

RÉPUBLIQUE ISLAMIQUE DE MAURITANIE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DE NOUAKCHOTT

FACULTÉ DES SCIENCES ET TECHNIQUES

DÉPARTEMENT D'INFORMATIQUE



كلية العلوم والتقنيات  
Faculté des Sciences et Techniques

## MÉMOIRE DE FIN D'ÉTUDES

En vue de l'obtention de la Licence en :

Méthodes Informatiques Appliquées à la Gestion des Entreprises (MIAGE)

Sujet :

Conception et développement d'une application web de mise  
en relation entre clients et freelancers



Encadré par :

Présenté par :

Moussa Mohamed Erebih

Dr. Cheikh Abdelkader Ahmed  
Telmoud

Ing. Salem Issa

Date de soutenance : Lundi 30 juin 2025

Membres du jury :

Président : Dr. Marieme Sid'ahmed

Encadrant Pédagogique : Dr. Cheikh Abdelkader Ahmed Telmoud

Membres : Dr. Mohamed El Galy Beheitt

Ing. Ahmed Mohamed Lemine Mamoune

Année universitaire :

2024-2025

# Dédicaces

Je tiens à dédier ce projet avant tout à mes parents, qui ont toujours été là pour moi, avec leur amour, leur soutien et leurs innombrables sacrifices. Sans eux, rien de tout ça n'aurait été possible.

Un grand merci aussi à toute ma famille, qui m'a soutenu d'une manière ou d'une autre, que ce soit par des mots d'encouragement, une présence rassurante ou simplement en croyant en moi.

Et bien sûr, je n'oublie pas toutes les personnes qui, de près ou de loin, ont joué un rôle dans mon parcours, que ce soit dans mes études ou dans ma vie en général. Votre aide, vos conseils et même vos petites attentions ont compté plus que vous ne l'imaginez. Merci à vous tous !

# Remerciements

Nous tenons à remercier tous ceux qui ont contribué à ce travail et qui nous ont aidés dans sa réalisation, en particulier :

- Avant tous, nous remercions ALLAH Tout-Puissant de nous avoir donné le courage et la force de faire ce travail et de nous avoir donné la volonté et la patience de continuer notre chemin éducatif malgré toutes les difficultés.
- Nous tenons à exprimer nos sincères remerciements à notre encadrant Cheikh Abdelkader Ahmed Telmoud.
- Et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et répondu à nos questions au cours de notre recherche.

Nous remercions tous nos professeurs qui nous ont accordé leur attention durant les années de nos études.

# Table des matières

Dédicaces	i
Remerciements	ii
Abréviations	vii
Résumé	viii
Introduction Générale	1
<b>1 Chapitre I : Présentation Générale</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Cadre Général du stage . . . . .	3
1.3 Présentation d'organisme d'accueil . . . . .	3
1.4 Présentation du sujet . . . . .	4
1.5 Choix de la méthodologie . . . . .	4
1.5.1 Présentation de l'UML . . . . .	4
1.5.2 Modèle en V . . . . .	5
1.6 Conclusion . . . . .	8
<b>2 Chapitre II : Étude Préalable</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Étude de L'existant . . . . .	9
2.2.1 La Description de L'existant . . . . .	9
2.2.2 La Critique de L'existant . . . . .	10
2.2.3 La Solution Proposée . . . . .	11
2.2.4 L'intérêt de l'Application . . . . .	11
2.3 Conclusion . . . . .	12
<b>3 Chapitre III : Spécification des besoins</b>	<b>13</b>
3.1 Introduction . . . . .	13
3.2 Etude des besoins . . . . .	13
3.2.1 Besoins fonctionnels . . . . .	13
3.2.2 Besoins non fonctionnels . . . . .	14
3.3 Identification des acteurs . . . . .	15

3.3.1	Client . . . . .	15
3.3.2	Freelancer . . . . .	15
3.3.3	Administrateur . . . . .	16
3.4	Conclusion . . . . .	16
<b>4</b>	<b>Chapitre IV : Conception</b>	<b>17</b>
4.1	Introduction . . . . .	17
4.2	Diagramme de cas d'utilisation . . . . .	18
4.3	Diagrammes de Séquence . . . . .	19
4.3.1	Diagramme de Séquence pour l'Authentification . . . . .	19
4.3.2	Diagramme de séquence pour la publication d'un projet . . . . .	20
4.3.3	Diagramme de séquence pour la candidature à un projet . . . . .	21
4.3.4	Diagramme de séquence pour la sélection d'un freelancer . . . . .	22
4.4	Diagramme de classe . . . . .	22
4.5	Conclusion . . . . .	23
<b>5</b>	<b>Chapitre V : Réalisation</b>	<b>24</b>
5.1	Introduction . . . . .	24
5.2	Environnement de développement . . . . .	24
5.2.1	Environnement matériel . . . . .	24
5.2.2	Environnement logiciel . . . . .	25
5.3	Interfaces graphiques . . . . .	31
<b>6</b>	<b>Conclusion Générale</b>	<b>55</b>
	<b>Webographie</b>	<b>56</b>

# Table des figures

1.1	Diagrammes UML . . . . .	5
1.2	Modèle en V . . . . .	6
2.1	Interface MostaqL . . . . .	10
4.1	Diagramme de cas d'utilisation . . . . .	18
4.2	Diagramme de Séquence pour l'Authentification . . . . .	19
4.3	Diagramme de séquence pour la publication d'un projet . . . . .	20
4.4	Diagramme de séquence pour la candidature à un projet . . . . .	21
4.5	Diagramme de séquence pour la sélection d'un freelancer . . . . .	22
4.6	Diagramme de classe . . . . .	22
5.1	Lucidchart . . . . .	25
5.2	Logo django + DRF . . . . .	25
5.3	Logo PostgreSQL . . . . .	26
5.4	Logo React . . . . .	27
5.5	Logo MUI . . . . .	27
5.6	Logo React Router . . . . .	28
5.7	Logo JWT . . . . .	28
5.8	Logo Git et Github . . . . .	29
5.9	Logo vs code . . . . .	29
5.10	Logo Postman . . . . .	30
5.11	Logo Swagger . . . . .	30
5.12	Page d'accueil . . . . .	31
5.13	Erreurs de saisie dans Information de base (Client) . . . . .	32
5.14	Information de base (Client) . . . . .	33
5.15	Informations complémentaires (Client) . . . . .	34
5.16	Inscription réussie! . . . . .	34
5.17	Email de vérification . . . . .	35
5.18	Renvoyer l'email de vérification . . . . .	35
5.19	Un nouveau email de vérification . . . . .	35
5.20	Email vérifié avec succès . . . . .	36
5.21	Erreurs de saisie dans Informations complémentaires (Freelancer) . . . . .	37
5.22	Confirmation et envoi des informations . . . . .	38
5.23	Connexion à votre compte . . . . .	38

5.24 Réinitialisation du mot de passe . . . . .	39
5.25 Email de réinitialisation envoyé . . . . .	39
5.26 Réinitialisation de votre mot de passe . . . . .	40
5.27 Nouveau mot de passe . . . . .	40
5.28 Réinitialisation réussie . . . . .	41
5.29 DASHBOARD CLIENT . . . . .	41
5.30 Interface “Mon Profil”(client) . . . . .	42
5.31 Modifier le profil (client) . . . . .	42
5.32 Interface “Publier un Projet” . . . . .	43
5.33 Exemple d’Interface “Publier un Projet” . . . . .	43
5.34 Notification de succès après la publication du projet . . . . .	44
5.35 Interface “Consulter Mes Projets” . . . . .	44
5.36 Accusé de réception des candidatures . . . . .	45
5.37 Interface “Gérer les candidatures des freelancers” . . . . .	45
5.38 Postulation refusée avec succès . . . . .	46
5.39 Notification de refus de sa candidature . . . . .	46
5.40 Postulation acceptée avec succès . . . . .	47
5.41 Notification d’acceptation de sa candidature . . . . .	47
5.42 Email(freelancer) . . . . .	48
5.43 Email(client) . . . . .	48
5.44 Évaluer le freelancer . . . . .	49
5.45 DASHBOARD FREELANCER . . . . .	49
5.46 Interface “Mon Profil”(freelancer) . . . . .	50
5.47 Modifier le profil (freelancer) . . . . .	51
5.48 Interface “Trouver des projets” . . . . .	52
5.49 Interface “Postulation à un projet” . . . . .	52
5.50 DASHBOARD ADMINISTRATEUR . . . . .	53
5.51 Interface “Mon Profil”(admin) . . . . .	53
5.52 Modifier le profil (admin) . . . . .	54
5.53 Interface de gestion des utilisateurs . . . . .	54

# Abréviations

- DRF : Django REST Framework
- MUI : Material-UI
- MIAAGE : Méthode Informatique Appliquée à la Gestion des Entreprises
- UML : Unified Modeling Language
- RAM : Random-Access Memory
- FST : Faculté des Sciences et Techniques
- CPU : Central Processing Unit
- MOE : Maître d'œuvre (prestataire)
- MOA : Maître d'ouvrage (client)



# Résumé

Ce rapport a été rédigé dans le cadre de notre stage de fin d'études chez Syskat Technologies, une entreprise mauritanienne bien établie depuis 2007, spécialisée dans les systèmes intégrés et les télécommunications. On a eu la chance d'y travailler avec une équipe super motivée, ce qui nous a bien aidés à avancer sur notre projet.

Le projet en question ? Créer une plateforme web pour mettre en relation des clients et des freelancers. Le but est simple : permettre aux gens de poster leurs besoins, aux freelancers de postuler, et aux clients de laisser des retours une fois le travail terminé. L'idée, c'est de rendre les échanges professionnels plus simples, efficaces et sécurisés.

Niveau technologique, on a misé sur du solide et moderne : on a utilisé Visual Studio Code pour coder, Django + DRF pour gérer le backend, et côté interface, on s'est appuyés sur React avec MUI. Cette combinaison nous a permis d'avoir une application web réactive, bien structurée et facile à maintenir.

# Introduction Générale

Durant ma L3 MIAE à la Faculté des Sciences et Techniques, j'ai eu la chance d'intégrer Syskat Technologies, une entreprise mauritanienne spécialisée dans les systèmes intégrés depuis 2007. Passionné par le domaine du web et l'innovation numérique, j'ai saisi cette opportunité pour réaliser mon projet de fin d'études dans un environnement où l'expertise informatique et télécommunication se conjugue avec la créativité et le dynamisme d'une équipe jeune et professionnelle.

Au sein de Syskat, j'ai travaillé sur la conception et le développement d'une application web destinée à faciliter la mise en relation entre clients et freelancers. Ce projet vise à proposer une plateforme intuitive et sécurisée où les clients peuvent publier des projets (avec description, budget, compétences requises, etc.) et sélectionner le freelancer le mieux adapté, tandis que les freelancers peuvent postuler en présentant leurs compétences et expériences. L'objectif de ce projet est double : répondre aux attentes des entreprises et particuliers en matière de solutions informatiques innovantes, tout en valorisant l'expertise des professionnels indépendants.

Dans le cadre de ce projet, j'ai eu l'occasion d'utiliser un ensemble de technologies modernes telles que React.js pour le développement du front-end, Django et DRF pour le back-end, ainsi que Postgresql pour la gestion de la base de données. Ces outils ont été choisis pour leur robustesse, leur flexibilité et leur capacité à assurer une scalabilité optimale de l'application.

Ce rapport est organisé de la manière suivante :

- Le premier chapitre présente le contexte de Syskat ainsi que l'environnement professionnel dans lequel s'inscrit ce projet, en mettant en avant les défis technologiques et organisationnels de l'entreprise.
- Le deuxième chapitre expose la problématique, la méthodologie adoptée et les différentes solutions envisagées pour répondre aux besoins identifiés.
- Le troisième chapitre détaille les exigences fonctionnelles et techniques du projet, en décrivant précisément les rôles des acteurs (Clients, freelancers, Administrateurs).
- Le quatrième chapitre se concentre sur la conception détaillée et la modélisation du système à l'aide du langage UML, en présentant notamment les architectures front-end et back-end.

- Enfin, Le dernier chapitre va présenter la phase de réalisation, l’environnement de travail, les applications et les langages de développements utilisés, ainsi que quelques interfaces de notre application.

Ce rapport retrace l’ensemble du processus de développement, depuis l’analyse des besoins jusqu’à la réalisation finale, en passant par la conception et la mise en œuvre, reflétant ainsi la contribution de Syskat dans l’innovation et l’informatisation des services.

# Chapitre I : Présentation Générale

## 1.1 Introduction

La présentation générale constitue une étape essentielle pour poser les fondements de ce rapport. Dans ce chapitre, nous commençons par un bref aperçu du contexte dans lequel s'inscrit notre projet. Ensuite, nous décrivons le cadre du stage, en présentant notamment l'organisme d'accueil, avant d'aborder plus précisément le sujet traité. Pour finir, nous expliquons les raisons qui ont guidé notre choix méthodologique.

## 1.2 Cadre Général du stage

Dans le cadre de ma Licence MIAGE, j'ai réalisé un stage de trois mois (du 07/03 au 01/06/2025) au sein de l'entreprise Syskat Technologies. Mon travail a principalement porté sur la conception et le développement d'une application web destinée à faciliter la mise en relation entre clients et freelancers.

