



*Ministère de l'Enseignement Supérieur et de la recherche scientifique  
Direction Générale des Etudes Technologiques  
Institut Supérieur des Etudes Technologiques de Sfax  
Département Technologies de l'Informatique*



**Code stage :**

**Soutenu le :**

# **STAGE De Perfectionnement**

*Organisme d'accueil : AZIIN Engineering Solutions*

*Elaboré par : Achraf Mkaouar*

*Cin : 11141357*

*Groupe : DSI23*

*Période : Du 2024-01-11 Au 2024-02-03*

*Encadrant : Hassen Mrabet*

# **Remerciements**

Je tiens à exprimer ma profonde reconnaissance à toutes les personnes qui ont contribué à la réalisation de ce travail.

Tout d'abord, je tiens à remercier mon encadreur A AZIIN, Mr. Mrabet Hassen, pour son accompagnement tout au long de mon stage.

Sa disponibilité et sa confiance m'ont permis d'accomplir mon travail dans les meilleures conditions. Mes sincères remerciements vont également à la personne qui a accepté de juger mon travail.

Enfin, je souhaite exprimer ma gratitude à tous les enseignants de l'Institut Supérieur des Etudes Technologiques de Sfax pour leur soutien et leurs précieux enseignements tout au long de mon cursus universitaire. Leurs conseils et leur expertise ont été essentiels pour la réussite de ce travail.

## Table des matières

Introduction Générale .....	1
1. Chapitre 1 : Cadre générale du projet .....	3
1.1. Introduction.....	3
1.2. Présentation de l'organisme d'accueil .....	3
1.2.1.Présentation générale .....	3
1.2.2.Domaine de la Société .....	3
1.3. Problématique et solution proposée .....	4
1.3.1.Problématique .....	4
1.3.2.Solution Proposée .....	5
1.4. Conclusion .....	5
2. Chapitre 2 : Lancement Et Analyse De Projet .....	6
2.1. Introduction.....	6
2.2. Spécification des besoins .....	6
2.2.1.Identification des acteurs .....	6
2.2.2.Identification des besoins fonctionnels .....	6
2.2.3.Identification des besoins non fonctionnels .....	7
2.3. Besoins techniques .....	8
2.3.1.Environnement matériel .....	8
2.3.2.Environnement logiciel .....	8
2.3.3.Environnement Base de données .....	9
2.3.4.Framework utilisée .....	9
2.3.5.Bibliothèque utilisé .....	10
2.3.6.Modélisation UML .....	10
2.3.7.Autre Technologies utilisées .....	10
2.4. Conclusion .....	12
3. Chapitre 3 : Modélisation UML .....	13
3.1. Introduction.....	13
3.2. Langage de modélisation .....	13
3.3. Modélisation du projet .....	13
3.3.1.Diagramme de cas d'utilisation .....	14
3.3.2.Diagramme de classe .....	16
3.4. Conclusion .....	16
4. Chapitre 4 : Réalisation .....	17
4.1. Introduction.....	17
4.2. Interface d'accueil .....	17
4.2.1.Page d'accueil .....	17
4.2.2.Page de S'inscrire .....	17
4.2.3.Page Connectez-vous .....	19

4.3. Interface de Candidat .....	22
4.3.1.Page Préinscription .....	22
4.3.2.Page Candidat Accueil .....	23
4.3.3.Page Candidat Profile .....	26
4.3.4.Page Candidat Demandes .....	27
4.3.5.Page Candidat Postuler .....	28
4.4. Interface Admin .....	30
4.4.1.Page Admin Accueil .....	30
4.4.2.Page List des Candidatures .....	30
4.4.3.Page Ajouter Stage .....	33
4.4.4.Page List des Stages .....	34
4.4.5.Page Demandes .....	36
4.5. Conclusion.....	40
Conclusion et perspectives.....	41

## Table des figures

Figure 1.1 – Logo de la société "Aziin Engineering Solutions" .....	3
Figure 2.1 – Logo Visual Studio.....	9
Figure 2.2 – Logo SQL Server.....	9
Figure 2.3 – Logo ASP.NET Core.....	9
Figure 2.4 – Logo React.....	10
Figure 2.5 – Logo StarUML.....	10
Figure 2.6 – Logo Docker.....	11
Figure 2.7 – Logo Swagger.....	11
Figure 2.8 – Logo Cloudinary.....	11
Figure 3.1 – Diagramme de cas d'utilisation pour L'Administrateur.....	14
Figure 3.2 – Diagramme de cas d'utilisation pour Candidat.....	15
Figure 3.3 – Diagramme de classe.....	16
Figure 4.1 – Page d'accueil.....	17
Figure 4.2 – Page de S'inscrire.....	17
Figure 4.2.1 – Api Register.....	18
Figure 4.2.2 – Résultat Api register.....	18
Figure 4.2.3 – Résultat dans la base de données.....	19
Figure 4.3 – Page Connectez-vous.....	19
Figure 4.3.1 – Api login.....	20
Figure 4.3.2 – Résultat Api login.....	21
Figure 4.3.3 – Loader login.....	21
Figure 4.4 – Page Préinscription.....	22
Figure 4.4.1 – Page Candidat Accueil.....	23
Figure 4.4.2 – Api Get all stages.....	23
Figure 4.4.3 – Résultat Api Get all stages.....	24
Figure 4.4.4 – Api Get User With Cookies.....	24
Figure 4.4.5 – Résultat Api Get User With Cookies.....	24
Figure 4.4.6 – Button Se Déconnecter.....	25

Figure 4.4.7 – Api Logout.....	25
Figure 4.4.8 – Résultat Api Logout.....	25
Figure 4.5.1 – Page Profile.....	26
Figure 4.5.2 – Api Profile.....	26
Figure 4.5.3 – Résultat Api Profile.....	27
Figure 4.6.1 – Page Demandes.....	27
Figure 4.6.2 – Api Demande.....	27
Figure 4.6.3 – Résultat Api Demande.....	28
Figure 4.7.1 – Page Postuler.....	28
Figure 4.7.2 – Api Postuler.....	29
Figure 4.7.3 – Résultat Api Postuler.....	29
Figure 4.8.1 – Page Admin.....	30
Figure 4.8.2 – Page List des Candidatures.....	30
Figure 4.8.3 – Api Users.....	31
Figure 4.8.4 – Résultat Api Users.....	31
Figure 4.8.5 – Page Infos Candidat.....	32
Figure 4.8.6 – Page Validation Suppression.....	32
Figure 4.8.7 – Page Ajouter Stage.....	33
Figure 4.8.8 – Api Ajouter Stage.....	33
Figure 4.8.9 – Résultat Api Ajouter Stage.....	34
Figure 4.8.10 – Page List des Stages.....	34
Figure 4.8.11 – Api Stages.....	35
Figure 4.8.12 – Résultat Api Stages.....	35
Figure 4.8.13 – Page Validation Suppression Stage.....	36
Figure 4.8.14 – Page Demandes.....	36
Figure 4.8.15 – Api Demandes.....	36
Figure 4.8.16 – Résultat Api Demandes.....	37
Figure 4.8.17 – Candidat Infos.....	37
Figure 4.8.18 – Stage Infos.....	38
Figure 4.8.19 – Api Décision demande.....	39

Figure 4.8.20 – Résultat Api Décision demande.....	39
Figure 4.8.21 – Résultat Décision demande.....	40
Figure 4.8.22 – Résultat Décision demande Interface Candidat.....	40

## **List des tableaux**

Table 2.1 – Environnement matériel utilisé.....	8
-------------------------------------------------	---

# Introduction Générale

**D**ans le contexte actuel, où les entreprises font face à des défis constants en matière de gestion des ressources humaines, l'efficacité dans le processus de gestion des candidatures est devenue cruciale. Pour répondre à cette problématique, un projet informatique a été conçu dans le but d'optimiser le processus de gestion des candidatures, avec pour objectif d'améliorer l'organisation et les performances générales de l'entreprise.

L'objectif de ce rapport est de présenter de manière détaillée le projet, en couvrant les différentes étapes, de l'analyse des besoins à la mise en œuvre et à l'évaluation de la solution. Nous explorerons également les enjeux spécifiques ainsi que les avantages associés à la mise en place de cette application, mettant en lumière son impact sur l'efficacité du recrutement et la collaboration au sein des équipes.

Dans un premier temps, nous exposerons le contexte et les objectifs du projet, soulignant les contraintes particulières de notre processus de recrutement et les exigences que la solution doit satisfaire. Ensuite, nous détaillerons l'architecture et les choix technologiques adoptés pour garantir la pertinence et la performance de l'application.

Par la suite, nous aborderons les principales fonctionnalités de l'application de gestion des candidatures, ainsi que les étapes clés du processus de développement et de déploiement. Enfin, nous évaluerons les résultats obtenus et les avantages concrets apportés par cette solution informatique, en termes d'amélioration de l'efficacité du recrutement et de satisfaction des utilisateurs. Nous conclurons en discutant des perspectives d'évolution et des axes d'amélioration envisageables pour assurer le succès continu de l'application.

Ce document suit une structure articulée autour de quatre chapitres synthétisant les différentes tâches réalisées tout au long du projet :

- Le premier chapitre, intitulé "**Cadre général du projet**" constitue une introduction dédiée à la présentation de l'organisme d'accueil, du contexte, des objectifs du projet, de la problématique, et de la solution proposée.
- Le deuxième chapitre, "**Lancement et analyse du projet**" vise à identifier les acteurs, les besoins fonctionnels et non fonctionnels du projet, ainsi que les outils et technologies utilisés au cours de sa réalisation.
- Le troisième chapitre, "**Modélisation UML**" se concentre sur l'identification des différents diagrammes utilisés pour la conception du projet.
- Le quatrième chapitre, "**Réalisation**" est dédié à l'évaluation des résultats obtenus et à la présentation concrète de l'application de gestion des candidatures.

En conclusion, ce rapport offre une vue d'ensemble du travail accompli tout en esquissant quelques perspectives.

# **Chapitre 1 : Cadre générale Du Projet**

## **1.1 – Introduction**

Ce chapitre est conçu pour situer le projet dans son cadre général. Nous débutons en fournissant un aperçu de l'organisme d'accueil, ainsi que du contexte du projet.

## **1.2 – Présentation de l'organisme d'accueil**

Dans cette section, nous présentons l'entreprise AZIIN ENGINEERING SOLUTIONS au sein de laquelle nous avons réalisé notre projet de perfectionnement sur une période d'un mois.

### **1.2.1 – Présentation générale**

Aziin Engineering Solutions, située à 1,5 kilomètre de la Route El Ain, occupe le 1er étage de l'immeuble, appartement N°1, sous la direction de Mme Ameni Atii.

Elle se spécialise dans la création de logiciels et de sites web sur mesure, en intégrant l'ensemble des compétences nécessaires à la conduite de projets.



Figure 1.1 – Logo de la société "Aziin Engineering Solutions"

### **1.2.2 – Domaine de la Société**

AZIIN met à profit tout son savoir-faire en analyse, recherche et conception pour concrétiser vos projets. En tant que société pionnière, elle excelle dans la satisfaction des clients les plus exigeants en leur offrant des solutions personnalisées, et ce, dans des délais records et à des coûts compétitifs.

