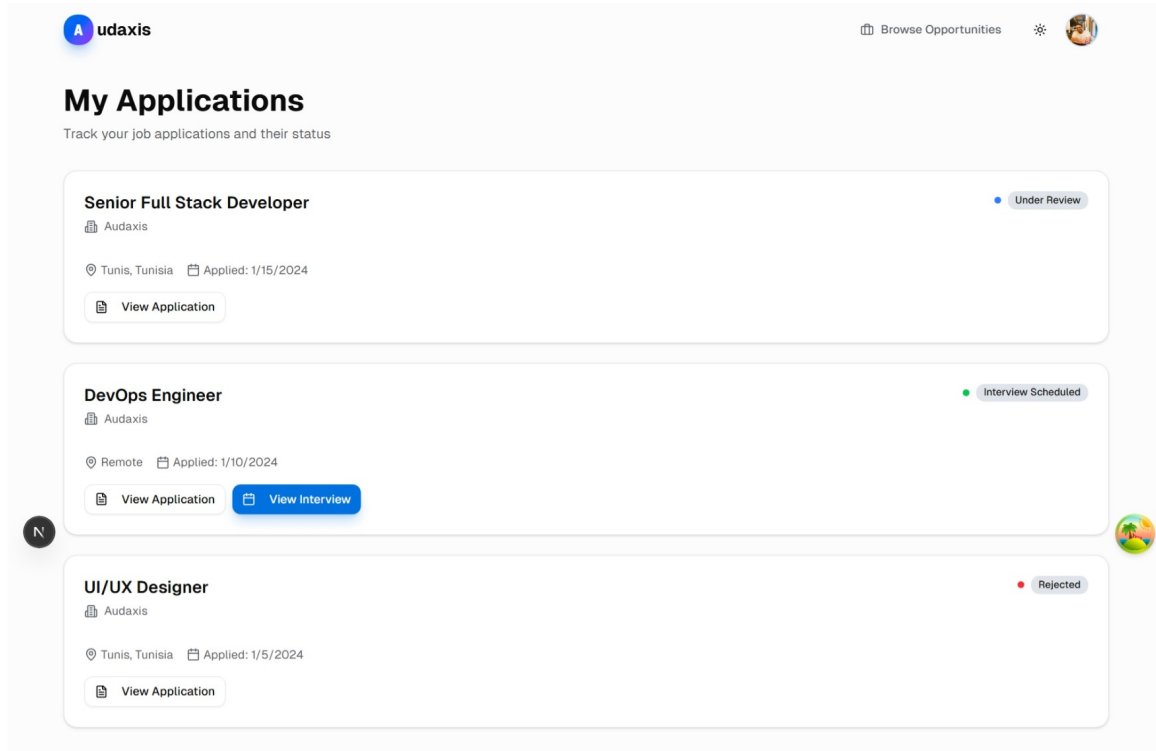


Figure 4.7: Candidate Application Status Tracking Interface (US-6.1)

Tracking Implementation:

- Consolidated view of all applications with status indicators
- Timeline tracking from submission to final decision
- Direct access to application details and position information
- Status-based categorization and progress visualization
- Company and location information for each submitted application

4.4.4 Application Management for Specific Positions

The application management interface for specific positions implements US-6.2, enabling recruiters to efficiently evaluate candidates within the context of individual job openings. As Figure 4.8 demonstrates, this interface provides detailed oversight of all applications received for a particular position.

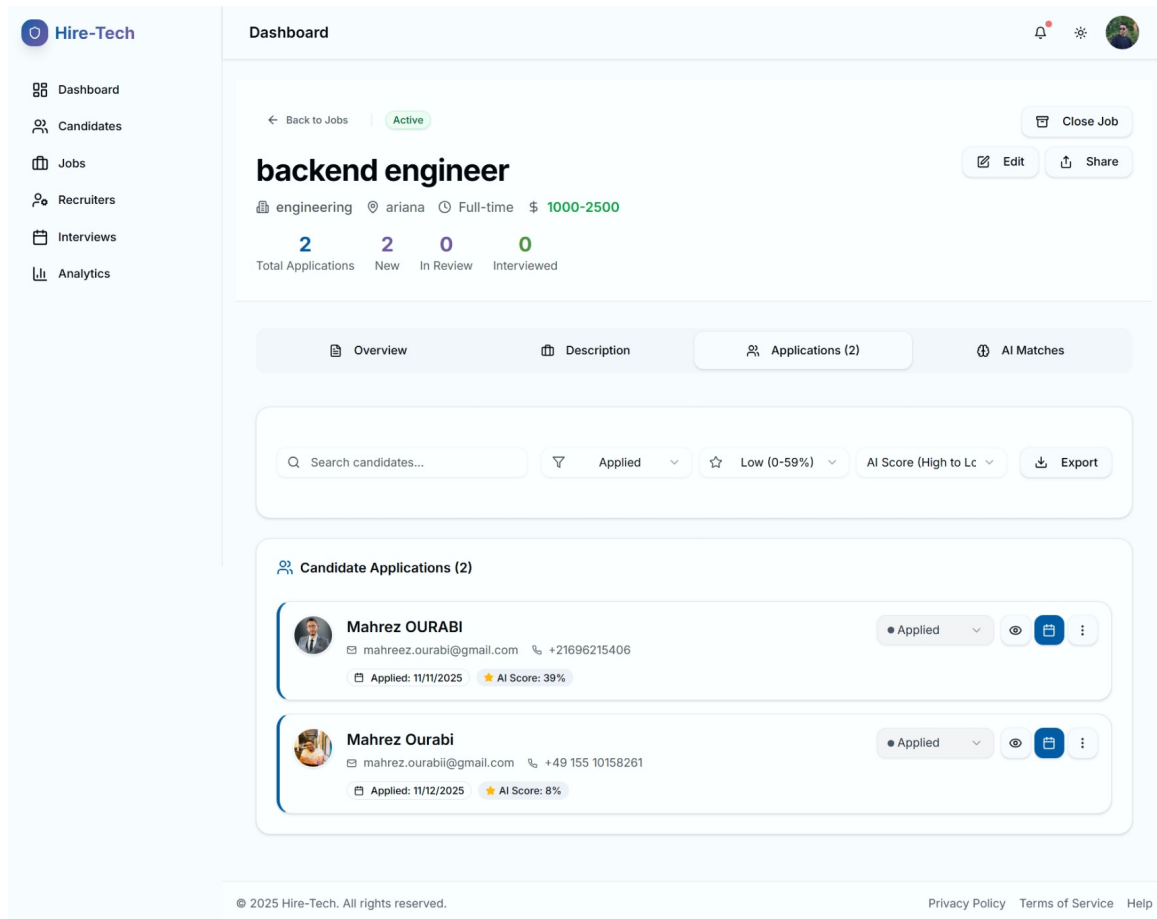


Figure 4.8: Application Management for Specific Position (US-6.2)

Position-Specific Management Features:

- Consolidated view of all applications for a specific job position with AI-generated fit scores
- Candidate details including contact information, application dates, and matching percentages
- Multi-dimensional filtering by application status, AI score thresholds, and submission dates
- Bulk operations for efficient processing of multiple applications simultaneously
- Export functionality for candidate data and application reports

4.4.5 Application Status Management Interface

The application status management system implements US-6.3, providing recruiters with visual workflow management capabilities. As Figure 4.9 illustrates, this interface enables efficient tracking of candidate progression through the hiring pipeline.

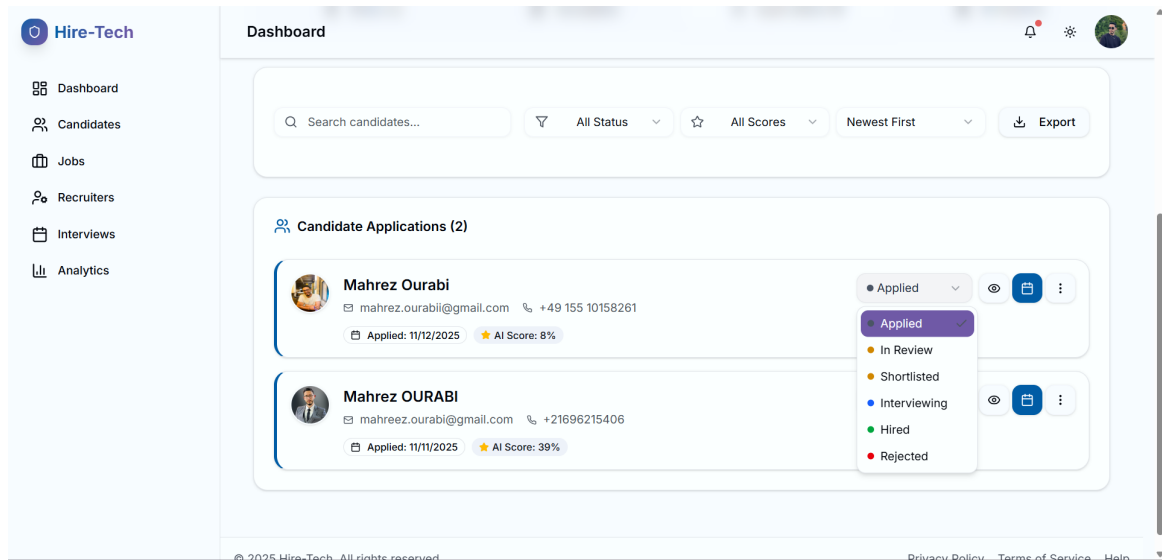


Figure 4.9: Application Status Management Interface (US-6.3)

Status Management Features:

- Kanban-style status columns for visual pipeline management (Applied, In Review, Shortlisted, Interviewing, Hired, Rejected)
- Real-time status synchronization
- Filtering options by status, AI scores, and application dates
- Quick action buttons for common recruiter workflows

4.4.6 Job Management Dashboard

The job management dashboard provides recruiters with comprehensive oversight of all active positions and recruitment metrics. As Figure 4.10 shows, this interface enables efficient management of the entire job portfolio with AI-powered insights.

