**Recruitment Optimizer**

**Project Specifications**

**Version *<1.0>***

***<05/02/2025>***

**Esprit Team Name : DevHive**

**Team member :**

Chaanbi Houda

Ammar Wassef

Chawech Fourat

Khmeri Oussema

Ouhichi Nourhene

## **Project Introduction**

1. **Background and Motivation**

Recruitment is essential for organizations, yet many face inefficiencies, high costs, and biased hiring decisions. HR teams struggle with managing applications, scheduling interviews, and on-boarding new hires.

With digital transformation and AI, there is a need for an intelligent system to optimize recruitment. This project streamlines hiring, reduces HR workload, and enhances decision-making through data-driven insights, improving talent acquisition, lowering costs, and enhancing the candidate experience.

1. **Study of the existing:**

|  |  |  |
| --- | --- | --- |
| Name of existing | function | existing limitations |
| **Weentime** | Weentime offers pricing details for its recruitment and HR management solutions, designed to streamline hiring processes and improve workforce management | **-Limited Advanced AI Features:** Weentime may rely on basic automation, lacking the deeper AI-driven functions (like candidate scoring and predictive analytics) that your project offers for more precise and unbiased hiring decisions.  **-Less Comprehensive Data Integration:** While your project aims to unify and fully leverage data for informed decision-making, Weentime might not integrate and analyze data as effectively, potentially limiting actionable insights. |
| **BambooHR** | is a comprehensive human resources (HR) software platform designed for small and medium-sized businesses. It centralizes various HR functions, including employee records, payroll, time tracking, and benefits management, into a single system. This integration enhances data accuracy, security, and coordination, streamlining HR processes for organizations | **-Limited Advanced AI Features**: BambooHR lacks advanced AI-powered resume screening and candidate matching capabilities, leading to more manual work during initial candidate selection.  **Limited Integrations**: a limited number of native integrations with third-party applications, potentially posing challenges for businesses that rely heavily on other tools for daily operations |

1. **Project Objective:**

This project aims to improve the efficiency of the recruitment process and optimize HR department operations by leveraging automation and data-driven decision-making.

Through the implementation of a digital platform, the system is designed to:

* **Automate candidate screening** by analyzing resumes and ranking applicants based on predefined criteria.
* **Accelerate the hiring process** using AI-powered algorithms to match job requirements with the most suitable candidates.
* **Support HR decision-making** by providing real-time analytic and performance insights for better workforce management.
* **Enhance the candidate experience** by ensuring a transparent, efficient, and user-friendly recruitment process.
* **Streamline administrative tasks** by automating interview scheduling, document management, and employee on-boarding.

By optimizing these processes, the project aims to create a more effective, time-efficient, and cost-saving HR management system, ultimately leading to improved workforce quality and better organizational performance.

## **Business Features Specifications**

1. **Functional Requirements:**

1. **Filter job applications**

Automated CV screening: Implement automated pre-selection based on predefined criteria (skills, experience).

2. **Manage application progress status**

Send email notifications to candidates (new, in progress, interview, offer, hiring).

**3. Manage communication between teams**

Facilitate exchanges and sharing of notes, feedback, and documents between HR, managers, and other stakeholders via an integrated messaging system.

4. **Manage departments, teams, and roles**

-Create/modify departments and teams.

-Assign custom access rights based on roles (HR, manager, admin).

**5. Manage leaves**

-Employees can submit leave requests.

-The department head can accept or reject requests via a dedicated interface.

-Automatic notifications to employees after validation.

6. **Manage events/workshops**

-Schedule and manage events:

-Create events (recruitment, internal training).

-Notifications sent to participants by email.

7. Manage the recruitment process

-An HR team member conducts interviews via an integrated Zoom link.

-Facial analysis during video interviews to generate a report on the candidate’s engagement.

8. Manage complaints

-Employees can submit complaints related to:

-Working conditions.

-Technical issues (e.g., denied access to a tool).

-Internal conflicts.

-Possible attachments (e.g., screenshots, documents).

1. **Non-Functional Requirements:**

1. Performance:

* Response Time: The main operations (CV filtering, status display, report generation) must be executed quickly to ensure a smooth user experience, even during peak loads.
* Data Processing: The system must be capable of handling a large volume of applications.

2. Scalability:

* Concurrent Capacity: The system must efficiently manage a large number of simultaneous connections, especially during periods of intense recruitment**.**

3. Security:

* Strong Authentication: Implement strong authentication (possibly multi-factor) and a role management system to restrict access to sensitive information to authorized users.