Réalisation :

#### **1/Développement des applications et sites web**

Tout en prenant en considération vos besoins fonctionnels , vos préférences techniques , votre planning et votre stratégie marketing .

notre équipe met en œuvre des compétences hautement qualifiées et des outils dédiés pour développer les applications et les sites web

## **2/Développement des applications mobiles**

Réels outils de communication, les applications mobiles viennent de plus en plus compléter le dispositif des plans médias.

Nous mettons notre expertise, passion, et savoir faire, au service de vos besoins ,afin de créer des solutions adéquates qui répondre à vos besoins.

## **3/Transformation Digitale**

La transformation digitale est le processus qui consiste à remplacer complètement les processus métier manuels existants par les toutes dernières alternatives numériques.

On vous accompagne dans vos projets ,on échange les idées, on les discute, on stimule l'imagination et la créativité, on les réinvente pour atteindre les solutions les plus adaptées et les plus performantes en terme de qualité et de temps.

## **1.3 –Problématique et solution proposée**

Ce projet, intitulé "Application de gestion des candidatures", s'inscrit dans le cadre du projet de perfectionnement. Il est réalisé au sein de la société AZIIN Engineering Solutions à Sfax, au sein d'une équipe dédiée au développement des applications Web.

### **1.3.1 –Problématique**

Dans le contexte actuel de compétitivité accrue et d'exigences de productivité toujours plus élevées, la gestion efficace des processus de recrutement, en particulier pour les stages, revêt une importance cruciale. Les méthodes traditionnelles de suivi des candidatures peuvent se révéler inefficaces, entravant la capacité des organisations à répondre aux défis modernes.

Comment alors mettre en place un système informatique de gestion des candidatures de stages adapté, favorisant l'amélioration du processus de sélection, la collaboration entre les équipes et la prise de décision, tout en tenant compte des contraintes et spécificités de l'entreprise?

Cette problématique soulève des enjeux à la fois techniques et organisationnels qui doivent être pris en considération pour développer une solution efficace, pérenne et évolutive. Pour remédier à ces problèmes, nous proposons la réalisation d'une plateforme web dédiée à la gestion des candidatures de stages. Cette plateforme facilitera le suivi des candidats, offrira des fonctionnalités de sauvegarde des informations liées aux missions travaillées, le tout à un coût minimal et avec une sécurité maximale.

### **1.3.2 –Solution Proposée**

Notre solution vise à simplifier le processus de gestion des candidatures de stages, offrant une interface conviviale pour le suivi et l'évaluation des candidats. Elle favorisera également la collaboration entre les différentes parties prenantes du processus de recrutement de stagiaires, tout en garantissant la sécurité des données sensibles associées aux candidatures. Grâce à cette plateforme, l'entreprise pourra optimiser ses processus de recrutement de stagiaires et prendre des décisions éclairées, tout en respectant les spécificités de son fonctionnement.

## **1.4 –Conclusion**

Dans ce chapitre, nous avons situé notre projet dans son contexte. En premier lieu, nous avons présenté l'organisme d'accueil. Avant de dévoiler la solution proposée, nous avons abordé la liste de nos problématiques.

Dans le chapitre suivant, nous entreprenons une étude théorique afin d'analyser les besoins et d'étudier la conception générale de notre application.

# **Chapitre 2 : Lancement Et Analyse De Projet**

## **2.1 –Introduction**

Dans ce chapitre, nous commençons par présenter la première phase du cycle de développement du logiciel, qui consiste à définir les besoins fonctionnels et non fonctionnels de notre plate-forme. Par la suite, nous exposons les choix technologiques utilisés durant la réalisation de notre projet.

## **2.1 –Spécification des besoins**

Dans cette section, nous commençons par identifier les acteurs de notre système. Ensuite, nous recensons les besoins fonctionnels et non fonctionnels.

### **2.2.1 –Identification des acteurs**

Un acteur représente l'abstraction d'un rôle joué par des entités qui interagissent directement avec le système étudié. L'identification des acteurs constitue la première étape de la phase de spécification des besoins. Cette étape permet de distinguer les utilisateurs ou tout autre système qui interagit avec notre solution. Nos acteurs sont les suivants :

- Développeur : Il est responsable de la conception et du développement du système. Les tâches qu'il effectue comprennent :
  - La gestion des Candidatures
- Administrateur/Recruteur : il s'occupe de la publication des offres, la gestion des services ainsi l'affectation des candidats retenus au service adéquat. Son rôle est la gestion des candidatures qui répondent aux offres publié par Lui Même, notifier les candidats par email pour les informer de leurs statuts, et insertion des résultats.
- Candidat : Toute personne qui suit et intéressée par les offres publiées, pour postuler à un appel de Stage, comme il peut suivre son statut et ses demandes.

### **2.2.2 –Identification des besoins fonctionnels**

L'application doit être en mesure de gérer deux types d'utilisateurs : l'administrateur et les candidats. L'application répondra à l'exigence suivante :

- L'administrateur sera chargé d'analyser les besoins au niveau du département afin de détecter les profils à recruter. Il pourra :

- ❖ Ajouter une offre en déterminant les caractéristiques du poste à pourvoir, telles que les missions et tâches principales, critères de performance, salaire, spécialité, type de contrat, le nombre de personnes à recruter, et la ville (lieu de travail).
- ❖ Déterminer le profil du poste en spécifiant la formation requise, les connaissances nécessaires, les expériences requises (junior, expérimenté ou fraîchement diplômé), les compétences clés, le comportement attendu, et les langues (français, anglais, etc.).
- ❖ Publier des offres avec la possibilité de les modifier, consulter ou supprimer.
- ❖ Visualiser l'ensemble des candidatures reçues avec la possibilité d'appliquer des filtres sur celles-ci.

Une fois les besoins en ressources humaines définis, l'administrateur pourra prendre les mesures nécessaires pour le processus de recrutement.

- Candidats :

- ❖ Les candidats doivent être en mesure de créer un compte en remplissant le formulaire d'inscription sur le site ou de se connecter en saisissant leur login et mot de passe.
- ❖ L'application doit permettre aux candidats de créer un profil, facilitant ainsi leur capacité à postuler à plusieurs offres d'emploi sans avoir à retaper leurs informations personnelles à chaque fois. Ils peuvent également télécharger les documents nécessaires tels que le CV et la lettre de motivation scannés.
- ❖ Les candidats peuvent visualiser l'ensemble des offres publiées, en filtrant celles qui sont encore valables (n'ayant pas atteint la date d'expiration), et choisir l'offre qui correspond à leurs attentes pour y postuler.
- ❖ Une fois qu'un candidat a postulé, il doit pouvoir consulter ou suivre l'état de ses demandes.

### **2.2.3 –Identification des besoins non fonctionnels**

À côté des besoins fonctionnels énumérés, notre application doit également répondre à plusieurs besoins non fonctionnels visant à améliorer l'expérience utilisateur, tels que :

- **La sécurité** : C'est un facteur primordial. Cela englobe la nécessité d'établir une connexion sécurisée pour accéder aux services de l'application, la gestion sécurisée des mots de passe, et la déconnexion automatique après une certaine durée d'inactivité.
- **L'ergonomie et la simplicité** : Notre application doit répondre aux standards d'ergonomie, et son interface utilisateur doit être simple, facile à comprendre et à manipuler.

- La portabilité et la compatibilité :** Notre application est totalement portable. Nous avons utilisé un Docker Compose qui regroupe toutes les images de nos micro-services et de notre client front-end. De plus, elle est compatible avec tous les navigateurs web et tous les dispositifs, que ce soit sur PC, iPad, etc.
- La fiabilité :** Notre application doit fonctionner de manière cohérente et sans erreurs, assurant ainsi une expérience utilisateur fiable.

## 2.3 –Besoins techniques

Les besoins techniques englobent ceux liés à l'environnement de travail. Pour la réalisation de notre application, nous avons utilisé plusieurs moyens matériels et logiciels.

### 2.3.1 –Environnement matériel

Pour mener à bien la réalisation, nous avons utilisé comme environnement matériel un poste de travail présentant les caractéristiques indiquées dans le tableau 2.1.

Caractéristiques	Ordinateurs
Marque	Dell
Processeur	Intel® Core™ i7-13650HX
Mémoire vive	24,00 GO
Disque Dur	512 SSD
Système d'exploitation	Windows 11 Home

Table 2.1 – Environnement matériel utilisé

### 2.3.2 –Environnement logiciel

Dans cette partie, nous présentons les logiciels et les langages de programmation utilisés pour la réalisation de notre plateforme.

#### — Visual Studio

L'IDE Visual Studio est un panneau de lancement créatif que vous pouvez utiliser pour modifier, déboguer et générer du code, puis publier une application. En plus de l'éditeur et du débogueur standard fournis par la plupart des IDE, Visual Studio inclut des compilateurs, des outils de complétion de code, des concepteurs graphiques et bien d'autres fonctionnalités pour améliorer le processus du développement de logiciels.



Figure 2.1 – Logo Visual Studio

### 2.3.3 –Environnement Base de données

#### — SQL Server

désigne couramment un serveur de base de données. La définition du SQL Server est étroitement liée à celle du langage SQL (Structured Query Language), un langage informatique permettant d'exploiter des bases de données.



Figure 2.2 – Logo SQL Server

### 2.3.4 –Framework utilisée

#### — ASP.NET Core

ASP.NET Core est un Framework Web gratuit et open-source, développé par Microsoft et la communauté. Il est plus performant qu'ASP.NET. C'est un Framework modulaire qui fonctionne à la fois avec le Framework .NET, sous Windows, et .NET en multiplateforme.

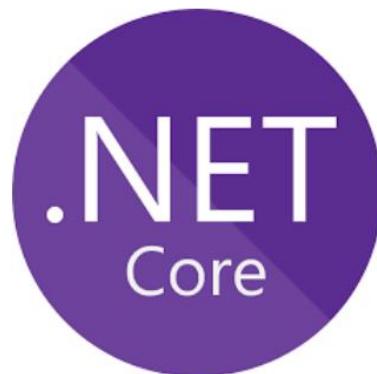


Figure 2.3 – Logo ASP.NET Core

### **2.3.5 –Bibliothèque utilisé**

#### **— React**

(aussi appelée React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook (maintenant Meta) depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'applications web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

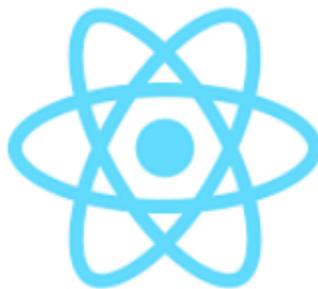


Figure 2.4 – Logo React

### **2.3.6 –Modélisation UML**

#### **— StarUML**

StarUML est un outil de génie logiciel dédié à la modélisation UML et édité par la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOS.



Figure 2.5 – Logo StarUML

### **2.3.7 –Autre Technologies utilisées**

#### **— Docker**

Docker est une plate-forme ouverte pour le développement, la livraison et l'exécution d'applications. Il permet de séparer nos applications de notre infrastructure, nous permettant ainsi de livrer rapidement des logiciels.