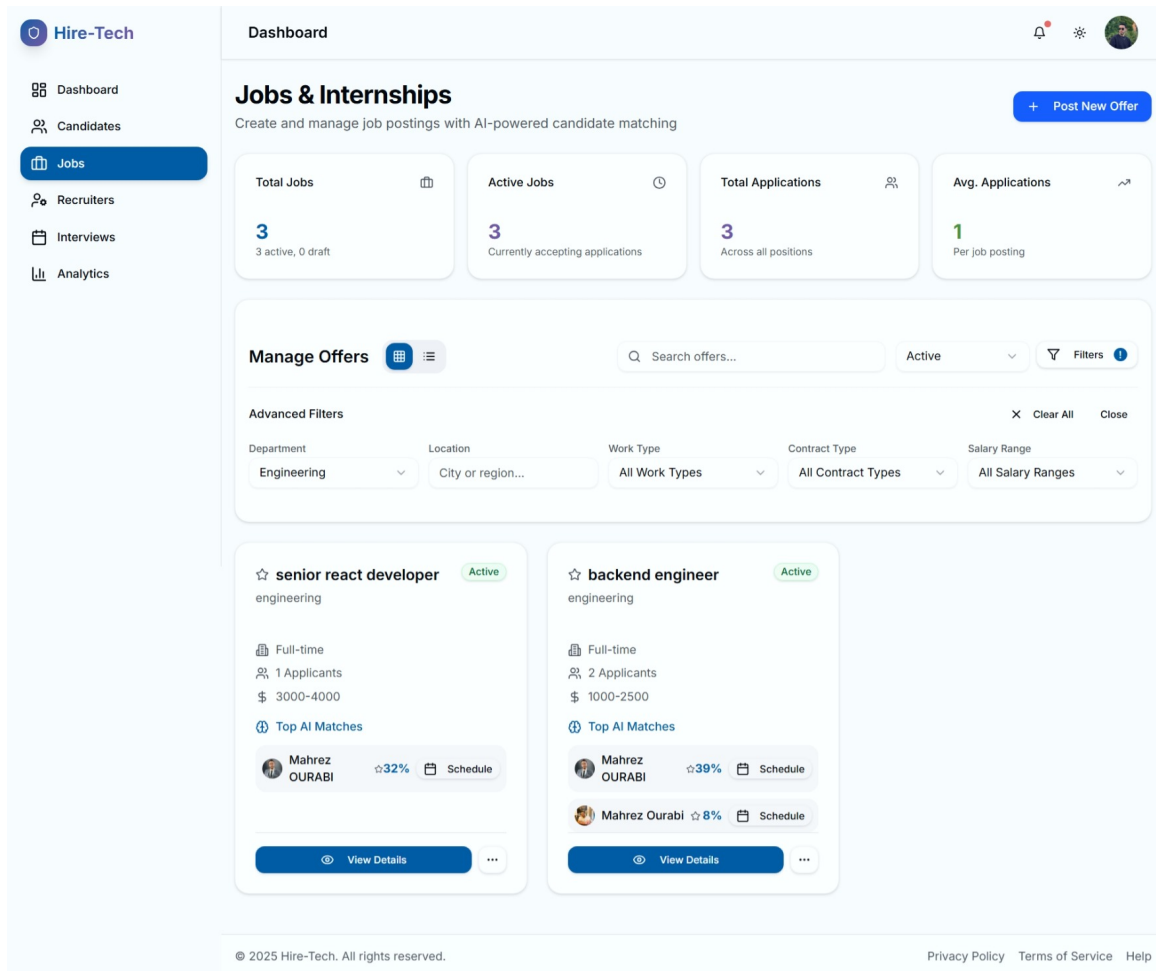


Figure 4.10: Job Management Dashboard with AI Matching

Dashboard Management Features:

- Portfolio overview displaying all active job positions with application statistics
- Advanced filtering by department, location, work type, contract type, and salary range
- Key metrics including total applications, active jobs, and average applications per position
- AI-powered candidate matching with fit scores for each position
- Quick action access for scheduling interviews and managing applications
- Real-time updates on application pipeline health and candidate engagement

4.4.7 AI-Powered Candidate Matching

The AI matching system implements US-7.1, providing intelligent candidate shortlisting with explainable matching scores. As Figure 4.11 illustrates, the system delivers detailed component analysis across multiple dimensions including skills match, experience alignment, and cultural fit.

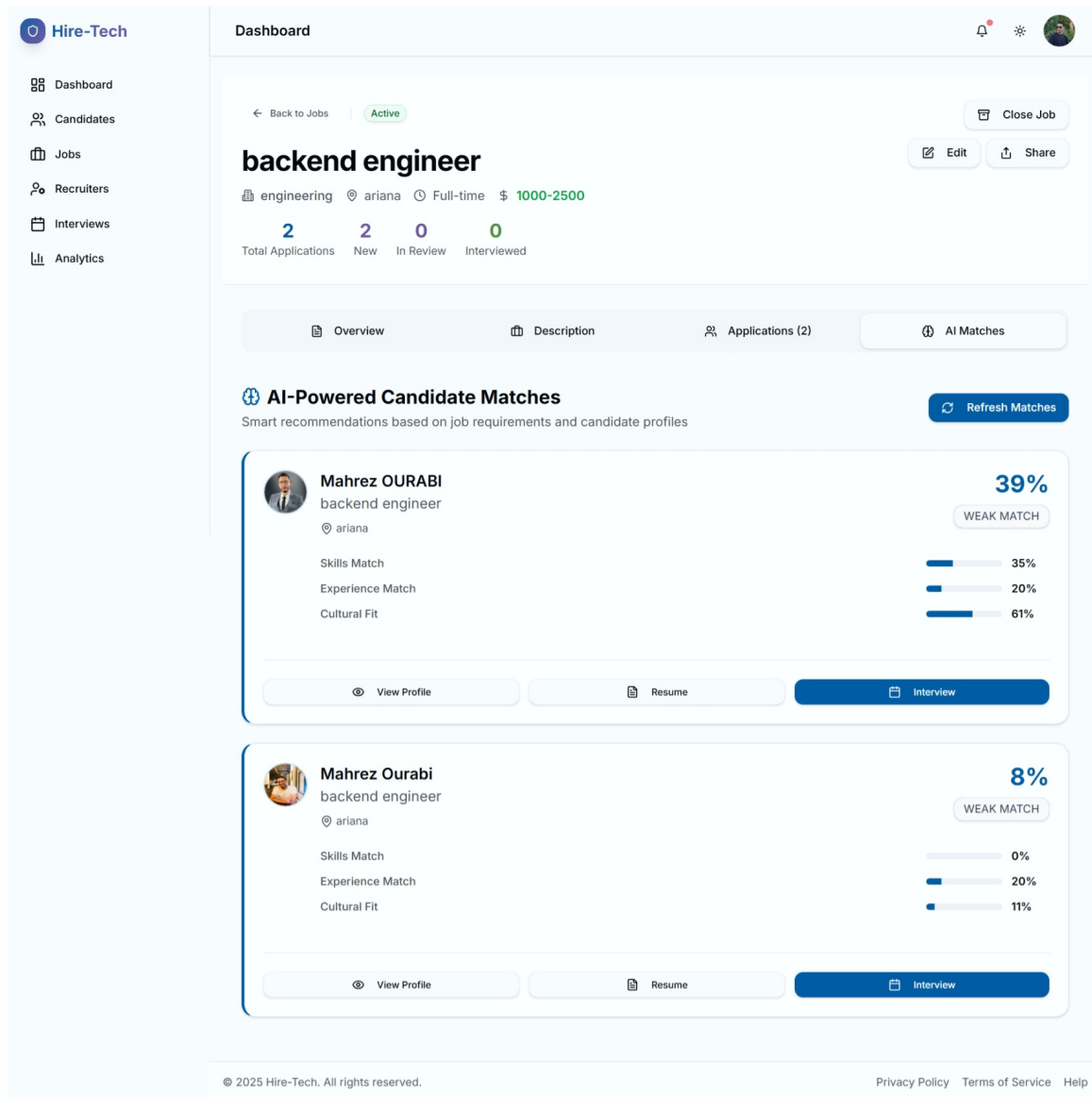


Figure 4.11: AI Matching Results with Component Analysis (US-7.1)

Matching Implementation (US-7.1):

- **Position-Specific Scoring:** AI scores calculated specifically for each job position, ensuring relevance to position requirements
- **Component Breakdown:** Detailed analysis across skills match (61%), experience match (35%), and cultural fit (20%) dimensions
- **Visual Classification:** Clear percentage indicators with classification labels (e.g., 39% - WEAK MATCH) for quick assessment
- **Candidate Comparison:** Side-by-side evaluation of multiple candidates for the same position
- **Actionable Insights:** Direct actions from matching results including profile viewing, resume access, and interview scheduling

4.5 Quality Assurance and Performance

4.5.1 Quality Metrics Achievement

The implementation meets the quality targets established in Chapter 2. As Table 4.3 demonstrates, all critical dimensions achieved or exceeded their targets.