## 1.3 Présentation d'organisme d'accueil

Spécialisée dans le domaine informatique et de la télécommunication, Syskat technologies est devenu le leader du développement des systèmes intégrés en Mauritanie depuis sa création en 2007.

Notamment grâce à notre équipe composée de jeunes cadres dynamiques et professionnels à votre écoute pour satisfaire tous vos besoins, nous vous proposons des solutions informatiques innovantes dans les domaines suivants :

- Conception, Développement et Maintenance des Systèmes d'Information Intégrés ;
- Hébergement site web (Sites, Modules, Design, Hébergement, ...etc.) ;
- Evaluation et audit des progiciels ;
- Formation (Standard et sur mesure, Élaboration des Plans de Formation, ...etc.) ;
- Accompagnement des organisations dans le processus d'informatisation ;
- Installation, Administration et Maintenance des Systèmes et Réseaux TIC ;

- Études et Conseils en Informatique et Télécoms ;
- Conception des outils d'enquêtes ;
- Réalisation, interprétation et exploitation des Statistiques ;
- Prestation de Services à Valeur Ajoutée (SMS, MMS, IVR, VOIP, ...etc.) ;
- Caméras de Surveillance et Contrôle d'accès ;
- Géolocalisation et suivi de véhicules (SVTS).[1]

## 1.4 Présentation du sujet

Aujourd'hui, dans un paysage économique qui valorise de plus en plus la flexibilité et la réactivité, le travail freelance s'impose comme une solution privilégiée, aussi bien pour les entreprises que pour les indépendants. Pour Syskat, l'enjeu est clair : il faut revoir ses méthodes pour mieux identifier les talents et simplifier les échanges entre clients et freelancers. Avec une concurrence toujours plus forte, le marché demande des outils à la fois efficaces et fiables pour recruter des profils qualifiés et piloter leurs missions sans perdre de temps.

Pour la réalisation de ce projet, nous prévoyons d'utiliser des technologies web modernes telles que Django + DRF pour gérer le backend, et côté interface, on s'est appuyés sur React avec MUI. Cette combinaison nous a permis d'avoir une application web réactive, bien structurée et facile à maintenir.

## 1.5 Choix de la méthodologie

Pour ce projet, notre méthodologie repose sur deux éléments clés : UML (Langage de Modélisation Unifié) pour la modélisation et le modèle en V pour le développement. Les sections suivantes détailleront leur mise en œuvre.

### 1.5.1 Présentation de l'UML

L'UML (Unified Modeling Language) est un outil de modélisation graphique qui permet de représenter visuellement l'architecture d'un système. On l'utilise surtout dans le domaine du génie logiciel et de la conception orientée objet. En gros, il repose sur différents types de diagrammes qui aident à formaliser la structure et le comportement d'un logiciel.

#### Types de diagrammes UML :

Comme le montre l'illustration ci-dessous, UML se distingue par la richesse et la diversité de ses diagrammes :

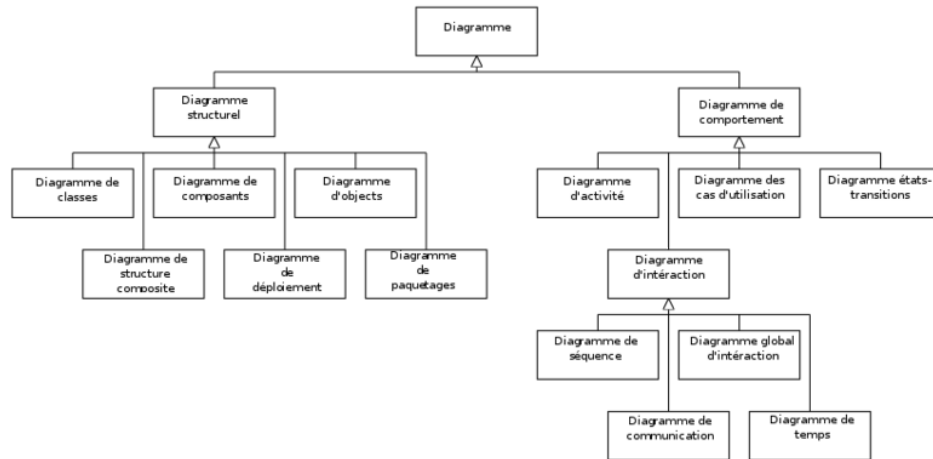


FIGURE 1.1 – Diagrammes UML

Dans ce projet, j'ai choisi d'utiliser uniquement les diagrammes UML essentiels afin de mettre en avant de manière claire et concise la logique fonctionnelle et les interactions clés entre clients, freelancers et administrateurs. Ce choix de modélisation minimaliste permet d'éviter une surcharge d'informations et facilite la compréhension des flux métiers et des échanges au sein du système. En se concentrant uniquement sur les diagrammes nécessaires, l'analyse reste ciblée sur les processus déterminants pour la réussite de l'application, garantissant ainsi que l'ensemble des besoins fonctionnels est correctement adressé et validé.

**Les diagrammes UML retenus pour ce projet sont les suivants :**

- **Diagramme de cas d'utilisation** : Le diagramme de cas d'utilisation sert à modéliser une fonctionnalité précise au sein d'un système. Son but est de visualiser les liens entre différentes fonctionnalités, tout en mettant en évidence les acteurs (internes et externes) qui interagissent avec elles.
- **Diagramme de séquence** : Les diagrammes de séquence, c'est un outil qui permet de visualiser les échanges entre différents objets, en montrant l'enchaînement des interactions. En gros, ils servent à représenter un scénario précis, en illustrant l'ordre dans lequel les objets communiquent entre eux.
- **Diagramme des classes** : Parmi les diagrammes UML, celui qui revient le plus souvent et qui sert de base à toute approche orientée objet, c'est le diagramme de classe. En gros, il représente les différentes classes d'un système, avec leurs attributs, leurs méthodes, et les liens entre elles.

### 1.5.2 Modèle en V

Le cycle en V, utilisé en gestion de projet, s'inspire du modèle en cascade, une approche développée dans les années 1970. Ce modèle permet d'organiser les processus de développement de façon séquentielle, étape par étape.

Principalement appliqué dans le domaine des logiciels, le cycle en V est une méthode linéaire où chaque phase de conception est associée à une phase de test correspondante. D'ailleurs, son nom vient justement de son double objectif : vérification (côté conception) et validation (côté tests).

Si on le représente visuellement, les étapes de développement et de test s'alignent de manière symétrique, formant une sorte de « V » – d'où son nom !

### Comment fonctionne le cycle en V ?

Le cycle en V est une méthode de développement logiciel structurée en deux phases complémentaires : d'une part, une série de phases de conception, et d'autre part, une série de phases de validation.

Pour résumer, tout commence par l'analyse des besoins : on définit précisément ce que le client attend. Ensuite, on passe à la conception du système, puis à l'architecture logicielle (histoire de savoir comment les différents modules vont communiquer entre eux), et enfin à la conception détaillée des composants. Tout ça, c'est la branche gauche du V. Une fois que le code est écrit, on bascule vers la partie droite, dédiée aux tests.

Et là, on enchaîne avec :

- Les tests unitaires (pour vérifier chaque brique du logiciel),
- Les tests d'intégration (pour s'assurer que tout fonctionne bien ensemble),
- Les tests système (histoire de valider les performances globales),
- Et enfin, les tests d'acceptation (où on confirme que le logiciel répond vraiment aux besoins des utilisateurs).

Ce modèle linéaire permet ainsi de suivre une approche structurée où chaque phase de conception est directement associée à une phase de validation correspondante, garantissant une qualité et une conformité améliorées du produit final.

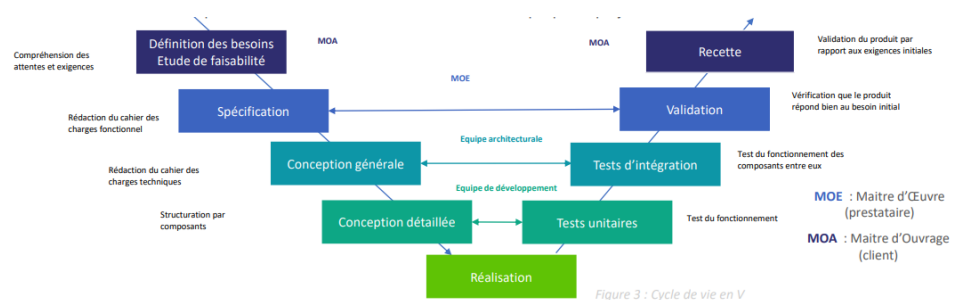


FIGURE 1.2 – Modèle en V

### Définition des besoins / Étude de faisabilité

- **Objectif** : Comprendre les attentes et les exigences du client.
- **Livrable** : Cahier des charges fonctionnel.
- **Acteurs** : MOA (Maître d'Ouvrage, c'est-à-dire le client).
- **But** : Vérifier si le projet est faisable techniquement et économiquement.

## Spécification

- **Objectif** : Formaliser les besoins du client sous forme de spécifications fonctionnelles.
- **Livrable** : Cahier des charges technique.
- **But** : Décrire précisément ce que doit faire le système sans encore dire comment.

## Conception générale

- **Objectif** : Traduire les besoins fonctionnels en une architecture système globale.
- **Acteurs** : Équipe architecturale.
- **But** : Structurer le système en composants majeurs et définir les interactions.

## Conception détaillée

- **Objectif** : Définir la conception technique précise de chaque composant.
- **Acteurs** : Équipe de développement.
- **But** : Préparer le travail de codage en détaillant les interfaces, classes, algorithmes, etc.

## Réalisation

- **Objectif** : Coder et implémenter le système selon la conception détaillée.
- **But** : Créer un logiciel fonctionnel prêt à être testé.

## Tests unitaires

- **Objectif** : Vérifier individuellement chaque composant développé.
- **But** : Détecter et corriger les bugs au niveau des modules.

## Tests d'intégration

- **Objectif** : Vérifier le bon fonctionnement des interactions entre composants.
- **But** : S'assurer que les modules intégrés fonctionnent correctement ensemble.

## Validation

- **Objectif** : Vérifier que le logiciel correspond aux besoins définis initialement.
- **But** : Contrôler la qualité du produit avant la livraison.

## Recette

- **Objectif** : Valider le produit final par rapport aux exigences du client.
- **But** : Autoriser la mise en production après validation complète.



### **Avantages du cycle en V :**

- C'est un modèle assez simple à appréhender et à appliquer. Comme chaque étape doit être finalisée avant de passer à la suivante, ça permet de mieux suivre l'avancement et d'atteindre les objectifs sans trop de surprises.
- La gestion est facilitée grâce à sa structure linéaire. On sait exactement où on en est, d'où on vient et quelle est la prochaine étape à aborder.
- C'est aussi un bon choix pour les projets avec des délais serrés, car il impose un cadre rigoureux et des jalons bien définis.

### **Inconvénients du cycle en V :**

- Un des gros points faibles, c'est qu'il ne permet pas vraiment de travailler en parallèle sur différentes phases.
- Autre problème : si une erreur est détectée dans une phase, ça peut obliger à revoir tout ce qui a été fait après... ce qui peut faire perdre pas mal de temps.

Pour ce projet de PFE, le modèle en V s'impose comme une solution pertinente, et voici pourquoi :

D'abord, il assure une traçabilité solide entre les besoins initiaux et les phases de test, ce qui est crucial pour la cohérence du projet.

Ensuite, sa documentation détaillée répond parfaitement aux attentes académiques – rien n'est laissé au hasard, et tout est formalisé.

Autre point fort : sa structure rigoureuse facilite une gestion qualité systématique et limite les risques d'erreurs, un vrai plus pour un projet technique.

Bref, ce choix méthodologique montre une démarche organisée et sérieuse, alignée à la fois sur les standards académiques et les contraintes d'un développement complexe.

## **1.6 Conclusion**

Pour conclure, ce premier chapitre pose les bases de notre rapport en décrivant le contexte général du stage, notre sujet d'étude ainsi que l'organisme qui nous a accueillis. Nous avons également expliqué les raisons derrière notre choix méthodologique. Ces éléments préparatoires nous permettront d'aborder, dans les prochains chapitres, des analyses plus approfondies.

# Chapitre II : Étude Préalable

## 2.1 Introduction

Pour concevoir et développer une application web qui met en relation clients et freelancers, il est essentiel de commencer par une étude approfondie du contexte actuel. Cette analyse préliminaire permet d'identifier les besoins et de proposer des solutions adaptées. On peut la décomposer en plusieurs parties : l'étude de l'existant, la description de l'existant, la critique de l'existant, la solution proposée et l'intérêt de l'application.

## 2.2 Étude de L'existant

### 2.2.1 La Description de L'existant

Actuellement, la façon dont les clients et les freelancers se connectent reste assez désorganisée. Les clients partagent leurs besoins un peu partout : sur les réseaux sociaux, dans des forums, par e-mail, voire même au téléphone. Certains se débrouillent avec des outils basiques, comme des tableurs Excel, pour essayer de garder une trace de leurs recherches.

Côté freelancers, c'est pareil : ils doivent surveiller plusieurs plateformes, compter sur leur réseau ou faire des recherches manuelles pour dénicher des opportunités. Bref, rien de très optimisé.