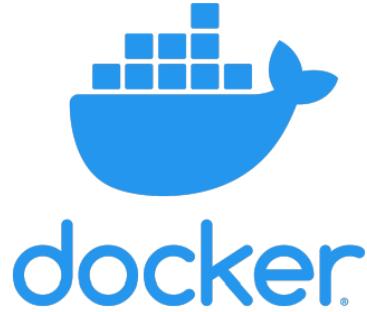


Figure 2.6 – Logo Docker

#### — Swagger

Swagger est un ensemble d'outils open source construits autour de la spécification OpenAPI qui peut nous aider à concevoir, créer, documenter et consommer des API REST.



Figure 2.7 – Logo Swagger

#### — Cloudinary

Dans le processus de développement, j'ai collaboré avec Cloudinary pour simplifier et optimiser le téléchargement de fichiers, notamment d'images, de CV au format PDF, et de lettres de motivation également au format PDF. L'intégration de Cloudinary a permis une gestion efficace des fichiers multimédias, offrant une solution rapide et évolutive pour le stockage et la manipulation des ressources. Cette collaboration a considérablement amélioré l'expérience utilisateur en facilitant le téléchargement de documents, tout en assurant une gestion efficiente des ressources multimédias au sein de l'application.



Figure 2.8 – Logo Cloudinary

## **2.4 –Conclusion**

Ce chapitre nous a permis de couvrir, dans la première partie, les acteurs principaux ainsi que les différents besoins fonctionnels et non fonctionnels. Par la suite, nous avons exposé les différentes technologies, frameworks et logiciels à utiliser lors du développement.

Afin de suivre un cheminement logique dans ce rapport, une modélisation UML constituera le sujet du prochain chapitre.

# **Chapitre 3 : Modélisation UML**

## **3.1 –Introduction**

Après avoir spécifié et analysé les besoins fonctionnels et non fonctionnels, nous entamons la conception de l'application.

## **3.2 –Langage de modélisation**

Pour modéliser notre solution, nous allons utiliser un langage de modélisation. Notre choix s'oriente vers le langage UML en raison des avantages qu'il offre, notamment sa standardisation et les divers diagrammes qu'il propose, ainsi que l'utilisation d'une notation graphique simple et compréhensible.

## **3.3 –Modélisation du projet**

Dans notre projet, nous allons nous focaliser sur deux des diagrammes les plus célèbres dans le langage UML : le diagramme de classe et le diagramme de cas d'utilisation, afin de mieux comprendre notre solution.

### **3.3.1 –Diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation est un diagramme UML de vue statique. Il peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. Lorsqu'ils sont bien conçus, les diagrammes de cas d'utilisation peuvent aider l'équipe à collaborer et représenter :

- Les scénarios dans lesquels notre application interagit avec des personnes, des organisations ou des systèmes externes.
- Les objectifs que notre système permet aux acteurs d'atteindre.
- La portée de votre système.

Le diagramme suivant (Figure 3.1) présente les cas d'utilisation illustrant les fonctionnalités offertes à l'administrateur (Recruteur) :

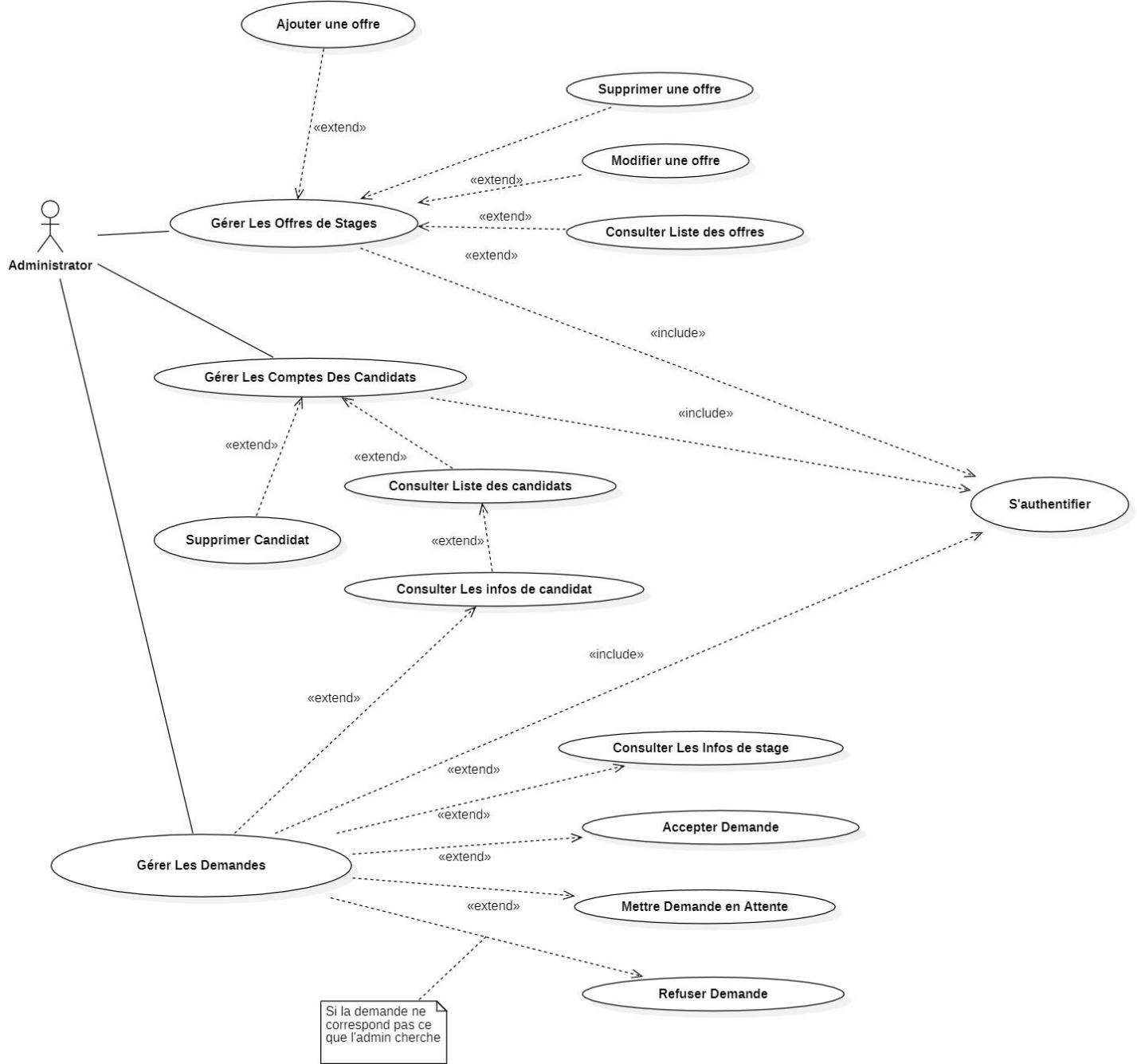


Figure 3.1 – Diagramme de cas d'utilisation pour L'Administrateur

La figure 3.1 illustre un Administrateur/Recruteur, une fois authentifié, bénéficie d'un ensemble de services globaux. Va recevoir l'ensemble des candidatures qu'il gère, puis peut choisir s'il accepte les demandes et peut aussi ajouter des autres offres de stages.

Le diagramme suivant (Figure 3.2) présente les cas d'utilisation illustrant les fonctionnalités offertes au Candidat (Client) :

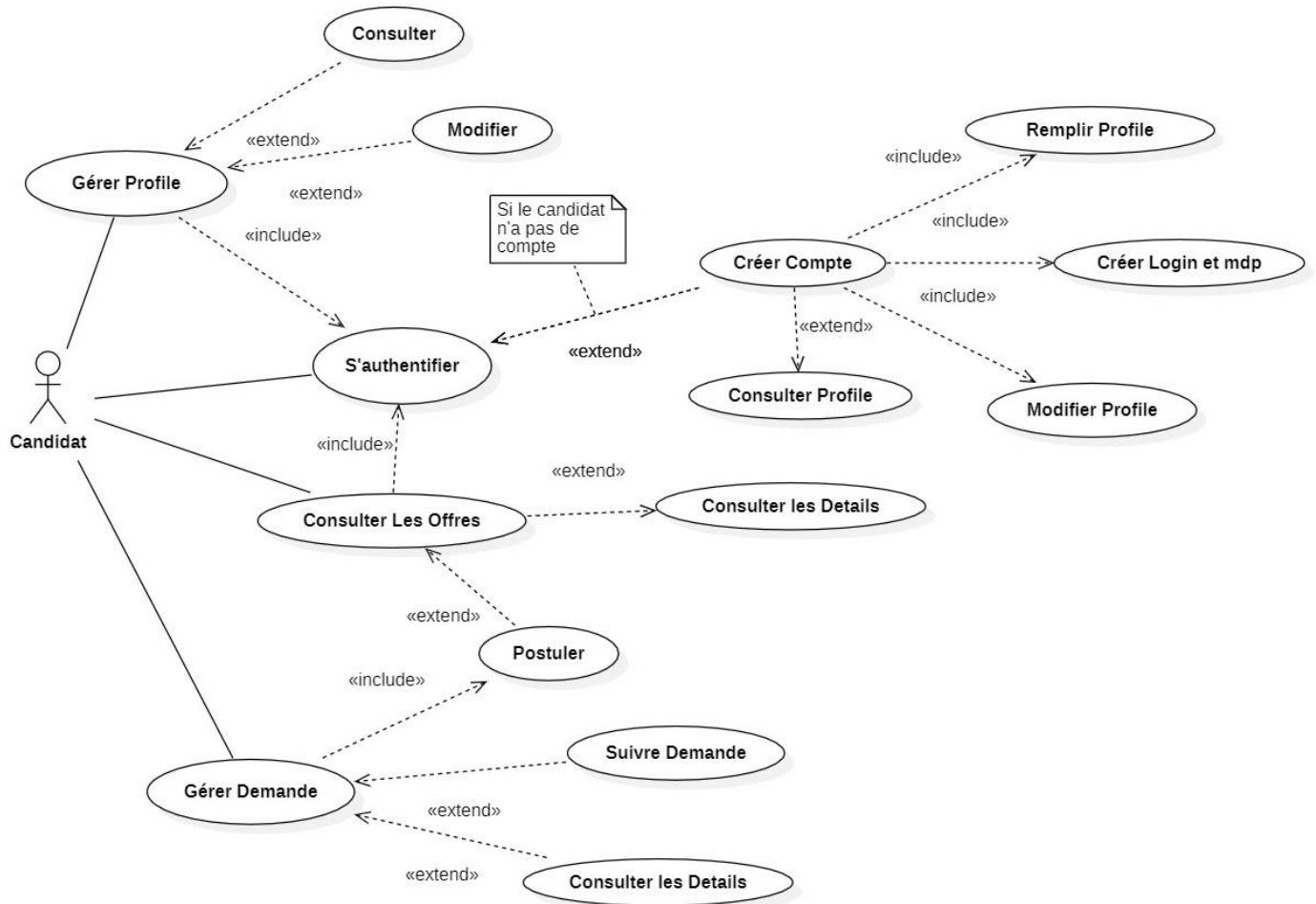


Figure 3.2 – Diagramme de cas d'utilisation pour Candidat

### 3.3.2 –Diagramme de classe

Un diagramme de classes dans le langage UML est un type de diagramme de structure statique qui décrit la structure d'un système en montrant les classes du système, leurs attributs, opérations (ou méthodes) et les relations entre les objets.