4. Maintainability:

* Continuous Integration and Automated Testing: Set up a CI/CD pipeline to automate tests (unit, integration, and performance) and deployments, ensuring consistent quality and rapid detection of regressions.

1. **User stories (product backlog):**

**Une image contenant texte, ligne, nombre, Police

Description générée automatiquement**

**System Design and Architecture**

## **High-Level Architecture Diagram:**

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

**2-Data Model:**

**Une image contenant diagramme, Plan, texte, Dessin technique

Le contenu généré par l’IA peut être incorrect.**

## **3-Technology Stack choice:**

The following technologies, frameworks, and tools have been selected for the development of the project:

#### ****Frontend****

* **React.js** – A JavaScript framework for building dynamic and reusable user interfaces. It ensures efficient rendering with a virtual DOM and supports component-based architecture.
* **React Router** – For client-side navigation and routing.
* **Axios** – For handling HTTP requests.
* **Styled Components / Tailwind CSS** (if applicable) – For efficient and maintainable styling.

#### ****Backend****

* **Node.js** – A JavaScript runtime that allows server-side execution of JavaScript for high-performance, scalable applications.
* **Express.js** – A lightweight and flexible web framework for building APIs and backend services.
* **JWT (JSON Web Token)** – For authentication and security.

#### ****Database****

* **MongoDB** (if using NoSQL) – A document-based database for flexible and scalable data storage.

#### ****DevOps / CI/CD Tools****

* **Git & GitHub** – For version control and collaborative development

1. **Security and Compliance Considerations**

**-Authentication and Access Control:**  
Implement strong authentication, ideally multi-factor, and a role management system to restrict access to sensitive data only to authorized users.

**-Data Encryption:**  
Ensure that data at rest (in databases) and in transit (via HTTPS) is encrypted to protect personal and confidential information.

**CI/CD and DevOps Strategy**

1. **Version Control Plan:**

The project follows a **Git-based version control strategy** using **GitHub** for repository management. The chosen **branching strategy** is **Git Flow**, which includes:

* **Main branch (**main**)** – Stable production-ready code.
* **Development branch (**dev**)** – Active development and integration.
* **Feature branches (**feature/branch-name**)** – For new features, merged into dev upon completion.
* **Bugfix branches (**bugfix/branch-name**)** – For fixing issues before merging into dev.
* **Release branches (**release/branch-name**)** – For final testing before merging into main.
* **Hotfix branches (**hotfix/branch-name**)** – For urgent fixes applied directly to main.

1. **CI/CD Pipeline Setup Plan:**

The **CI/CD pipeline** is designed to automate the development lifecycle, ensuring efficient and reliable deployment. It includes:

**Continuous Integration (CI):**

* + **GitHub Actions** triggers on push and pull requests.
  + **Automated testing** (unit and integration tests) runs on every commit.
  + **Linting and static code analysis** for code quality.

**Continuous Deployment (CD):**

* + **Build automation** using Docker.
  + **Deployment to staging environment** for final validation.
  + **Production deployment** via cloud services (e.g., AWS, Heroku, or Vercel for frontend).

1. **Testing Strategy:**

To ensure code reliability and system stability, the following testing strategy is implemented:

* **Unit Testing:**
  + Frontend: Jest + React Testing Library.
  + Backend: Mocha/Chai or Jest for API and business logic validation.
* **Integration Testing:**
  + API tests using Postman or Supertest.
  + Database testing with mock data.
* **End-to-End (E2E) Testing:**
  + Cypress or Selenium for simulating real user interactions.
* **Automated Testing in CI/CD:**
  + Tests run automatically on every push and merge request.
  + Failed tests block deployments to ensure stability.

This strategy ensures a streamlined development process, reducing manual intervention and increasing deployment efficiency.

## **Timeline and Sprint Planning**

1. **Sprint Overview:**

The project follows an **Agile Scrum methodology**, with **bi-weekly sprints** to ensure continuous development, testing, and iteration. Each sprint focuses on specific business features to gradually build a fully functional recruitment optimization system.

|  |  |  |
| --- | --- | --- |
| Sprint | Duration (weeks) | Main Focus |
| 1 | 1-2 | |  | | --- | |  |  |  | | --- | | Environment Setup, Auth, Basic User Management | |
| 2 | 2 | Communication, Departments & Roles Send a response for a complaint Leaves Management Events/Workshops |
| 3 | 3-4 | CV Filtering, Application Progress Zoom Integration, Facial Analysis |