#### Comment ça se passe concrètement ?

- **Publication des projets**

Au lieu d'utiliser une plateforme unique, les clients postent leurs demandes un peu partout. Un coup sur LinkedIn, un autre dans un groupe Facebook spécialisé, parfois même par e-mail à des contacts directs. Résultat : les annonces sont dispersées, et les freelancers peuvent facilement passer à côté.

- **La chasse aux freelancers**

Sans outil centralisé, trouver le bon freelancer relève souvent du parcours du combattant. Les clients fouillent Google, demandent des recommandations à leurs proches

ou parcourent plusieurs sites en parallèle. Un vrai travail de fourmi, qui prend un temps fou.

#### — Retour d'expérience : le grand flou

Après la mission, les feedbacks restent souvent informels. Pas de système standardisé pour évaluer les freelancers, ce qui rend difficile la construction d'une réputation solide et comparable.

Des solutions émergent pour simplifier tout ça.

Heureusement, certaines plateformes commencent à répondre à ces problèmes. Par exemple, Mostaql propose une application web qui centralise toute la démarche : publication de projets, recherche de talents, gestion des contrats et même évaluation des prestations. L'avantage ? Un gain de temps énorme, plus de transparence, et surtout, des critères objectifs pour juger de la qualité du travail.

Bref, l'idée est de passer d'un système artisanal à quelque chose de bien plus professionnel.

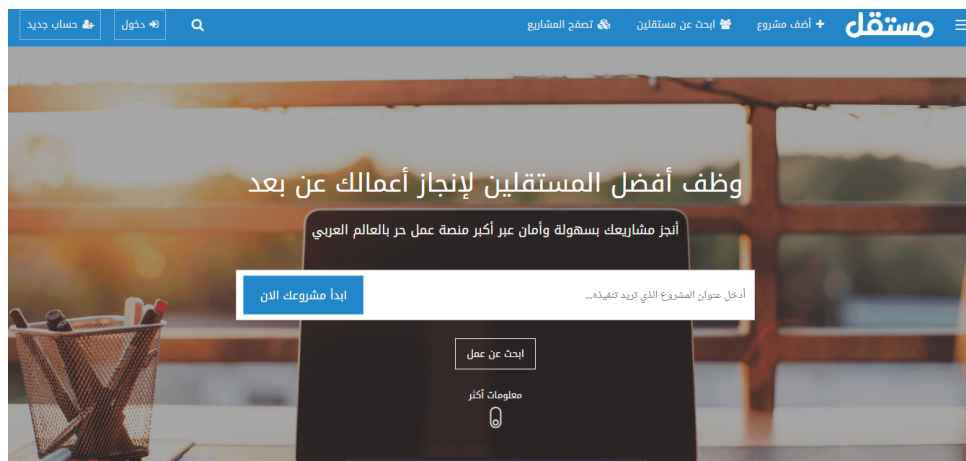


FIGURE 2.1 – Interface Mostaql

## 2.2.2 La Critique de L'existant

**Mostaql est une plateforme performante et bien conçue, mais elle n'est pas sans défauts.** Malgré ses nombreux atouts, certains aspects peuvent poser problème, selon le profil de l'utilisateur.

#### — Une interface pas toujours intuitive

Certes, la plateforme propose énormément d'options, mais c'est aussi ce qui la rend difficile à prendre en main. Les nouveaux arrivants, ou ceux qui cherchent une solution simple, peuvent se sentir perdus. Résultat : il faut du temps pour s'y familiariser, parfois même une formation.

#### — Le prix, un vrai sujet

Les tarifs, surtout pour les fonctionnalités premium, ne sont pas accessibles à tous. Pour un freelancer qui débute ou une petite structure, l'abonnement peut représenter un budget trop élevé, alors que leurs besoins sont souvent basiques.

- **Un temps d'adaptation long**

Avec autant de possibilités, il faut forcément un moment pour tout maîtriser. Or, tout le monde n'a pas le temps ou l'envie de se former sur un outil avant de l'utiliser efficacement.

- **Pas toujours adapté aux besoins simples**

Pour ceux qui veulent juste une plateforme de mise en relation classique, Mostaqi peut sembler surchargé. Certaines fonctionnalités, aussi utiles soient-elles, ne servent à rien si on n'en a pas l'usage.

## 2.2.3 La Solution Proposée

Notre application, qui se positionne comme une version simplifiée et épurée de ce modèle, se propose de résoudre ces limitations en offrant :

- Une interface plus simple et intuitive, idéale pour les utilisateurs recherchant une prise en main rapide.
- Une solution économique sans les coûts liés aux fonctionnalités avancées inutilisées par un public ciblé.
- Une personnalisation facilitée, permettant d'adapter les outils à des besoins spécifiques sans surcharge fonctionnelle.

Cette approche permet ainsi de répondre aux attentes d'un segment de marché souhaitant bénéficier des avantages d'une plateforme de mise en relation tout en évitant la complexité et les coûts induits par une solution entièrement exhaustive.

## 2.2.4 L'intérêt de l'Application

L'application développée offre un ensemble d'avantages significatifs pour Syskat :

- **Amélioration de la Mise en Relation Professionnelle**

Grâce à l'automatisation des processus de recrutement, l'application simplifie la recherche et la sélection de freelancers qualifiés. Cela permet à Syskat de gagner un temps précieux tout en renforçant son efficacité opérationnelle. En facilitant l'accès à des profils compétents, elle contribue à une meilleure productivité globale.

- **Innovation et Adaptation aux Nouvelles Tendances**

Le développement de cette solution numérique démontre la capacité de Syskat à innover et à suivre les évolutions du marché du travail. En se positionnant comme un acteur technologique moderne, l'entreprise répond aux besoins changeants des professionnels, consolidant ainsi sa crédibilité dans le secteur.

- **Expérience Utilisateur Optimisée**

Avec une interface intuitive et des fonctionnalités pensées pour les utilisateurs (gestion de projets, candidatures en ligne, feedback post-mission), l'application améliore significativement la satisfaction des clients et des freelancers. Cette approche favorise la fidélisation et renforce la réputation de Syskat.

## 2.3 Conclusion

L'étude préalable a permis de mettre en lumière les limitations du système existant. En développant une application web de mise en relation entre clients et freelancers, nous visons à améliorer l'expérience utilisateur à travers une interface simplifiée et intuitive, et à mieux répondre aux besoins simples en intégrant un mode rapide de publication et de candidature.

Cette solution, développée dans le cadre de notre stage au sein de Syskat et reposant sur des technologies modernes et éprouvées, constitue un investissement stratégique pour l'entreprise. Elle permet d'optimiser la gestion des projets entre clients et freelancers, tout en contribuant à l'amélioration continue des services numériques proposés.

# Chapitre III : Spécification des besoins

## 3.1 Introduction

La première étape de notre processus de développement consiste à spécifier les besoins du système. Cette phase est cruciale, car elle permet de définir l'étude fonctionnelle et de clarifier ce que le système devra accomplir en termes métier (son comportement attendu).

Dans ce chapitre, nous allons identifier les besoins fonctionnels et non fonctionnels de l'application, mais aussi les différents acteurs impliqués.

## 3.2 Etude des besoins

Dans cette section, on va établir les fondements pour identifier les besoins du système à développer. Afin de bien cerner les attentes des utilisateurs, on abordera à la fois les exigences fonctionnelles et les aspects non fonctionnels.

### 3.2.1 Besoins fonctionnels

En gros, les besoins fonctionnels décrivent ce que l'application doit faire, sans rentrer dans les détails techniques de comment ça sera codé. Concrètement, le système qu'on doit développer doit permettre aux utilisateurs de réaliser les actions suivantes :

- **Créer un compte** : Permettre aux utilisateurs de s'inscrire en renseignant leurs informations personnelles pour accéder à la plateforme en tant que Client ou free-lancer.
- **Gérer le profil** : Offrir aux utilisateurs la possibilité de consulter et modifier leurs informations personnelles et professionnelles pour maintenir un profil à jour.
- **S'authentifier (se connecter / se déconnecter)** : Assurer l'accès sécurisé au compte via l'identification (email/mot de passe).
- **Publier un projet** : Permettre aux Clients de créer et diffuser une offre de projet en spécifiant les besoins, le budget et les compétences requises.

- **Consulter la liste des freelancers** : Offrir aux Clients la possibilité de rechercher et visualiser les profils de freelancers selon différents critères (compétences, évaluations, localisation, etc.).
- **Sélectionner un freelancer** : Faciliter la comparaison des propositions et la décision de choisir le freelance le plus adapté au projet du Client.
- **Évaluer le freelancer** : Permettre aux Clients d'attribuer une note après la réalisation d'un projet afin de garantir la qualité des prestations.
- **Consulter la liste des projets** : Offrir aux freelancers la possibilité de visualiser et filtrer les offres de projets correspondants à leurs compétences.
- **Soumettre une proposition** : Permettre aux freelancers de répondre à une offre de projet en proposant des conditions définies.
- **Finaliser la mission** : Offrir aux clients la possibilité de clôturer la mission une fois exécutée, marquant ainsi la fin du projet et initiant le processus d'évaluation.
- **Notifications** : Informer en temps réel les utilisateurs des mises à jour pertinentes (nouveaux projets, réponses, évaluations, etc.).
- **Administration de l'application** : Permettre à l'administrateur de gérer les comptes utilisateurs, le contenu et de surveiller la plateforme pour garantir sa sécurité et son respect des règles en vigueur.

### 3.2.2 Besoins non fonctionnels

Les besoins non-fonctionnels correspondent aux contraintes qu'il faut intégrer pour proposer une solution adaptée aux attentes des concepteurs d'architectures dynamiques.

Notre application se doit de garantir impérativement ces exigences :

- **Sécurité** : L'application doit garantir la confidentialité des données utilisateurs. Un système d'authentification robuste (identifiant/mot de passe) est nécessaire pour limiter l'accès aux seules personnes autorisées.
- **Performance** : Le temps de réponse doit être optimal, quelle que soit l'action (recherche de profils, envoi de messages). Les délais d'attente doivent rester minimaux pour une expérience fluide.
- **Convivialité** : L'interface doit être intuitive, même pour des utilisateurs non techniques. Une navigation simple et des fonctionnalités bien organisées sont essentielles pour faciliter les interactions entre clients et freelancers.
- **Maintenance** : Le code doit être structuré et documenté pour permettre des mises à jour efficaces. Une base propre et modulaire simplifiera les évolutions futures.
- **Disponibilité** : La plateforme doit être accessible en tout temps, y compris pendant les périodes de forte demande. Une haute disponibilité est cruciale pour maintenir la confiance des utilisateurs.
- **Scalabilité** : Le système doit pouvoir gérer une augmentation du nombre d'utilisateurs et de transactions sans dégradation des performances.
- **Compatibilité** : L'application doit fonctionner sur les navigateurs et appareils courants (mobile, tablette, desktop) pour toucher un public large.

## 3.3 Identification des acteurs

Dans cette section, nous identifions les différents acteurs qui interagiront directement avec le système.

Avant de les présenter, rappelons qu'un acteur représente une entité (utilisateur, système ou dispositif) qui interagit avec l'application étudiée, en jouant un rôle spécifique.

Le fonctionnement global du système repose sur trois types d'acteurs principaux :

### 3.3.1 Client

Les clients sont les utilisateurs à l'origine des projets. Ils utilisent la plateforme pour publier des projets, gérer leurs publications et sélectionner les freelancers les plus appropriés selon les candidatures reçues.

Leurs principales fonctionnalités sont :

- Créer un compte et s'authentifier.
- Publier un projet avec ses détails (titre, description, budget, délai, etc.).
- Consulter la liste de leurs projets publiés.
- Examiner les propositions soumises par les freelancers.
- Sélectionner un freelancer pour un projet donné.
- Évaluer et noter les freelancers après la finalisation du projet.
- Mettre à jour leur profil.
- Réinitialiser leur mot de passe.
- Se déconnecter.

### 3.3.2 Freelancer

Les freelancers sont des professionnels indépendants à la recherche de projets. Ils utilisent la plateforme pour consulter les offres disponibles, proposer leurs services et collaborer avec les clients.

Leurs principales fonctionnalités sont :

- Créer un compte et s'authentifier.
- Rechercher et consulter les projets disponibles.
- Postuler à un projet en soumettant une proposition détaillée.
- Mettre à jour leur profil.
- Réinitialiser leur mot de passe.
- Se déconnecter.



### 3.3.3 Administrateur

L'administrateur est responsable de la supervision globale du système. Il garantit la sécurité, la fiabilité et le bon fonctionnement de la plateforme, tout en assurant une gestion efficace des utilisateurs.

Ses principales fonctionnalités sont :

- S'authentifier de manière sécurisée.
- Accéder à un tableau de bord administratif.
- Gérer les comptes clients et freelancers.
- Surveiller les activités de la plateforme.
- Réinitialiser son mot de passe.
- Se déconnecter.

## 3.4 Conclusion

Ce chapitre marque une étape clé dans le développement de notre application. Après avoir analysé les exigences, nous avons pu établir une liste détaillée des **besoins fonctionnels et non fonctionnels**, tout en identifiant les **acteurs** concernés.

