| HIGHER INSTITUTE OF TECHNOLOGICAL STUDIES OF CHARGUIA |  |
|---|---|

# *INTEGRATION PROJECT*


***By: Mohamed Anas Ayari***

***Mohamed Taleb Mouelhi***

***Mohamed Amine Jabou***

***Yassine Ghzela***

***Taissir Bouzidi***


*Applied License in Information Technology*

*Development of Information Systems*

*2024 - 2025*

# Table of Contents

# Tables of Figures

# Table of Tables

# General Introduction

In today's competitive job market, students and recent graduates face significant challenges in securing internships or job placements within a short time frame. Traditional application processes are often time-consuming and complex, leading to frustration as applicants try to connect with the right opportunities. Furthermore, the sheer volume of applications can make it difficult for recruiters to identify the most suitable candidates. To address these issues, we have developed a web application designed to streamline and digitize the entire recruitment process.

Our solution leverages an AI-powered system that allows users to apply for positions online, simply by uploading their resumes. The application's intelligent algorithms analyze each CV, identify key skills and qualifications, and match candidates with job openings that align with their profiles. Additionally, the system provides personalized recommendations, guiding users toward positions that best suit their experience and career goals. By integrating digital tools and AI, our platform aims to facilitate quicker, more efficient job matching, ultimately easing the journey from application to employment for candidates and recruiters alike.

# Chapter 1: Project Framework

This chapter establishes the groundwork for the project by exploring the identified problem, reviewing current solutions, and presenting our innovative approach. It aims to set the stage for the entire project by highlighting its primary objective and defining the essential requirements that will shape the development journey.

## I. Problem

In the current landscape, the recruitment process is often lengthy, complex, and inefficient. Recruiters face challenges in quickly finding suitable candidates and are required to manually sift through large volumes of applications. Similarly, candidates spend a considerable amount of time searching for job postings that align with their skills, which extends the job search process.

**Key Challenges :**

- Difficulty for recruiters in quickly filtering qualified candidates.
- Lack of visibility for candidates on job opportunities suited to their skills.
- Manual and tedious management of applications by recruiters.

## II. Proposed Solution

The solution is to develop a platform powered by artificial intelligence (AI) that can automatically match candidates' skills with available job offers. Recruiters will be able to post job opportunities and easily filter applications, while candidates will receive automatic notifications for job offers that match their profiles.

## III. Specifications

### 1. Project Objectives

The main objective is to simplify and automate both the recruitment and job search processes by offering smart matching between candidates and available positions.

**Specific Objectives :**

- Allow job postings with customized filters (location, salary, contract type).
- Enable candidates to submit their resumes and automatically analyze their skills.
- Leverage AI to suggest matches between candidates’ skills and job requirements.

- Send automatic notifications to candidates about new, relevant job postings.

## 2. Targets

The intended users of the platform include:

- Recruiters: Seeking to automate and optimize their recruitment processes.
- Job seekers: Looking to easily find job offers that correspond to their skillsets.

## 3. Functional requirements

- Job posting system with filters for location, salary, and contract type.
- Resume submission system with automatic skills analysis.
- AI-based matching engine to align candidates' skills with job descriptions.
- Automatic notifications sent to candidates when relevant job opportunities are posted.

## 4. Functional non-requirements

- **Performance**: The platform should quickly process searches and resume analyses, ensuring a smooth user experience.
- **Security**: Protecting personal data (GDPR compliance) is crucial. Candidate and recruiter data should be encrypted and safeguarded against cyberattacks.
- **Scalability**: The platform must be scalable to accommodate an increasing number of users and job postings.
- **Usability**: The interface should be simple, accessible, and easy to use for non-technical users.

# IV. Backlog Prioritization and Sprint Planning

## 1. Product Bocklog

Table 1- Product Backlog

| ID | User story | BV | Effort | Priority |
|---|---|---|---|---|
| 3 | As a recruiter, I want to receive recommendations for candidates with transferable skills so that I can consider a broader pool of candidates. | High | 8 | 1 |
| 1 | As a candidate, I want the system to automatically match my skills with job descriptions so that I can find relevant offers easily. | High | 8 | 2 |
| 6 | As a recruiter, I want to manage job postings by adding, deleting, and modifying specific requirements for each position so that the AI engine can use these criteria to improve candidate matching and refine the selection process. | High | 5 | 3 |
| 9 | As a candidate, I want to view a list of job matches ranked by relevance so that I can prioritize my applications. | High | 5 | 4 |
| 8 | As a candidate, I want to receive feedback on my resume status so that I can understand my progress in the recruitment process. | Medium | 5 | 5 |
| 4 | As a recruiter, I want to receive notifications when candidates matching my job requirements apply so that I can review them promptly. | Medium | 5 | 6 |
| 2 | As a recruiter, I want to modify job offers after posting so that I can make necessary updates. | Medium | 3 | 7 |
| 7 | As a recruiter, I want to view previously posted jobs so that I can track candidate responses. | Medium | 3 | 8 |
| 10 | As a candidate, I want the system to suggest skill-building resources when I fall short of job requirements so that I can improve my profile. | Low | 8 | 9 |

| | | | | |
|---|---|---|---|---|
| 5 | As a recruiter, I want to automatically refresh expired job postings so that the offers remain visible. | Low | 5 | 10 |
| | | **velocity** | 55 | |

## 2. Distribution of User Stories by Sprint

Table 2- Sprint Planning

| Id | Effort (Fibonacci) | Sprint number | Period |
|---|---|---|---|
| 1 | 8 | 1 | September 23rd,2024-October 7th, 2024 |
| 3 | 8 | | |
| 6 | 5 | 2 | October 7th,2024- October 21st,2024 |
| 9 | 5 | | |
| 8 | 5 | | |
| 4 | 5 | | |
| 2 | 3 | 3 | October 28th,2024 – November 12th,2024 |
| 7 | 3 | | |
| 10 | 8 | | |
| 5 | 5 | | |

This chapter sets a solid foundation for the project by defining recruitment challenges and presenting an AI-driven solution to optimize both hiring and job-seeking processes. The detailed objectives, specifications, and structured planning provide a comprehensive roadmap for developing an efficient and scalable platform.

# Chapter 2: Sprint1

Chapter 2 outlines the first sprint, focusing on implementing key features for candidates and recruiters. It covers the planning, design, implementation, and testing phases, providing a roadmap for future development.

## I. Sprint backlog

The Sprint Backlog below details all tasks prioritized for completion in the first sprint. Each user story is broken down into specific tasks to clarify work requirements and ensure the team stays focused on delivering core functionalities within the sprint timeframe. This organized approach helps streamline the workflow and track progress effectively.

| ID | Description | Tasks | Time Estimation |
|---|---|---|---|
| 4 | As a recruiter, I want to receive recommendations for candidates with transferable skills so that I can consider a broader pool of candidates. | **T1**: Defining Requirements<br>**T2**: Analyzing Requirements<br>**T3**: Solution Design: Class Diagram & Relational Schema<br>**T4**: Backend Development & Unit Testing<br>**T5**: Frontend Development & Unit Testing<br>**T6** : Documentation | 1 H<br>30min<br>2H<br>10min<br>4H<br>30min |
| 1 | As a candidate, I want the system to automatically match my skills with job descriptions so that I can find relevant offers easily. | **T1**: Backend Development & Unit Testing<br>**T2**: Frontend Development & Unit Testing | 5 H<br>1H |

| | | | |
|---|---|---|---|
| | | **T3** : Documentation | 1H |

# II. Analysis

## 1. Use Case Diagram

This use case diagram illustrates interactions between candidates, recruiters, and an AI system within a job-matching platform. Candidates can view, filter, and apply for job matches, while recruiters receive candidate recommendations that include an analysis of transferable skills, facilitated by the AI system.

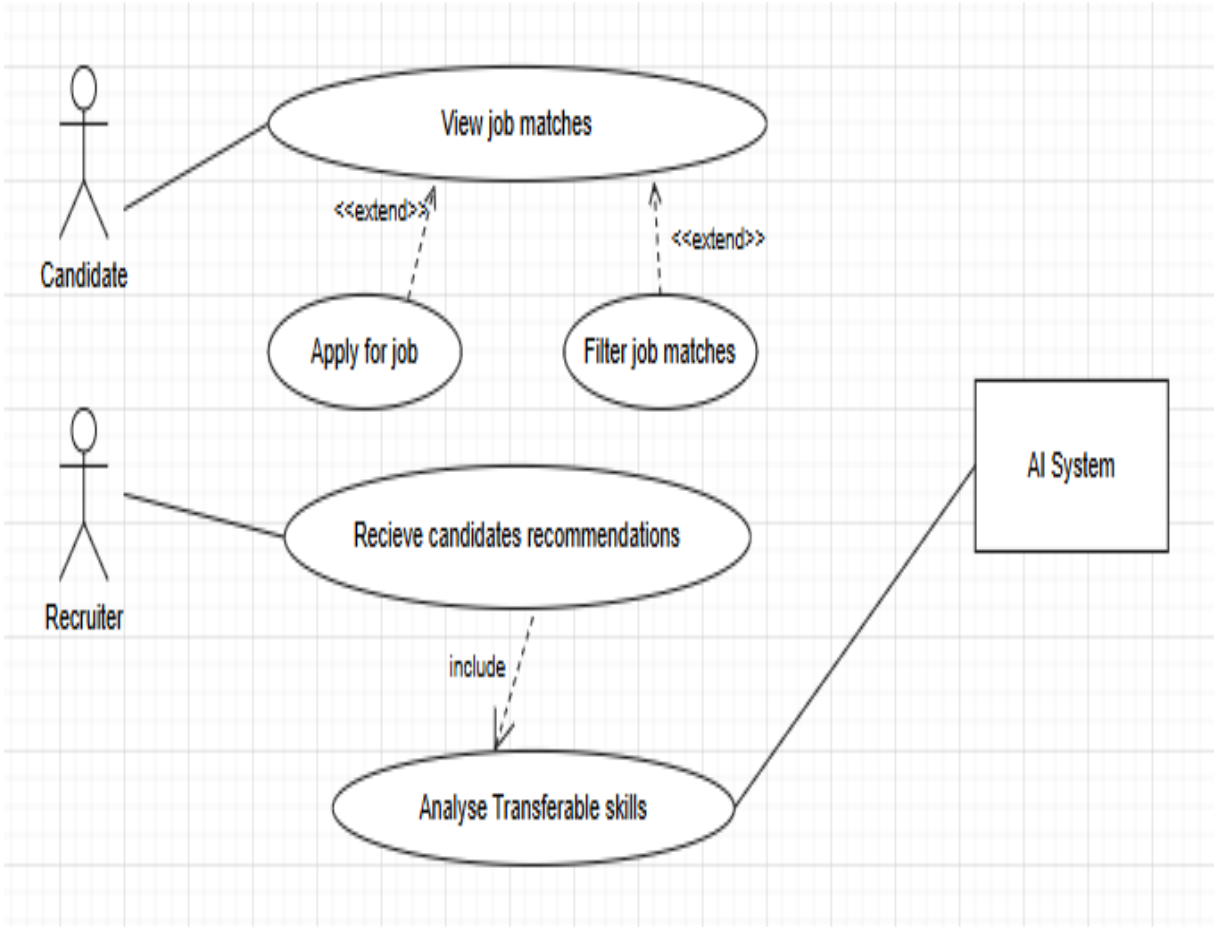


**Figure 1- Sprint 1 : use case**

2. **Object Sequence diagram <<View job matches>>**

This diagram illustrates a job matching system where the Candidate's skills are compared against multiple JobOffers through a loop process. The system retrieves job skills for each offer, compares them with the candidate's skills, and returns a list of matching jobs.

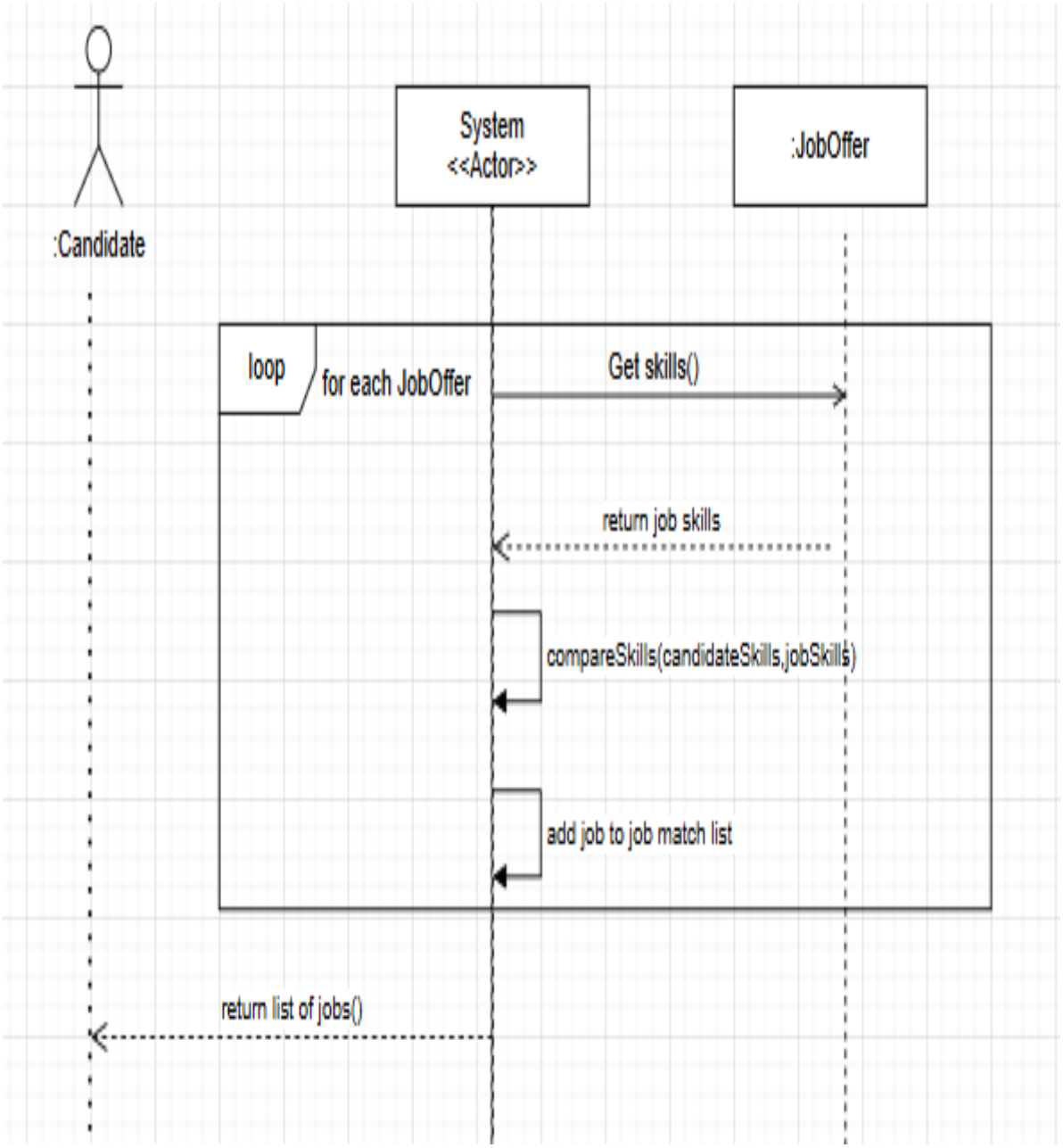


**Figure 2- Objet Sequence Diagram For "view job matches"**

# II. Conception

## 1. Class diagram

This diagram illustrates a job matching system where the Candidate's skills are compared against multiple JobOffers through a loop process. The system retrieves job skills for each offer, compares them with the candidate's skills, and returns a list of matching jobs.