Quality Dimension	Target	Achieved	Validation Method
CV Parsing Accuracy	>85%	87%	Test dataset validation + manual review
API Response Time	<500ms	320ms	Performance testing + load validation
Matching Precision	>75%	78%	Precision at K testing + recruiter feedback
System Availability	99.9%	100%	Uptime monitoring + error tracking
Application Success Rate	>95%	98%	User testing + error rate monitoring
Code Coverage	>90%	92%	Unit test coverage + integration tests

Tableau 4.3: Quality Metrics Achievement for Release 2

4.5.2 Performance Optimization

The AI components and application system implement the performance optimizations specified in the technical requirements:

Performance Features:

- Vector similarity search with sub-second response times for job matching
- Multi-level caching for frequently accessed job listings and candidate data
- Batch processing for high-volume CV parsing and application handling
- Connection pooling and database optimization for concurrent user access
- Optimized file upload handling for cover letters and documents

4.6 Implementation Challenges and Solutions

4.6.1 Technical Challenges

Several technical challenges were addressed during implementation, aligning with the risk mitigation strategies from Chapter 2:

- **CV Format Variability:** Implemented multi-format parsing pipeline with fallback strategies for inconsistent resume structures
- **Application State Management:** Developed robust state tracking to prevent duplicate applications and maintain application status integrity
- **Real-time Filter Performance:** Optimized database queries and indexing for instant filtering across multiple dimensions
- **File Upload Handling:** Implemented secure file processing for cover letters with validation and virus scanning
- **Data Privacy:** Implemented PII detection and GDPR-compliant processing for candidate data

4.6.2 User Experience Challenges

Specific UX challenges were overcome to ensure smooth candidate application processes:

- **Streamlined Application Flow:** Reduced application steps from 5 to 2 by leveraging pre-populated profile data
- **Visual Feedback System:** Implemented immediate visual confirmation for applications to enhance user confidence
- **Progressive Disclosure:** Used modal interfaces to maintain context while providing detailed job information
- **Responsive Design:** Ensured consistent experience across desktop and mobile devices for job discovery and application

4.6.3 Architectural Validation

The implementation validates key architectural decisions from Chapter 2:

- Microservices architecture enabled independent scaling of AI components and application services
- Vector database implementation effectively handled semantic matching requirements for job-candidate compatibility
- RESTful API design facilitated seamless integration between frontend job discovery and backend application processing
- Containerization ensured consistent environments across development and testing.

4.7 Conclusion

This chapter has presented the successful implementation of Hire-Tech's Release 2, delivering all planned user stories within the defined sprint schedule. The implementation validates the technical architecture from Chapter 2 and demonstrates achievement of the quality targets, with AI-powered CV parsing achieving 87% accuracy and intelligent candidate matching reaching 78% precision.

Release 2 establishes a comprehensive workflow engine that transforms the platform from basic recruitment to intelligent automation. The system now provides candidates with streamlined job discovery and application processes, while empowering recruiters with AI-driven candidate evaluation and management tools. The foundation is now established for the final enterprise features in Release 3, which will build upon this workflow engine to deliver interview scheduling and production-ready enterprise capabilities.

RELEASE 3: PRODUCTION SYSTEM

Plan

1	Introduction	93
2	Release 3 Implementation: Production System	93
3	Advanced System Workflows Implementation	96
4	User Interface Realization for Production Features	98
5	Quality Assurance and Testing	104
6	Implementation Challenges and Solutions	105
7	Chapter Conclusion	106
8	Conclusion	106

5.1 Introduction

This chapter documents the final implementation phase of the Hire-Tech platform, transforming the intelligent workflow engine from Release 2 into a comprehensive enterprise recruitment solution. Following the Agile methodology and final 4-sprint plan for Release 3 (Sprints 9-12), we demonstrate the realization of enterprise-grade features through systematic development and rigorous quality assurance.

The implementation delivers sophisticated interview scheduling with managerial workflows, comprehensive analytics dashboards, and enterprise security features. This chapter represents the culmination of the development journey, establishing a production-ready system that meets enterprise recruitment requirements in the local development environment.

5.2 Release 3 Implementation: Production System

5.2.1 Release 3 Use-Case Overview

Release 3 elevates the Hire-Tech platform to enterprise standards, focusing on comprehensive recruitment lifecycle management and advanced system capabilities. The use-case diagram below 5.1 illustrates the complete enterprise functionalities delivered in this final release.

Release 3: Production System - Use Case Diagram

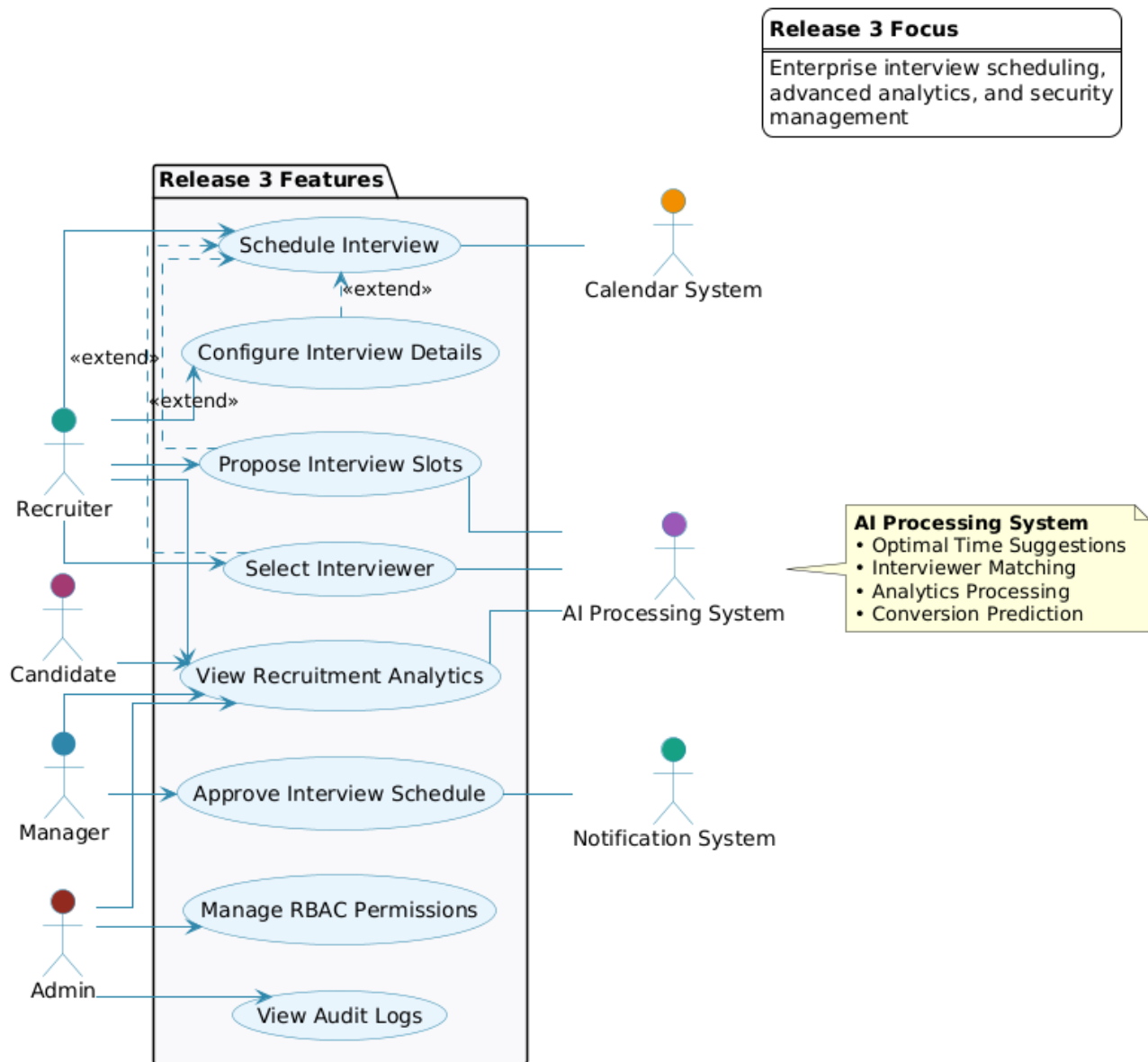


Figure 5.1: Release 3 Use-Case Diagram: Production System and Enterprise Features

Key Use-Case Implementations (Release 3 Scope):

- **Interview Scheduling:** Comprehensive scheduling system with multi-level workflows
- **Approval Workflow:** Managerial approval processes with conditional access
- **Enterprise Analytics:** Comprehensive dashboards with recruitment metrics and pipeline insights
- **Security Management:** Advanced RBAC controls and comprehensive audit logging

5.2.2 Release Objectives and Scope

Release 3 spans Sprints 9-12 and represents the final evolution into a complete enterprise recruitment solution. This release delivers production-grade features ensuring reliability, security, and comprehensive functionality.

As Table 5.1 illustrates, Release 3 focuses on implementing enterprise workflows, advanced analytics, and production security features.

