2024-2025

Optimizing recruitement Process and HR department

Elaborated by :

Chaima Naouali

Bechir Garali

Amenallah Belhouichet

Ayoub Hanfi

Sommaire

# Project Introduction

## Introduction

## Problem Statement

## Study of the Existing

### Description of the existing solution

### Criticism of the existing solution

## Project Objective

# Business features specifications

## Actors identification

## Functional requirements

## Non Functional requirements

## Product Backlog

# System Design and architecture

## High-level architecture Diagram

## Data Model

## Technology stack choice

## Security and compliance considerations

# CI/CD and Devops Strategy

## Version control plan

## CI/CD pipeline setup plan

## Testing Strategy

# Timeline and Sprint Planning

## Sprint Overview

## Project Introduction :

## Introduction

## Recruitment and HR management are critical processes that impact the efficiency and success of organizations. However, traditional methods often face challenges such as bias, inefficiency, and limited assessment accuracy. This project aims to optimize the recruitment process and HR management by improving candidate evaluation, reducing biases, and enhancing decision-making. Through structured assessments and data-driven insights, the system seeks to streamline hiring and workforce management, ensuring better job placements and employee performance.

## Background motivation

## How can the recruitment process and HR management be optimized to improve efficiency, reduce biases, and enhance decision-making while ensuring a better experience for both candidates and employees?

## Study of the Existing solutions

## 1.Description of the existing solutions

## Currently, recruitment and HR management rely on a mix of **manual processes, traditional HR software, and standardized psychological tests** to assess candidates. These solutions, while widely used, have several limitations:

## ***Traditional HR Software*** : Limited integration with **psychological assessments** and advanced candidate evaluation methods.

## ***Psychological Tests in Recruitment*** : not commonly used but when used They lack **real-time analysis.**

## ***Challenges in Existing Solutions :*** Many traditional systems do not effectively mitigate unconscious biases in hiring decisions and Limited Data Utilization.

## 2. Criticism of the existing solutions

**1-** **Limited Integration of Psychological Assessments**

-Traditional HR software lacks seamless integration with **psychological testing**, reducing the depth of candidate evaluation.

-As a result, hiring decisions rely heavily on resumes and interviews rather than comprehensive personality and cognitive assessments.

**2-** **Lack of Real-Time Analysis in Psychological Tests**

-Psychological assessments, when used, often employ **static and outdated scoring methods** without real-time adaptation to responses.

-This limits the ability to gain **deep, personalized insights** into a candidate’s personality, work ethic, and job fit.

**3-Bias in Hiring Decisions**

-Many traditional systems fail to effectively **detect and mitigate unconscious biases**, leading to **subjective hiring decisions** that may not prioritize the most qualified candidates.

-Relying solely on resumes and manual screening increases the risk of **favoritism, gender bias, and cultural bias** in recruitment.

**4-** **Limited Data Utilization**

-HR systems often **do not fully leverage candidate and employee data** to optimize recruitment and workforce management.

-The lack of **advanced analytics** prevents organizations from identifying hiring trends, improving decision-making, and enhancing long-term employee performance.

## Project Objective :

Our project addresses these limitations by using **dynamic question selection, sophisticated response analysis, and comprehensive validation**. Additionally, it enhances recruitment by **filtering resumes** based on qualifications, experience, and relevance to job requirements. The system also gathers and analyzes the **historical data of previous employees**, focusing on their **behavioral patterns and performance**. This allows HR teams to make **data-driven hiring decisions**, ensuring better candidate-job fit and improving the overall recruitment process and the good management of HR department.

## Business features Specifications

## Actors identification :

## Candidate : have the possibility to pass psychological tests and apply to a job.

## Human Ressource(HR) : post jobs, can consult users’ test information, and users’ job results and give candidates a feedback about the result of their application.

## Super Human Ressource(HR) : have the same role as HR; in addition, they can manage Human Resources.

## Functional requirements :

|  |  |
| --- | --- |
| Actor | Functionnality |
| Candidate | • **Authentication:** The candidate must be able to register and log in to the application .• **Psychological Tests:**– The candidate must be able to take psychological tests– The candidate must be able to view their psychological test results.• **Job Search and Application:** – The candidate must be able to search for job listings.– The candidate must be able to apply for job listings.– The candidate must be able view the feedback and the result of his/her application. |
| Human Ressource(HR) | • **Authentication:** The HR must be able to register and log in to the application.• **Job Listings Management:**– HR must be able to create, modify, and delete job listings.– HR must be able to view applications received for job listings.• **User Information Consultation:** HR must be able to view user information, including psychological test results and job application results. |
| Super Human Ressource (HR) | • **Authentication:** The Super HR must be able to register and log in to the application.• **All HR Functionalities:** Super HR must be able to perform all HR tasks.• **Human Resources Management:** Super HR must be able to add and manage HR accounts. |

## Non Functional requirements :

## **Security** : Ensuring the protection of the system and its data is essential. This includes **robust user authentication, data encryption, and role-based access control** to prevent unauthorized access and potential security breaches.