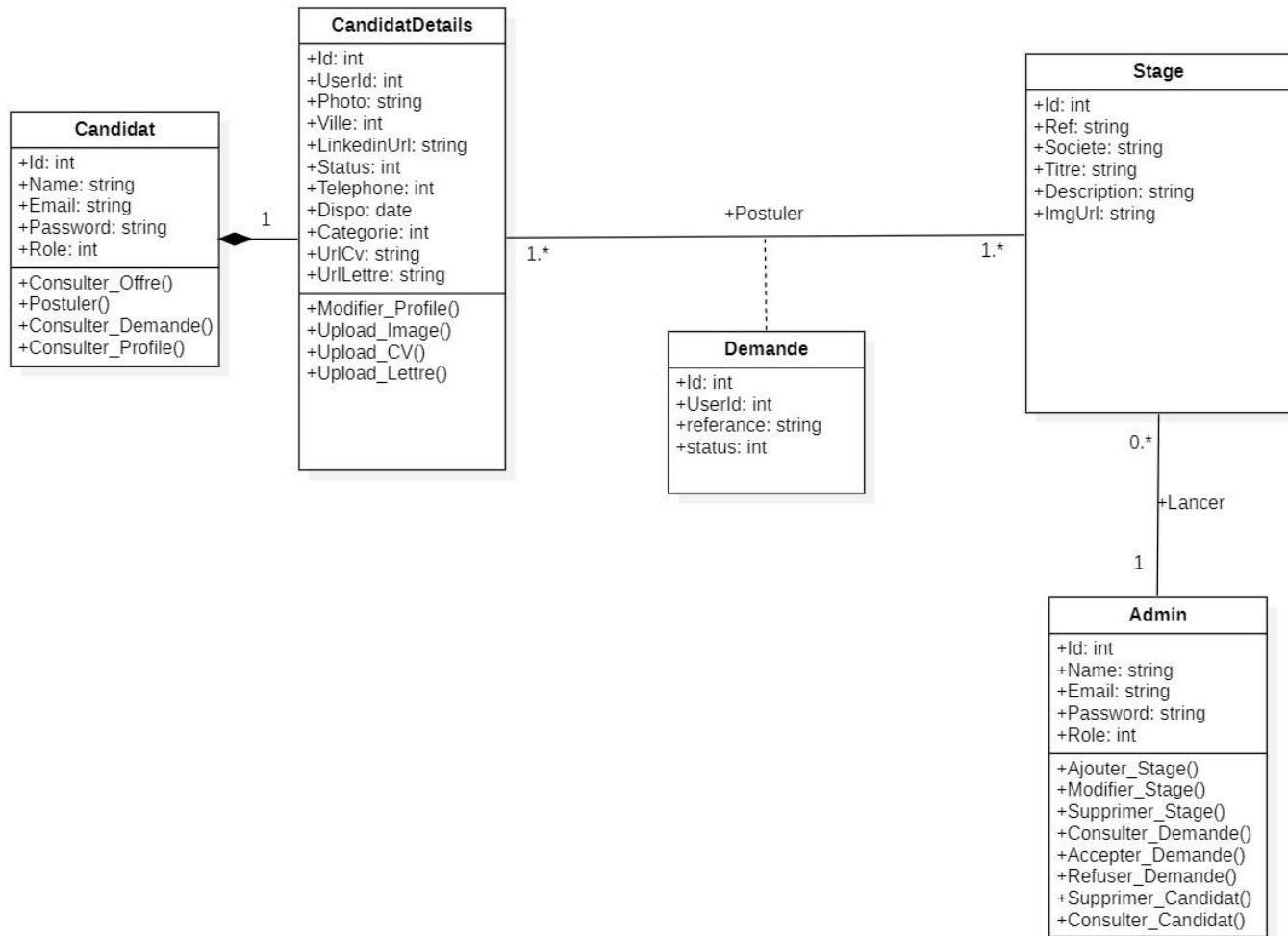


Figure 3.3 – Diagramme de classe

### 3.4 –Conclusion

Au cours de ce chapitre, nous avons défini la conception de notre projet en spécifiant le diagramme de classe et le diagramme de cas d'utilisation. Dans le chapitre suivant, nous aborderons la liste des différentes interfaces de notre application.

## Chapitre 4 : Réalisation

### 4.1 –Introduction

Dans ce chapitre, nous aborderons les différentes interfaces réalisées dans notre application.

## 4.2 –Interface d'accueil

### 4.2.1 –Page d'accueil



Figure 4.1 – Page d'accueil

### 4.2.2–Page de S'inscrire

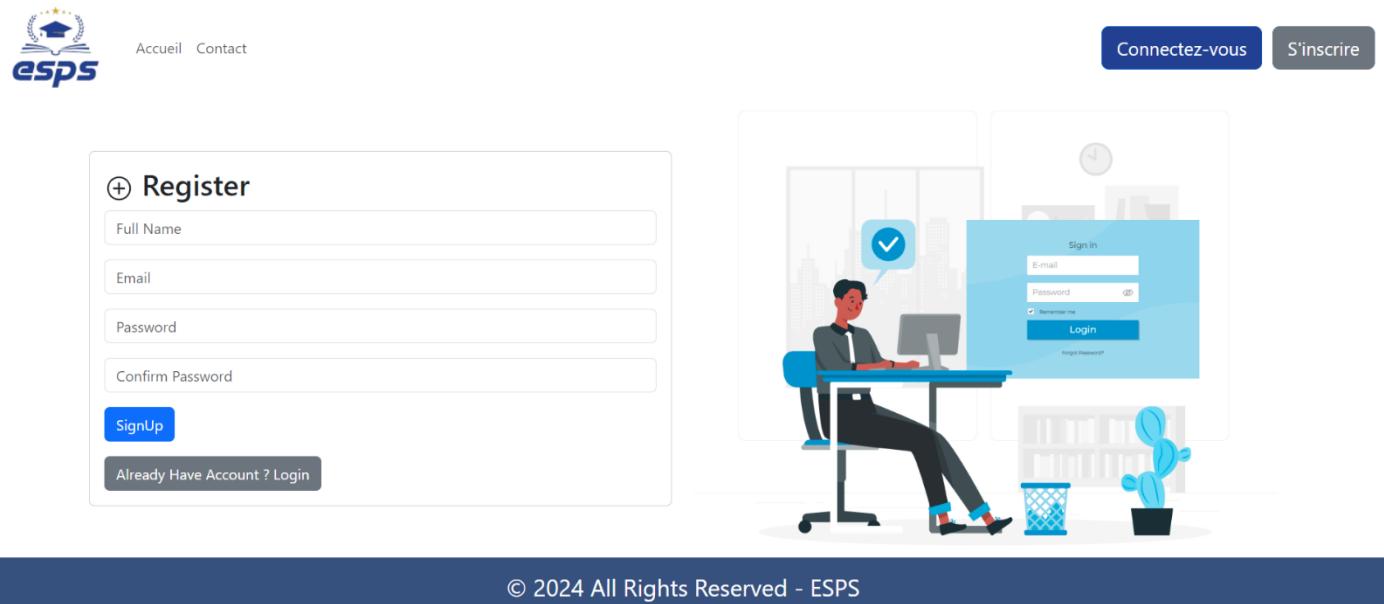


Figure 4.2 – Page de S'inscrire

Pendant mon travail sur le développement de la fonction d'inscription d'une page, j'ai collaboré avec la bibliothèque Bcrypt pour sécuriser les mots de passe des utilisateurs. En utilisant cette technologie de hachage robuste, j'ai renforcé la sécurité des informations sensibles. De plus, j'ai

intégré Swagger pour faciliter le processus de test et de documentation. Cette combinaison de Bcrypt et Swagger a permis de créer une solution fiable, assurant la confidentialité des données des utilisateurs tout en simplifiant le processus de développement et de test.

**POST /api/register**

**Parameters**

No parameters

**Request body**

application/json

**Example Value | Schema**

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "role": 0
}
```

**Responses**

Code	Description	Links
200	Success	No links

Figure 4.2.1 – Api register

Curl

```
curl -X 'POST' \
'http://localhost:5135/api/register' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "anis",
  "email": "an@an.com",
  "password": "123",
  "role": 0
}'
```

Request URL

<http://localhost:5135/api/register>

Server response

Code	Details
201	Undocumented

Response body

```
{
  "id": 2002,
  "name": "anis",
  "email": "an@an.com",
  "role": 0,
  "userDetails": null
}
```

Download

Response headers

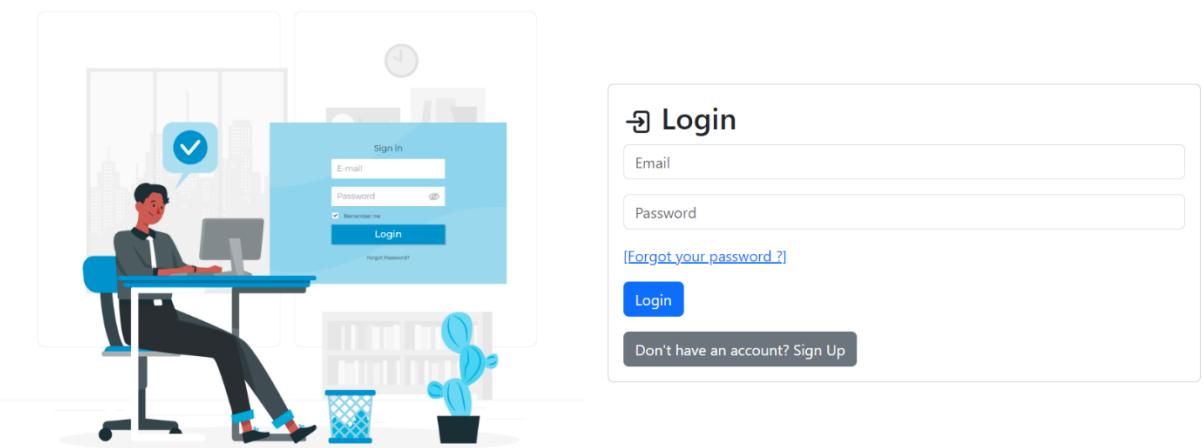
```
content-type: application/json; charset=utf-8
date: Mon, 11 Mar 2024 11:52:38 GMT
location: success
server: Kestrel
transfer-encoding: chunked
```

Figure 4.2.2 – Résultat Api register

	Id	Name	Email	Password	Role
►	1	a	a@a.com	\$2a\$11\$VWn3c0cH/ABwmcHw23V6dunvY1m...	1
	2	achraf	b@b.com	\$2a\$11\$jl1mcOLXZ3xUq5jjBA1LLeQo.CeYMYg...	0
	1002	g	g@g.com	\$2a\$11\$vR11xH3ZFoL1Z9kKy5D1TOuS7xCThJ...	0
	2002	anis	an@an.com	\$2a\$11\$kcwQXQ3ccBKqgHhvrlB0dhnlrvIOY...	0
∅	NULL	NULL	NULL	NULL	NULL

Figure 4.2.3 – Résultat dans la base de données

#### 4.2.3 –Page Connectez-vous



The screenshot shows the ESPS login page. At the top left is the ESPS logo. To its right are links for "Accueil" and "Contact". On the far right are buttons for "Connectez-vous" and "S'inscrire". Below this, there's a large illustration of a person sitting at a desk, working on a computer, with a checkmark icon above them. To the right of the illustration is a blue rectangular area containing a "Sign in" form with fields for "E-mail" and "Password", a "Remember me" checkbox, and a "Login" button. To the right of the form is a "Forgot Password?" link. Below the illustration is a trash can and a small potted plant. To the right of the illustration is a "Login" form with fields for "Email" and "Password", a "Forgot your password?" link, a "Login" button, and a "Don't have an account? Sign Up" link. At the bottom of the page is a dark blue footer bar with the text "© 2024 All Rights Reserved - ESPS".

Figure 4.3 – Page Connectez-vous

Dans le cadre de la mise en place de la page de connexion, j'ai également travaillé avec **JSON Web Tokens (JWT)** pour renforcer le mécanisme d'authentification. En utilisant des tokens et des cookies, j'ai mis en place un système sécurisé permettant aux utilisateurs de se connecter de manière fiable et efficace. Les JWT ont été intégrés pour garantir l'authenticité des informations d'identification, offrant ainsi une couche supplémentaire de sécurité. Cette approche a permis une gestion transparente des sessions utilisateur tout en renforçant la protection contre les menaces potentielles liées à la sécurité des données.



**HEADER**  
ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

**PAYOUT**  
DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

**SIGNATURE**  
VERIFICATION

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload), secretKey)
```