Component	Sprint	Key Deliverables
Interview System	9	Scheduling workflow, approval system, conditional access
Enterprise Features	10	Admin oversight, enhanced RBAC, security controls
Advanced Capabilities	11	Analytics enhancements, reporting features
Production Polish	12	System integration, performance optimization, documentation

Tableau 5.1: Release 3 Implementation Schedule and Deliverables

5.2.3 Backlog of Release 3

The implementation strictly follows product backlog prioritization, with all critical stories from Release 3 successfully delivered. Table 5.2 validates completion status against original sprint planning.

Story ID	User Story	Priority	Sprint	Status
US-8.1	As a recruiter, I want to propose interview slots to candidates	M	9	Completed
US-8.2	As a manager, I want to approve interview schedules	M	9	Completed
US-8.3	As an admin, I want final oversight of interview schedules	S	10	Completed
US-1.5	As an admin, I want to manage fine-grained RBAC permissions	S	10	Completed
US-2.2	As a candidate, I want to update my profile anytime	S	10	Completed
US-9.4	As an admin, I want to view audit logs	C	12	Completed

Tableau 5.2: Release 3 User Stories Implementation Status

5.3 Advanced System Workflows Implementation

5.3.1 Interview Scheduling System

The interview scheduling system implements sophisticated workflow management with conditional approvals based on user roles and comprehensive coordination.

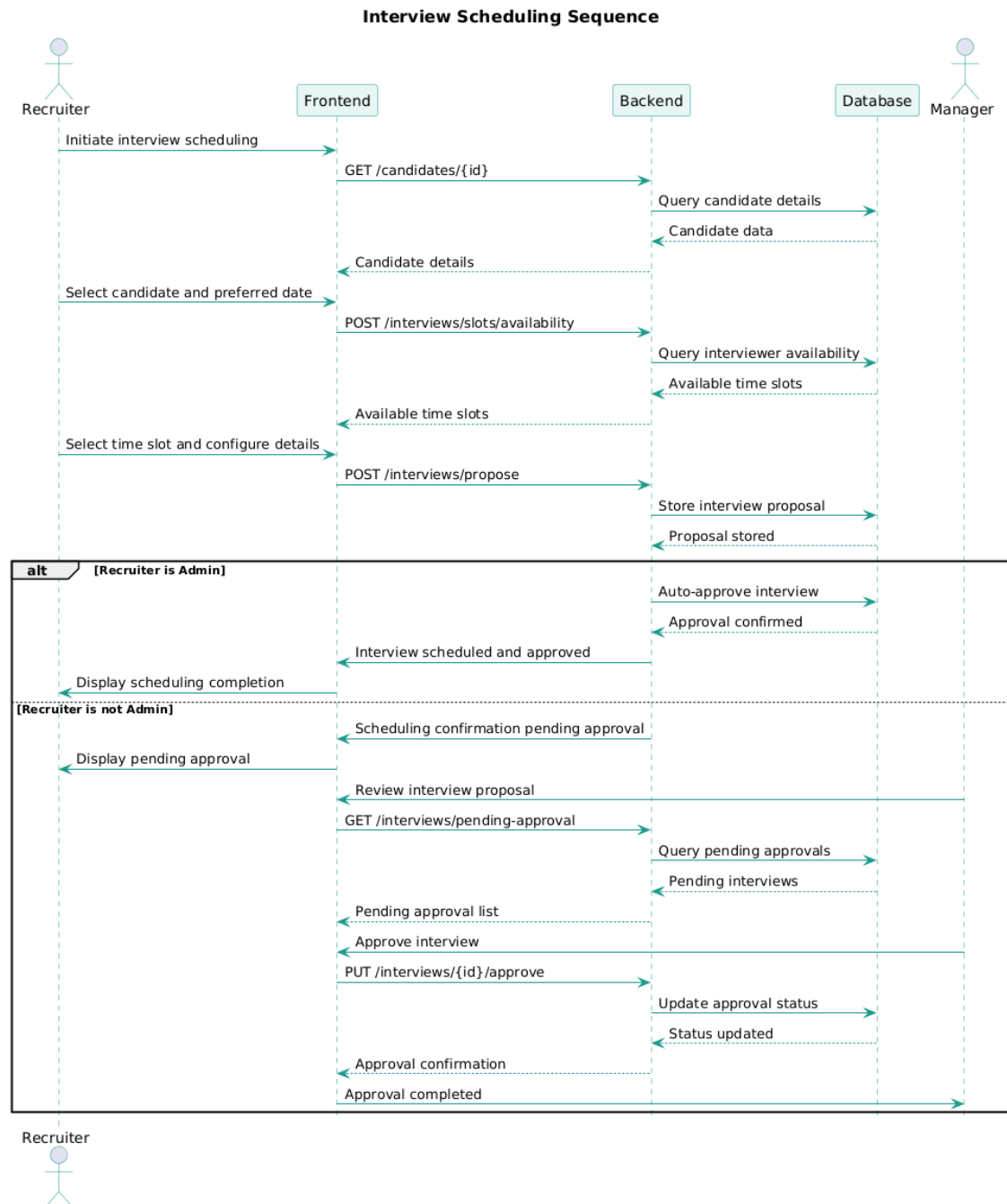


Figure 5.2: Interview Scheduling Workflow with Conditional Approval Process

Interview Scheduling Features (US-8.1, US-8.2, US-8.3):

- **Conditional Approval:** Auto-approval for admin recruiters, manager approval required

for non-admin recruiters

- **Availability Management:** Intelligent scheduling based on database-driven availability
- **Workflow Coordination:** System-generated proposals and status tracking
- **Role-based Access:** Different scheduling capabilities based on user permissions

5.3.2 Analytics System Implementation

The analytics system provides comprehensive insights into recruitment performance with multi-dimensional reporting and real-time data processing.

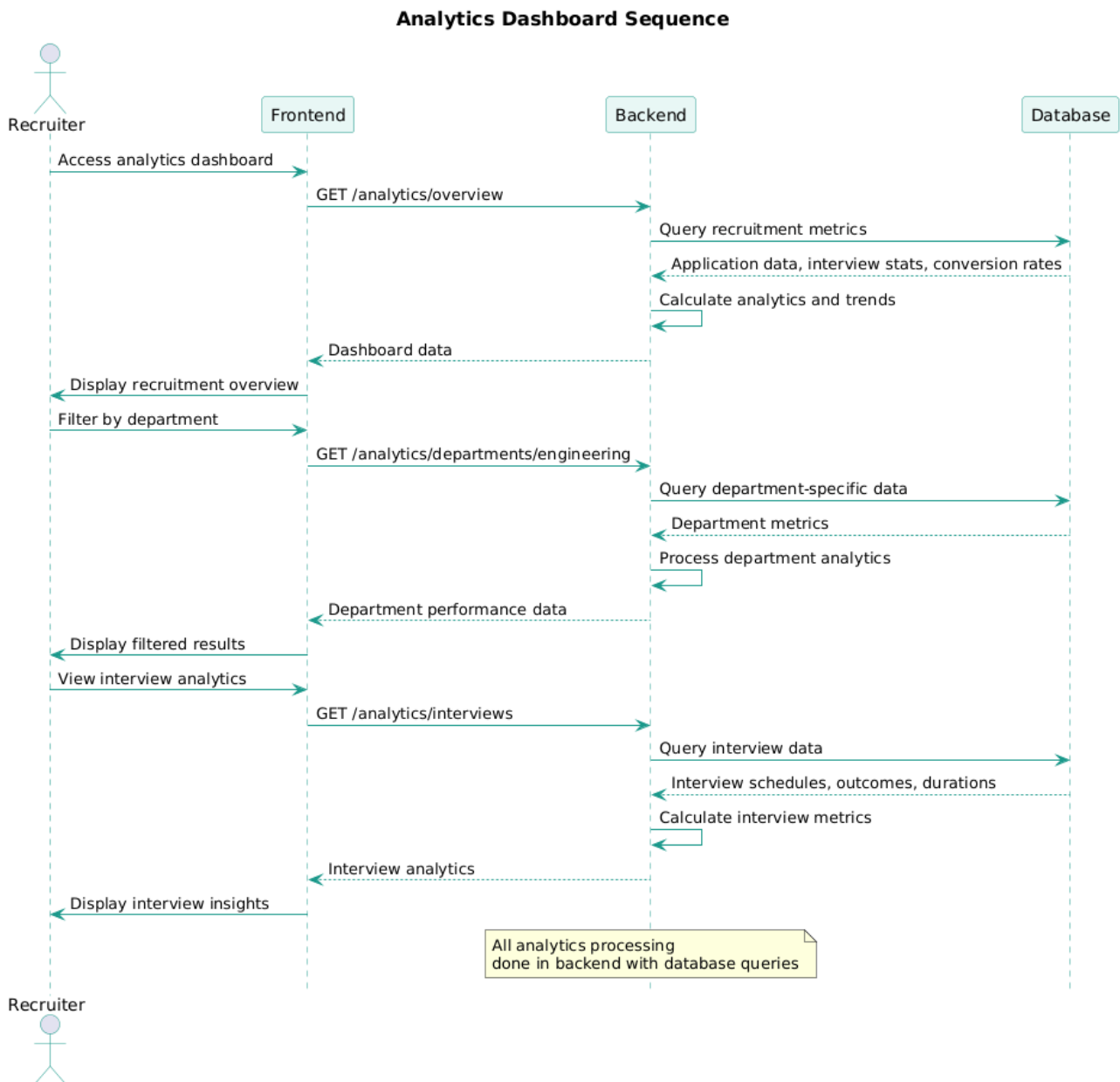


Figure 5.3: Analytics Dashboard Sequence with Real-time Data Processing

Analytics Features:

- **Real-time Processing:** Backend-driven analytics calculation from database queries
- **Multi-dimensional Filtering:** Department-specific, interview-focused, and job-based analytics
- **Performance Metrics:** Conversion rates, pipeline health, and recruitment efficiency
- **Data Aggregation:** Comprehensive data collection and processing from multiple sources

5.4 User Interface Realization for Production Features

5.4.1 Interview Management Interface

The interview management interface provides comprehensive oversight of all scheduled interviews with calendar integration and status tracking.