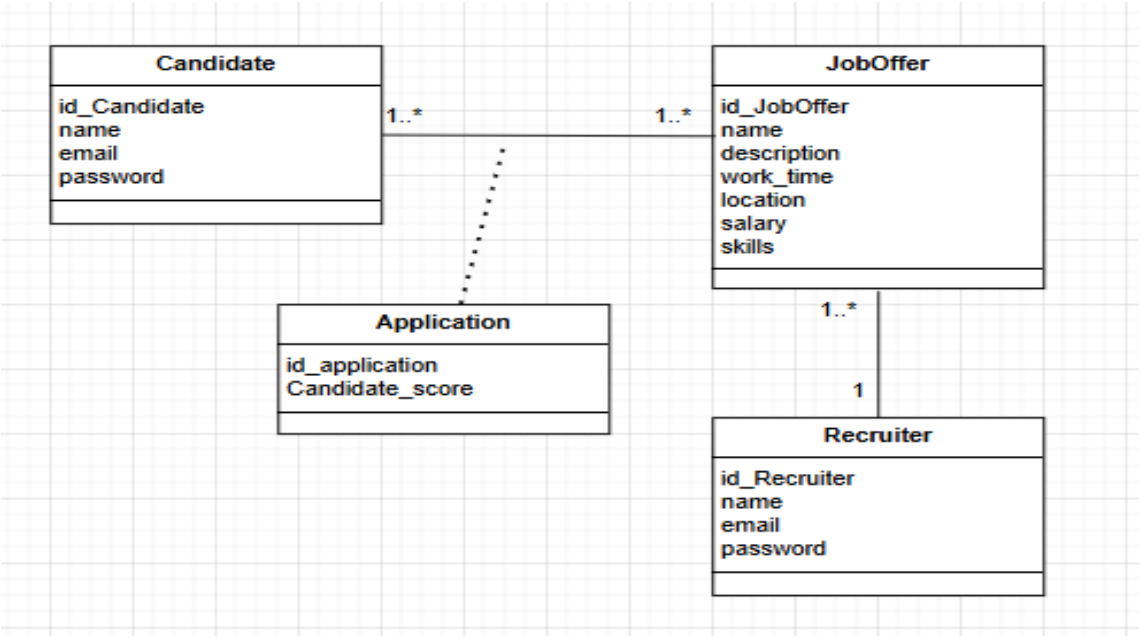


**Figure 3: Sprint 1-class diagram**

## *2.* Database schema

The database schema is carefully designed to mirror the class diagram, capturing essential data relationships to optimize both storage and retrieval. By structuring entities and their connections thoughtfully, the schema enhances data integrity and supports efficient querying, ensuring a robust and scalable backend for the system.

**Candidate** (id_candidate,name,email,password)

**Recruiter** (id_recruiter,name,email)

**Joboffer**(id_joboffer,
name,description,worktime,location,Salary,skills,#id_recruiter,#id_candidate,#id_application)

**Application** (id_application, candidate_score, #id_Candidate, #id_joboffer)

# III. Implementation and testing

This section outlines the technical environment and testing procedures utilized during the first Sprint. It presents the tools and technologies used to develop the initial features and describes the tests conducted to verify that functionalities perform as expected.

## 1. Software Environment

The table below outlines the key components of our development environment for Oasis, highlighting the technologies and tools chosen to ensure efficiency, scalability, and support for agile practices.

| Component | Technology |
|---|---|
| Frontend | Angular |
| Backend | Django |
| Database | Mysql |
| Development Environment | Visual Studio Code |
| API Testing | Postman |

## 2. Tests

Testing using Postman the user story 4 ensured that recruiters receive accurate and relevant candidate recommendations based on transferable skills.

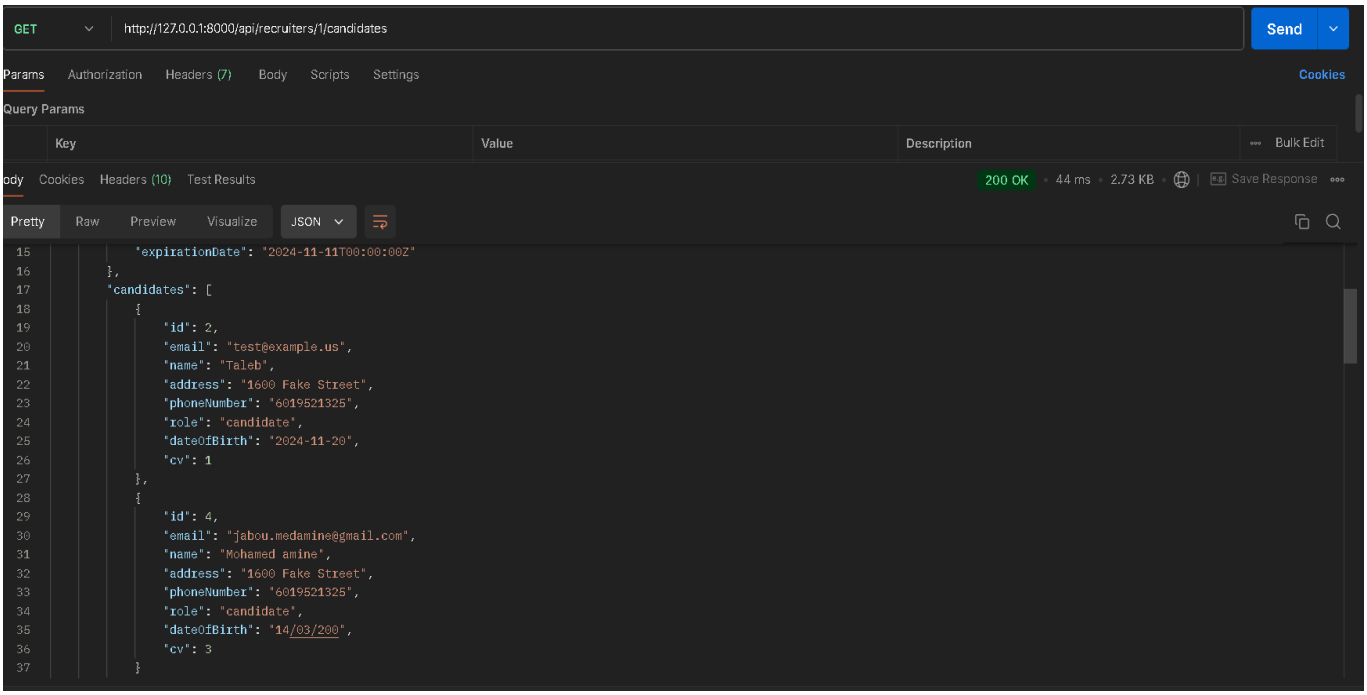


**Figure 4: Sprint 1 : unit test**

Testing the user story 1 verified that candidates receive accurate job matches based on their skills.

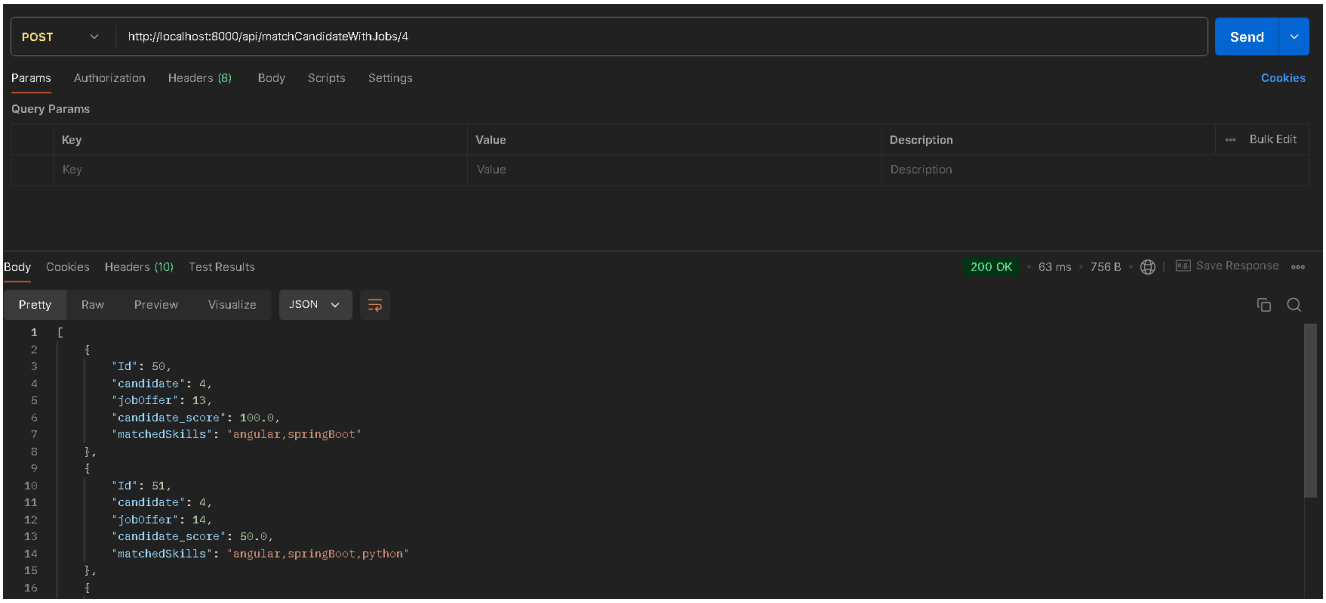


**Figure 5: Sprint 1 unit test**

# IV. Sprint Review

## 1. Interfaces

This list of jobs matching dashboard enables recruiters to consider candidates with transferable skills by displaying candidate profiles and their skill.

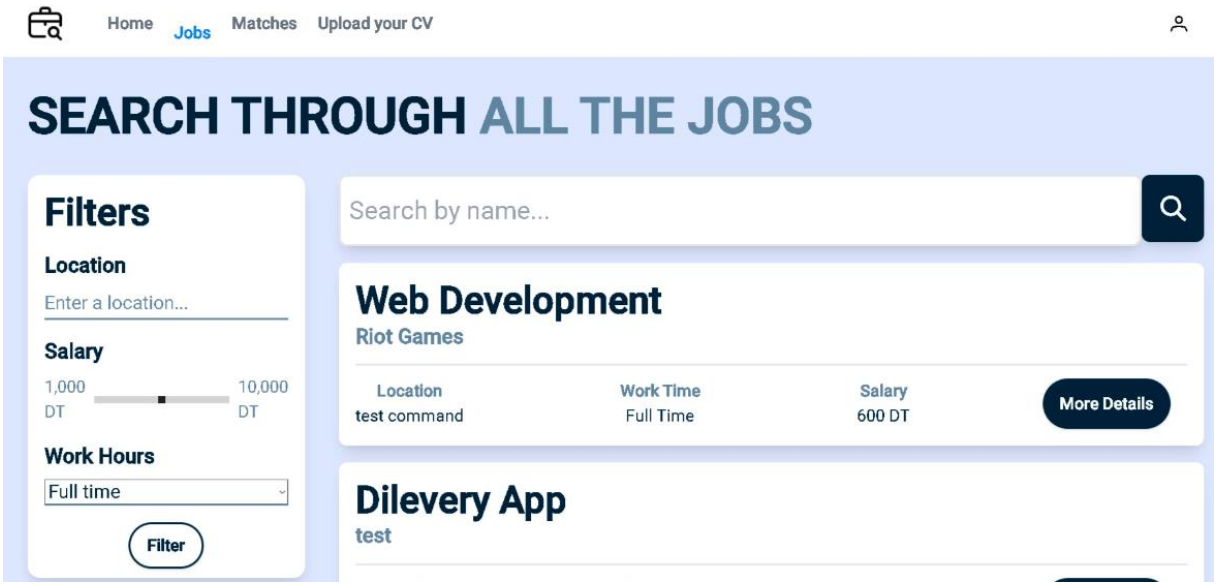


**Figure 6: List of Job Offer**

The system's ability to identify qualified applicants beyond an exact skills match supports the recruiter in building a diverse talent pool.

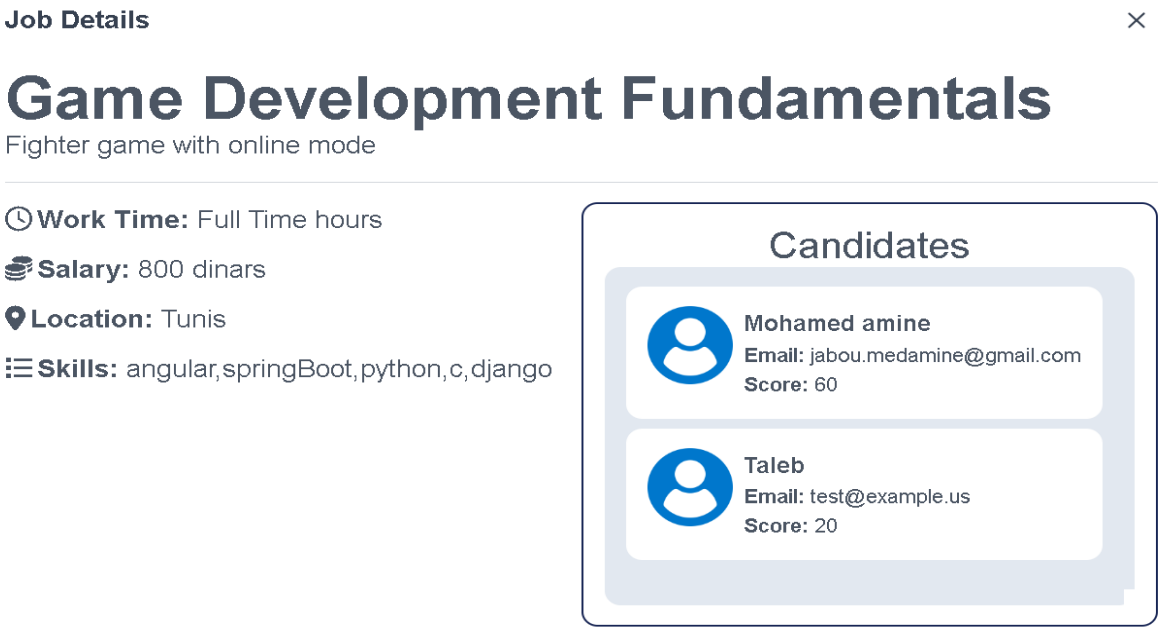


**Figure 7: Candidate Talent Matching Dashboard**

### 2. Updated Product Backlog

The updated product backlog incorporates the tasks completed in Sprint 1, along with new user stories identified and the remaining tasks to be addressed in upcoming sprints. It prioritizes functionalities according to the evolving needs of the project.