La prochaine partie sera consacrée à la **conception**. Nous y aborderons notamment le **diagramme de cas d'utilisation**, ainsi que les **diagrammes de séquence** et le **diagramme de classe**, qui permettront de structurer notre solution de manière plus concrète.

# Chapitre IV : Conception

## 4.1 Introduction

La conception est une étape clé dans le développement d'un projet logiciel. C'est là qu'on définit précisément l'architecture du système, comment les différents éléments interagissent, et comment l'application doit se comporter globalement.

Dans cette partie, on va utiliser UML (Unified Modeling Language) pour modéliser notre plateforme web qui connecte clients et freelancers. L'idée, c'est de visualiser à la fois la structure et le comportement du système, pour que tout le monde puisse bien comprendre son fonctionnement.

## 4.2 Diagramme de cas d'utilisation

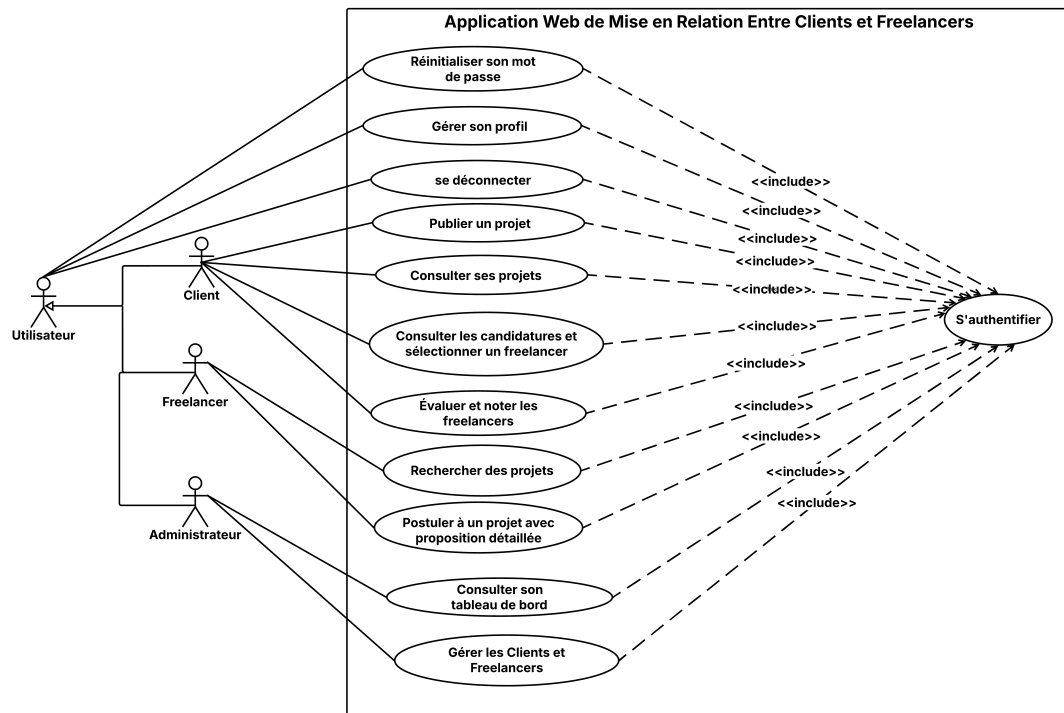


FIGURE 4.1 – Diagramme de cas d'utilisation

## 4.3 Diagrammes de Séquence

### 4.3.1 Diagramme de Séquence pour l'Authentification

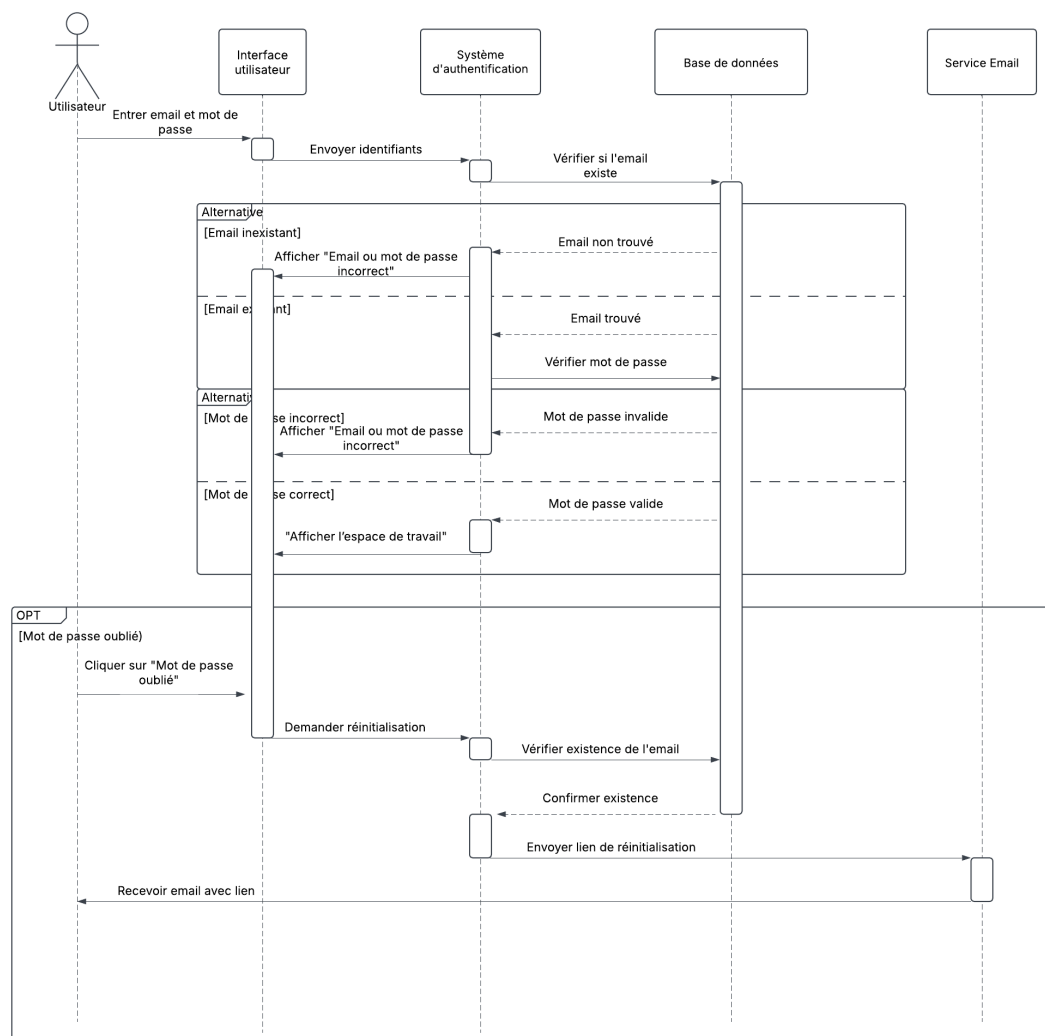


FIGURE 4.2 – Diagramme de Séquence pour l'Authentification

### 4.3.2 Diagramme de séquence pour la publication d'un projet

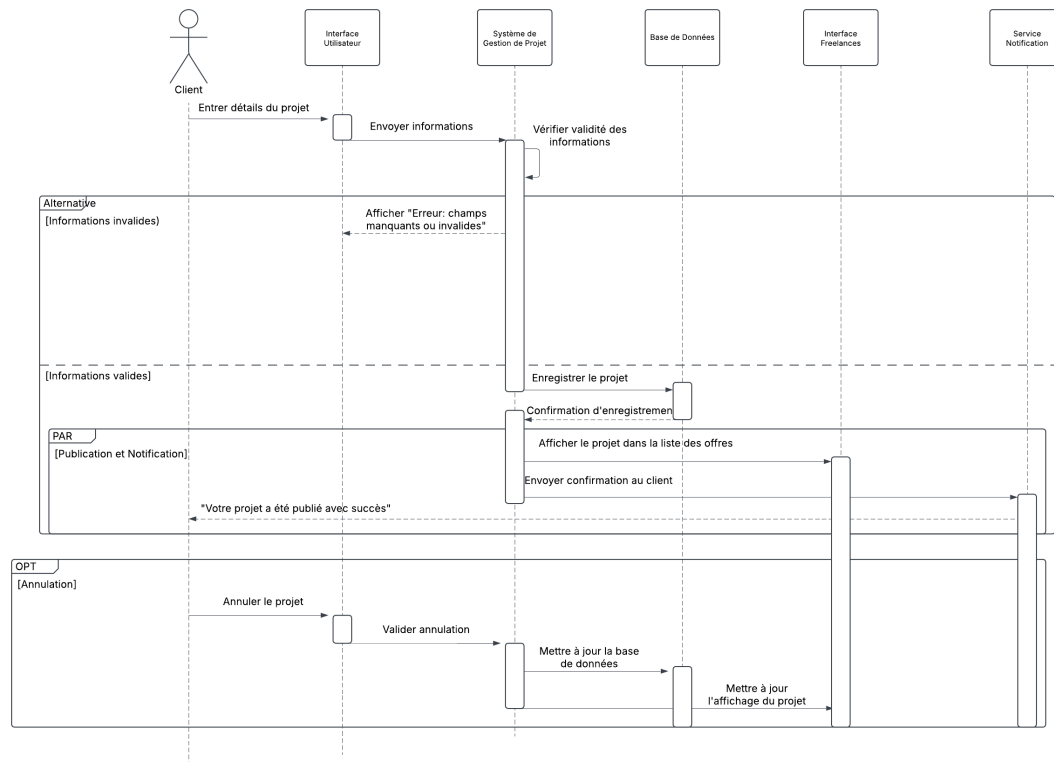


FIGURE 4.3 – Diagramme de séquence pour la publication d'un projet

### 4.3.3 Diagramme de séquence pour la candidature à un projet

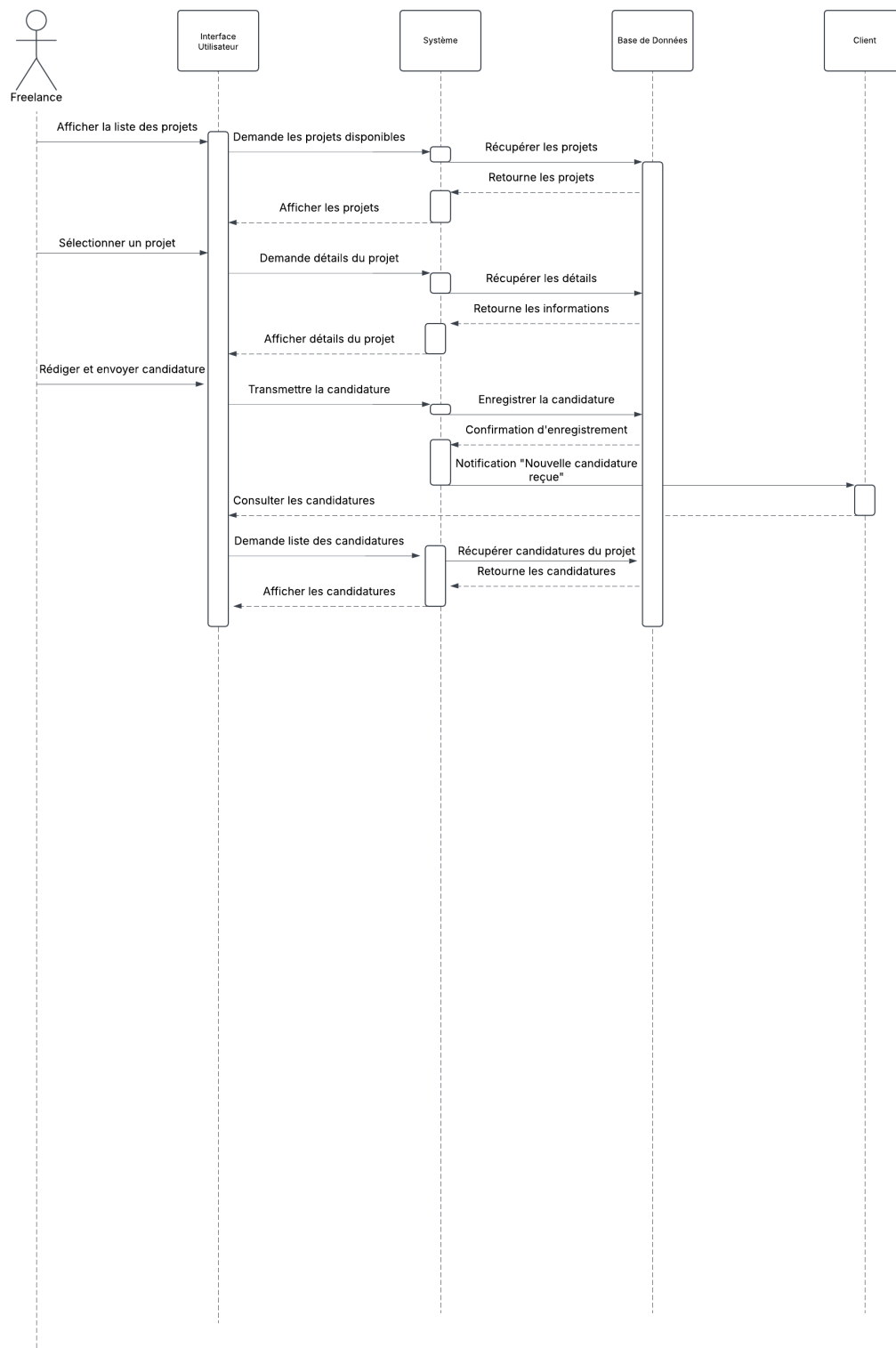


FIGURE 4.4 – Diagramme de séquence pour la candidature à un projet

#### 4.3.4 Diagramme de séquence pour la sélection d'un freelance

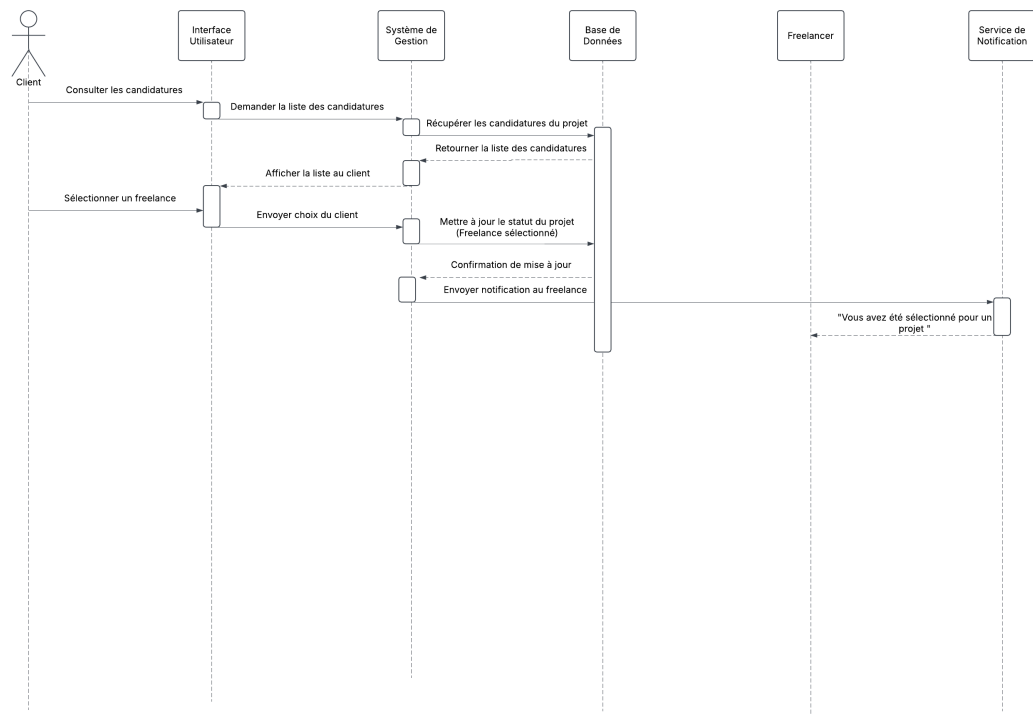


FIGURE 4.5 – Diagramme de séquence pour la sélection d'un freelance

#### 4.4 Diagramme de classe

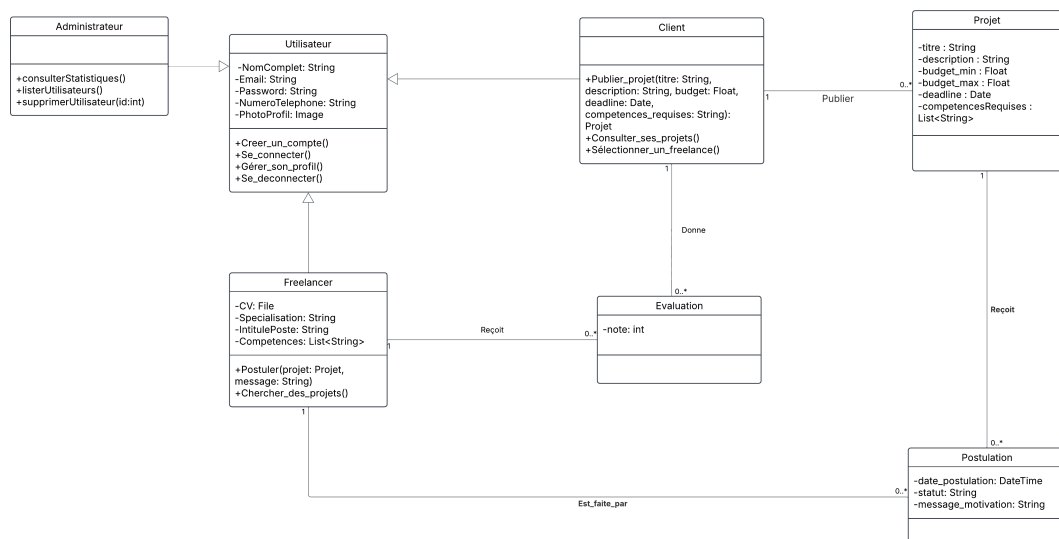


FIGURE 4.6 – Diagramme de classe

## 4.5 Conclusion

Dans ce chapitre, nous avons décrit les diagrammes de cas d'utilisation, de classe et de séquence, afin de préparer un terrain favorable pour la prochaine étape.

Maintenant, notre application est prête à être codée. Dans le chapitre suivant, nous passerons à la dernière phase du cycle de vie de notre application : la réalisation.



# Chapitre V : Réalisation

## 5.1 Introduction

Dans le chapitre précédent, nous avons abordé les différentes étapes de conception de l'application, en présentant notamment le diagramme de cas d'utilisation. Par la suite, nous avons approfondi cette phase en détaillant le diagramme de classe, ainsi que les diagrammes de séquences.

Ce chapitre, quant à lui, marque la fin de cette section du rapport. Son objectif principal est de présenter les réalisations concrètes issues de notre travail.

Pour commencer, nous décrirons l'environnement utilisé, tant au niveau matériel que logiciel. Ensuite, nous illustrerons les fonctionnalités de l'application à travers plusieurs captures d'écran.

## 5.2 Environnement de développement

Ici, nous présentons l'environnement de développement de l'application, en détaillant les configurations matérielles et logicielles qui ont été utilisées.

### 5.2.1 Environnement matériel

Pour donner une meilleure idée des conditions de travail, il nous semble pertinent de préciser les caractéristiques techniques de la machine utilisée pour le développement. Notre application a été codée sur un Lenovo ThinkPad T470, un portable assez performant dont voici les **spécifications techniques principales** :

- **Processeur** : Intel® Core™ i5-7300U CPU @ 2.60GHz 2.71 GHz
- **Mémoire vive (RAM)** : 16,0 Go (15,8 Go utilisable)
- **Système d'exploitation** : Windows 11 Professionnel, 64 bits
- **Disque dur** : SSD de 256 Go

### 5.2.2 Environnement logiciel

Dans cette section, nous avons passé en revue les différents outils employés tout au long du projet, que ce soit pour l'analyse ou la réalisation de notre application.

#### Outil de modélisation UML



FIGURE 5.1 – Lucidchart

Lucidchart est un environnement de travail visuel qui associe création de diagrammes, visualisation de données et fonctionnalités de collaboration afin de faciliter la communication et stimuler l'innovation.[2]

Il permet de créer divers types de diagrammes, notamment UML, organigrammes, logigrammes, cartes conceptuelles, et bien plus encore, grâce à une interface intuitive basée sur le cloud. Les utilisateurs peuvent collaborer en temps réel, partager leurs travaux et intégrer Lucidchart avec des applications populaires telles que Google Workspace, Microsoft Teams, Jira et Confluence.[3]

#### Technologies, Bibliothèques et Outils de Développement

##### Django et DRF



FIGURE 5.2 – Logo django + DRF

Django est un framework web Python de haut niveau qui encourage le développement rapide et une conception propre et pragmatique. Il est conçu pour aider les développeurs à créer des applications web rapidement et avec un minimum de code. Django REST Framework (DRF) est une bibliothèque puissante et flexible pour construire des API Web. Elle facilite la création d'API robustes et bien structurées, offrant des fonctionnalités telles que la sérialisation des données, la gestion des permissions, l'authentification, et la navigation dans l'API via une interface web conviviale.[4]