POST /api/login

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
{
  "email": "string",
  "password": "string"
}
```

Responses

Code	Description	Links
200	Success	No links

Figure 4.3.1 – Api login

Curl

```
curl -X 'POST' \
'http://localhost:5135/api/login' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "email": "an@an.com",
  "password": "123"
}'
```

Request URL

<http://localhost:5135/api/login>

Server response

Code	Details	Links
200	<p>Response body</p> <pre>{   "message": "success" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon, 11 Mar 2024 12:15:00 GMT server: Kestrel transfer-encoding: chunked</pre>	 
Responses		Links
Code	Description	Links
200	Success	No links

Figure 4.3.2 – Résultat Api login

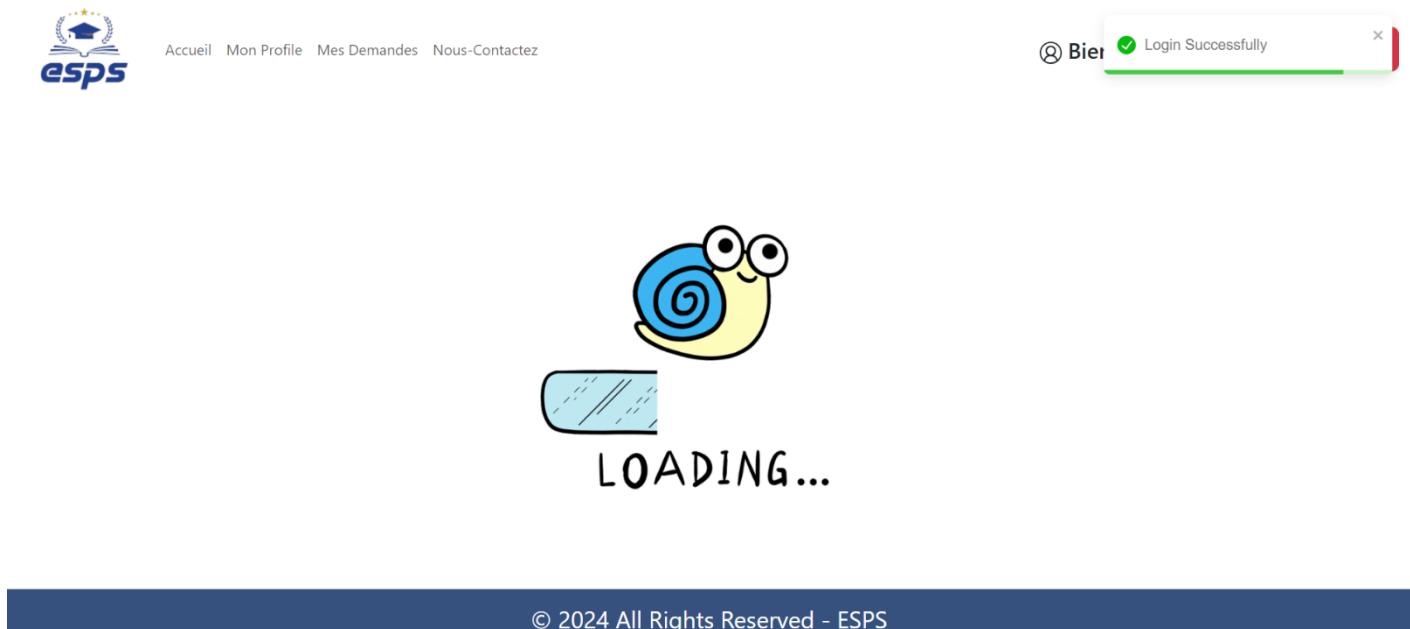


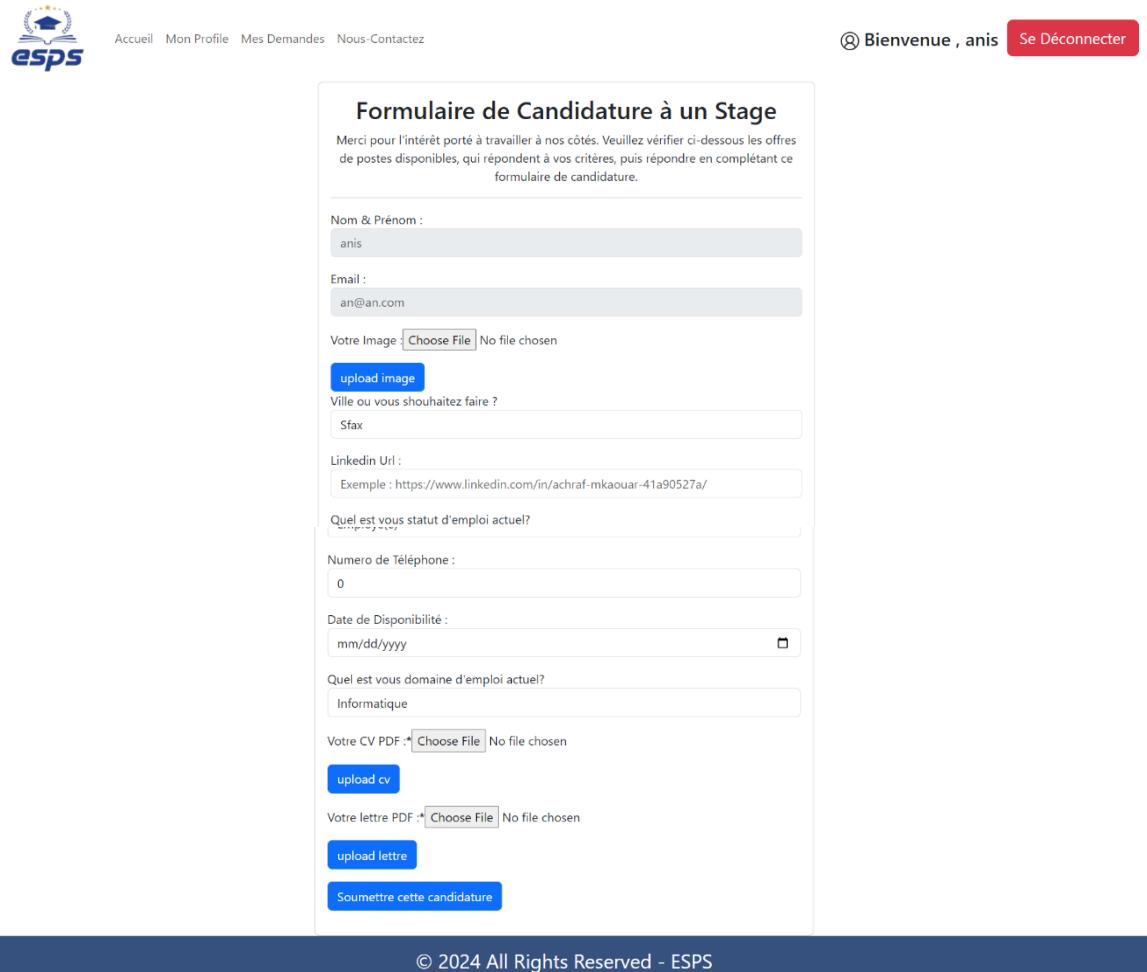
Figure 4.3.3 – Loader login

J'ai fait cette page de Loading pour Vérifier si User est Admin ou Candidat et aussi pour Vérifier s'il est Candidat alors vérifier si le profile est complété ou non . On va voir dans le sous chapitre suivant Interface de Candidat .

## 4.3 –Interface de Candidat

1<sup>er</sup> cas : Si Le Candidat n'a pas complété Son Profile .

### 4.3.1 –Page Préinscription



The screenshot shows the 'Formulaire de Candidature à un Stage' (Stage Application Form) on the ESPS website. The form is divided into several sections:

- Identité :** Nom & Prénom : anis, Email : an@an.com.
- Image :** Votre Image [Choose File] No file chosen, upload image button.
- Localisation :** Ville ou vous souhaitez faire ? Sfax.
- LinkedIn :** Linkedin Url : Exemple : <https://www.linkedin.com/in/achraf-mkaouar-41a90527a/>.
- Statut d'emploi :** Quel est votre statut d'emploi actuel? Choisir...
- Contact :** Numéro de Téléphone : 0.
- Date :** Date de Disponibilité : mm/dd/yyyy.
- Domaine d'emploi :** Quel est votre domaine d'emploi actuel? Informatique.
- CV :** Votre CV PDF: [Choose File] No file chosen, upload cv button.
- Lettre de motivation :** Votre lettre PDF: [Choose File] No file chosen, upload lettre button.
- Action :** Soumettre cette candidature button.

At the bottom, a dark blue footer bar contains the text: © 2024 All Rights Reserved - ESPS.

Figure 4.4 – Page Préinscription

2ème cas : Si Le Candidat a complété Son Profile .

### 4.3.2 –Page Candidat Accueil

The screenshot shows the candidate's profile on a platform. On the left, there is a sidebar with a user icon, the letter 'A', the email 'a@a.com', and two links: 'Mon profile' and 'Mes Demandes'. The main area displays a job listing for 'AZIIN Engineering Solutions' with the subject 'Création d'une application e-learning'. The listing includes a brief description: 'Conception et réalisation d'un environnement d'apprentissage en ligne avec une combinaison de vidéo, de tests à choix multiples et de gamification. L'obtention d'une attestation électronique nécessite le passage d'un test d'évaluation.' A 'Postulez Maintenant' button is at the bottom.

Figure 4.4.1 – Page Candidat Accueil

The screenshot shows the 'GET /api/stages' API endpoint in a developer tool. The 'Parameters' section indicates 'No parameters'. Below it are 'Execute' and 'Clear' buttons. The 'Responses' section is currently empty.

Figure 4.4.2 – Api Get all stages

Curl