Table 3: Product Backlog -version 2

| ID | User story | BV | Effort | Priority |
|---|---|---|---|---|
| 11 | As a candidate, I want to submit my resume so that the AI system can analyze it | High | 5 | 1 |
| 6 | As a recruiter, I want to manage job postings by adding, deleting, and modifying specific requirements for each position so that the AI engine can use these criteria to improve candidate matching and refine the selection process. | High | 5 | 2 |
| 9 | As a candidate, I want to view a list of job matches ranked by relevance so that I can prioritize my applications. | High | 5 | 3 |

| | | | | |
|---|---|---|---|---|
| 8 | As a candidate, I want to receive feedback on my resume status so that I can understand my progress in the recruitment process. | Medium | 5 | 4 |
| 4 | As a recruiter, I want to receive notifications when candidates matching my job requirements apply so that I can review them promptly. | Medium | 5 | 5 |
| 2 | As a recruiter, I want to modify job offers after posting so that I can make necessary updates. | Medium | 3 | 6 |
| 7 | As a recruiter, I want to view previously posted jobs so that I can track candidate responses. | Medium | 3 | 7 |
| 10 | As a candidate, I want the system to suggest skill-building resources when I fall short of job requirements so that I can improve my profile. | Low | 8 | 8 |
| 5 | As a recruiter, I want to automatically refresh expired job postings so that the offers remain visible. | Low | 5 | 9 |
| 12 | As a recruiter I want to have authentication for security purposes | Low | 3 | 10 |

### 3. Remaining Sprints Planning

Table 4: Sprint 2 &3 planning

| ID | Effort (Fibonacci) | Sprint Number | Period |
|---|---|---|---|
| 11 | 5 | 2 | October 7th,2024- October 21st,2024 |
| 6 | 5 | | |
| 9 | 5 | | |
| 8 | 5 | | |
| 4 | 5 | | |
| 2 | 3 | | |

| | | | |
|---|---|---|---|
| 7 | 3 | 3 | October 28th,2024 – November 12th,2024 |
| 10 | 8 | | |
| 5 | 5 | | |
| 12 | 3 | | |

## IV. Sprint Retrospective

The sprint retrospective reviewed the processes and approaches used in Sprint 1, focusing on areas for growth. The team reflected on what worked well, what challenges arose, and identified strategies to enhance performance and teamwork in the upcoming sprint.

### 1. Retrospective Table

The retrospection table captures key takeaways from Sprint 1, emphasizing successes, areas needing improvement, and action steps for the next sprint.

| Went Good | To Improve | Action Plan |
|---|---|---|
| -Communication | -Point Estimation<br>-Stress Management<br>-Time Management<br>-Fast Problem<br>-Resolve Punctuality | -Set Channels for Daily Scrum<br>-Set Deadline for meets<br>-Do better research before assigning point estimation |

### 2. Burn Down Chart

The burndown chart displays the remaining work relative to the sprint timeline, offering insights into the team's task completion efficiency.

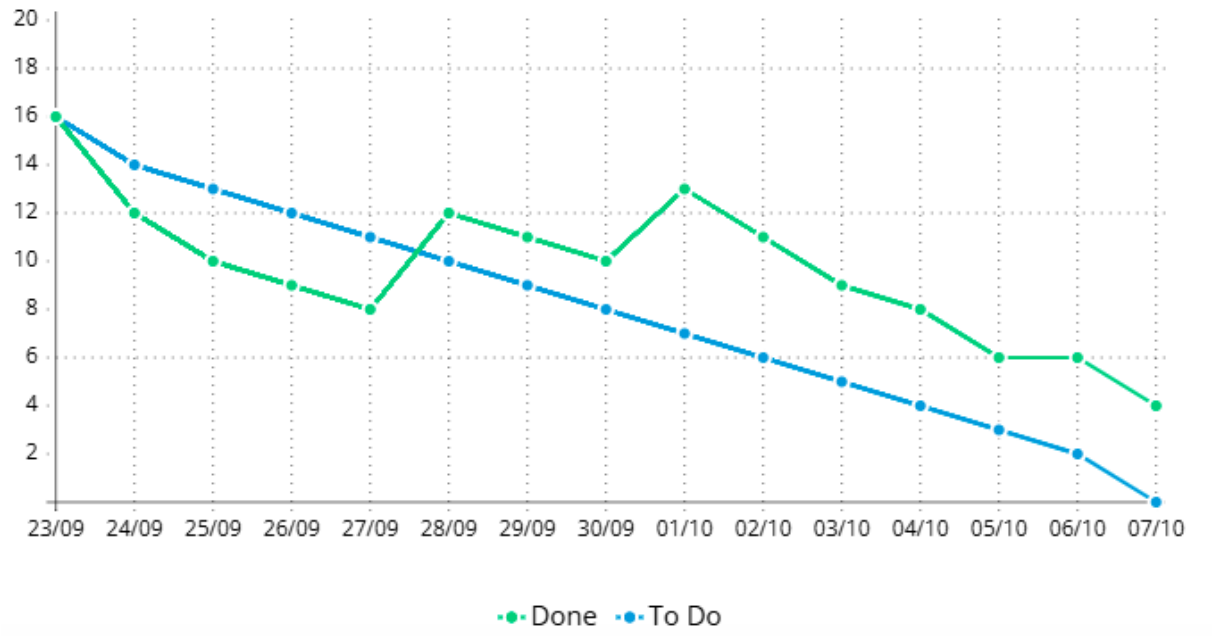


**Figure 8: Sprint 1-Burndown Chart**

## 3. Burn Up Chart

The burn-up chart highlights progress toward the project goals, displaying both completed tasks and the remaining workload.

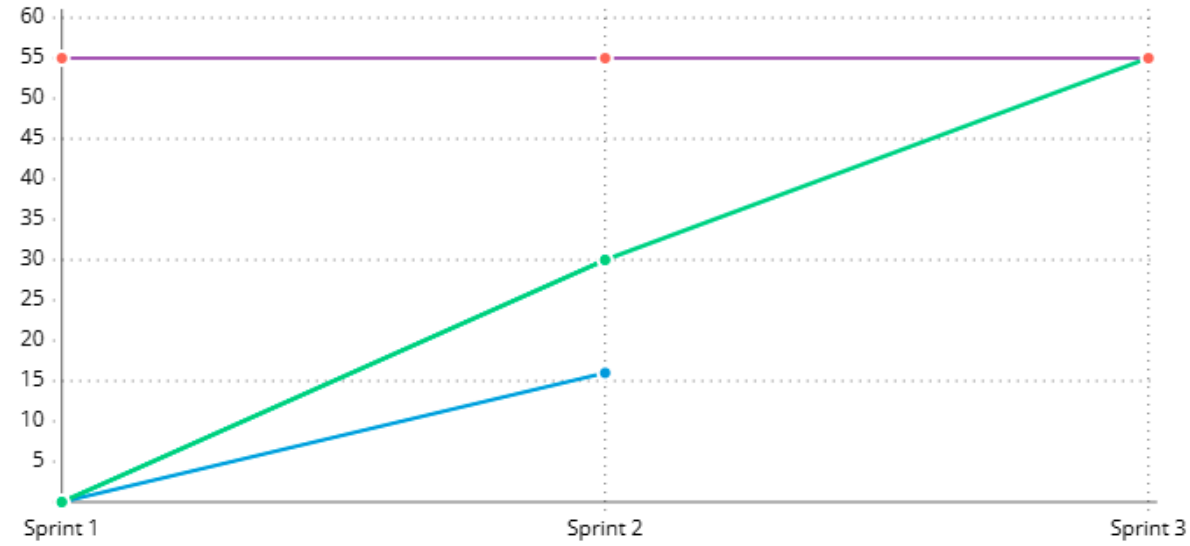


**Figure 9: Sprint 1- Brun up chart**

Chapter 2 highlights the accomplishments and challenges of Sprint 1, offering insights into areas for improvement. It sets a strong foundation for future sprints, refining processes and aligning with the project's overall goals.

# Chapter 3: Sprint2

This chapter presents the second sprint in the development process, outlining the tasks completed and the deliverables necessary to achieve the goals. It serves as a practical guide to understanding the stages of project development.

## I. Sprint Backlog

This section offers a detailed overview of the tasks and objectives established for Sprint 2, emphasizing user stories that align with the project's continuous development. The tasks are prioritized to ensure the team remains focused on delivering key components effectively within the sprint timeline.

| ID | Description | Tasks | Time Estimation |
|---|---|---|---|
| 11 | As a candidate, I want to submit my resume so that the AI system can analyze it | **T1**: Defining Requirements<br>**T2**: Analyzing Requirements<br>**T3**: Solution Design: Class Diagram & Relational Schema<br>**T4**: Backend Development & Unit Testing<br>**T5**: Frontend Development & Unit Testing<br>**T6** : Documentation | 1 H<br>30min<br>2H<br>1 H<br>2H<br>1H |
| 6 | As a recruiter, I want to manage job postings by adding, deleting, and modifying specific requirements for each position so that the AI engine can use these criteria to improve | **T1**: Backend Development & Unit Testing<br>**T2**: Frontend Development & Unit Testing<br>**T3 :** Integrale Test<br>**T4** : Documentation | 2H<br>2H<br>1H<br>1 H |

| | | | |
|---|---|---|---|
| | candidate matching and refine the selection process. | | |
| 9 | As a candidate, I want to view a list of job matches ranked by relevance so that I can prioritize my applications. | Same tasks as user Story 6 | 6H |
| 8 | As a candidate, I want to receive feedback on my resume status so that I can understand my progress in the recruitment process. | Same tasks as user Story 6 | 6H |
| 4 | As a recruiter, I want to receive notifications when candidates matching my job requirements apply so that I can review them promptly. | Same tasks as user Story 6 | 6H |

# II. Analysis

## 1. Use Case Diagram

The use case diagram illustrates the system's evolution in Sprint 2, highlighting newly introduced functionalities. As additional features were implemented, the diagram was updated to represent the increasing complexity and shifting requirements of the system.

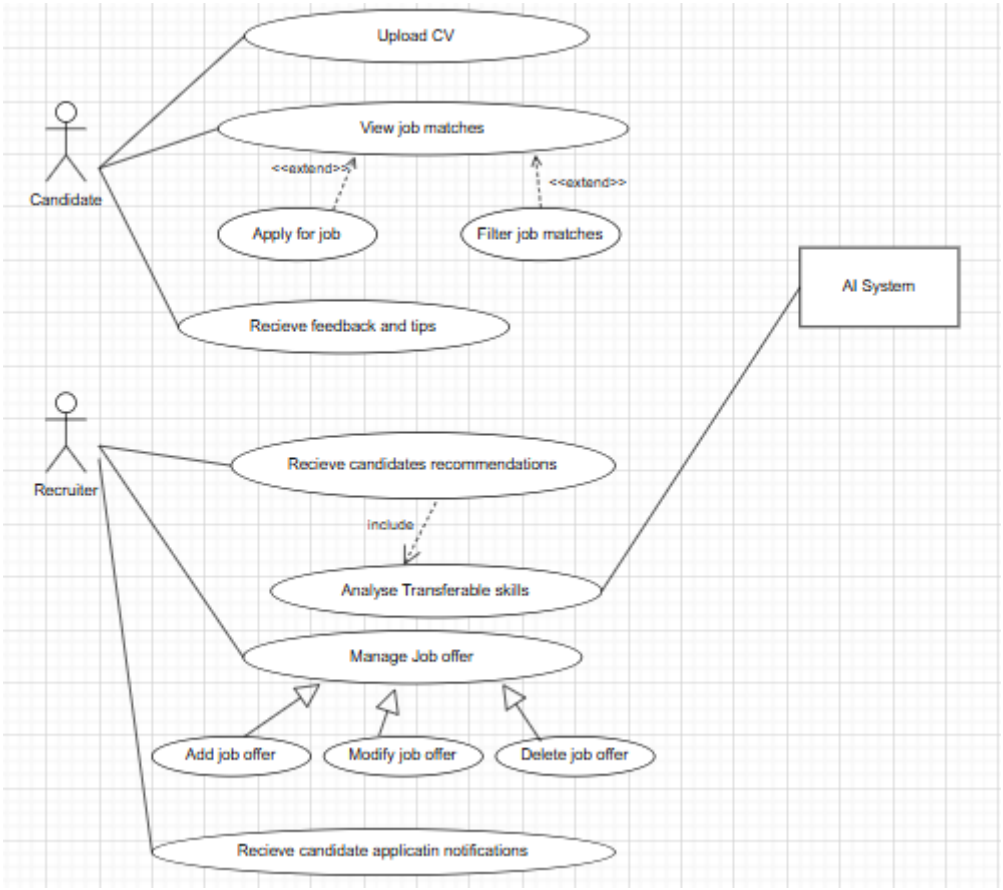


**Figure 10- Sprint 2 : Use Case Diagram**

# III. Conception

## 1. Class Diagram

The class diagram shows the relationships between entities in the recruitment system, including Candidate, Recruiter, JobOffer, and Application, with added features like CV and Match Skills for enhanced functionality. It illustrates how candidates, job offers, and applications interact through notifications and matching scores.