### **PostgreSQL**



FIGURE 5.3 – Logo PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle objet (ORDBMS) open source puissant qui utilise et étend le langage SQL, combiné à de nombreuses fonctionnalités permettant de stocker et de faire évoluer en toute sécurité les charges de travail de données les plus complexes. Les origines de PostgreSQL remontent à 1986 dans le cadre du projet POSTGRES à l'Université de Californie à Berkeley, et il bénéficie de plus de 35 ans de développement actif sur la plateforme principale.[5]

### **React.js**



FIGURE 5.4 – Logo React

React est une bibliothèque JavaScript déclarative, efficace et flexible pour construire des interfaces utilisateurs (UI). Elle vous permet de composer des UI complexes à partir de petits morceaux de code isolés appelés « composants ».[6]

### **Material UI (MUI)**



FIGURE 5.5 – Logo MUI

**Material UI** est une bibliothèque de composants React open-source qui implémente le Material Design de Google. Elle comprend une collection complète de composants prédéfinis prêts à être utilisés en production dès la sortie de la boîte et offre une suite d'options de personnalisation qui facilitent la mise en œuvre de votre propre système de conception personnalisé sur nos composants.[7]

### **React Router**



FIGURE 5.6 – Logo React Router

React Router est une bibliothèque de routage pour React qui permet de gérer la navigation dans les applications monopages (SPA). Elle offre une approche déclarative pour définir les routes, facilitant ainsi la gestion de la navigation et des vues en fonction de l'URL. React Router est conçu pour être flexible et s'adapte à divers environnements, y compris le web et les applications mobiles. Il prend en charge des fonctionnalités avancées telles que le chargement de données, les transitions de vues et la gestion de l'historique de navigation.

### JWT (JSON Web Tokens)

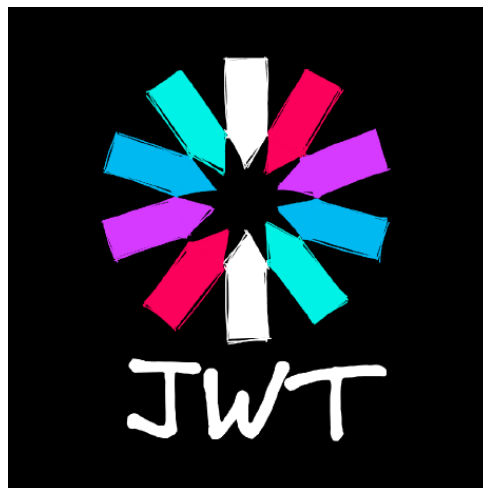


FIGURE 5.7 – Logo JWT

JSON Web Token (JWT) est une norme ouverte (RFC 7519) qui définit un moyen compact et autonome de transmettre des informations de manière sécurisée entre les parties sous forme d'objet JSON. Ces informations sont vérifiables et fiables grâce à leur signature numérique. Les JWT peuvent être signés à l'aide d'une clé secrète (avec l'algorithme HMAC) ou d'une paire de clés publique/privée utilisant RSA ou ECDSA.

Bien que les JWT puissent être chiffrés pour garantir la confidentialité entre les parties, nous nous concentrerons sur les jetons signés. Les jetons signés peuvent vérifier l'intégrité des revendications qu'ils contiennent, tandis que les jetons chiffrés les masquent aux autres parties. Lorsque les jetons sont signés à l'aide de paires de clés publique/privée, la signature certifie également que seule la partie détenant la clé privée est celle qui les a signés.[8]

### Git et Github



FIGURE 5.8 – Logo Git et Github

Git est un système de contrôle de version distribué open-source, largement utilisé par les développeurs pour gérer les modifications apportées au code source de leurs projets. Il permet de suivre l'historique des changements, de collaborer efficacement en équipe et de revenir à des versions antérieures du code si nécessaire.

GitHub, quant à lui, est une plateforme basée sur le cloud qui héberge des dépôts Git. Elle offre des fonctionnalités supplémentaires telles que le suivi des problèmes, les demandes de tirage (pull requests), la gestion des tâches et des wikis pour chaque projet. GitHub facilite la collaboration entre développeurs en permettant de partager du code, de réviser les modifications et de gérer les versions de manière centralisée.[9]

### Visual Studio Code



FIGURE 5.9 – Logo vs code

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js, et dispose d'un riche écosystème d'extensions pour d'autres langages et environnements d'exécution (tels que C++, C#, Java, Python, PHP, Go, .NET).[10]

### Postman



FIGURE 5.10 – Logo Postman

**Postman** est une plateforme collaborative pour le développement d'API, utilisée par des millions de développeurs et plus de 500 000 entreprises dans le monde. Elle permet de concevoir, tester, documenter et surveiller des API de manière efficace. Grâce à ses outils tels que les collections, les environnements, les scripts de test et les intégrations CI/CD, Postman facilite la gestion du cycle de vie complet des API, de la conception à la production.[11]

### Swagger



FIGURE 5.11 – Logo Swagger

Swagger est un ensemble d'outils open source construits autour de la spécification OpenAPI, qui peuvent vous aider à concevoir, construire, documenter et consommer des API REST. Les principaux outils Swagger incluent : Swagger Editor, Swagger UI et Swagger Codegen.[12]

## 5.3 Interfaces graphiques

L'interface graphique joue un rôle clé dans le développement d'une application web réussie. Elle doit être intuitive et agréable, afin d'offrir une expérience utilisateur fluide et engageante.