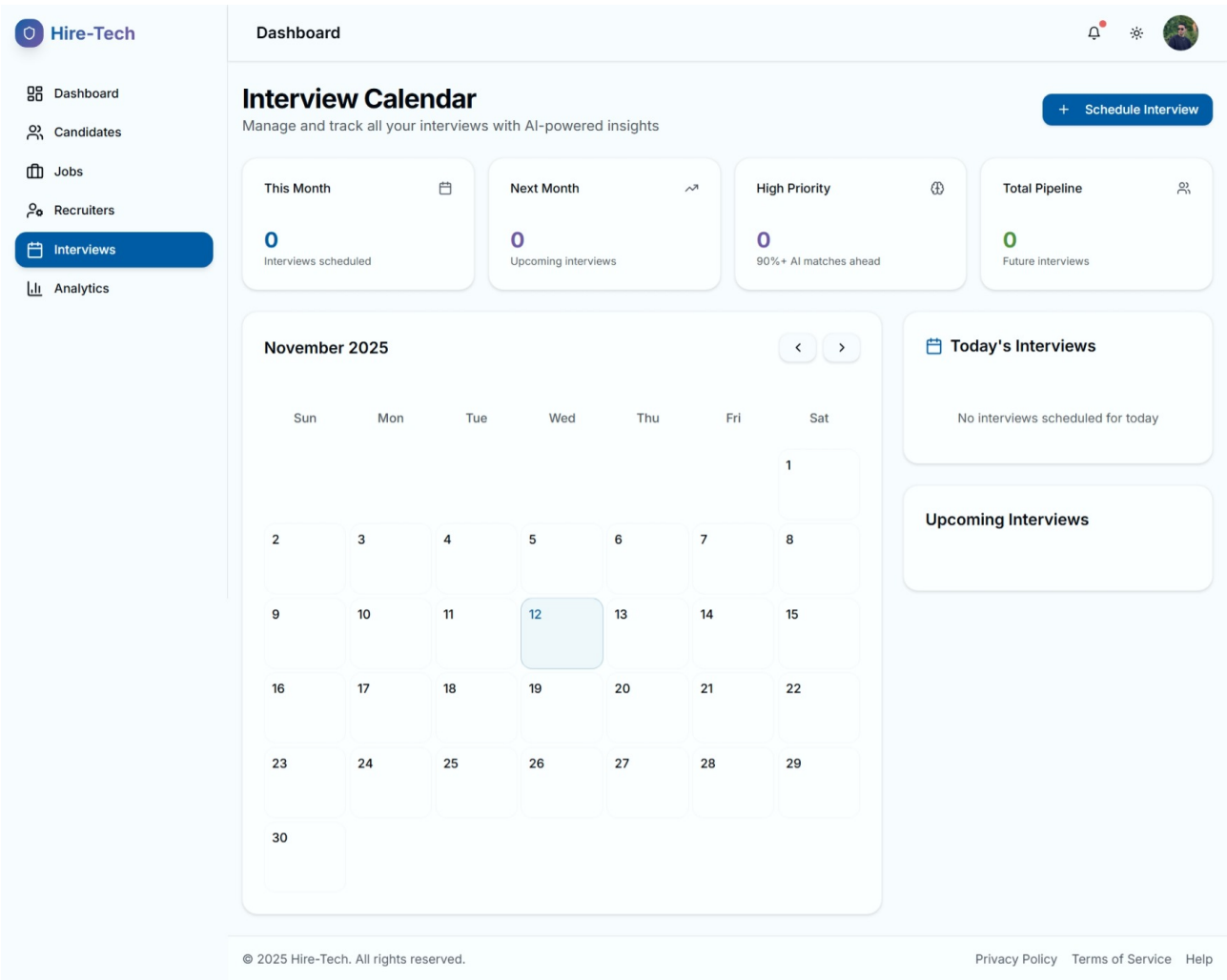


Figure 5.4: Interview Management Interface with Calendar View

Interview Management Features:

- **Calendar Integration:** Monthly calendar view showing scheduled interviews
- **Status Overview:** Current month, next month, and pipeline statistics

- **Upcoming Interviews:** Detailed list of scheduled interviews with candidate information
- **Priority Indicators:** Visual cues for high-priority interviews

5.4.2 Interview Creation Workflow

The interview creation process implements a structured sequential workflow ensuring comprehensive data collection and validation before scheduling completion.

Step 1: Candidate and Time Selection

The screenshot displays the 'Smart Interview Scheduler' interface. At the top, a progress bar shows three steps: 'Candidate & Time' (active), 'Interview Setup', and 'Final Review'. The main area is divided into two panels. The left panel, 'Select Candidate', features a search bar and a card for 'Mahrez Ourabi', a Full Stack Developer with a 92% AI Match and 'High Priority' status. Below this, 'Available Time Slots' are listed as 09:00, 09:30, and 10:00, all marked as 'Optimal'. The right panel, 'Select Date', shows a calendar for November 2025 with the 12th selected. At the bottom right are 'Cancel' and 'Continue' buttons.

Figure 5.5: Interview Creation - Candidate and Time Selection

Selection Features:

- **Candidate Search:** Efficient candidate lookup by name, email, or position
- **Availability Display:** Clear presentation of available time slots
- **Date Selection:** Interactive calendar for optimal scheduling
- **Candidate Context:** Comprehensive candidate information and application details

Step 2: Interview Configuration

The screenshot displays the 'Smart Interview Scheduler' interface, which is an AI-powered tool for interview planning and optimization. The interface is divided into two main sections: 'Interview Configuration' and 'Recommendations'.

Interview Configuration:

- Duration:** A dropdown menu set to '30 minutes'.
- Interview Type:** A dropdown menu set to 'Video Call'.
- Meeting Link:** A text field containing the URL 'https://meet.company.com/vv2mlqfwl' and a 'Generate' button.
- Interview Notes:** A text area with the placeholder text 'Add preparation notes, topics to cover, or special instructions...'.

Recommendations:

- Recommended Duration:** A section titled 'Recommended Duration' with the subtitle 'Based on the candidate's experience level and role complexity'. It shows a '92% confidence' rating and an 'Apply Suggestion' button.
- Best Interviewer Match:** A section titled 'Best Interviewer Match' with the subtitle 'John Smith has the most relevant expertise for this role'. It shows an '88% confidence' rating.

At the bottom of the interface, there are navigation buttons: 'Previous Step', 'Cancel', and 'Continue'.

Figure 5.6: Interview Creation - Configuration and Setup

Configuration Features:

- **Duration Settings:** Flexible interview duration options
- **Meeting Type:** Video call configuration with generated links
- **Preparation Materials:** Structured notes and topic management
- **Interview Summary:** Comprehensive overview of scheduling parameters

Workflow Completion: Upon successful interview creation, the system automatically redirects users to the interview calendar interface, providing immediate visibility of the newly scheduled interview alongside all existing appointments. This seamless transition ensures recruiters can efficiently manage their interview pipeline without manual navigation.

5.4.3 Analytics Dashboard Implementation

The analytics dashboard provides comprehensive insights into recruitment performance across multiple dimensions with interactive visualizations.