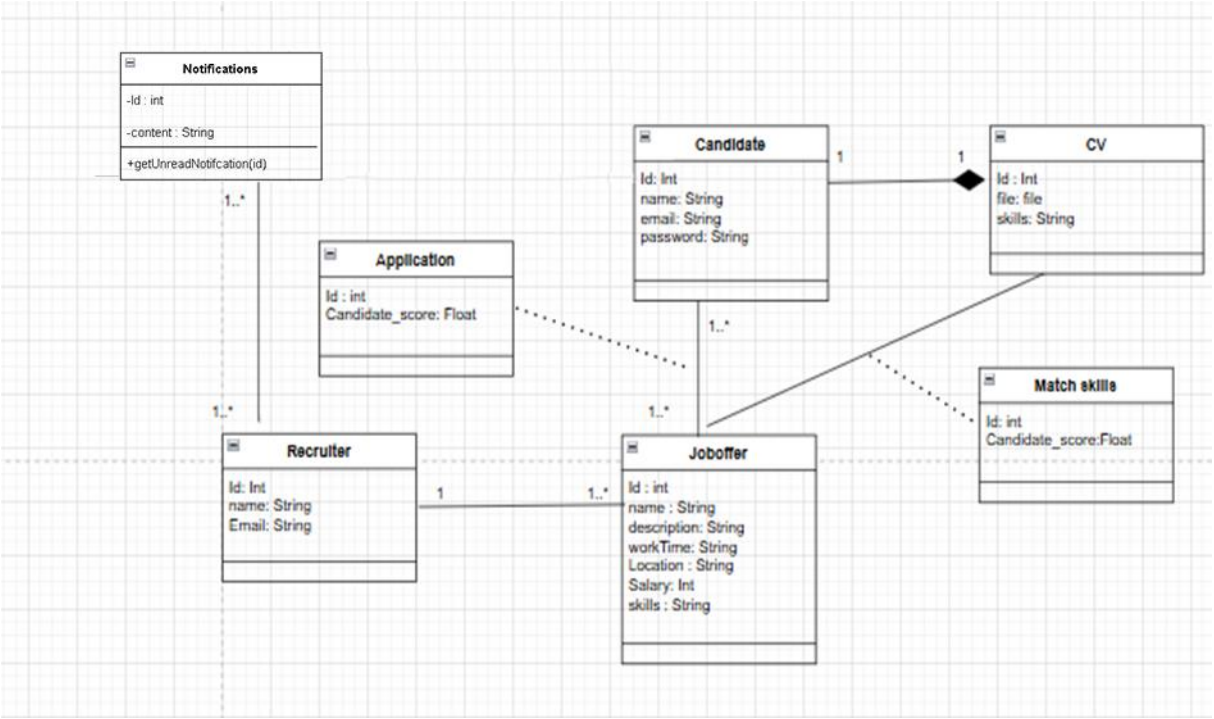


**Figure 11: Sprint 2 : Final Class Diagram**

## 2. Database schema

The database schema is carefully designed to mirror the class diagram, capturing essential data relationships to optimize both storage and retrieval. By structuring entities and their connections thoughtfully, the schema enhances data integrity and supports efficient querying, ensuring a robust and scalable backend for the system

**Candidate** (id_candidate,name,email,password,#cv_id)

**Recruiter** (id_recruiter,name,email)

**Joboffer**(id_joboffer, name,description,worktime,location,Salary,skills,#id_recruiter)

**Application** (id_application, candidate_score, #id_Candidate, #id_joboffer)

**Cv** (id_cv, file,skills)

**Notifications** (id_content, candidate_score, ,#id_recruiter, #id_Candidate, #id_joboffer)

**Match_Skills** (id_match, candidate_score, #id_Candidate, #id_joboffer)

# IV. Implementation and testing

## 1. Software Environment

For this sprint, we kept the same software environment as in the previous one. This consistency ensured stability and helped prevent compatibility issues, allowing the team to focus on implementing new features and improving existing functionality. Maintaining this setup minimized disruptions and enhanced productivity throughout the sprint.

## 2. Tests

Testing user story 9 confirmed that job matches are correctly ranked by relevance for candidates

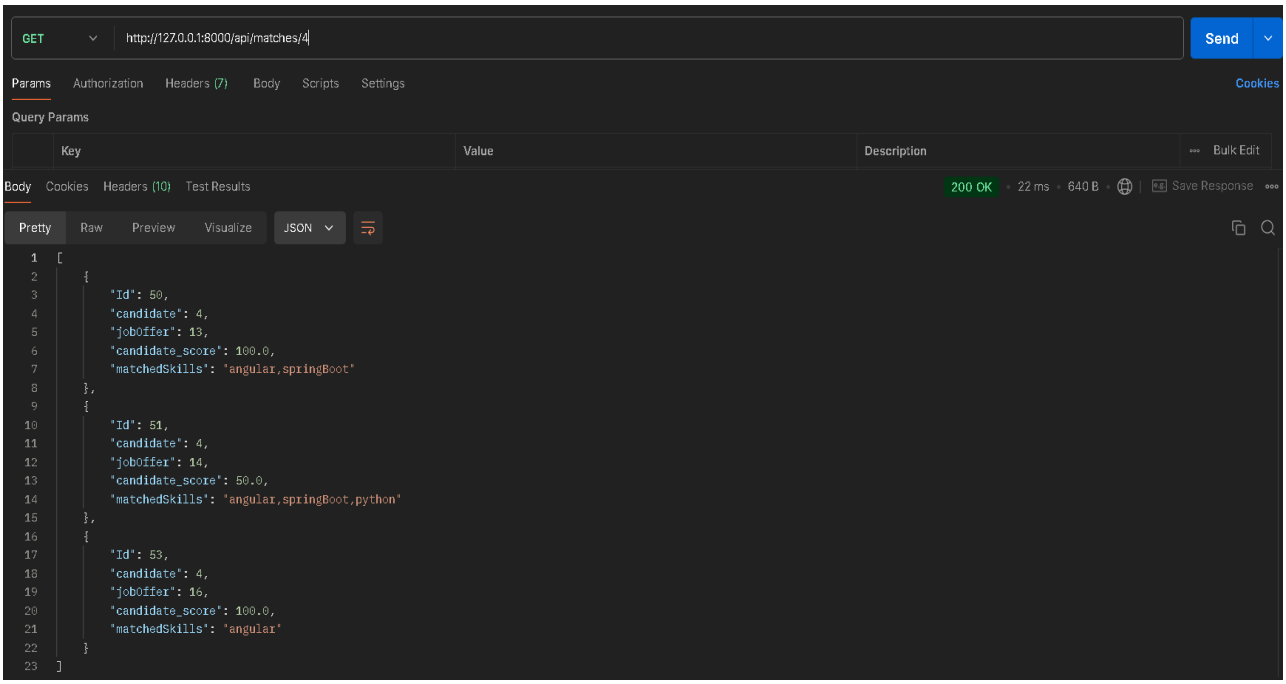


**Figure 12:Sprint 2 -unit test**

Testing user story 6 ensured that recruiters receive timely notifications when candidates matching job requirements apply.

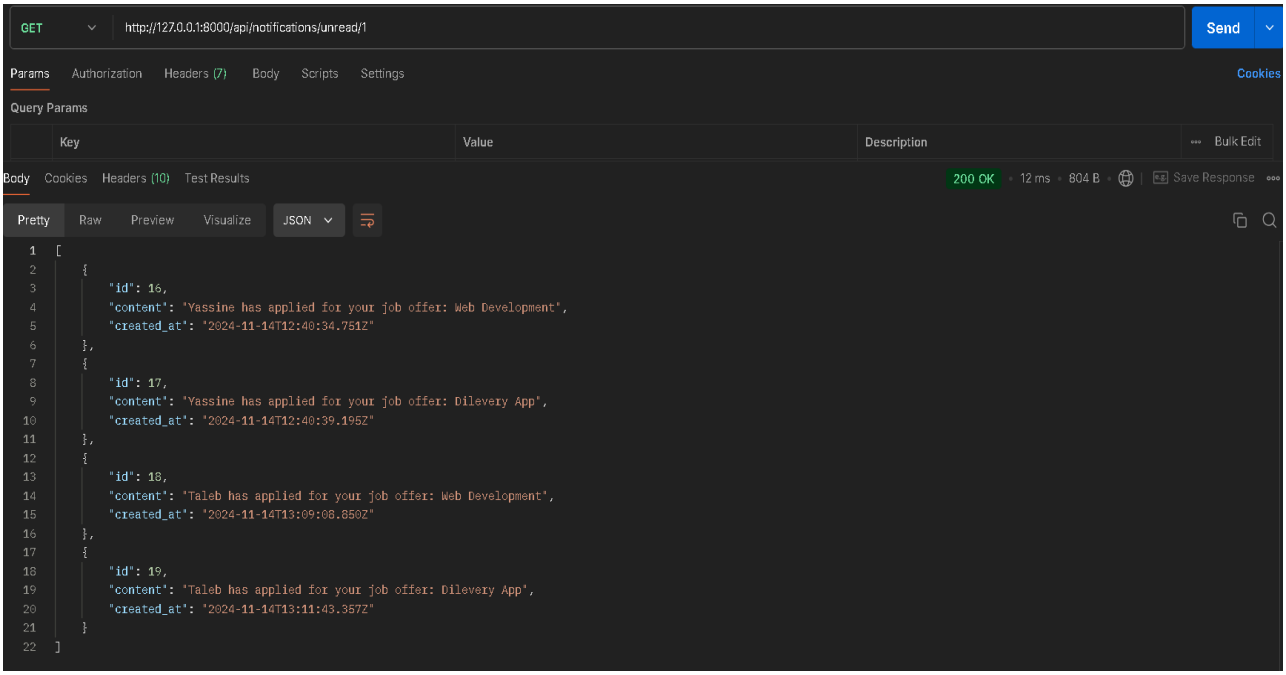


**Figure 13: Sprint 2- unit test**

# V. Sprint Review

## 1. Interfaces

The interface allows managing job postings - adding new ones, editing and deleting existing ones. This data can then be used by an AI engine to improve candidate matching and the selection process.

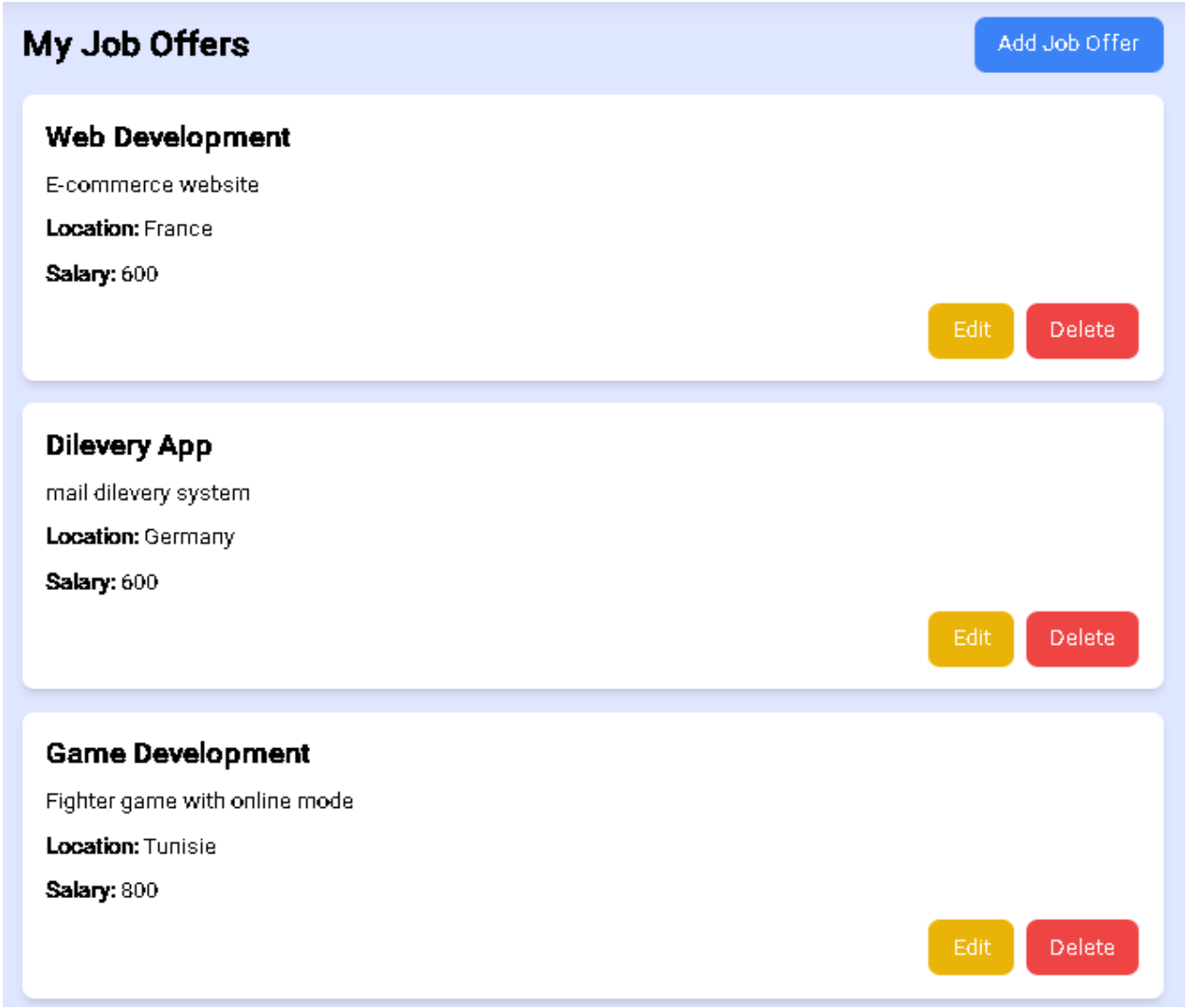


**Figure 14: Recruiter Interface : "Manage Job Postings"**

This interface displays job matches based on the candidate's skills, allowing them to easily identify the most relevant opportunities. The matching score is shown for each job, enabling the candidate to focus on the best fit positions.

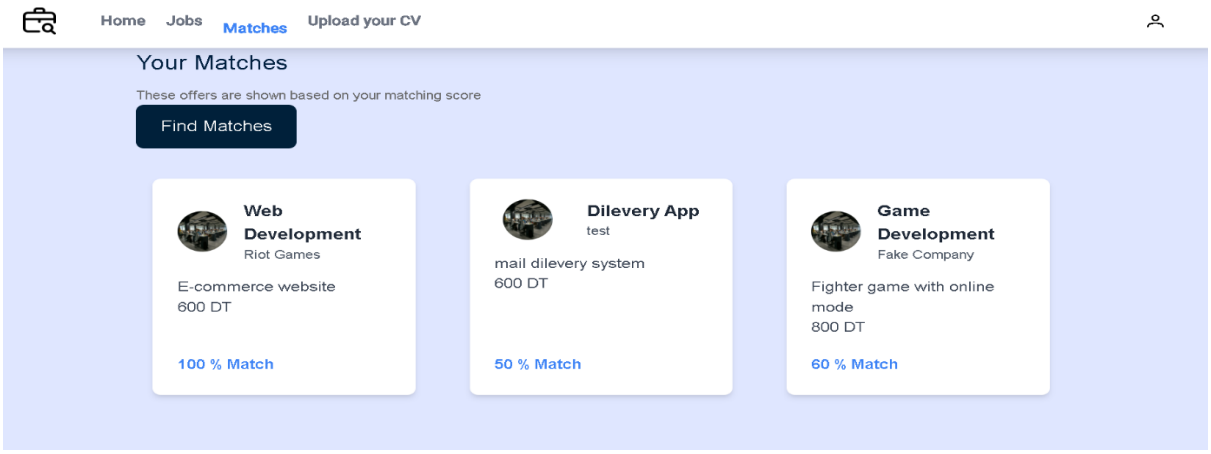


**Figure 15: Job Offer Matching ranked for candidate**

The interface provides recruiters with notifications when candidates who match job requirements apply, enabling timely review and response.