Dans cette section, nous présentons une série de captures d'écran illustrant les fonctionnalités principales de notre application.

### Page d'accueil

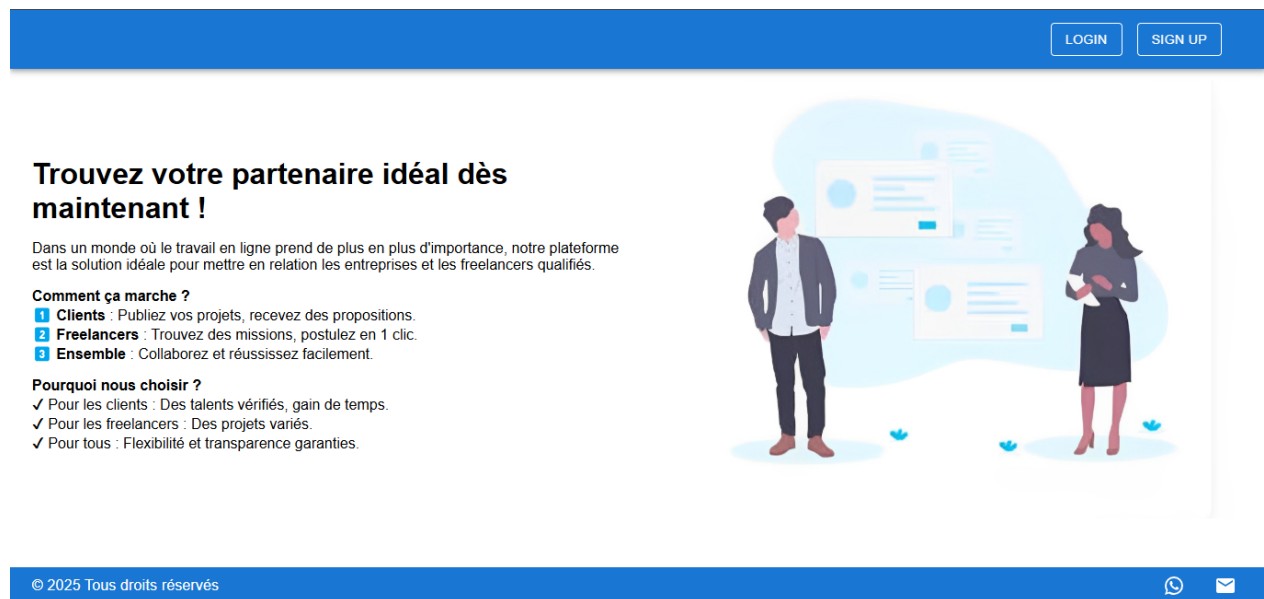


FIGURE 5.12 – Page d'accueil

### Création de comptes

Pour assurer une expérience utilisateur fluide et structurée, nous avons conçu une interface de création de compte en **trois étapes**, adaptée aussi bien aux clients qu'aux freelancers. La première étape, "**Informations de base**", est commune aux deux types d'utilisateurs et permet de saisir des données essentielles telles que le nom, l'adresse email et le mot de passe. La deuxième étape, "**Informations complémentaires**", est personnalisée selon le type d'utilisateur et permet de recueillir des informations spécifiques (par exemple, les compétences pour les freelancers). Enfin, la dernière étape, "**Confirmation**", présente un récapitulatif des données saisies afin de permettre à l'utilisateur de les vérifier avant l'envoi. Une fois le formulaire validé, un email de vérification est automatiquement envoyé à l'adresse fournie. Si l'utilisateur ne reçoit pas cet email, une option "**Renvoyer l'email**" est disponible pour lui permettre de générer un nouveau lien de confirmation

*Pour le client :*



# Créer un compte

1

2

3

Informations de baseInformations complémentairesConfirmation

Type d'utilisateur  
Client

Nom Complet \*  
Ahmed Dine

Email \*  
ahmed.dine.21397  
Veuillez entrer une adresse email valide

Mot de passe \*  
12  
Doit contenir au moins 8 caractères

Force du mot de passe:  
Très faible

Numéro de Téléphone \*  
2  
Format invalide. Ex: 44076356, +(222)44076356, +22244076356

PHOTO DE PROFIL (OPTIONNELLE, MAX 2MB)

RETOURSUIVANT

Vous avez déjà un compte ? [SE CONNECTER](#)

FIGURE 5.13 – Erreurs de saisie dans Information de base (Client)

# Créer un compte

1

2

3

Informations de baseInformations complémentairesConfirmation

Type d'utilisateur

Client

Nom Complet \*

Ahmed Dine

Email \*

ahmed.dine.21397@gmail.com

Mot de passe \*

Hoqn\*C21397

Doit contenir 8+ caractères, majuscule, minuscule, chiffre et caractère spécial

Force du mot de passe:

Très fort

Numéro de Téléphone \*

26225035

Formats acceptés: 44076356, +(222)44076356, +22244076356

AHMED DINE.JPG297.3 KB

RETOUR

SUIVANT

Vous avez déjà un compte ?

SE CONNECTER

FIGURE 5.14 – Information de base (Client)

The screenshot shows a registration form titled "Créer un compte". It has a progress bar with three steps: 1. Informations de base (checked), 2. Informations complémentaires (active), and 3. Confirmation. Below the progress bar, it states "Aucune information supplémentaire requise pour les clients." (No additional information required for clients). There are two buttons: "RETOUR" (Return) on the left and "SUIVANT" (Next) on the right. At the bottom, it says "Vous avez déjà un compte ? [SE CONNECTER](#)" (Do you already have an account? [Log in](#)).

FIGURE 5.15 – Informations complémentaires (Client)

The screenshot shows a confirmation page with a green header "Inscription réussie!". The main text says "Un email de vérification a été envoyé à ahmed.dine.21397@gmail.com. Veuillez cliquer sur le lien dans l'email pour activer votre compte." (A verification email has been sent to ahmed.dine.21397@gmail.com. Please click on the link in the email to activate your account.). Below this, it asks "Vous n'avez pas reçu l'email?" (Did you not receive the email?). There is a blue button labeled "RENOYER L'EMAIL" (Resend email). At the bottom, there is a link "ALLER À LA PAGE DE CONNEXION" (Go to the login page).

FIGURE 5.16 – Inscription réussie!

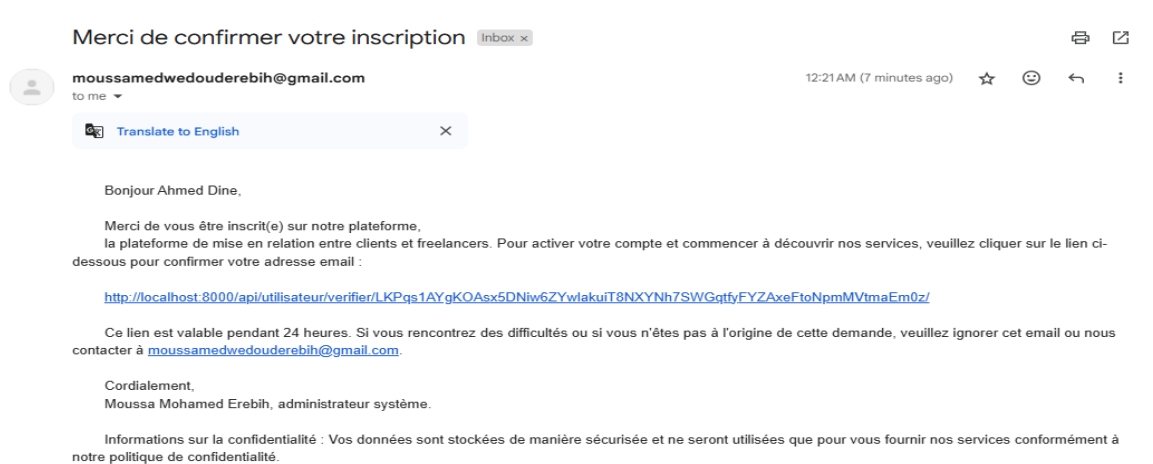


FIGURE 5.17 – Email de vérification

*Dans le cas où vous cliquez sur Renvoyer l'email :*

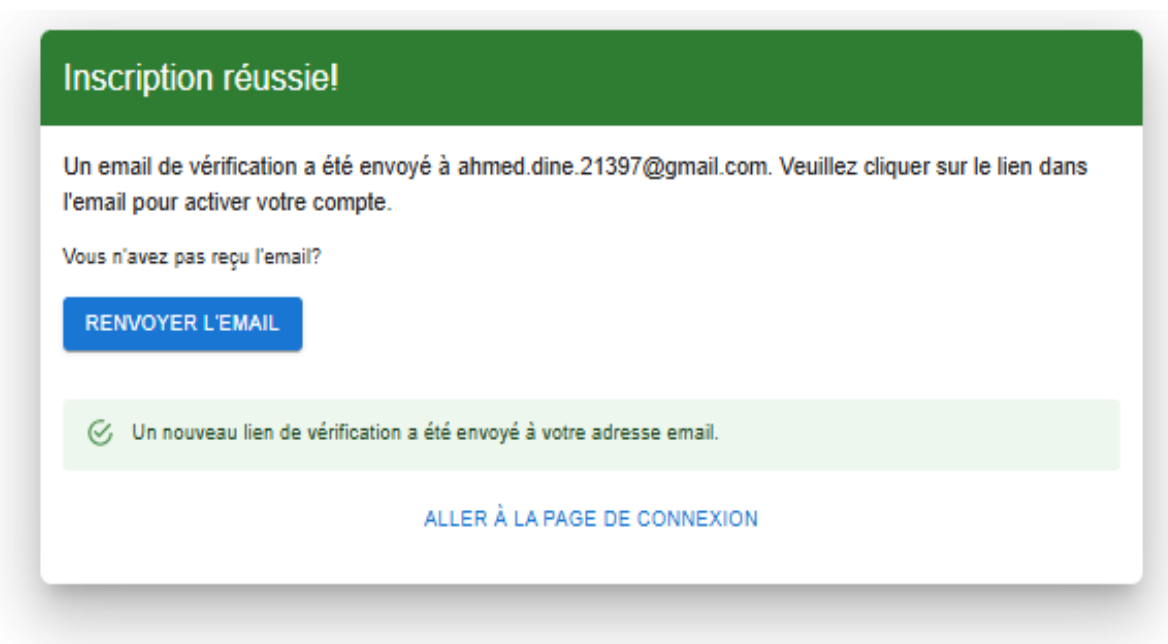


FIGURE 5.18 – Renvoyer l'email de vérification

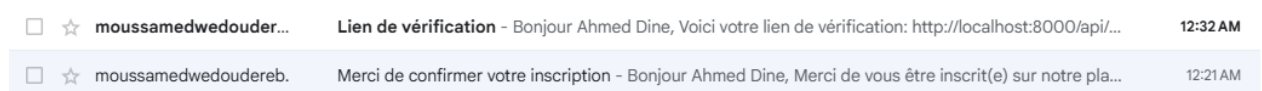


FIGURE 5.19 – Un nouveau email de vérification

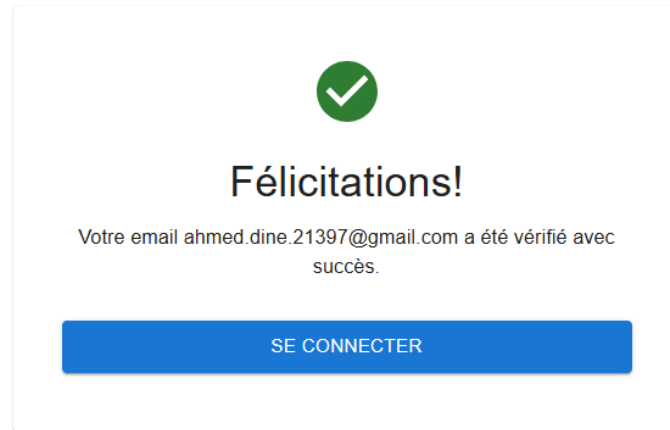


FIGURE 5.20 – Email vérifié avec succès

***Pour le freelancer :***

Le processus de création de compte est identique pour les deux types d'utilisateurs — **client** et **freelancer** — et suit les trois mêmes étapes. La seule différence réside dans la deuxième étape, "**Informations complémentaires**", qui est adaptée au rôle sélectionné. Dans le cas d'un **freelancer**, cette étape permet de renseigner des informations professionnelles essentielles telles que les **compétences**, etc ...

## Créer un compte

✓

2


3

Informations de baseInformations complémentairesConfirmation

ⓘ


Veuillez corriger les erreurs avant de continuer

- Un CV est requis
- La spécialisation est requise
- L'intitulé de poste est requis

 TÉLÉverser VOTRE CV (PDF ou DOC, MAX 5MB)

Un CV est requis

Spécialisation \*



La spécialisation est requise

Intitulé de poste \*

L'intitulé de poste est requis

Compétences (séparées par des virgules)

RETOUR

SUIVANT

Vous avez déjà un compte ?