## **Performance** : It should ensure that performance is ultra-responsive without a hint of disruption, reducing response times, load times, and downtime to an absolute minimum for optimal delivery of the desired seamless user experience

## **Maintainability**: challenging to accomplish without a straightforward yet reliable update and bugfix modal architecture that emphasizes long-term usefulness with this system modification. As new needs arise, the existing ones can be changed and new ones put into effect.

## **Scalability** : It should be possible to increase the number of users and the amount of data on the platform over time. It should also be able to integrate new functionalities that are developed and support them.

## Product Backlog:

## 

## System Design and architecture

## High-Level Architecture Diagram

1. **Frontend :**

* Provides an interface for **candidates, recruiters, and HR personnel**.
* Allows users to **apply for jobs, take psychological tests, and review analytics**.
* Implements **role-based access** (admin, recruiter, HR, candidate).

1. **Backend :**

* Manages authentication and role-based authorization.
* Handles job postings, applications, and AI-based rankings.
* Processes psychological test data and HR analytics.

1. **Database Layer**

* Tables based on UML:
  + **User**
  + **Candidate**
  + **Recruiter**
  + **Job**
  + **Application**
  + **AI\_Ranking**
  + **PsychologicalTest**
  + **HR\_Analytics**

1. **AI & Machine Learning**

* **Resume Filtering**: Scores candidates based on job requirements.
* **Psychological Test Analysis**: Adapts questions dynamically and analyzes responses.
* **AI Ranking System**: Ranks candidates based on **test results & AI-driven recommendations**.

1. **External Integrations**

* **Calendar API** for scheduling interviews.

|  |  |  |
| --- | --- | --- |
| **Microservice** | **Description** | **Tech Stack** |
| Auth Service | Recruiter,employee,candidate authentication and user roles. | Node.js+express.js |
| User Service | Manages user profiles ,permissions and HR roles. |  |
| Recruitement Service | Job postings, Candidate applications and resume parsings. |  |
| Psychological test service | Personnality assessments and emotional intelligence tests for candidates. |  |
| Frontend | UI for recruiters ,HR managers and employees |  |
| AI Matching Service | Candidate ranking base on their results |  |
|  |  |  |
|  |  |  |

## Data Model:

## 

## Class Diagram

## 

## Use Case Diagram

## Technology Stack choice:

## ***Frontend :*** React

## ***Backend :*** Node js+ express js

## ***DataBase :*** MongoDB

## ***Devops/CI/CD tools :***

## - **Docker** for containerization.

***-* CI/CD Pipelines** for continuous deployment.

***-* Cloud Hosting** (AWS/GCP/Azure) for scalability.

## Security and compliance Considerations :

## 1) User Authentication & Access Control

* **JWT Authentication** ensures secure access.
* **Data Encryption** secures user credentials and sensitive data

2) Data Protection & Encryption

* **End-to-End Encryption (E2EE):** Encrypt sensitive data, including resumes, test results, and AI ranking

## System CI/CD and Devops Strategy

## A . Version control plan :

Version Control System (VCS): Utilize Git as the primary version control system, hosted on platforms such as GitHub or GitLab.

***Branching Strategy:***

Main Branch (main): Stable code, always deployable to production.

* ***Development Branch (develop):*** Integration of new features before merging to main.
* ***Feature Branches:*** Each new feature is developed in its own branch (feature/feature-name).
* ***Hotfix Branches:*** For urgent bug fixes in the production environment (hotfix/issue-name).
* ***Release Branches:*** Prepare code for production releases (release/version-number).

***Code Review & Merge Requests:***

## Implement mandatory code reviews via pull/merge requests, ensuring peer review before merging.

***Commit Message Convention:***

## Follow the conventional commits standard for clear commit history (e.g., feat: add new login feature, fix: resolve login bug).