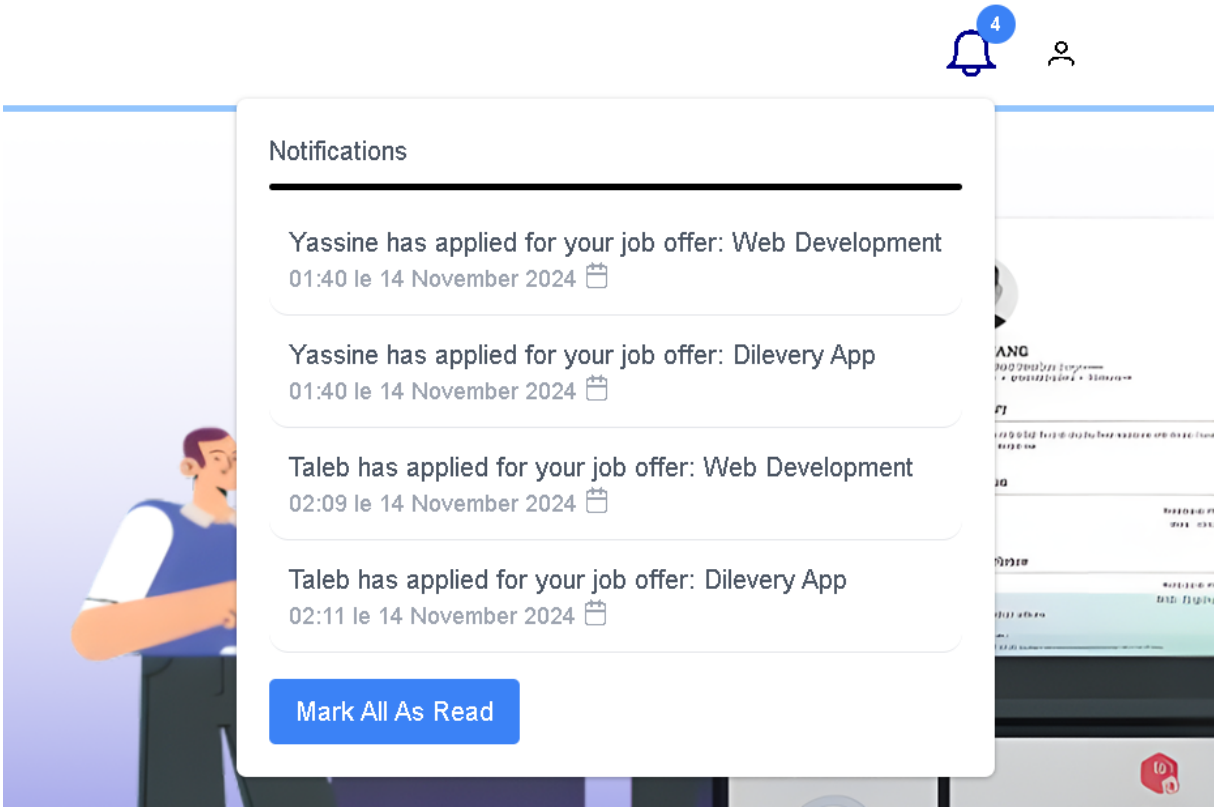


**Figure 16: List of notifications**

The interface allows candidates to submit their resumes, enabling the AI system to analyze their qualifications and match them with suitable job opportunities.

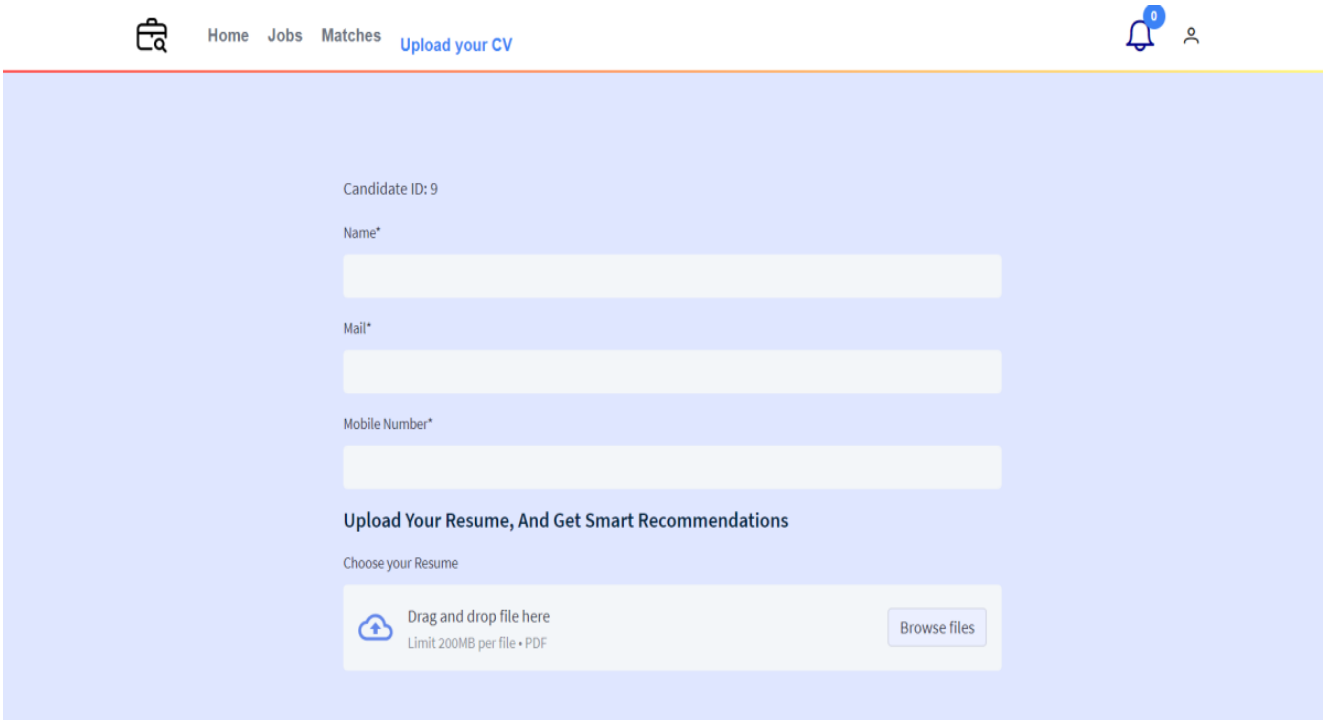


**Figure 17 : Upload Cv**

## 2. Update Product Backlog

| ID | User story | BV | Effort | Priority |
|---|---|---|---|---|
| 2 | As a recruiter, I want to modify job offers after posting so that I can make necessary updates. | Medium | 3 | 1 |
| 7 | As a recruiter, I want to view previously posted jobs so that I can track candidate responses. | Medium | 3 | 2 |
| 10 | As a candidate, I want the system to suggest skill-building resources when I fall short of job requirements so that I can improve my profile. | Low | 8 | 3 |
| 5 | As a recruiter, I want to automatically refresh expired job postings so that the offers remain visible. | Low | 5 | 4 |
| 12 | As a recruiter I want to have authentication for security purposes | Low | 3 | 5 |

| | | | | |
|---|---|---|---|---|
| 13 | As a candidate I want to have authentication for security purposes | Low | 3 | 6 |

### 3. Remaining Sprints Planning

| ID | Effort (Fibonacci) | Sprint Number | Period |
|---|---|---|---|
| 2 | 3 | 3 | October 28th,2024 – November 12th,2024 |
| 10 | 8 | | |
| 5 | 5 | | |
| 12 | 3 | | |
| 7 | 3 | | |

## V. Sprint Retrospective

The objectives of Sprint 2 were largely achieved, with the team successfully implementing candidate prioritization, notification systems. Minor adjustments to the notification logic were identified and scheduled for the following sprint

### 1. Retrospective Table

The retrospection table captures key takeaways from Sprint 1, emphasizing successes, areas needing improvement, and action steps for the next sprint.

**Table 5: Sprint 2 - Retrospective table**

| Went Good | To Improve | Action Plan |
|---|---|---|
| Development<br>Testing<br>Work Division<br>Cooperation | -Stress Management<br>- Better testing strategies | Daily Channel |

### 2. Burn Down Chart

The burndown chart displays the remaining work relative to the sprint timeline, offering insights into the team's task completion efficiency.

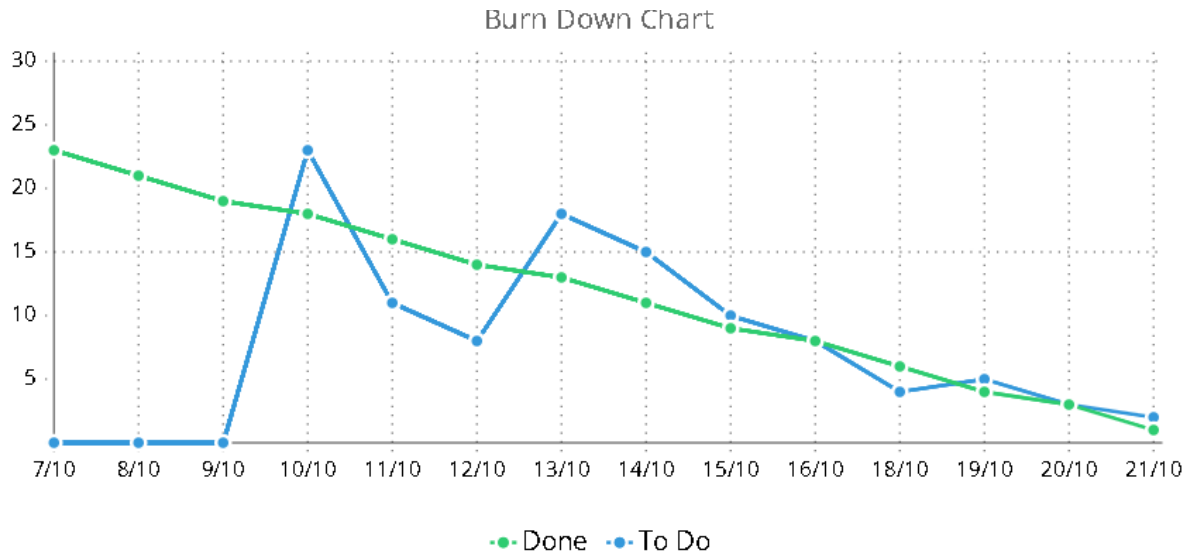


**Figure 18: Sprint 2- Burn Down Chart**

## 3. Burn Up Chart

The burn-up chart highlights progress toward the project goals, displaying both completed tasks and the remaining workload.

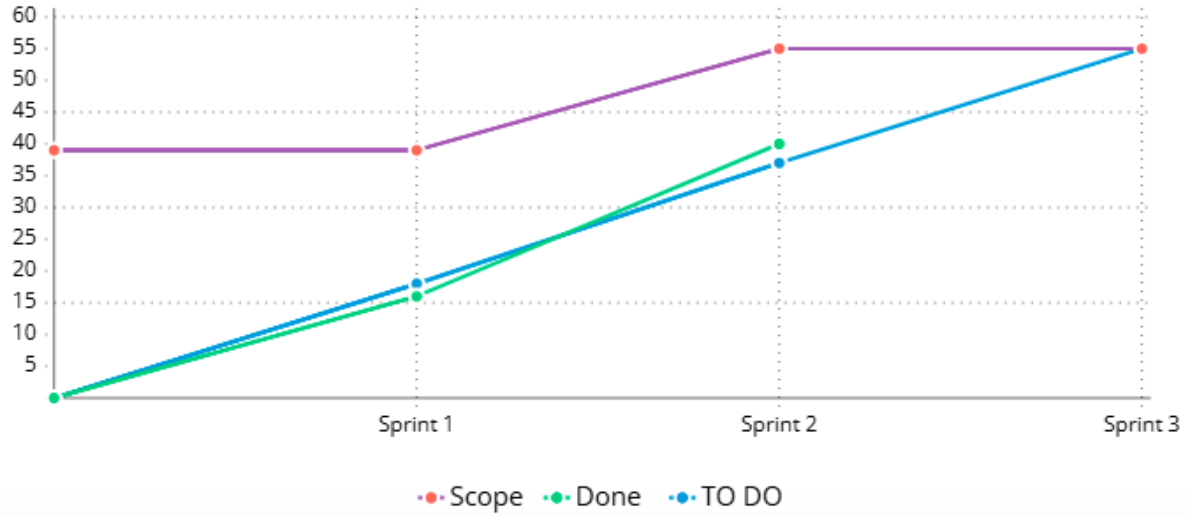


**Figure 19: Burn up chart**

In conclusion, Chapter 3 outlines the progress made during Sprint 2, covering key activities in analysis, conception, and testing. It also highlights the review process and adjustments made to the product backlog, ensuring continued alignment with project goals.

# Chapter 4: Sprint3

This chapter documents the progress, tasks, and deliverables achieved in Sprint 3, highlighting the steps taken to further develop the project. It includes the backlog, analysis, implementation, and evaluation of this sprint.

## I. Sprint Backlog

This section provides an overview of the tasks and objectives for Sprint 3, focusing on the relevant user stories that align with the project's continuous development. The tasks are prioritized to ensure the team remains focused on delivering key components effectively within the sprint timeline.

| ID | User story | Tasks | Time Estimation |
|---|---|---|---|
| 2 | As a recruiter, I want to modify job offers after posting so that I can make necessary updates. | **T1**: Defining Requirements<br>**T2**: Analyzing Requirements<br>**T3**: Solution Design: Class Diagram & Relational Schema<br>**T4**: Backend Development & Unit Testing<br>**T5**: Frontend Development & Unit Testing<br>**T6** : Documentation | 1 H<br>30min<br>2H<br>1 H<br>2H<br>1H |
| 7 | As a recruiter, I want to view previously posted jobs so that I can track candidate responses. | **T1**: Backend Development & Unit Testing<br>**T2**: Frontend Development & Unit Testing<br>**T3 :** Integrale Test<br>**T4** : Documentation | 2H<br>2H<br>1H<br>1 H |

| | | | |
|---|---|---|---|
| 10 | As a candidate, I want the system to suggest skill-building resources when I fall short of job requirements so that I can improve my profile. | Same tasks as user story 7 | 6H |
| 5 | As a recruiter, I want to automatically refresh expired job postings so that the offers remain visible. | Same tasks as user story 7 | 6H |
| 12 | As a recruiter I want to have authentication for security purposes | Same tasks as user story 7 | 6H |

# II. Analysis

## 1. Use Case Diagram

This diagram represents the final use cases of a recruitment management application powered by AI. It shows functionalities for candidates, such as uploading CVs, viewing job matches, and applying for jobs, as well as for recruiters, like managing job offers and receiving candidate recommendations. Key features include personalized feedback and skills analysis by the AI system. Both candidates and recruiters are authenticated to access these features.