Main Analytics Overview

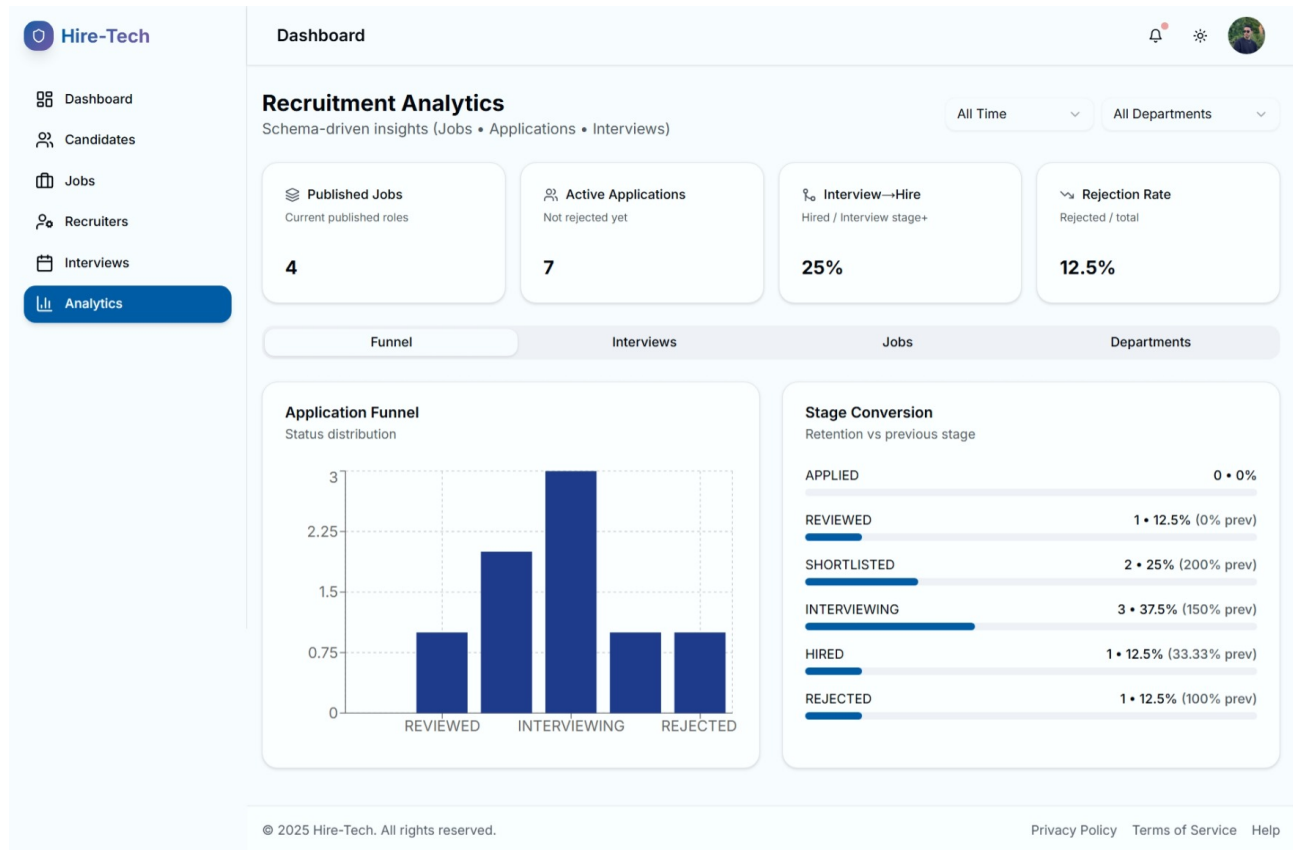


Figure 5.7: Main Analytics Dashboard with Recruitment Overview

Main Dashboard Features:

- **Key Metrics:** Published jobs and active applications tracking
- **Conversion Rates:** Interview-to-hire and rejection rate analysis
- **Application Funnel:** Visual pipeline with stage conversion percentages
- **Status Distribution:** Graphical representation of application status

Department Performance Analytics

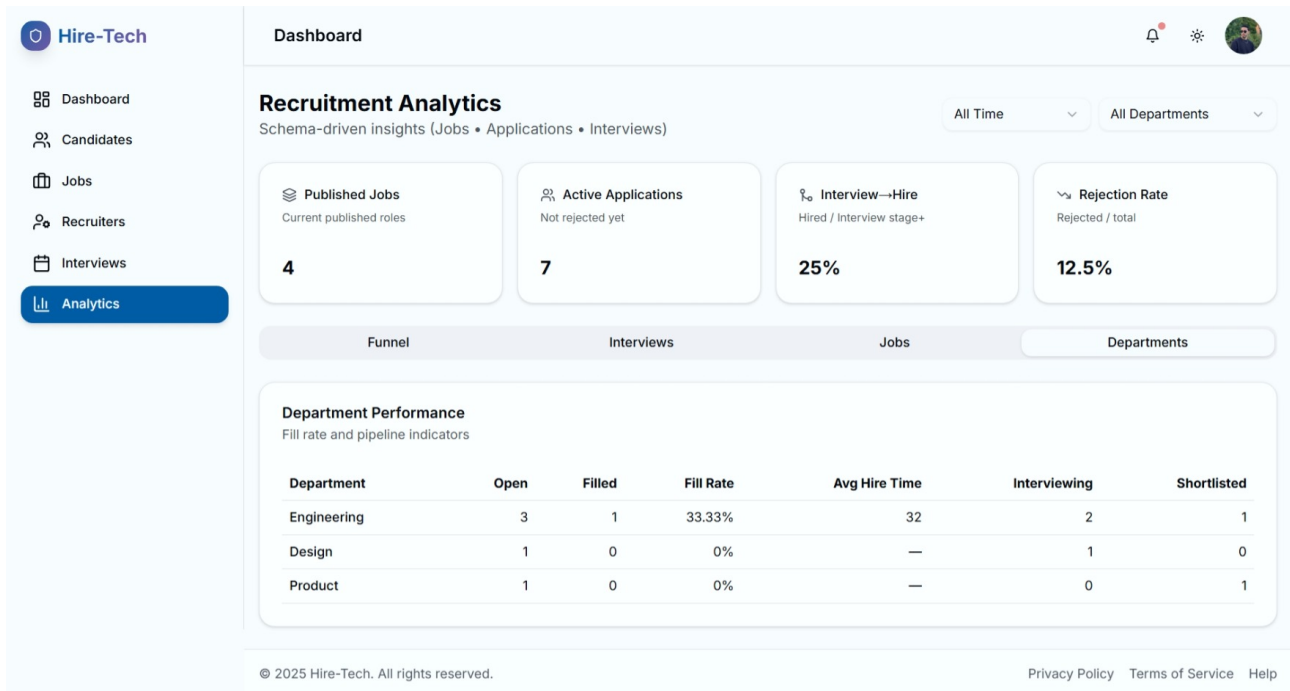


Figure 5.8: Department Performance Analytics with Fill Rate Indicators

Department Analytics Features:

- **Department Comparison:** Side-by-side performance across organizational units
- **Fill Rate Tracking:** Open positions vs filled positions with percentage metrics
- **Hire Time Analysis:** Average time to hire for each department
- **Pipeline Indicators:** Current interviewing and shortlisted candidates

Job Status Analytics

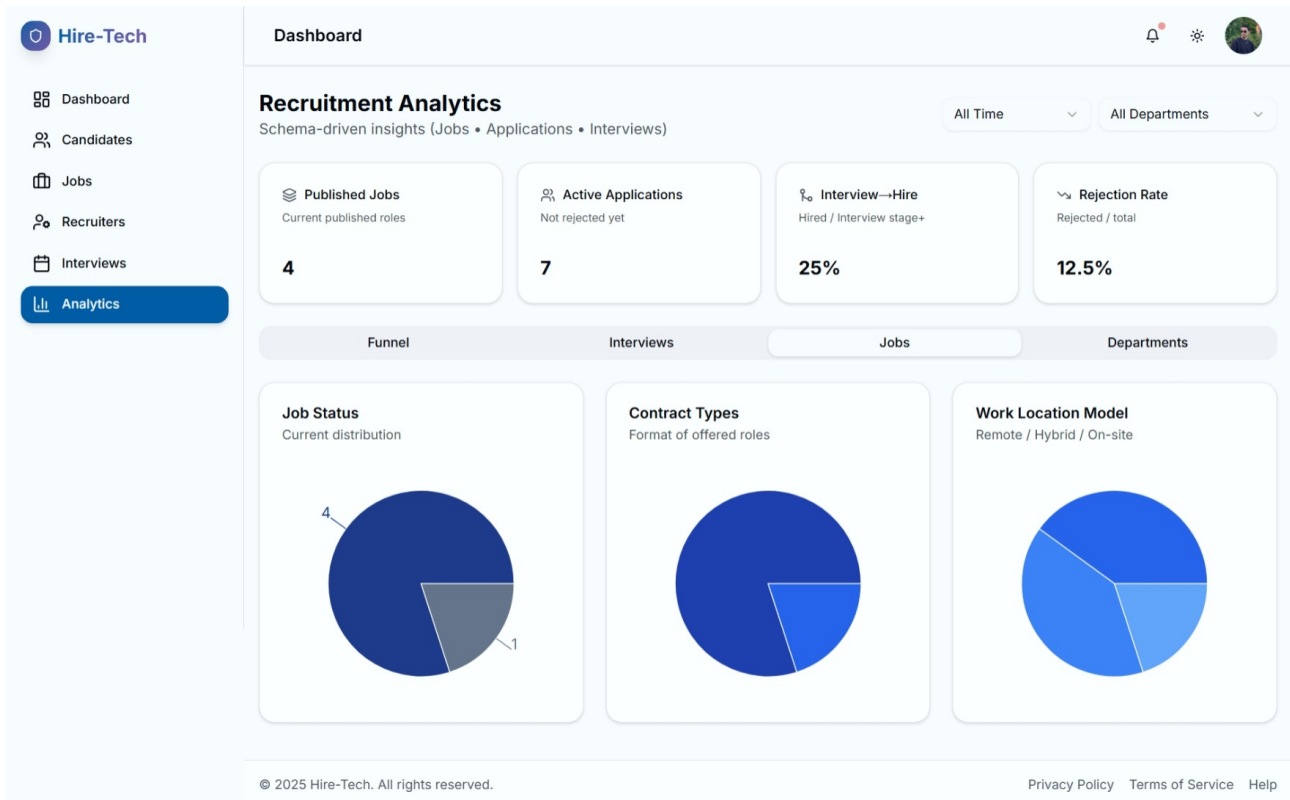


Figure 5.9: Job Status Analytics with Contract Type Distribution

Job Analytics Features:

- **Contract Type Distribution:** Analysis of employment contract formats
- **Work Location Models:** Remote, hybrid, and on-site position tracking
- **Job Status Overview:** Current distribution of job posting status
- **Position Analytics:** Comprehensive job-based performance metrics