***Tagging and Versioning:***

## Use semantic versioning (e.g., v1.0.0) for consistent release management

## CI/CD pipeline setup plan :

## **CI/CD Tools:**

## Continuous Integration: Utilize GitHub Actions or GitLab CI for automating builds and tests tailored for JavaScript projects.

## Continuous Deployment: Use Docker for containerization and Kubernetes for orchestration, with deployment to cloud platforms like AWS, GCP, or Azure.

## **Pipeline Stages:**

## 1 .Build Stage:

## Install Node.js dependencies using npm or yarn.

## Compile React frontend and Node.js backend code.

## Build Docker images for both frontend and backend services.

## 2.Test Stage:

## Run unit tests using Jest (frontend) and Mocha/Chai (backend).

## Execute integration tests to ensure proper interaction between React, Node.js, and MongoDB.

## Perform static code analysis using tools like ESLint and SonarQube.

## 3.Deploy Stage:

## Deploy to a staging environment for further testing and validation.

## Upon approval, deploy to the production environment.

## **Automation and Triggers:**

## Automate pipeline triggers on push or pull requests to the develop and main branches.

## Require manual approval for deploying to production.

## **Environment Configuration:**

## Use environment variables to manage sensitive information securely.

## Maintain separate configuration files for development, staging, and production environments.

## **Monitoring & Logging:**

## Implement real-time system health checks using Prometheus and Grafana.

## Utilize centralized logging with the ELK Stack (Elasticsearch, Logstash, Kibana) for efficient log management.

## Testing strategy :

## **Types of Testing:**

## **Unit Testing**: Test individual components using Jest for the React frontend and Mocha/Chai for the Node.js backend.

## **Integration Testing**: Ensure seamless interaction between the React frontend, Node.js backend, and MongoDB database.

## **End-to-End (E2E) Testing**: Simulate real user scenarios using Cypress.

## **Performance Testing**: Use tools like Artillery or JMeter to test system performance under load.

## **Security Testing**: Conduct regular vulnerability scans using OWASP ZAP.

## **Testing in CI/CD Pipeline:**

## Automatically run unit and integration tests during the CI phase.

## Execute E2E and performance tests in the staging environment before production deployment.

## **Code Coverage:**

## Aim for at least 80% code coverage.

## Use tools like Istanbul/nyc for tracking code coverage.

## **Test Data Management:**

## Utilize mock data for unit tests.

## Use anonymized real data for integration and E2E tests to ensure realistic scenarios.

## **Bug Tracking and Reporting:**

## Integrate with project management tools like Jira to track bugs and issues.

## Generate automated test reports and share them after each pipeline run.

## 

## Timeline and Sprint Planing:

## Sprint Overview

## **Sprint 0: Preliminary Study *(1 week)***

## **Goal:** Define project scope, architecture, and initial setup. **Tasks:**

## Finalize technical stack & microservices architecture.

## Identify key business needs.

## Define database schema & API structure.

## **Sprint 1: Auth, User Management & UI *(3 week)***

## **Goal:** Implement core user authentication and management features. **Tasks:**

## **Auth Service:** JWT & OAuth for authentication.

## **User Management:** Role-based access control (Candidate, Recruiter, HR, Admin).

## **Landing Page & UI Setup:** React.js + Redux for frontend structure.

## **CI/CD & Infrastructure:** Initial DevOps setup (Docker, deployment pipeline).

## **Sprint 2 & 3: Core Business Features (Business - Recruitment and Application) *(8 week total)***

## **Goal:** Develop the core functionalities for job postings, applications, and candidate processing. **Tasks:**

## **Job Posting System:** Recruiters create and manage job listings.

## **Candidate Application:** Apply to jobs, upload resumes.

## **Application Tracking System (ATS):** Status updates (applied, reviewed, interview, hired/rejected).

## **Recruiter Dashboard:** View applications & shortlist candidates.

## **HR Management Panel:** Manage recruitment workflows & candidate progress.

## **Notification System:** Email/SMS alerts for applications, interviews.

## **Sprint 4: AI-Powered Features *(2 week)***

## **Goal:** Implement AI-powered candidate ranking & psychological

## assessments. **Tasks:**

## **AI Matching Service:** Develop ML models for candidate-job fit ranking.

## **Psychological Test Service:** Implement adaptive testing (cognitive & emotional intelligence).

## **Automated Candidate Scoring:** AI-driven ranking for recruiters.

## **HR Analytics Dashboard:** Workforce trends, hiring insights, efficiency tracking.