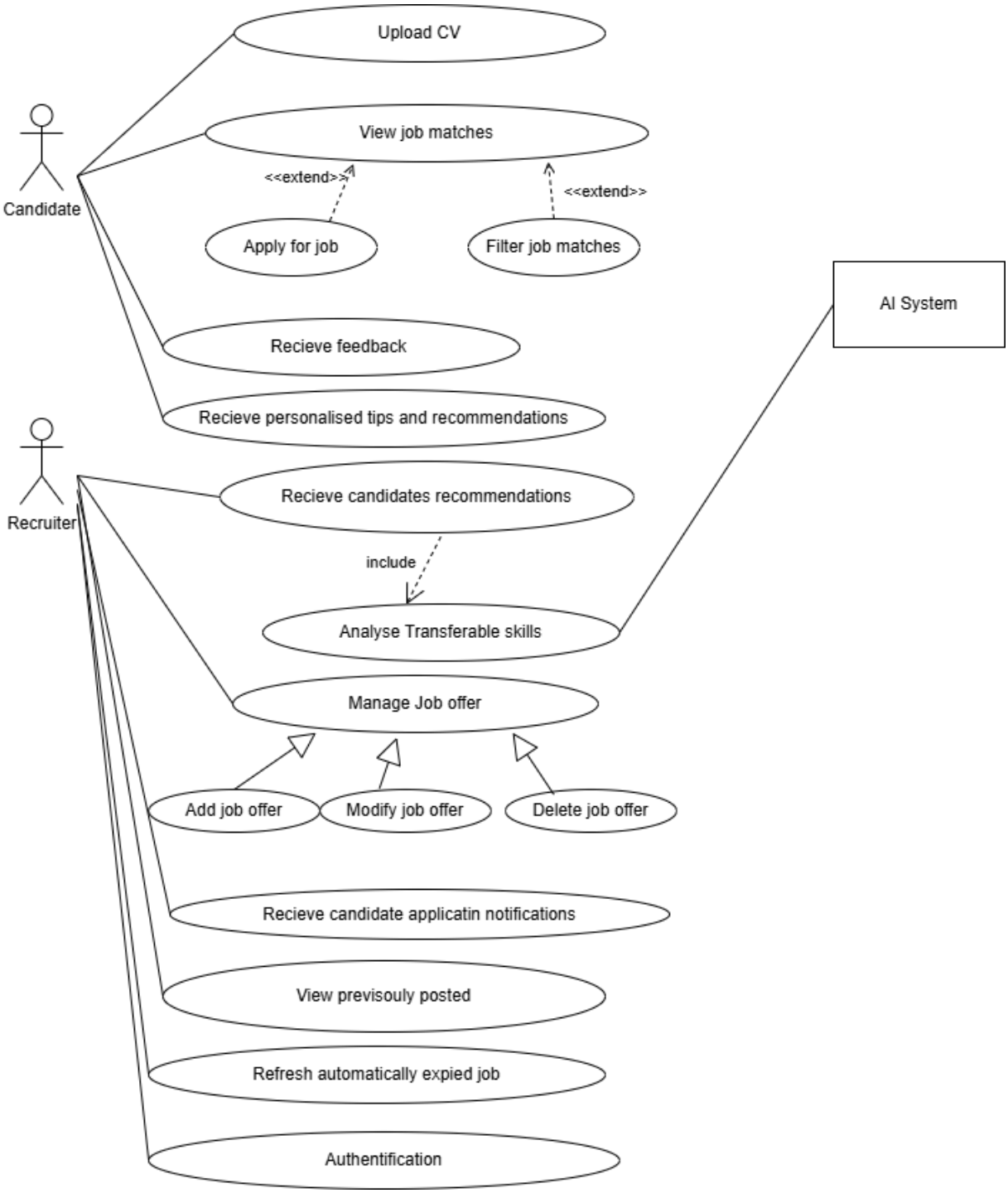


**Figure 20- Sprint 3: Use Case Diagram**

# III. Conception

## 1. Class Diagram

This class diagram represents a recruitment management application with main entities like **Candidate**, **Recruiter**, **CV**, **JobOffer**, **Application**, **Match skills**, and **Notifications**. Candidates have CVs and can apply for job offers, which are managed by recruiters. **Application** and **Match skills** classes handle job matching and scoring. **Notifications** keeps users informed of relevant updates.

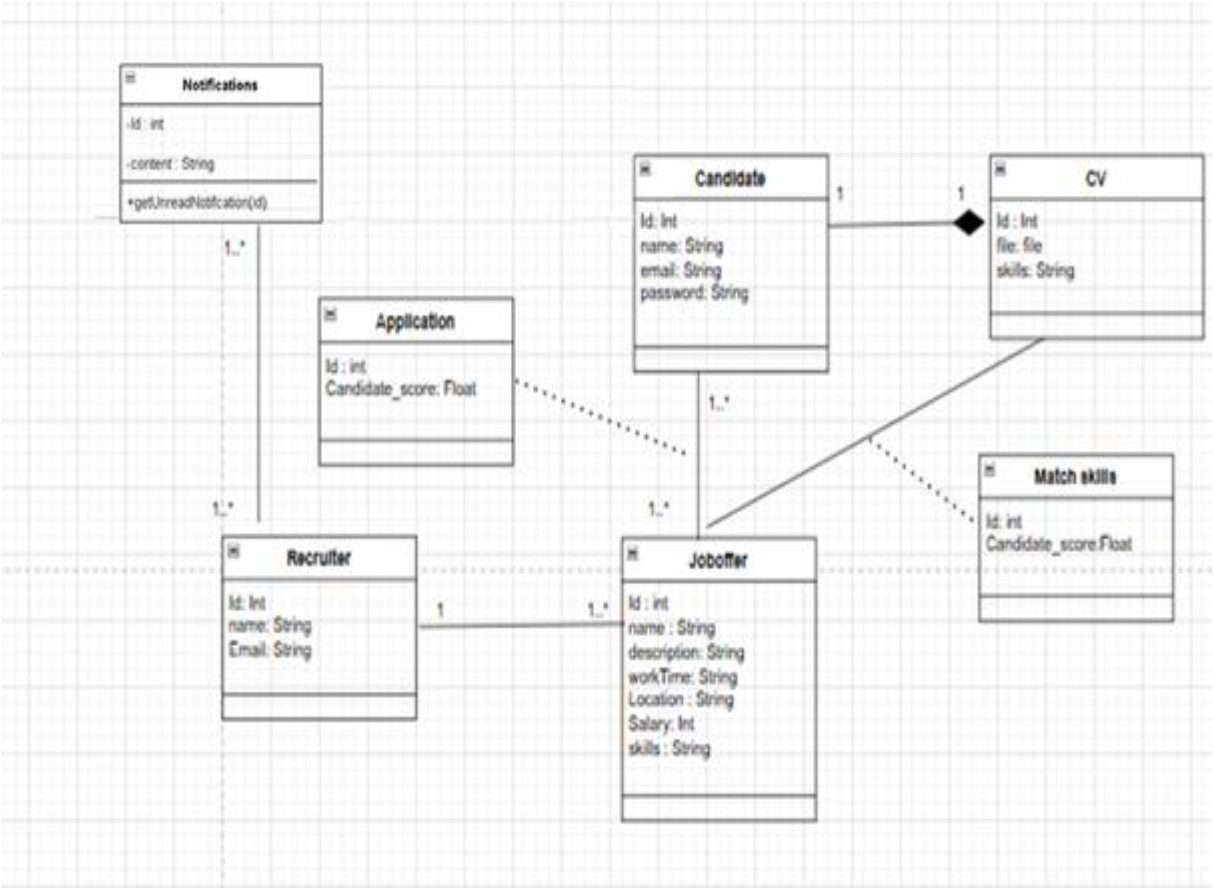


**Figure 21: Sprint 3 : Final Class Diagram**

# IV. Implementation and testing

## 1. Software Environment

For this sprint, we kept the same software environment as in the previous one. This consistency ensured stability and helped prevent compatibility issues, allowing the team to focus on implementing new features and improving existing functionality. Maintaining this setup minimized disruptions and enhanced productivity throughout the sprint.

## 2. Tests

Testing user story 2 validated that recruiters can successfully modify job offers after posting.

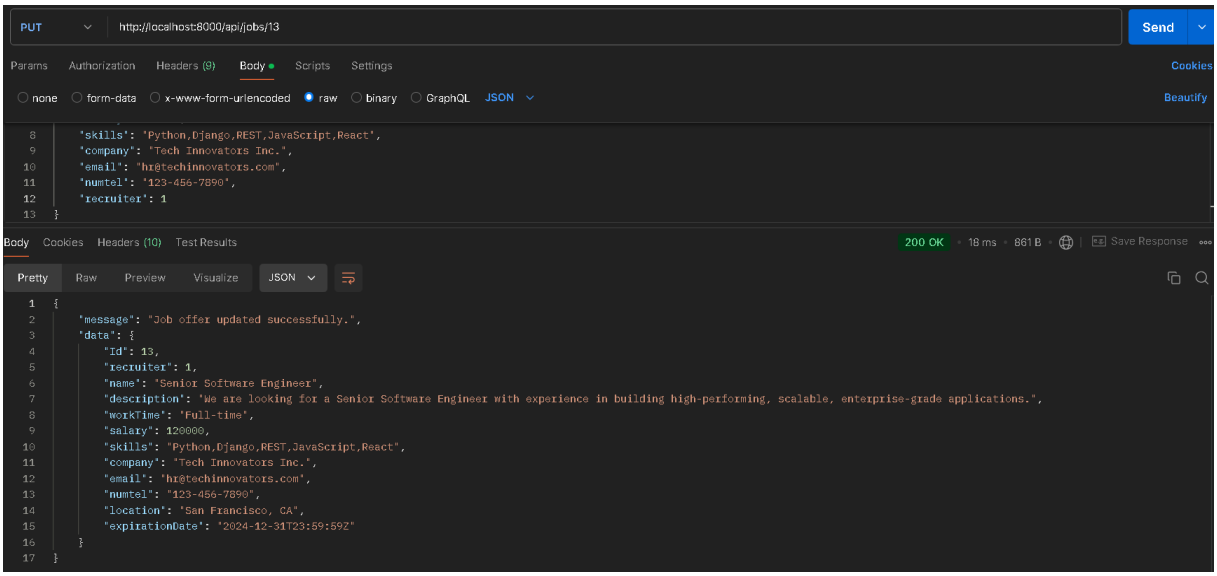


**Figure 22: Sprint 3- unit Test**

Testing user story 12 confirmed that recruiters can securely authenticate using JWT for access control.

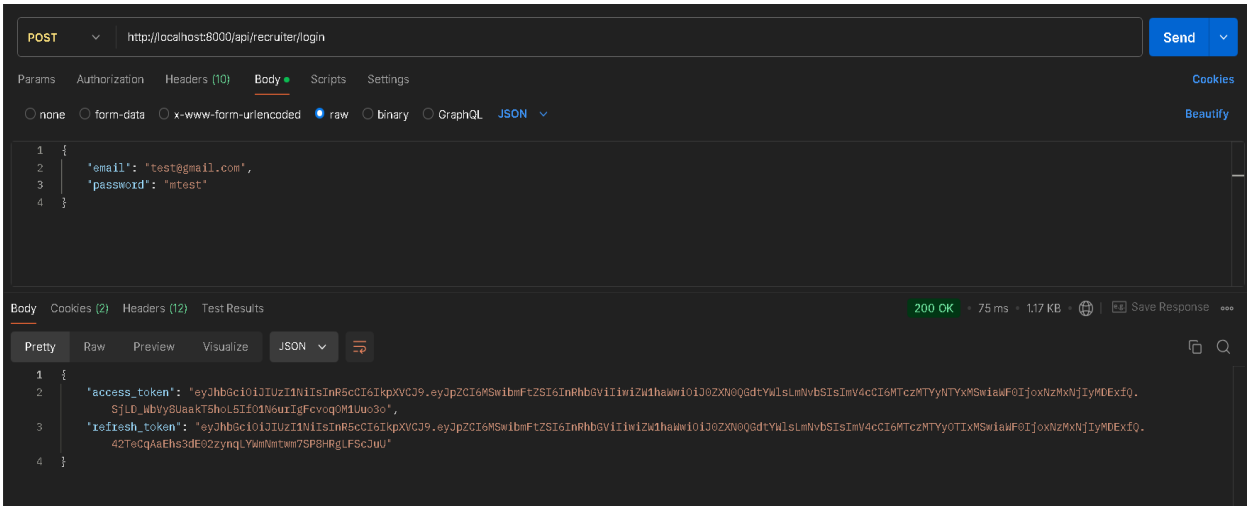


**Figure 23: Sprint 3- unit Test**

# V. Sprint Review

## 1. Interfaces

The interface allows recruiters to update existing job offers after posting, ensuring that any necessary changes are reflected for candidates viewing the job listings.

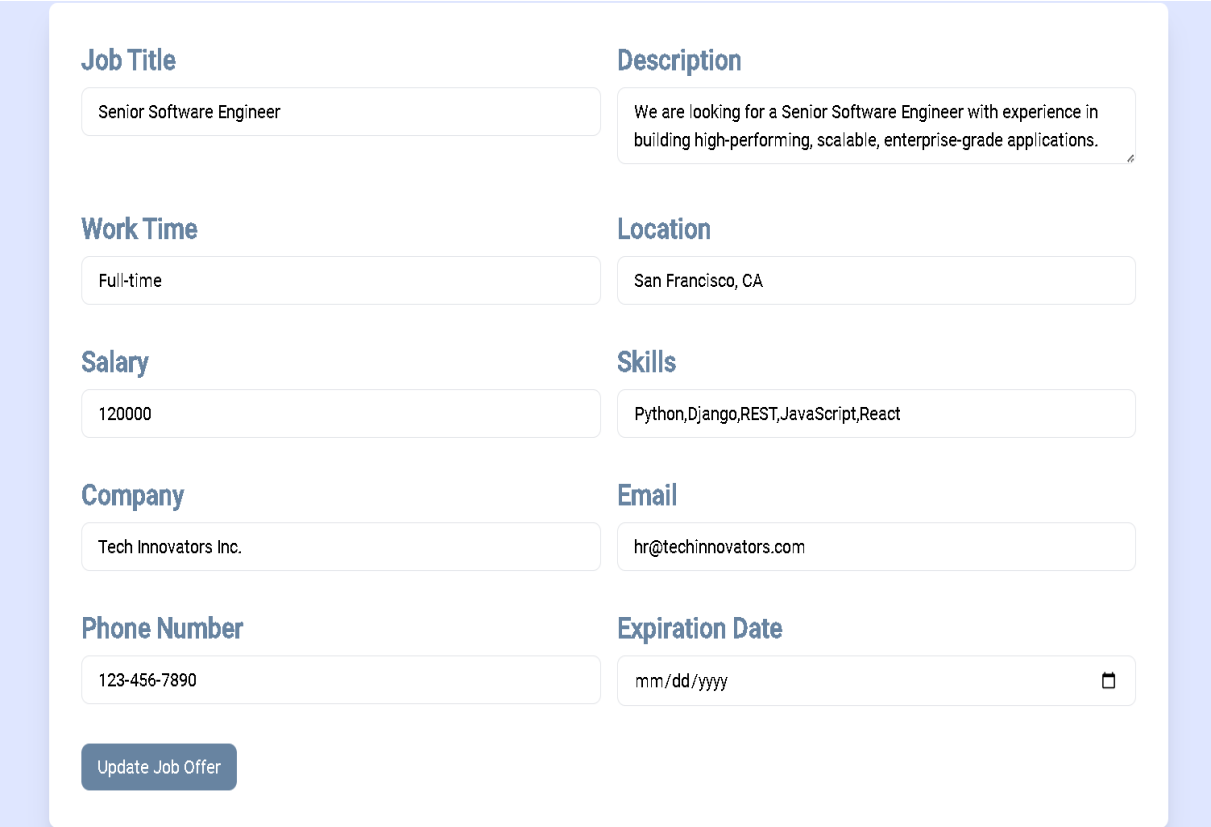


**Figure 24: Update existing job**

The interface includes secure authentication for recruiters, ensuring only authorized users can access and manage job postings.