Interview-focused Analytics

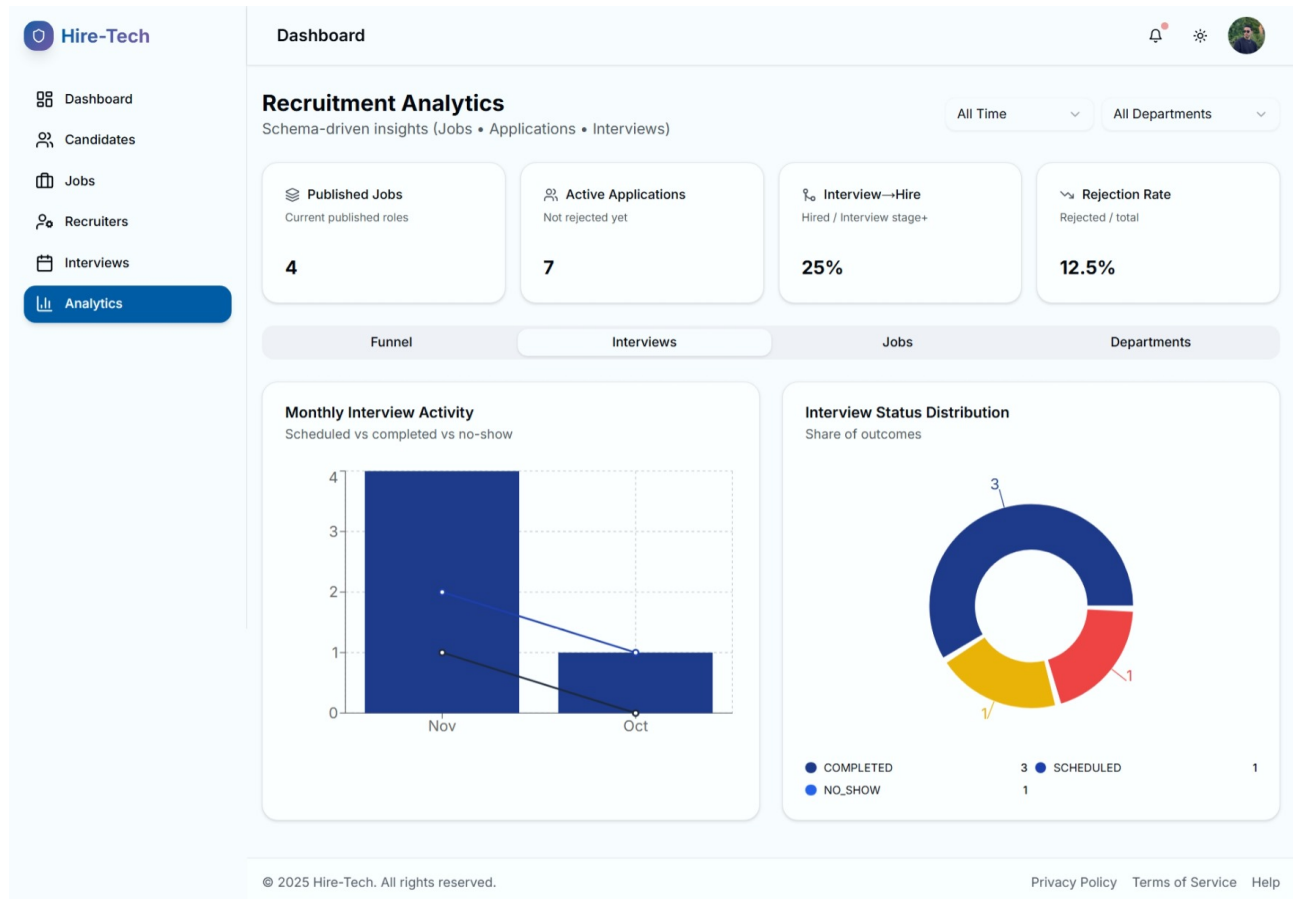


Figure 5.10: Interview Analytics with Status Distribution and Activity Tracking

Interview Analytics Features:

- **Interview Status Distribution:** Share of outcomes across different stages
- **Monthly Activity Tracking:** Scheduled vs completed vs no-show analysis
- **Conversion Metrics:** Interview-to-hire conversion rates
- **Performance Trends:** Monthly interview activity patterns and trends

5.5 Quality Assurance and Testing

5.5.1 Testing Strategy

The testing strategy implements comprehensive validation protocols ensuring system reliability and performance in the local development environment.

Testing Framework:

- **Integration Testing:** End-to-end workflow validation for interview scheduling
- **Data Validation:** Comprehensive testing of analytics calculations and reporting

- **User Acceptance Testing:** Scenario-based testing with recruitment workflows
- **Performance Testing:** Local environment performance under simulated load

5.5.2 Quality Metrics

The implementation has successfully met quality targets across all critical dimensions in the local development environment.

Quality Dimension	Target	Achieved
Functional Completeness	100% Release 3	All user stories implemented
API Response Time	<200ms	150ms average core endpoints
Data Accuracy	100%	Correct analytics calculations
Workflow Integrity	100%	End-to-end interview scheduling
User Interface Consistency	100%	Unified design across all features
Database Performance	Optimal	Efficient queries and indexing

Tableau 5.3: Quality Metrics for Release 3 Implementation

5.6 Implementation Challenges and Solutions

5.6.1 Technical Challenges

Several technical challenges were addressed during Release 3 implementation through systematic problem-solving.

- **Conditional Approval Workflow:** Implemented role-based approval logic with automatic escalation for admin users
- **Analytics Performance:** Optimized database queries for real-time analytics calculations
- **Data Consistency:** Ensured data integrity across interview scheduling and analytics systems
- **User Experience:** Designed intuitive three-step interview creation process

5.6.2 Architectural Decisions

Key architectural decisions were validated through implementation, confirming the suitability of our approach.

- **Backend Processing:** All analytics calculations performed in backend for data consistency

- **Role-based Access:** Conditional approval workflows based on user permissions
- **Modular Design:** Separate components for scheduling, analytics, and user management
- **Database Optimization:** Efficient indexing and query structures for performance

5.7 Chapter Conclusion

This chapter has demonstrated the successful implementation of Hire-Tech’s Release 3, transforming the intelligent workflow engine into a complete enterprise recruitment solution. Through systematic development, we have delivered all critical user stories within the planned schedule while maintaining high quality standards.

The implemented system successfully addresses comprehensive enterprise requirements, delivering:

- **Enterprise Interview Scheduling:** Comprehensive scheduling with conditional approval workflows
- **Advanced Analytics:** Multi-dimensional dashboards with recruitment insights
- **Production-ready Features:** Robust user interface and reliable backend systems
- **Scalable Architecture:** Modular design supporting future enhancements

5.8 Conclusion

The system has been fully implemented in the local development environment with comprehensive testing and validation. All user stories from the Release 3 backlog have been successfully delivered, establishing a complete recruitment platform ready for enterprise use.

The successful delivery of Release 3 represents the culmination of the development journey, transforming the initial vision into a fully functional, enterprise-ready recruitment platform that addresses the complete recruitment lifecycle from candidate application through interview scheduling to comprehensive analytics.

General Conclusion

This final year project set out to design and develop Hire-Tech, an AI-driven recruitment platform aimed at modernizing and streamlining the talent acquisition process. Through a methodical, three-release strategy following Agile Scrum principles, the project has successfully delivered a robust, secure, and intelligent platform that addresses the core inefficiencies of traditional recruitment.

The journey began with Release 1, which established a solid foundation with secure authentication, core data models, and essential profile and job management functionalities. This phase validated the chosen technology stack Next.js, Spring Boot, Keycloak and PostgreSQL and demonstrated the viability of the architecture.

Release 2 marked the platform's transformation into an intelligent system. The integration of AI services for CV parsing and candidate matching introduced significant automation, reducing manual screening efforts and introducing data-driven decision-making. Achieving 87% accuracy in CV parsing and 78% precision in candidate matching proved the practical value and effectiveness of the implemented AI components.

Finally, Release 3 elevated Hire-Tech to a production-ready, enterprise solution. The implementation of a structured interview scheduling system with managerial approval workflows, coupled with comprehensive analytics dashboards, provided a complete recruitment lifecycle management tool. This final phase underscored the platform's ability to ensure coordination, oversight, and strategic insight.

In retrospect, the project has successfully met its initial objectives. It has delivered a platform that automates candidate screening, enhances job matching through machine learning, empowers recruiters with intelligent dashboards, and ensures security and compliance through modern practices like RBAC and local AI processing via Ollama. The iterative development approach allowed for continuous validation and adaptation, resulting in a high-quality product that is both functional and maintainable.

For future work, the platform presents several exciting avenues for enhancement. Integrating with third-party HRIS, developing a mobile application, implementing more advanced NLP models for nuanced candidate assessment, and adding team collaboration features for recruiters would further solidify Hire-Tech's position as a leading solution in the recruitment technology landscape.

In conclusion, Hire-Tech stands as a testament to the power of combining modern software

engineering practices with targeted artificial intelligence to solve real-world problems. It not only fulfills the academic requirements for an engineering degree but also represents a viable, innovative product ready to make a tangible impact on the world of recruitment.