[SE CONNECTER](#)

FIGURE 5.21 – Erreurs de saisie dans Informations complémentaires (Freelancer)

**Créer un compte**

Informations de base Informations complémentaires **3** Confirmation

**Vérifiez vos informations**

Type: Freelancer  
Nom: Nabil Karim  
Email: nabil.karim.21397@gmail.com  
Téléphone: 23322897  
Spécialisation: Développeur  
Intitulé de poste: Développeur MERN  
Compétences: MongoDB, Express.js, React, Node.js, Tailwind.CSS

RETOUR ENREGISTREMENT...

Vous avez déjà un compte ? [SE CONNECTER](#)

FIGURE 5.22 – Confirmation et envoi des informations

### Connexion au compte

**Connexion à votre compte**

Adresse email \*  
moussamedwedouderebih@gmail.com

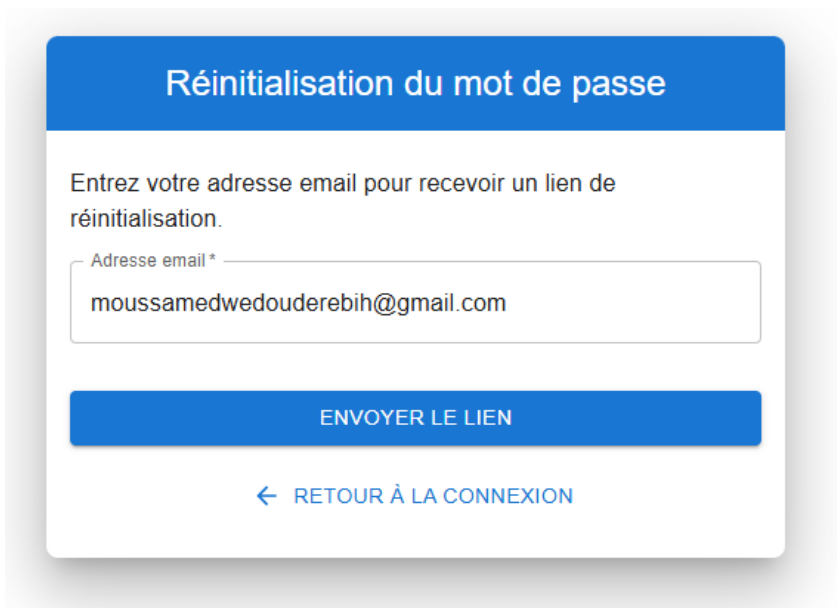
Mot de passe \*

SE CONNECTER

[Mot de passe oublié?](#)

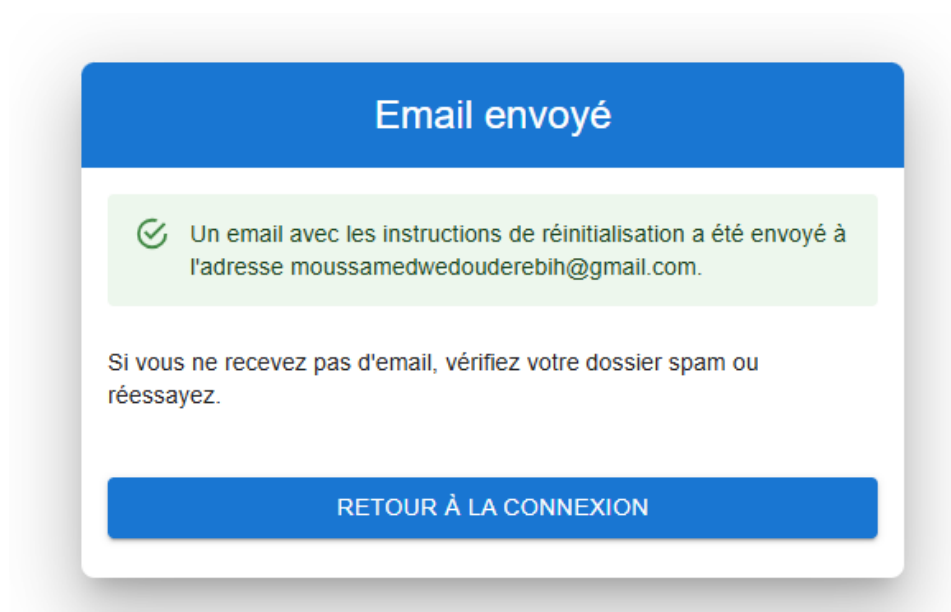
FIGURE 5.23 – Connexion à votre compte

### Dans le cas où le mot de passe est oublié



The screenshot shows a web form titled "Réinitialisation du mot de passe" (Password Reset). The form has a blue header bar with the title. Below the header, there is a text prompt: "Entrez votre adresse email pour recevoir un lien de réinitialisation." (Enter your email address to receive a reset link). A text input field contains the email address "moussamedwedouderebih@gmail.com". Below the input field is a blue button labeled "ENVOYER LE LIEN" (Send Link). At the bottom of the form is a blue link with a left-pointing arrow and the text "RETOUR À LA CONNEXION" (Return to Login).

FIGURE 5.24 – Réinitialisation du mot de passe



The screenshot shows a confirmation screen titled "Email envoyé" (Email sent) in a blue header bar. Below the header, there is a green success message with a checkmark icon: "Un email avec les instructions de réinitialisation a été envoyé à l'adresse moussamedwedouderebih@gmail.com." (An email with reset instructions has been sent to the address...). Below this message is a text prompt: "Si vous ne recevez pas d'email, vérifiez votre dossier spam ou réessayez." (If you do not receive an email, check your spam folder or try again.). At the bottom of the screen is a blue button labeled "RETOUR À LA CONNEXION" (Return to Login).

FIGURE 5.25 – Email de réinitialisation envoyé





FIGURE 5.26 – Réinitialisation de votre mot de passe

A form titled "Nouveau mot de passe" with a blue header. The text "Veuillez entrer votre nouveau mot de passe." is displayed. There are two input fields: "Nouveau mot de passe \*" and "Confirmer le mot de passe \*", both with masked text (dots) and a toggle icon. A blue button at the bottom says "RÉINITIALISER LE MOT DE PASSE".

FIGURE 5.27 – Nouveau mot de passe

✓ Mot de passe réinitialisé avec succès! ✕

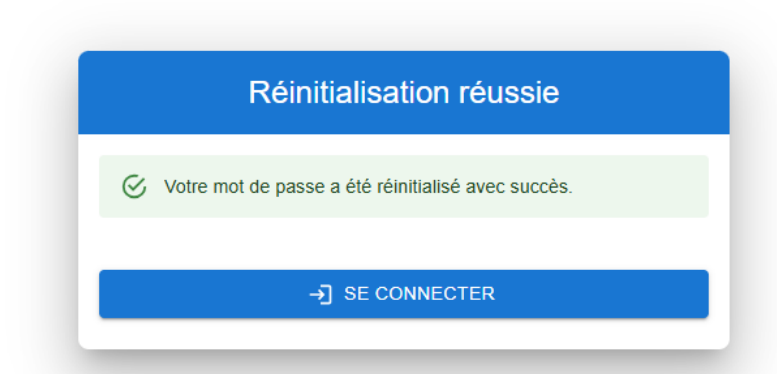


FIGURE 5.28 – Réinitialisation réussie

## Interfaces de l'espace Client

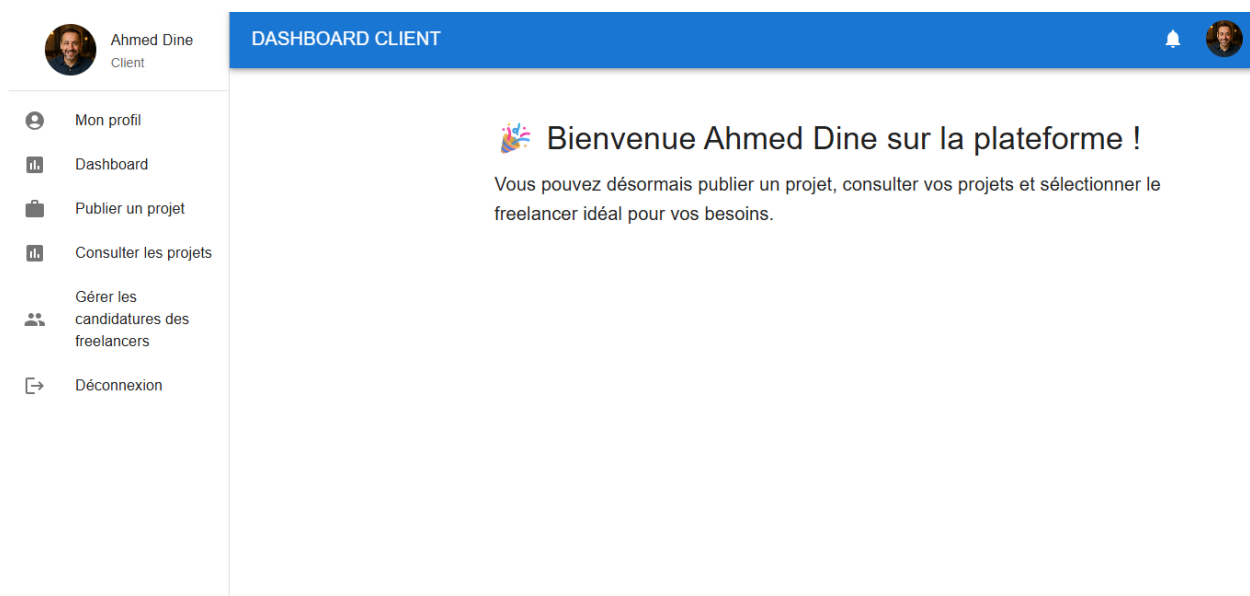
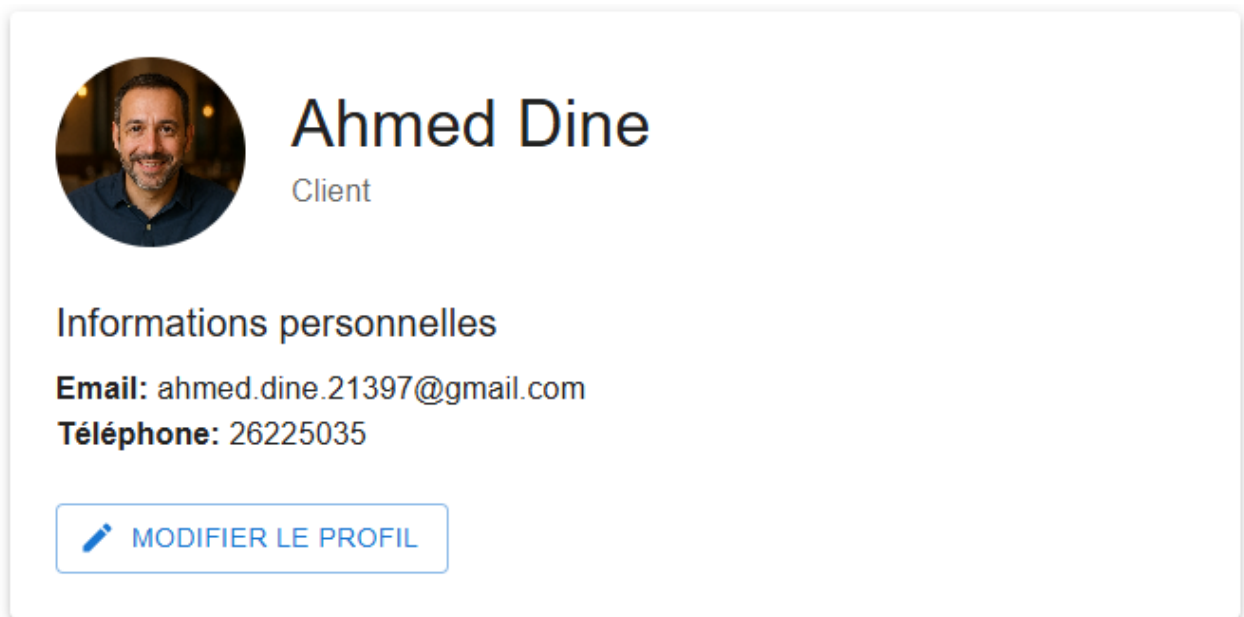



FIGURE 5.29 – DASHBOARD CLIENT

*Interface “Mon Profil” :*



A user profile card for Ahmed Dine. It features a circular profile picture of a man with a beard. To the right of the photo, the name 'Ahmed Dine' is displayed in a large, bold font, with 'Client' written below it in a smaller font. Underneath, the section 'Informations personnelles' is followed by the email 'ahmed.dine.21397@gmail.com' and the phone number '26225035'. At the bottom, there is a button with a pencil icon and the text 'MODIFIER LE PROFIL'.