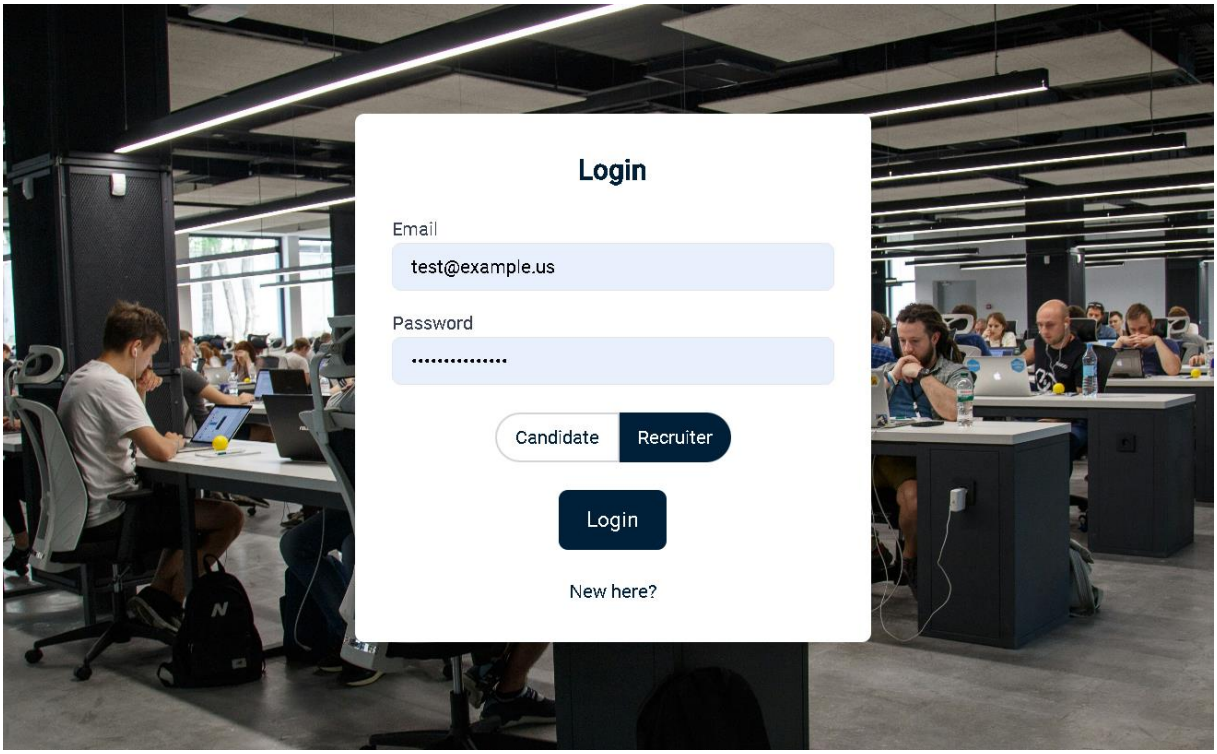


**Figure 25: Authentification**

The interface enables recruiters to view previously posted jobs, allowing them to track candidate responses and engagement over time.

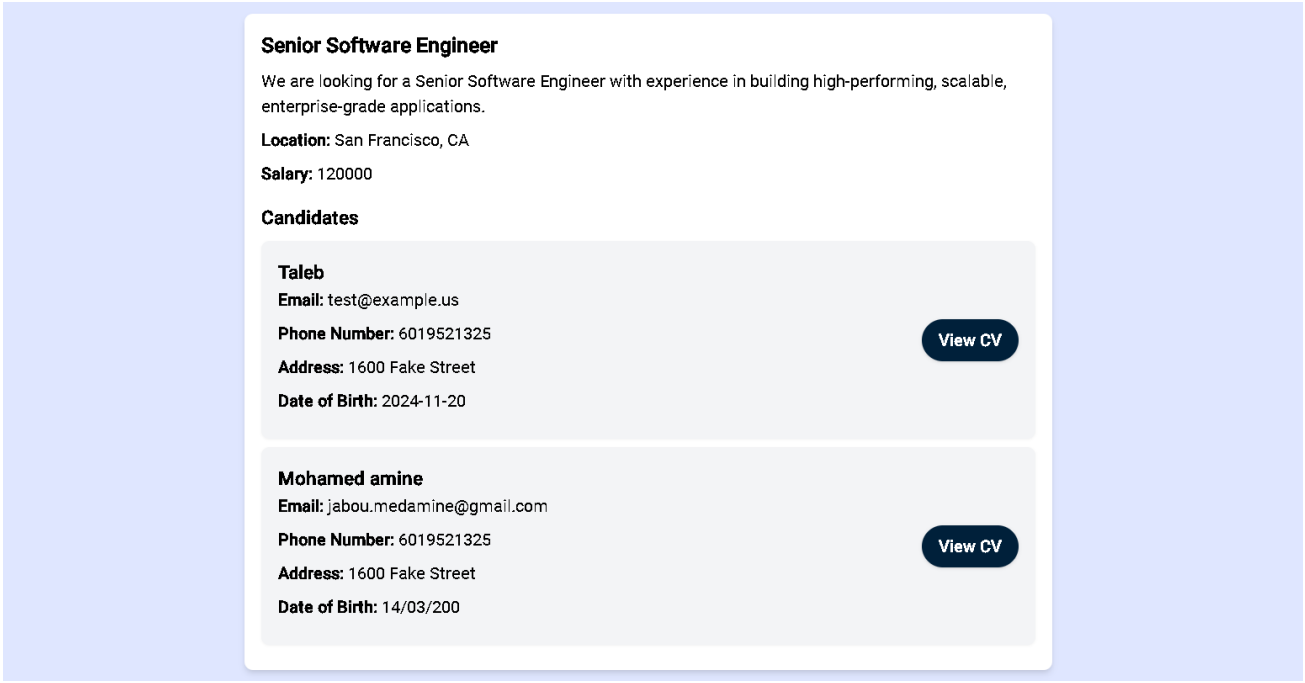


**Figure 26: list of previously jobs**

## VI. Sprint Retrospective

The objectives of Sprint 2 were largely achieved, with the team successfully implementing candidate prioritization, notification systems. Minor adjustments to the notification logic were identified and scheduled for the following sprint

### 1. Retrospective Table

The retrospection table captures key takeaways from Sprint 1, emphasizing successes, areas needing improvement, and action steps for the next sprint.

**Table 6: Sprint 3- Retrospective Table**

| Went Good | To Improve | Action Plan |
|---|---|---|
| Timely completion of tasks Collaboration | Setting clear meeting deadlines | Prepare meeting agendas in advance to ensure focused discussions. |

### 2. Burn Down Chart

The burndown chart displays the remaining work relative to the sprint timeline, offering insights into the team's task completion efficiency.

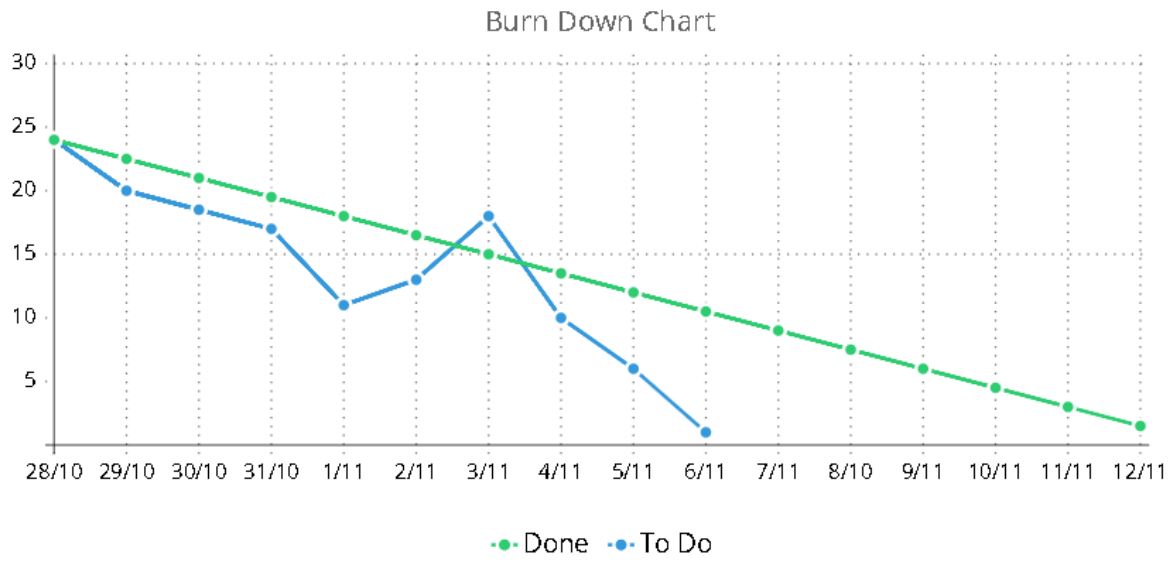


**Figure 27: Sprint 3- Burn Down Chart**

# VII. Application security with SSL and deployment on NGINX

To achieve this step, we used the reverse proxy approach with SSl.

## 1. What is reverse proxy approach

The reverse proxy is an intermediary server on the network that receives requests and forwards them to the target computer. In enterprise networks, this approach is widely used to provide more secure and controllable Internet access to users.

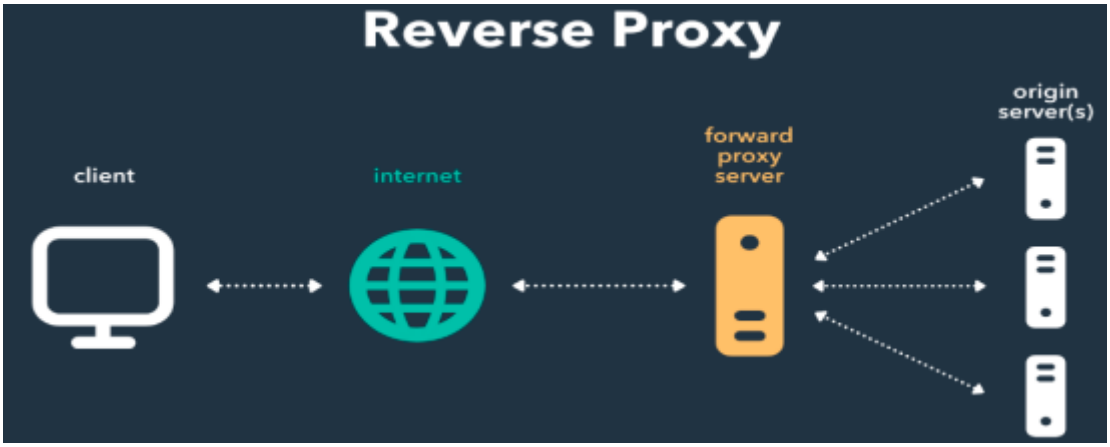


**Figure 28: Reverse Proxy**

## 2. Definition of SSL/TLS

SSL (Secure Sockets Layer) is a security protocol that allows connections to be established secure between a client and a server on the Internet. It ensures the confidentiality, intergrity and the authenticity of the data exchanged by encrypting the information passing between the two parts. Ssl is widely used to secure online transactions, such as payments and the exchange of sensitive data.

## 3. Configuration and deployment steps

### 3.1.Initial setup of NGINX

Here are the commands we placed in order to deploy and configure NGINX.

**Table 7: setup of ngnix**

| Commande | Signification |
|---|---|
| apt update<br>apt install ngnix | Simulation d'une application en arrière-plan, qui sera remplacée par la réelle application ensuite |
| Nc -k -lp 42000 | Mise en place du fichier configuration : désactivation du site par défaut, nommage de |

| | |
|---|---|
| | la nouvelle configuration puis activation de cette dernière. |
| Cd /ect/ngnix | Accéder au répertoire NGINX |
| Ls sites-enabled | Lister les fichiers dans le répertoire sitesenabled |
| rm sites-enabled/default | Supprimer le lien symbolique par défaut |
| Mv sites-available/default sitesavailable/project | Renommer le fichier de configuration |
| ln -s /etc/ngnix/sites-available/project / etc /ngnix/sites-enabled | Activation du site NGINX |

Cette image illustre le résultat de la commande 'ls', laquelle liste les fichiers et répertoires présents dans ce répertoire spécifique de configuration de Nginx, afin de confirmer l'exécution correcte des commandes précédemment effectuées

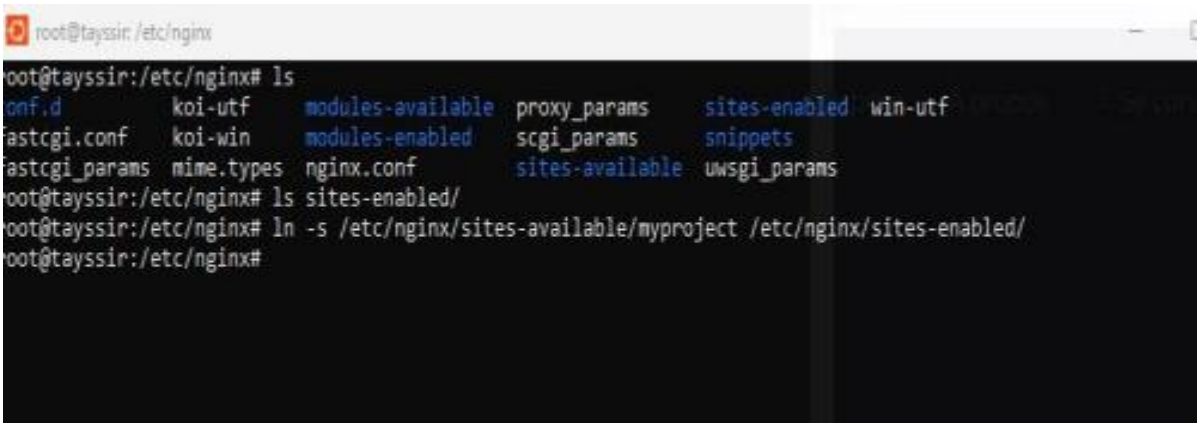


**Figure 29: Configuration ngnix**

Après avoir exécuté la commande suivante dans le terminal : « « sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginxselfsigned.key -out /etc/nginx/ssl/nginx-selfsigned.crt » », nous sommes autorisés à ajouter le certificat à Nginx, comme le montre l'image.