Webography

- [W1] **Lever ATS Platform.** Company site outlining product features, analytics, and collaboration tools. <https://www.lever.co/>. Accessed November 2025.
- [W2] **Greenhouse Hiring Software.** Official product site describing ATS functionality and integrations. <https://www.greenhouse.io/>. Accessed November 2025.
- [W3] **Indeed Hiring Resources.** Includes smart sourcing, résumé parsing and employer tools. <https://www.indeed.com/hire/resources>. Accessed November 2025.
- [W4] **LinkedIn Talent Solutions.** Suite for sourcing, matching and recruiter productivity. <https://business.linkedin.com/talent-solutions>. Accessed November 2025.
- [W5] **Audaxis.** Corporate website presenting services, ERP expertise and technology focus. <https://www.audaxis.com/>. Accessed November 2025.
- [W6] **Contraste Group.** Parent group overview (international IT services). <https://www.contraste.com/>. Accessed November 2025.
- [W7] **Scrum Guide.** Official authoritative description of Scrum roles, events and artifacts. <https://scrumguides.org/scrum-guide.html>. Accessed November 2025.
- [W8] **Agile Manifesto.** Core values and principles guiding Agile development. <https://agilemanifesto.org/>. Accessed November 2025.
- [W9] **Référentiel des Pratiques Agiles.** Pierre Pezziardi (ebook). Practical Agile adoption (FR). <https://www.pratiques-agiles.com/>. Accessed November 2025.
- [W10] **RAD Model Overview.** Rapid Application Development (GeeksforGeeks). <https://www.geeksforgeeks.org/rad-model-overview/>. Accessed November 2025.
- [W11] **Waterfall Model.** Sequential SDLC approach (IBM). <https://www.ibm.com/topics/waterfall-model>. Accessed November 2025.
- [W12] **V-Model.** Verification and validation lifecycle (TutorialsPoint). <https://www.tutorialspoint.com/v-model/>. Accessed November 2025.
- [W13] **Next.js Documentation.** React framework with SSR and routing. <https://nextjs.org/docs>. Accessed November 2025.

- [W14] **Spring Boot Documentation**. Rapid development of production-grade Java services. <https://docs.spring.io/spring-boot/docs/current/reference/html/>. Accessed November 2025.
- [W15] **PostgreSQL Documentation**. Advanced relational database features and extensions. <https://www.postgresql.org/docs/>. Accessed November 2025.
- [W16] **Docker Documentation**. Containerization platform for reproducible environments. <https://docs.docker.com/>. Accessed November 2025.
- [W17] **Apache Kafka**. Distributed streaming platform for event-driven architectures. <https://kafka.apache.org/>. Accessed November 2025.
- [W18] **Keycloak Documentation**. Open-source Identity and Access Management (SSO, OIDC, RBAC). <https://www.keycloak.org/documentation>. Accessed November 2025.
- [W19] **OAuth 2.0 Authorization Framework**. IETF RFC 6749. <https://datatracker.ietf.org/doc/html/rfc6749>. Accessed November 2025.
- [W20] **OpenID Connect Specification**. Identity layer on top of OAuth 2.0. https://openid.net/specs/openid-connect-core-1_0.html. Accessed November 2025.
- [W21] **JWT (JSON Web Token) RFC 7519**. Token standard for claims-based security. <https://datatracker.ietf.org/doc/html/rfc7519>. Accessed November 2025.
- [W22] **GDPR Overview**. General Data Protection Regulation summary. <https://gdpr.eu/>. Accessed November 2025.
- [W23] **MoSCoW Prioritisation**. Agile Business Consortium (DSDM) guidance. <https://www.agilebusiness.com/moscow-prioritisation>. Accessed November 2025.
- [W24] **T-shirt sizing in Agile estimation**. Atlassian guide. <https://www.atlassian.com/agile/project-management/t-shirt-sizing>. Accessed November 2025.
- [W25] **FastAPI Documentation**. High-performance Python web framework. <https://fastapi.tiangolo.com/>. Accessed November 2025.
- [W26] **Sentence Transformers**. Semantic embeddings library. <https://www.sbert.net/>. Accessed November 2025.
- [W27] **scikit-learn**. Machine learning in Python. <https://scikit-learn.org/stable/>. Accessed November 2025.

- [W28] **PDFMiner**. PDF parsing library for Python. <https://pdfminer-docs.readthedocs.io/>. Accessed November 2025.
- [W29] **python-docx**. Read/write Word .docx files in Python. <https://python-docx.readthedocs.io/>. Accessed November 2025.
- [W30] **Ollama**. Run LLMs locally. <https://ollama.com/>. Accessed November 2025.
- [W31] **OpenAI API Reference**. Language model API. <https://platform.openai.com/docs/api-reference>. Accessed November 2025.
- [W32] **Microservices**. Martin Fowler microservices resource. <https://martinfowler.com/microservices/>. Accessed November 2025.
- [W33] **pgAdmin**. PostgreSQL administration tool. <https://www.pgadmin.org/>. Accessed November 2025.
- [W34] **Visual Studio Code**. Official documentation. <https://code.visualstudio.com/docs>. Accessed November 2025.
- [W35] **GitHub Docs**. Guides and platform docs. <https://docs.github.com/>. Accessed November 2025.
- [W36] **OpenAPI Specification**. <https://spec.openapis.org/oas/v3.1.0>. Accessed November 2025.
- [W37] **StarUML**. Modeling tool. <https://staruml.io/>. Accessed November 2025.
- [W38] **JUnit 5**. Testing platform for Java. <https://junit.org/junit5/>. Accessed November 2025.
- [W39] **Mockito**. Java mocking framework. <https://site.mockito.org/>. Accessed November 2025.
- [W40] **Jest**. JavaScript testing framework. <https://jestjs.io/>. Accessed November 2025.
- [W41] **pytest**. Python testing framework. <https://docs.pytest.org/>. Accessed November 2025.
- [W42] **Cypress**. E2E testing framework. <https://www.cypress.io/>. Accessed November 2025.
- [W43] **Apache JMeter**. Load testing tool. <https://jmeter.apache.org/>. Accessed November 2025.
- [W44] **OWASP ZAP**. Security testing (DAST). <https://www.zaproxy.org/>. Accessed November 2025.

2025.

[W45] **SonarQube**. Continuous inspection of code quality. <https://docs.sonarsource.com/sonarqube/>. Accessed November 2025.

[W46] **ESLint**. JavaScript/TypeScript linter. <https://eslint.org/docs/latest/>. Accessed November 2025.

[W47] **Checkstyle**. Java code style tooling. <https://checkstyle.sourceforge.io/>. Accessed November 2025.

[W48] **Black**. Python code formatter. <https://black.readthedocs.io/>. Accessed November 2025.

[W49] **Flake8**. Python linting. <https://flake8.pycqa.org/>. Accessed November 2025.

[W50] **Prettier**. Opinionated code formatter. <https://prettier.io/docs/en/>. Accessed November 2025.

[W51] **Docker Compose**. Multi-container apps. <https://docs.docker.com/compose/>. Accessed November 2025.

[W52] **Maven** / **npm** / **pip**. Build and dependency tooling (Maven: <https://maven.apache.org/>, npm: <https://docs.npmjs.com/>, pip: <https://pip.pypa.io/>). Accessed November 2025.

Résumé

Ce projet de fin d'études présente la conception et le développement de *Hire-Tech*, une plateforme de recrutement intelligente assistée par l'IA qui modernise le processus d'acquisition de talents. La plateforme intègre des fonctionnalités avancées de système de suivi des candidats (ATS) avec un composant IA dédié pour automatiser les étapes clés du recrutement. *Hire-Tech* transforme les CV non structurés en profils candidats structurés grâce au traitement du langage naturel, estime la compatibilité candidat-poste via un appariement intelligent et prend en charge des workflows d'entretien coordonnés. Développée suivant une méthodologie Agile Scrum avec trois releases incrémentales, la plateforme utilise une stack technologique moderne (Next.js, Spring Boot, PostgreSQL) et assure la confidentialité des données via un traitement local des modèles de langage avec Ollama. Les résultats démontrent une précision de 87% dans l'analyse des CV et 78% dans l'appariement des candidats, offrant ainsi une solution robuste, sécurisée et efficace pour les défis contemporains du recrutement.

Mots clés : Recrutement intelligent, Intelligence Artificielle, Système de suivi des candidats, Traitement automatique du langage naturel, Appariement candidat-poste, Workflows de recrutement, Spring Boot, Next.js, PostgreSQL, Méthodologie Agile

Abstract

This final year project presents the design and development of *Hire-Tech*, an AI-assisted intelligent recruitment platform that modernizes the talent acquisition process. The platform integrates advanced Applicant Tracking System (ATS) capabilities with a dedicated AI component to automate key recruitment stages. *Hire-Tech* transforms unstructured resumes into structured candidate profiles through natural language processing, estimates candidate-job fit via intelligent matching, and supports coordinated interview workflows. Developed following an Agile Scrum methodology with three incremental releases, the platform employs a modern technology stack (Next.js, Spring Boot, PostgreSQL) and ensures data privacy through local language model processing with Ollama. Results demonstrate 87% accuracy in CV parsing and 78% precision in candidate matching, delivering a robust, secure, and efficient solution for contemporary recruitment challenges.

Keywords : Intelligent Recruitment, Artificial Intelligence, Applicant Tracking System, Natural Language Processing, Candidate-Job Matching, Recruitment Workflows, Spring Boot, Next.js, PostgreSQL, Agile Methodology