```
curl -X 'GET' \
'http://localhost:5135/api/stages' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5135/api/stages
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "id": 2,     "ref": "HL-01",     "societe": "AZIIN Engineering Solutions",     "titre": "Création d'une application e-learning",     "description": "Conception et réalisation d'un environnement d'apprentissage en ligne avec une combinaison de vidéo, de tests à choix multiples et de gamification. L'obtention d'une attestation électronique nécessite le passage d'un test d'évaluation.",     "imgUrl": "http://res.cloudinary.com/dujxr3zp/image/upload/v1710168748/dwgahxrrgk7h5pzy21.jpg"   },   {     "id": 3,     "ref": "HL-02",     "societe": "AZIIN Engineering Solutions",     "titre": "Application spécifique de don pour les associations",     "description": "L'utilisateur après inscription et création de profil, peut publier directement sur la plateforme l'image et l'état de leurs vieux objets . Les associations inscrites sur la plateforme, peuvent obtenir automatiquement les différents produits selon leurs besoins saisis préalablement..",     "imgUrl": "http://res.cloudinary.com/dujxr3zp/image/upload/v1710161334/r3pq7jotlvxd3soxftbi.jpg"   } ]</pre> <p><a href="#">Copy</a> <a href="#">Download</a></p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon,11 Mar 2024 12:49:42 GMT server: Kestrel transfer-encoding: chunked</pre>

Figure 4.4.3 – Résultat Api Get all stages

GET /api/user

Parameters

No parameters

[Execute](#) [Clear](#)

Figure 4.4.4 – Api Get User With Cookies

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5135/api/user' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5135/api/user
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 2002,   "name": "anis",   "email": "an@an.com",   "role": 0,   "userDetails": null }</pre> <p><a href="#">Copy</a> <a href="#">Download</a></p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon,11 Mar 2024 12:55:55 GMT server: Kestrel transfer-encoding: chunked</pre>

Responses

Code	Description	Links
200	Success	No links

Figure 4.4.5 – Résultat Api Get User With Cookies

Bienvenue , anis Se Déconnecter

Figure 4.4.6 – Button Se Déconnecter

POST /api/logout

Parameters

No parameters

Responses

Code	Description	Links
200	Success	No links

Figure 4.4.7 – Api Logout

Responses

Curl

```
curl -X 'POST' \
'http://localhost:5135/api/logout' \
-H 'accept: */*' \
-d ''
```

Request URL

```
http://localhost:5135/api/logout
```

Server response

Code	Details
200	Response body { "message": "success" }  Response headers content-type: application/json; charset=utf-8 date: Mon,11 Mar 2024 13:03:55 GMT server: Kestrel transfer-encoding: chunked

Responses

Code	Description	Links
200	Success	No links

Figure 4.4.8 – Résultat Api Logout

### 4.3.3 –Page Candidat Profile

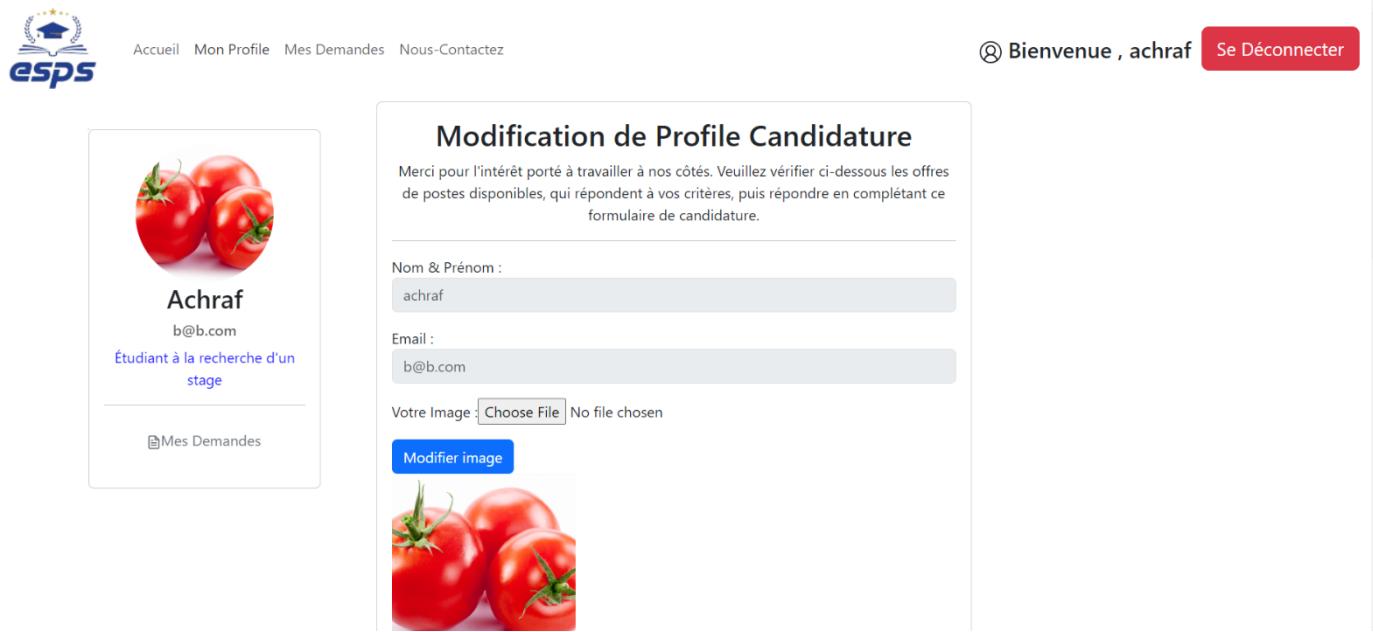


Figure 4.5.1 – Page Profile

The screenshot shows a Swagger UI interface for a 'PUT /api/user/{id}/update' endpoint. The 'Parameters' section shows a required parameter 'id' of type 'integer(\$int32)' with a placeholder 'id'. The 'Request body' section is set to 'application/json'. Below it is an 'Example Value' section containing a JSON schema for the user profile:

```
{
  "userId": 0,
  "photo": "string",
  "ville": 0,
  "linkedinUrl": "string",
  "status": 0,
  "telephone": 0,
  "dispo": "2024-03-11T21:50:04.900Z",
  "categorie": 0,
  "urlCv": "string",
  "urlLettre": "string"
}
```

Figure 4.5.2 – Api Profile

Curl

```
curl -X 'PUT' \
  'http://localhost:5135/api/user/2002/update' \
  -H 'accept: */*' \
  -H 'Content-type: application/json' \
  -d '{
    "userId": 2002,
    "photo": "string",
    "ville": 0,
    "linkedinUrl": "string",
    "status": 0,
    "telephone": 0,
    "dispo": "2024-03-11T21:51:36.926Z",
    "categorie": 0,
    "urlCV": "string",
    "urllettre": "string"
  }'
```

Request URL

```
http://localhost:5135/api/user/2002/update
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "message": "Mise à jour réussie",   "userDetails": {     "id": 2,     "userId": 2002,     "photo": "string",     "ville": 0,     "linkedinUrl": "string",     "status": 0,     "telephone": 0,     "dispo": "2024-03-11T21:51:36.926Z",     "categorie": 0,     "urlCV": "string",     "urllettre": "string",     "user": null   } }</pre> <p> </p>

Figure 4.5.3 – Résultat Api Profile

#### 4.3.4 –Page Candidat Demandes



The screenshot shows the ESPS website's user profile section. At the top, there is a logo with three stars above a graduation cap, followed by the text 'Accueil Mon Profile Mes Demandes Nous-Contactez'. To the right, there is a welcome message 'Bienvenue , achraf' and a 'Se Déconnecter' button. Below this, the 'Mes Demandes' section is displayed with a table:

Demande Numéro	Référence du Stage	Détails du Stage	Status
1	hl10	<a href="#">Détails du Stage</a>	Refusé(e)

At the bottom of the page, a footer bar contains the text '© 2024 All Rights Reserved - ESPS'.

Figure 4.6.1 – Page Demandes

GET /api/demande/all

Parameters

No parameters

Try it out

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5135/api/demande/all' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5135/api/demande/all
```

Figure 4.6.2 – Api Demande

**Server response**

Code	Details	Links
200	<b>Response body</b> <pre>[ { "id": 1, "userId": 2, "reference": "hl10", "status": 2 } ]</pre> <div style="text-align: right; margin-top: -20px;"> <a href="#">Copy</a> <a href="#">Download</a> </div> <b>Response headers</b> <pre>content-type: application/json; charset=utf-8 date: Mon, 11 Mar 2024 21:54:13 GMT server: Kestrel transfer-encoding: chunked</pre>	
<b>Responses</b>		
Code	Description	Links
200	Success	No links

Figure 4.6.3 – Résultat Api Demande

#### 4.3.5 –Page Candidat Postuler

The screenshot shows a web application interface for a candidate named Achraf. On the left, there's a sidebar with a profile picture of two tomatoes, the name 'Achraf', the email 'b@b.com', and a status 'Étudiant à la recherche d'un stage'. Below this are links for 'Mon profile' and 'Mes Demandes'. The main content area has a header 'Stage Référence : #HL-01'. It displays information about a job reference '#HL-01' at 'AZIIN Engineering Solutions' for a subject 'Création d'une application e-learning'. The description details the creation and realization of an environment for online learning, involving video, multiple-choice tests, and gamification. A confirmation dialog box is overlaid on the page, asking if the user wants to apply for the subject 'Création d'une application e-learning?' with 'ANNULER' and 'CONFIRMER' buttons.

Figure 4.7.1 – Page Postuler

POST /api/demande/add

**Parameters**

No parameters

**Request body** application/json

Example Value | Schema

```
{
  "id": 0,
  "userId": 0,
  "reference": "string",
  "status": 0
}
```

**Responses**

Code	Description	Links
200	Success	No links

Figure 4.7.2 – Api Postuler

Curl

```
curl -X 'POST' \
'http://localhost:5135/api/demande/add' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "id": 0,
  "userId": 2002,
  "reference": "HL-02",
  "status": 0
}'
```

Request URL

<http://localhost:5135/api/demande/add>

Server response

Code	Details	Links
200	<p>Response body</p> <pre>{   "message": "x",   "demande": {     "id": 2,     "userId": 2002,     "reference": "HL-02",     "status": 0   } }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon, 11 Mar 2024 21:58:09 GMT server: Kestrel transfer-encoding: chunked</pre>	<a href="#">Copy</a> <a href="#">Download</a>

Responses

Code	Description	Links
------	-------------	-------

Figure 4.7.3 – Résultat Api Postuler

## 4.4 –Interface Admin

Si Rôle est 1 alors C'est L'admin :

### 4.4.1 –Page Admin Accueil

The screenshot shows the Admin Home page. At the top, there is a logo for 'esps' with a graduation cap icon. To its right are links for 'Accueil', 'Mon Profile', 'Mes Demandes', and 'Nous-Contactez'. On the far right, there is a 'Bienvenue, a' message with a user icon and a red 'Se Déconnecter' button. Below this, a dark blue header bar contains a user icon labeled 'Admin', followed by 'Accueil', 'Tous Les Candidateurs', 'Ajouter Stage', 'List des Stages', and 'Demandes'. The main content area has a heading 'Bienvenue Admin' and a timestamp 'Current time: 11:01:00 PM'. At the bottom, a dark blue footer bar displays the copyright notice '© 2024 All Rights Reserved - ESPS'.