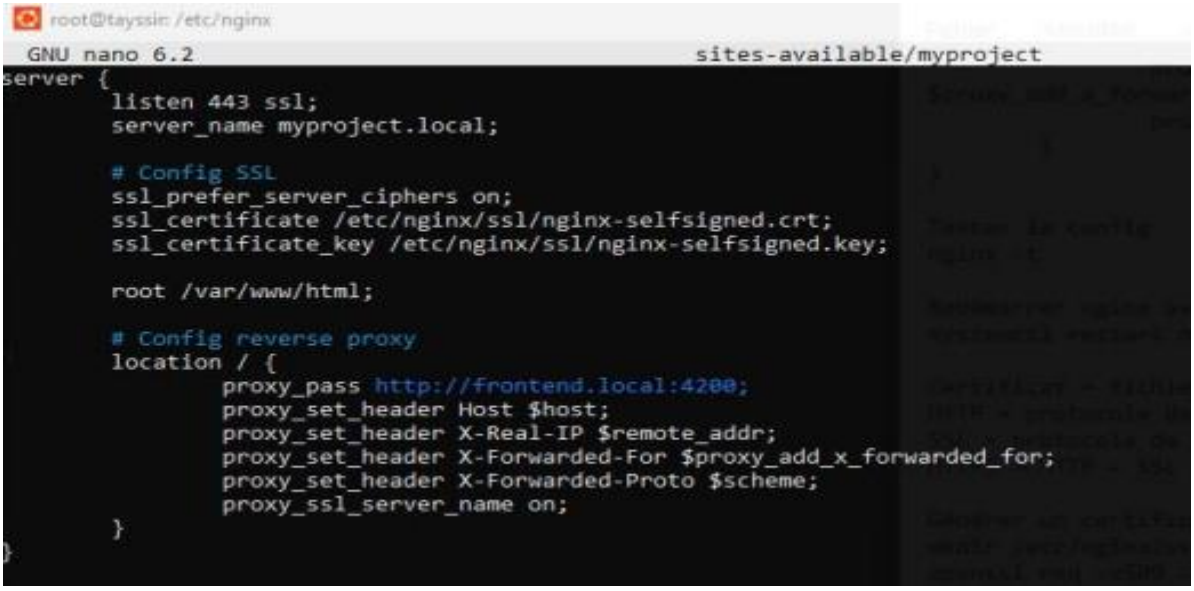


**Figure 30: configuration ssl**

We cloned our GitHub repository link named "AI-Resume-Analyzer" and navigated into its directory.

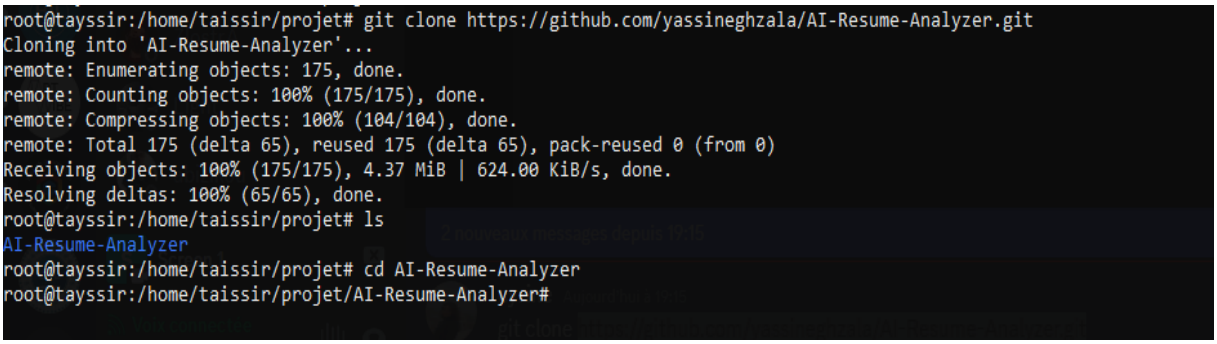


**Figure 31: AI System**

After installing system requirements, we deployed the AI Resume Analyzer as a systemd service which is now running successfully on both internal (192.168.83.50:8501) and external (197.16.186.6:8501) URLs through Streamlit.

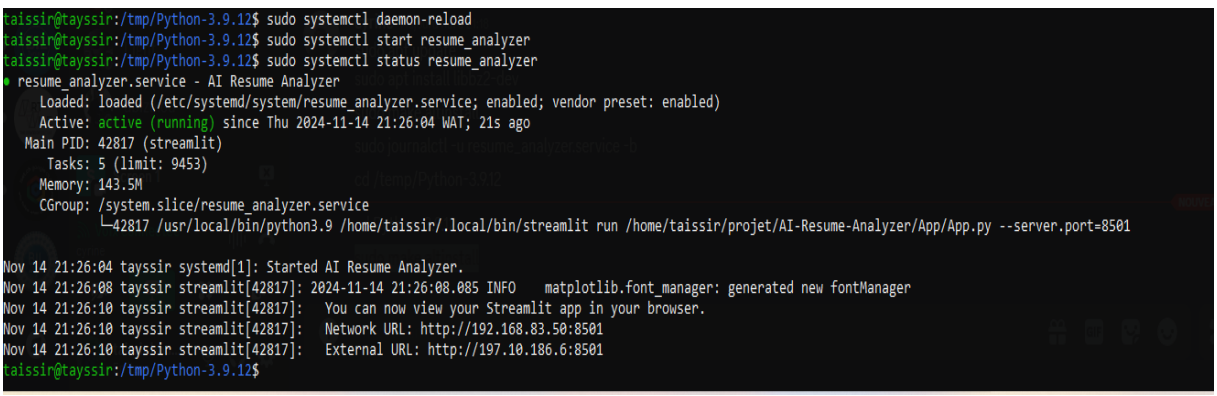


**Figure 32: Systemd Service**

Attempted to clone a GitHub repository but encountered permission errors, then successfully cloned projetGannarBackend using sudo permissions.

```
taissir@tayssir:~/projet$ git clone https://github.com/yassineghzala/projetGannarBackend.git
fatal: could not create work tree dir 'projetGannarBackend': Permission denied
taissir@tayssir:~/projet$ git clone https://github.com/yassineghzala/projetGannarBackend.git
fatal: could not create work tree dir 'projetGannarBackend': Permission denied
taissir@tayssir:~/projet$ sudo git clone https://github.com/yassineghzala/projetGannarBackend.git
[sudo] password for taissir:
Cloning into 'projetGannarBackend'...
remote: Enumerating objects: 10908, done.
remote: Counting objects: 100% (10908/10908), done.
remote: Compressing objects: 100% (7025/7025), done.
remote: Total 10908 (delta 3416), reused 10148 (delta 2656), pack-reused 0 (from 0)
Receiving objects: 100% (10908/10908), 16.52 MiB | 517.00 KiB/s, done.
Resolving deltas: 100% (3416/3416), done.
Updating files: 100% (7096/7096), done.
taissir@tayssir:~/projet$ ls
AI-Resume-Analyzer  projetGannarBackend
taissir@tayssir:~/projet$ cd projetGannarBackend
```

**Figure 33: Backend**

This shows a systemd service configuration file (/etc/systemd/system/backend.service) for the projetGannarBackend application, setting up a Gunicorn server with 3 workers under the taissir user in the www-data group, running from the project's working directory.

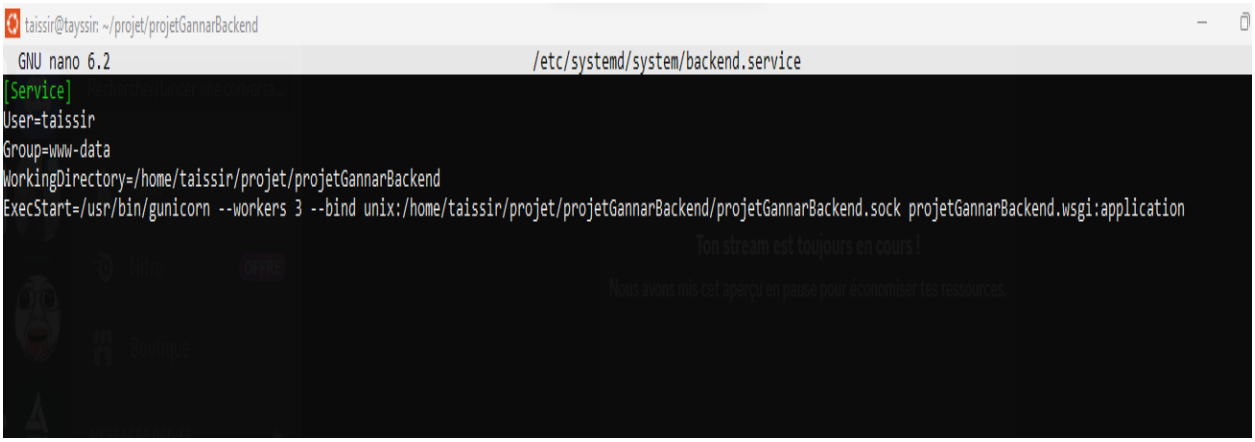


**Figure 34: Backend Service Configuration**

Backend Service Deployment Status: Successfully started Gunicorn server using Python3, running on localhost:8000, actively operational.

```
__init__.py  __pycache__  asgi.py  settings.py  urls.py  wsgi.py
(pjenv) taissir@tayssir:~/projet/projetGannarBackend/projetGannarBackend$ sudo nano settings.py
(pjenv) taissir@tayssir:~/projet/projetGannarBackend/projetGannarBackend$ sudo systemctl restart backend.service
(pjenv) taissir@tayssir:~/projet/projetGannarBackend/projetGannarBackend$ sudo systemctl status backend.service
● backend.service
     Loaded: loaded (/etc/systemd/system/backend.service; static)
     Active: active (running) since Fri 2024-11-15 00:31:40 CET; 12s ago
   Main PID: 46679 (gunicorn)
      Tasks: 4 (limit: 9453)
     Memory: 135.4M
     CGroup: /system.slice/backend.service
             ├─46679 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 projetGannarBackend.wsgi:application
             ├─46680 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 projetGannarBackend.wsgi:application
             ├─46681 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 projetGannarBackend.wsgi:application
             └─46682 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind 0.0.0.0:8000 projetGannarBackend.wsgi:application

Nov 15 00:31:40 tayssir systemd[1]: Started backend.service.
Nov 15 00:31:41 tayssir gunicorn[46679]: [2024-11-15 00:31:41 +0100] [46679] [INFO] Starting gunicorn 20.1.0
Nov 15 00:31:41 tayssir gunicorn[46679]: [2024-11-15 00:31:41 +0100] [46679] [INFO] Listening at: http://0.0.0.0:8000 (46679)
Nov 15 00:31:41 tayssir gunicorn[46679]: [2024-11-15 00:31:41 +0100] [46679] [INFO] Using worker: sync
Nov 15 00:31:41 tayssir gunicorn[46680]: [2024-11-15 00:31:41 +0100] [46680] [INFO] Booting worker with pid: 46680
Nov 15 00:31:41 tayssir gunicorn[46681]: [2024-11-15 00:31:41 +0100] [46681] [INFO] Booting worker with pid: 46681
Nov 15 00:31:41 tayssir gunicorn[46682]: [2024-11-15 00:31:41 +0100] [46682] [INFO] Booting worker with pid: 46682
(pjenv) taissir@tayssir:~/projet/projetGannarBackend/projetGannarBackend$
```

**Figure 35: Backend Service Status**

## VIII. Architecture Implementation

Our project is structured using a 5-tier architecture, which organizes the system into distinct layers, each with its own responsibilities. This design enhances maintainability, scalability, and security by clearly separating concerns and allowing each layer to evolve independently.

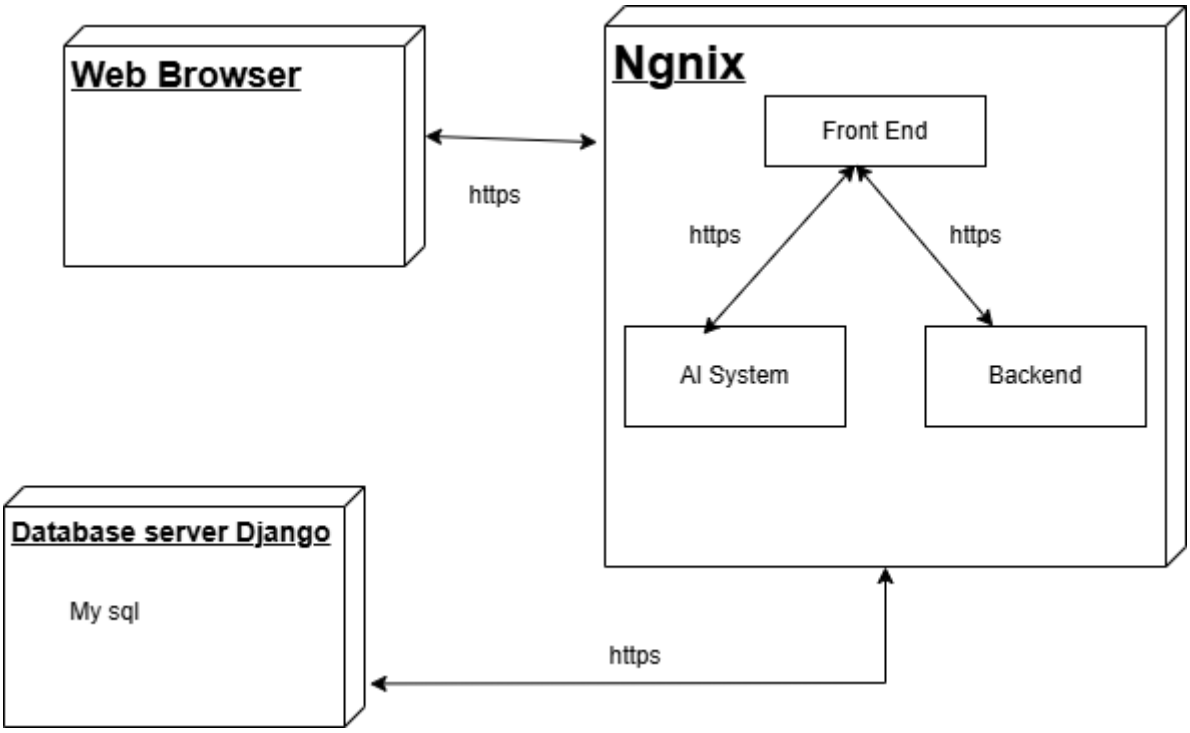


**Figure 36: deployment diagram**

In conclusion, this chapter details the progression and outcomes of Sprint 3, highlighting key achievements in analysis, conception, and implementation. It also emphasizes the integration of security measures and deployment steps, ensuring a robust and scalable platform.

# General Conclusion

In conclusion, our project addresses the challenges faced by students and recent graduates in securing internships or job placements. Traditional recruitment processes are often complex and time-consuming, making it difficult for both applicants and recruiters. Our AI-powered web application simplifies this process by analyzing resumes, matching candidates with suitable job opportunities, and providing personalized recommendations to guide users toward the most relevant positions.

This initiative marks our first use of agile methodologies and the Scrum framework, allowing us to take an iterative, flexible approach to development. The collaborative nature of agile has helped us quickly adapt to challenges and make continuous improvements, ensuring that the platform meets both user and recruiter needs efficiently.

As we move forward, we believe this solution will significantly streamline the recruitment process, benefiting both candidates and employers. The experience gained in this project lays a strong foundation for future developments, and we are committed to refining the platform to better meet the evolving demands of the job market.