Figure 4.8.1 – Page Admin

### 4.4.2 –Page List des Candidatures

The screenshot shows the 'List des Candidature' page. At the top, there is a logo for 'esps' with a graduation cap icon. To its right are links for 'Accueil', 'Mon Profile', 'Mes Demandes', and 'Nous-Contactez'. On the far right, there is a 'Bienvenue, a' message with a user icon and a red 'Se Déconnecter' button. Below this, a dark blue header bar contains a user icon labeled 'Admin', followed by 'Accueil', 'Tous Les Candidateurs', 'Ajouter Stage', 'List des Stages', and 'Demandes'. The main content area has a heading 'List des Candidature' and a table listing four candidates. The table columns are 'Numéro', 'Nom & Prénom', 'Email', and 'Complété'. Each row includes a green checkmark or a red X in the 'Complété' column, and two buttons: 'Infos' (blue) and 'Supprimer' (red). At the bottom, a dark blue footer bar displays the copyright notice '© 2024 All Rights Reserved - ESPS'.

Figure 4.8.2 – Page List des Candidatures

GET /api/users

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5135/api/users' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5135/api/users
```

Figure 4.8.3 – Api Users

http://localhost:5135/api/users

Server response

Code Details

200 Response body

```
[  
  {  
    "id": 1,  
    "name": "a",  
    "email": "a@a.com",  
    "role": 1,  
    "userDetails": null  
  },  
  {  
    "id": 2,  
    "name": "achraf",  
    "email": "b@b.com",  
    "role": 0,  
    "userDetails": null  
  },  
  {  
    "id": 1002,  
    "name": "g",  
    "email": "g@g.com",  
    "role": 0,  
    "userDetails": null  
  },  
  {  
    "id": 2002,  
    "name": "anis",  
    "email": "an@an.com",  
    "role": 0,  
    "userDetails": null  
  }]
```

Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 13 Mar 2024 22:04:32 GMT  
server: Kestrel  
transfer-encoding: chunked
```

Download

Figure 4.8.4 – Résultat Api Users

**Infos Candidat Numéro : 2**

Nom & Prénom : **achraf**

Ville : **Sfax**

Status : **Employé(e)**

Telephone : **2015548**

Disponibilité : **2024-02-27**

Domaine : **Informatique**

LinkedIn : [hvjhvjh](#)

Cv : [Voir](#)

Lettre : [Voir](#)

Tous les informations de Candidature

**OK**

Figure 4.8.5 – Page Infos Candidat

Are you sure you want to delete this user Number 1002?  
You can't undo this operation

**NO** **YES**

**© 2024 All Rights Reserved - ESPS**

Figure 4.8.6 – Page Validation Suppression

#### 4.4.3 –Page Ajouter Stage

The screenshot shows a form titled "Ajoutez Stage". It includes fields for "Reference" (with a building icon), "Société" (with a person icon), "Titre De Sujet" (with a document icon), and "description" (with a list icon). There is also a file input field for "Choose File" with "No file chosen" and a "upload image" button. A "Valider" button is at the bottom.

Figure 4.8.7 – Page Ajouter Stage

The screenshot shows the API endpoint `/api/stage/add`. It indicates "No parameters" and has a "Request body" section set to `application/json`. An "Example Value" is provided as a JSON object:

```
{
  "id": 0,
  "ref": "string",
  "societe": "string",
  "titre": "string",
  "description": "string",
  "imgUrl": "string"
}
```

Figure 4.8.8 – Api Ajouter Stage

Curl

```
curl -X 'POST' \
  'http://localhost:5135/api/stage/add' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "ref": "string",
    "societe": "string",
    "titre": "string",
    "description": "string",
    "imgUrl": "string"
  }'
```

Request URL

```
http://localhost:5135/api/stage/add
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "message": "Mise à jour réussie",   "stage": {     "id": 4,     "ref": "string",     "societe": "string",     "titre": "string",     "description": "string",     "imgUrl": "string"   } }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon, 11 Mar 2024 22:11:41 GMT server: Kestrel transfer-encoding: chunked</pre>

Figure 4.8.9 – Résultat Api Ajouter Stage

#### 4.4.4 –Page List des Stages



Accueil Mon Profile Mes Demandes Nous-Contactez

Bienvenue , a Se Déconnecter

**Admin** Accueil Tous Les Candidateurs Ajouter Stage List des Stages Demandes

#Référence	Image	Société	Titre	Description	
HL-02		AZIIN Engineering Solutions	Application spécifique de don pour les associations	L'utilisateur après inscription et création de profil, peut publier directement sur la plateforme l'image et l'état de leurs vieux objets . Les associations inscrites sur la plateforme, peuvent obtenir automatiquement les différents produits selon leurs besoins saisis préalablement.	<button>Supprimer</button>
HL-01		AZIIN Engineering Solutions	Création d'une application e-learning	Conception et réalisation d'un environnement d'apprentissage en ligne avec une combinaison de vidéo, de tests à choix multiples et de gamification. L'obtention d'une attestation électronique nécessite le passage d'un test d'évaluation.	<button>Supprimer</button>

© 2024 All Rights Reserved - ESPS

Figure 4.8.10 – Page List des Stages

GET /api/stages

**Parameters**

No parameters

**Responses**

**Curl**

```
curl -X 'GET' \
'http://localhost:5135/api/stages' \
-H 'accept: */*'
```

**Request URL**

```
http://localhost:5135/api/stages
```

**Server response**

**Code** Details

Figure 4.8.11 – Api Stages

**Curl**

```
curl -X 'GET' \
'http://localhost:5135/api/stages' \
-H 'accept: */*'
```

**Request URL**

```
http://localhost:5135/api/stages
```

**Server response**

**Code** Details

200

**Response body**

```
[
  {
    "id": 2,
    "ref": "HL-01",
    "societe": "AZIIN Engineering Solutions",
    "titre": "Création d'une application e-learning",
    "description": "Conception et réalisation d'un environnement d'apprentissage en ligne avec une combinaison de vidéo, de tests à choix multiples et de gamification. L'obtention d'une attestation électronique nécessite le passage d'un test d'évaluation..",
    "imgUrl": "http://res.cloudinary.com/dujxr3zp/image/upload/v1710160748/dugahkxrgrgk7h5pzzz21.jpg"
  },
  {
    "id": 3,
    "ref": "HL-02",
    "societe": "AZIIN Engineering Solutions",
    "titre": "Application spécifique de don pour les associations",
    "description": "L'utilisateur après inscription et création de profil, peut publier directement sur la plateforme l'image et l'état de leurs vieux objets . Les associations inscrites sur la plateforme, peuvent obtenir automatiquement les différents produits selon leurs besoins saisis préalablement..",
    "imgUrl": "http://res.cloudinary.com/dujxr3zp/image/upload/v1710161334/r3pq7jotlvxd3soxftbi.jpg"
  }
]
```

**Response headers**

```
content-type: application/json; charset=utf-8
date: Mon, 11 Mar 2024 12:49:42 GMT
server: Kestrel
transfer-encoding: chunked
```

Figure 4.8.12 – Résultat Api Stages

The screenshot shows a list of stages in a table. A modal dialog box is open over the table, asking for confirmation to delete a stage named 'HL-02'. The dialog contains the text: 'Stage Référence : HL-02', 'Vous voulez vraiment Supprimer ce Stage HL-02?', and 'Vous ne pouvez pas annuler cette opération'. At the bottom of the dialog are two buttons: 'NON' (blue) and 'OUI' (red). In the top right corner of the dialog, there is a small text note: 'Publier directement sur la plateforme l'image et l'état de la plateforme, peuvent obtenir automatiquement les informations.' Below the dialog, the table rows are visible.

#Référence	Image	Société	Titre	Description
HL-02		AZIIN Engineering Solutions	Application de don pour les	Vous pouvez publier directement sur la plateforme l'image et l'état de la plateforme, peuvent obtenir automatiquement les informations.
HL-01		AZIIN Engineering Solutions	Création d'une application e-learning	tests à choix multiples et de gamification. L'obtention d'une attestation électronique nécessite le passage d'un test d'évaluation.

© 2024 All Rights Reserved - ESPS

Figure 4.8.13 – Page Validation Suppression Stage

#### 4.4.5 –Page Demandes

The screenshot shows a list of stage requests in a table. The columns are: #, Référence de Stage, Numéro du Candidat, and Status. The table has two rows:

- Row 2: Référence de Stage is 'HL-02', Numéro du Candidat is '2002', Status is 'En Cours de Traitement' (yellow button), Infos Candidat (blue button), Infos Stage (green button), and Decision (black button).
- Row 1: Référence de Stage is 'hl10', Numéro du Candidat is '2', Status is 'Refusé(e)' (red button), Infos Candidat (blue button), Infos Stage (green button), and Decision (black button).

© 2024 All Rights Reserved - ESPS

Figure 4.8.14 – Page Demandes

The screenshot shows the API documentation for the `/api/demande/all` endpoint. It includes sections for Parameters, Responses, and Curl examples.

**GET /api/demande/all**

**Parameters**

No parameters

**Responses**

**Curl**

```
curl -X 'GET' \
  'http://localhost:5135/api/demande/all' \
  -H 'accept: */*'
```

**Request URL**

```
http://localhost:5135/api/demande/all
```

Figure 4.8.15 – Api Demandes

Curl

```
curl -X 'GET' \
'http://localhost:5135/api/demande/all' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5135/api/demande/all
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "id": 1,     "userId": 2,     "reference": "hl10",     "status": 2   },   {     "id": 2,     "userId": 2002,     "reference": "HL-02",     "status": 0   } ]</pre> <p><a href="#">Copy</a> <a href="#">Download</a></p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon,11 Mar 2024 22:28:40 GMT server: Kestrel transfer-encoding: chunked</pre>

Figure 4.8.16 – Résultat Api Demandes

Bienvenue , a Se Déconnecter

Accueil Mon Profile Mes Demandes Nous

Admin Accueil Tous Les Candidateurs Ajouter Stage

Infos Candidat Numéro : 2

Nom & Prénom : **achraf**

Ville : **Sfax**

Status : **Employé(e)**

Telephone : **2015548**

Disponibilité : **2024-02-27T00:00:00**

Domaine : **Informatique**

**Linkedin :** [hivjhvjh](#)

Cv : [Voir](#)

Lettre : [Voir](#)

Tous les informations de Candidature

OK

Figure 4.8.17 – Candidat Infos

The screenshot shows a web application interface for managing job stages. At the top, there's a navigation bar with links: Accueil, Mon Profile, Mes Demandes, Nous-Contactez, Bienvenue, Se Déconnecter, Admin, Accueil, Tous Les Candidateurs, and Ajouter Stage.

A modal window is open, titled "Stage Référence : HL-02". It displays the following details:

- Société :** AZIIN Engineering Solutions
- Titre de Sujet :** Application spécifique de don pour les associations
- Description :** L'utilisateur après inscription et création de profil, peut publier directement sur la plateforme l'image et l'état de leurs vieux objets . Les associations inscrites sur la plateforme, peuvent obtenir automatiquement les différents produits selon leurs besoins saisis préalablement..
- Tous les informations de Stage HL-02**
- OK** button at the bottom right of the modal.

Figure 4.8.18 – Stage Infos

The screenshot shows the same web application interface as the previous one, with the same navigation bar and Admin menu.

A modal window is open, titled "Demande Numéro : 2". It displays the following information:

- Candidat Numéro 2002 Pour Stage Référence : HL-02
- Buttons: En Cours de Traitement (yellow), Retenu(e) (green), Refusé(e) (red)
- Text: Confirmer Votre Choix ci-dessous :
- Buttons: NON (red) and VALIDER (blue) at the bottom right.

Figure 4.8.18 – Page Décision

PUT /api/demande/{id}/update

Parameters

Name Description

**id** \* required  
integer(\$int32) id  
(path)

Request body

application/json

Example Value | Schema

```
{  
    "status": 0  
}
```

Figure 4.8.19 – Api Décision demande

Responses

Curl

```
curl -X 'PUT' \  
  'http://localhost:5135/api/demande/2/update' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "status": 1  
}'
```

Request URL

<http://localhost:5135/api/demande/2/update>

Server response

Code Details

200 Response body

```
{  
    "message": "Mise à jour réussie",  
    "demande": {  
        "id": 2,  
        "userId": 2002,  
        "reference": "HL-02",  
        "status": 1  
    }  
}
```

Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 11 Mar 2024 22:32:13 GMT  
server: Kestrel  
transfer-encoding: chunked
```

Figure 4.8.20 – Résultat Api Décision demande



Accueil Mon Profile Mes Demandes Nous-Contactez

Bienvenue , a Se Déconnecter

Admin

Accueil Tous Les Candidateurs Ajouter Stage List des Stages Demandes

#	Référence de Stage	Numéro du Candidat	Status	Infos Candidat	Infos Stage	Decision
2	HL-02	2002	Retenu(e)	Infos Candidat	Infos Stage	Decision
1	hl10	2	Refusé(e)	Infos Candidat	Infos Stage	Decision

© 2024 All Rights Reserved - ESPS

Figure 4.8.21 – Résultat Décision demande



Accueil Mon Profile Mes Demandes Nous-Contactez

Bienvenue , anis Se Déconnecter

### Mes Demandes

Demande Numéro	Référence du Stage	Détails du Stage	Status
2	HL-02	Détails du Stage	Retenu(e)

© 2024 All Rights Reserved - ESPS

Figure 4.8.22 – Résultat Décision demande Interface Candidat

## 4.5 –Conclusion

Ce chapitre a présenté les différentes interfaces réalisées dans notre application.

## **Conclusion et perspectives**

Le projet informatique dédié à la gestion des candidatures de stages a considérablement amélioré l'efficacité du processus de recrutement, renforçant ainsi la compétitivité de l'entreprise grâce à une organisation plus rigoureuse et à une collaboration optimisée entre les équipes. Les résultats obtenus indiquent un impact positif tant sur la performance globale de l'entreprise que sur la satisfaction des utilisateurs.

Cependant, il est essentiel de percevoir ce projet comme le point de départ d'une amélioration continue. Les retours des utilisateurs et les évolutions des besoins de l'entreprise doivent être pris en compte pour garantir la pérennité et l'adaptabilité de la solution.

En envisageant les perspectives d'évolution, il serait opportun d'intégrer des outils d'intelligence artificielle pour affiner les prévisions et les recommandations. De plus, l'amélioration des interfaces utilisateur devrait être envisagée pour faciliter davantage l'expérience des utilisateurs. En demeurant attentif aux tendances du marché et en adaptant le système en conséquence, l'entreprise pourra continuer à optimiser sa gestion des candidatures de stages et à renforcer sa position concurrentielle à long terme. L'engagement envers l'amélioration continue garantira que la solution évolue en harmonie avec les besoins changeants de l'entreprise, assurant ainsi son efficacité à long terme dans le processus de recrutement des stagiaires.

**CAHIER DE STAGE OBLIGATOIRE**

Année universitaire : 2023/2024

Stage :  Initiation  Perfectionnement  
 Date : Du 11/01/2024 au 03/02/2024

<b><u>L'étudiant</u></b>	
Nom et Prénom :	MKAOUAR ACHRAF
Adresse :	ROUTE EL AIN KM6.5
N° de la carte d'étudiant :	
N° d'inscription :	
Téléphone :	25366610
E-mail :	ACHROUFA.TN@GMAIL.COM
Département :	Informatique
Groupe :	DSI23

<b><u>La structure d'accueil</u></b>	
Etablissement :	Aziin Engineering Solutions
Adresse :	Route Ain km 1.5 Kassas Numéro 5, Immeuble Nour, première étage, appartement numéro 1.
Représenté par :	Hassen Mrabet
Qualité du représentant :	
Téléphone :	+216 20 129 410
E-mail :	

## **Finalités du stage**

Le stage obligatoire a pour finalité de permettre l’application, dans un cadre professionnel, des acquis d’apprentissage du cursus suivi dans les établissements universitaires. Les étudiants doivent réaliser un véritable travail au sein de la structure d’accueil qui leur permet d’acquérir une expérience, de développer leurs aptitudes à mener de manière opérationnelle une activité professionnelle, de renforcer leur employabilité et de faciliter leur insertion professionnelle.

Le stage obligatoire permet de placer l’étudiant en situation d’activité professionnelle et de réalisation d’activités d’un niveau de complexité, d’autonomie et de responsabilité garanties par le diplôme universitaire conformément à la classification nationale des qualifications.

## **Obligations de l’étudiant pendant le stage**

L’étudiant doit respecter le règlement interne et la culture de la structure d’accueil. La ponctualité, le sérieux dans le travail et le respect de la confidentialité sont des qualités demandées à l’étudiant en stage obligatoire. L’étudiant doit tenir à jour, remplir et conserver soigneusement ce cahier indispensable à la validation régulière du stage.

La charte de stage obligatoire prévue à l’article 10 de la loi 2009-21 définit les principes généraux du stage obligatoire et les conditions permettant leur réalisation au sein des structures d’accueil. Elle fixe les modalités garantissant la qualité du stage obligatoire, sa conformité avec les objectifs pédagogiques du cursus de formation suivi par l’étudiant dans l’établissement d’enseignement supérieur et de recherche et son intérêt pour toutes les parties concernées.

## ORGANISATION GENERALE DU STAGE OBLIGATOIRE

### Description du travail demandé

Notre solution vise à simplifier le processus de gestion des candidatures de stages, offrant une interface conviviale pour le suivi et l'évaluation des candidats. Elle favorisera également la collaboration entre les différentes parties prenantes du processus de recrutement de stagiaires, tout en garantissant la sécurité des données sensibles associées aux candidatures. Grâce à cette plateforme, l'entreprise pourra optimiser ses processus de recrutement de stagiaires et prendre des décisions éclairées, tout en respectant les spécificités de son fonctionnement.

### Objectifs recherchés et résultats attendus

**Les objectifs recherchés pour mon stage à la Aziin étaient les suivants :**

- Développer une application de gestion informatisée des Candidatures
- Élaborer une base de données pour la gestion des Candidatures
- Mettre en place un système de Recrutement des stagiaires

**Les résultats attendus étaient les suivants :**

- Une gestion plus efficace et rapide des Candidatures
- Une meilleure organisation et gestion des données relatives aux Candidatures

Ces objectifs et résultats attendus étaient qualitatifs et quantitatifs. Les aspects qualitatifs étaient liés à l'amélioration de la qualité du service rendu aux Candidatures, à une meilleure organisation et gestion des données, et à une réduction des Formes Papiers .

Grâce à cette plateforme, l'entreprise pourra optimiser ses processus de recrutement de stagiaires et prendre des décisions éclairées, tout en respectant les spécificités de son fonctionnement.

### Directives particulières

(Documentation, règlementation particulière)

## Calendrier des activités de l'étudiant

Date	Description de l'activité réalisée	Observations
11/01/2024	Introduction de la société « Aziin»	
11/01/2024	Introduction Au Domaine de Société	
11/01/2024	Initiation sur le programme utilisé « StarUML»	
12/01/2024	Initiation sur la langage utilisé « UML»	
12/01/2024	Identification des acteurs	
12/01/2024	Installation Environnement logiciel	
13/01/2024	Installation Environnement Base de données	
13/01/2024	Installation Framework utilisée	
13/01/2024	Installation Bibliothèque utilisé	
14/01/2024	Initiation sur Langage de modélisation	
14/01/2024	Modélisation du projet	
14/01/2024	Faire Le Diagramme de cas d'utilisation	
15/01/2024	Faire Le Diagramme de classe	
16/01/2024	Faire Page d'accueil Avec ReactJs	
17/01/2024	Faire Page de S'inscrire Avec ReactJs	
17/01/2024	Initiation Sur MVC API .NET Core	
18/01/2024	Faire Api Pour La Page S'inscrire Avec .Net Core	
18/01/2024	Faire La Page Page Connectez-vous Avec ReactJS	
18/01/2024	Initiation sur La fonctionnalités JWT	
18/01/2024	Initiation sur Bibliothèque Bcrypt	
19/01/2024	Faire Api Pour La Page Connectez-vous Avec .Net Core	
20/01/2024	Faire La Page Préinscription Avec ReactJS	
20/01/2024	Faire Api Pour La Page Préinscription Avec .Net Core	
Date	Description de l'activité réalisée	Observations
21/01/2024	Faire La Page Candidat Accueil Avec ReactJS	

21/01/2024	Faire Api Pour La Page Candidat Accueil Avec .Net Core	
22/01/2024	Faire La Page Candidat Profile Avec ReactJS	
23/01/2024	Faire Api Pour La Page Candidat Profile Avec .Net Core	
24/01/2024	Faire La Page Candidat Demandes Avec ReactJS	
24/01/2024	Faire Api Pour La Page Demandes Avec .Net Core	
25/01/2024	Faire La Page Candidat Postuler Avec ReactJS	
26/01/2024	Faire Api Pour La Page Postuler Avec .Net Core	
27/01/2024	Faire La Page Admin Accueil Avec ReactJS	
28/01/2024	Faire La Page List Candidatures Avec ReactJS	
29/01/2024	Faire Api Pour La Page List Candidatures Avec .Net Core	
30/01/2024	Faire La Page Ajouter Stage Avec ReactJS	
30/01/2024	Faire Api Pour La Page Ajouter Stage Avec .Net Core	
31/01/2024	Faire La Page List Stages Avec ReactJS	
31/01/2024	Faire Api Pour La Page List Stages Avec .Net Core	
01/02/2024	Faire La Page Demandes Avec ReactJS	
02/02/2024	Faire Api Pour La Page Demandes Avec .Net Core	
02/02/2024	Tester Les différentes fonctionnalités du projet	
03/02/2024	Clôture du stage: Bilan et perspectives	
04/02/2024	Rédaction du rapport de stage et préparation de la présentation finale	