**Ahmed Dine**  
Client

Informations personnelles

**Email:** ahmed.dine.21397@gmail.com  
**Téléphone:** 26225035


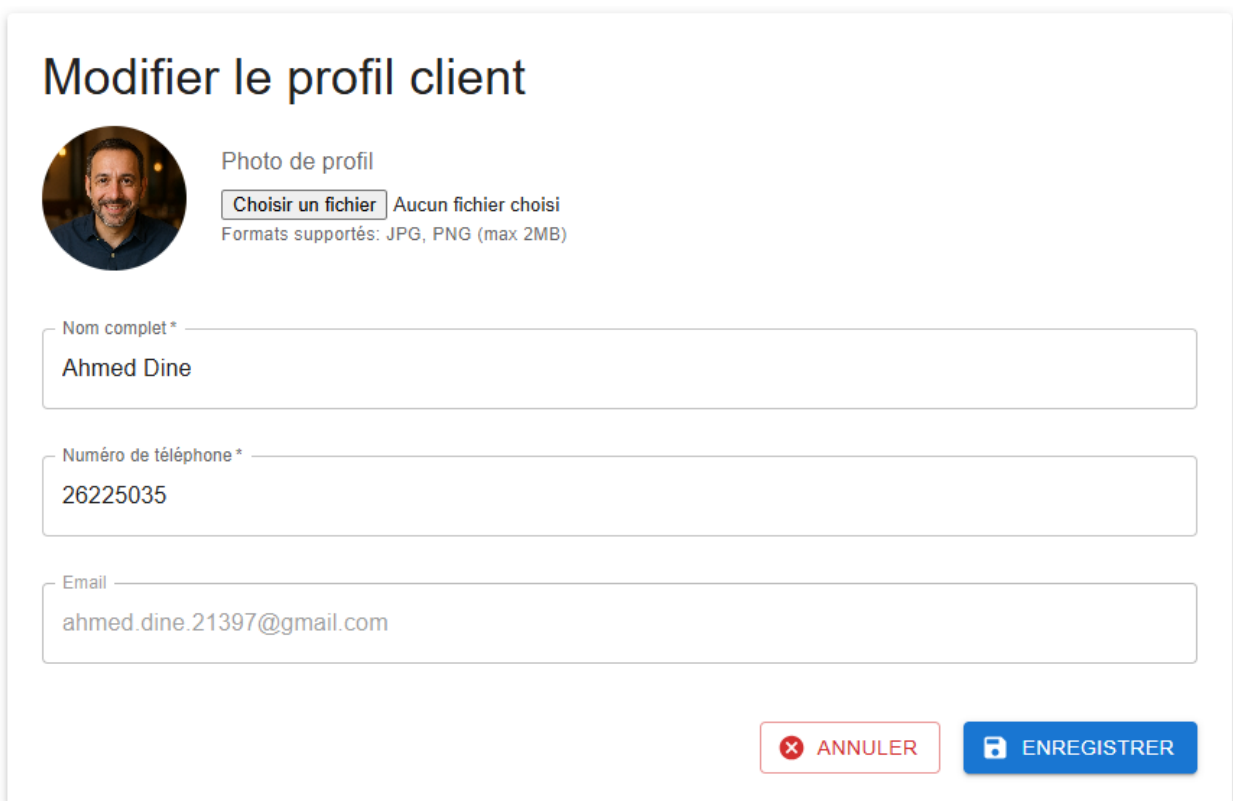
 **MODIFIER LE PROFIL**


FIGURE 5.30 – Interface “Mon Profil”(client)

*Interface “Modifier le profil ” :*



A form titled 'Modifier le profil client'. It includes a profile picture placeholder with a 'Choisir un fichier' button and text indicating no file is chosen and supported formats (JPG, PNG, max 2MB). Below are three input fields for 'Nom complet \*' (Ahmed Dine), 'Numéro de téléphone \*' (26225035), and 'Email' (ahmed.dine.21397@gmail.com). At the bottom right are two buttons: 'ANNULER' (red with a close icon) and 'ENREGISTRER' (blue with a save icon).

## Modifier le profil client

 Photo de profil  
**Choisir un fichier** Aucun fichier choisi  
Formats supportés: JPG, PNG (max 2MB)

Nom complet \*  
Ahmed Dine

Numéro de téléphone \*  
26225035

Email  
ahmed.dine.21397@gmail.com



 **ANNULER**  **ENREGISTRER**

FIGURE 5.31 – Modifier le profil (client)

### Interface “Publier un Projet” :

The screenshot shows the 'DASHBOARD CLIENT' interface. On the left is a sidebar with the user's profile 'Ahmed Dine Client' and navigation links: 'Mon profil', 'Dashboard', 'Publier un projet', 'Consulter les projets', 'Gérer les candidatures des freelancers', and 'Déconnexion'. The main content area is titled 'Publier un nouveau projet' and contains a form with the following fields: 'Titre du projet \*', 'Budget minimum (MRU) \*', 'Budget maximum (MRU) \*', 'Description détaillée \*', 'Date limite' (with a calendar icon), 'Compétences requises' (a dropdown menu), and a blue button labeled '✓ PUBLIER LE PROJET'.

FIGURE 5.32 – Interface “Publier un Projet”

This screenshot shows the same 'Publier un nouveau projet' form as Figure 5.32, but with example data entered. The 'Titre du projet \*' field contains 'Développement d'un site vi'. The 'Budget minimum (MRU) \*' field contains '8000' and the 'Budget maximum (MRU) \*' field contains '16000'. The 'Description détaillée \*' field contains the text: 'Nous recherchons un développeur frontend pour créer un site vitrine responsive de 4 à 5 pages (Accueil, À propos, Services, Contact) destiné à une entreprise locale. Le design sera fourni via Figma. L'objectif est d'avoir un site moderne, rapide, et facile à maintenir.' The 'Date limite' field shows '30/05/2025'. The 'Compétences requises' dropdown menu is open, showing a list of skills: 'Développement Web', 'Design Graphique', 'UX/UI Design', 'SEO', and 'Gestion de Projet'. The '✓ PUBLIER LE PROJET' button is still present at the bottom.

FIGURE 5.33 – Exemple d’Interface “Publier un Projet”

### Notification de succès après la publication du projet :

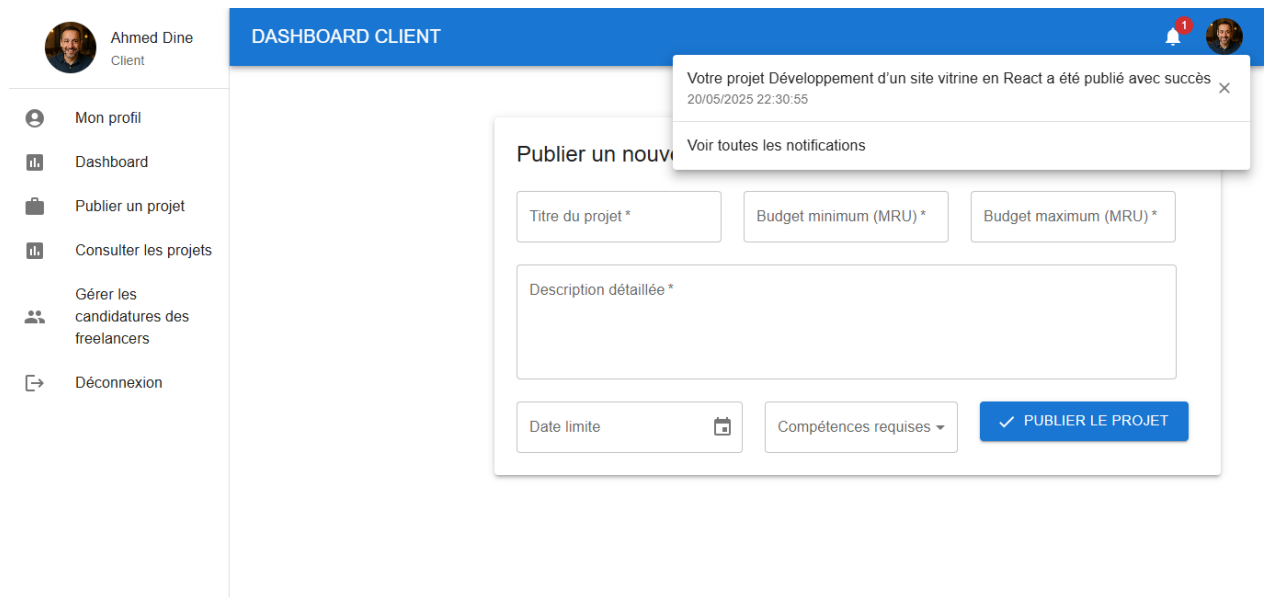


FIGURE 5.34 – Notification de succès après la publication du projet

### Interface “Consulter Mes Projets” :

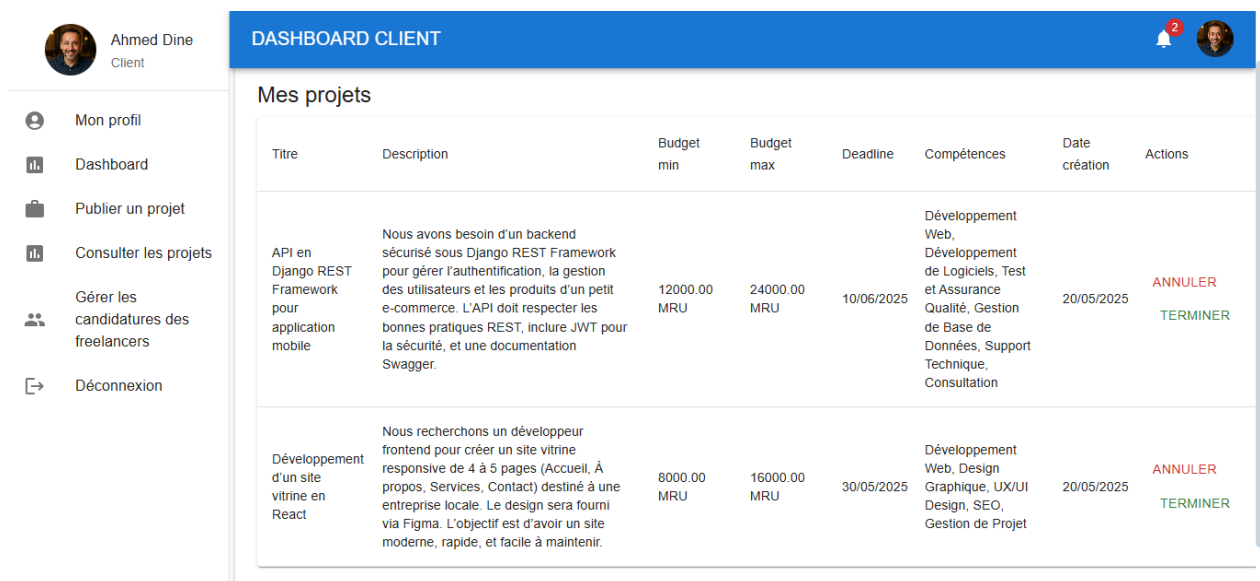


FIGURE 5.35 – Interface “Consulter Mes Projets”

### Accusé de réception des candidatures :

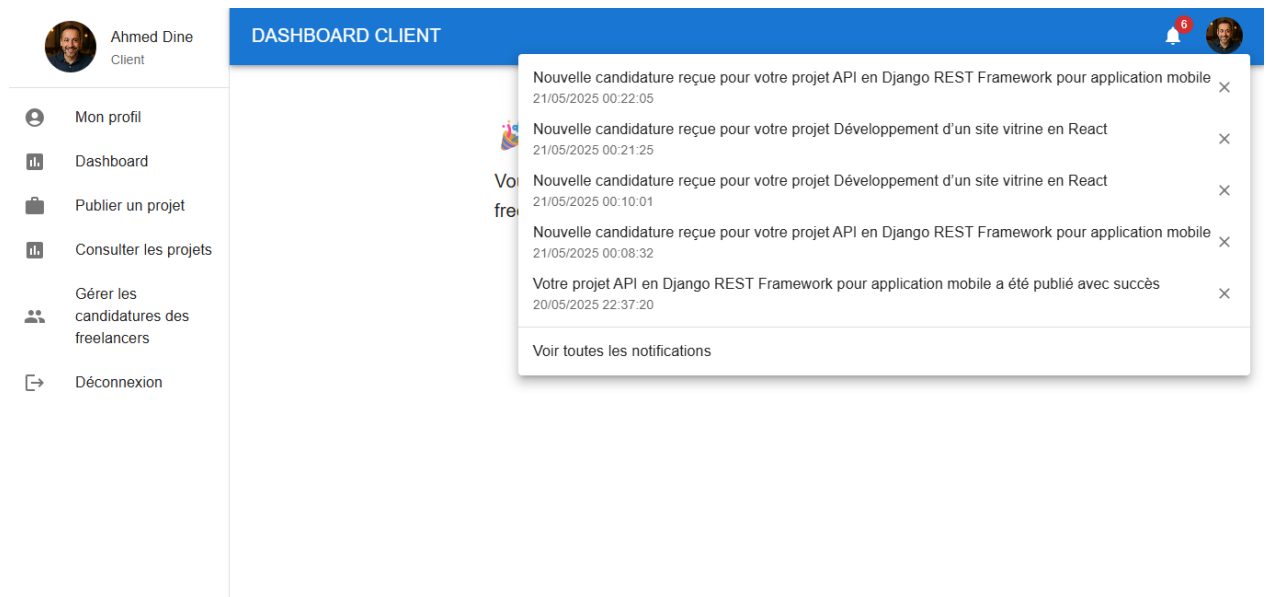


FIGURE 5.36 – Accusé de réception des candidatures

### Interface “Gérer les candidatures des freelancers” :

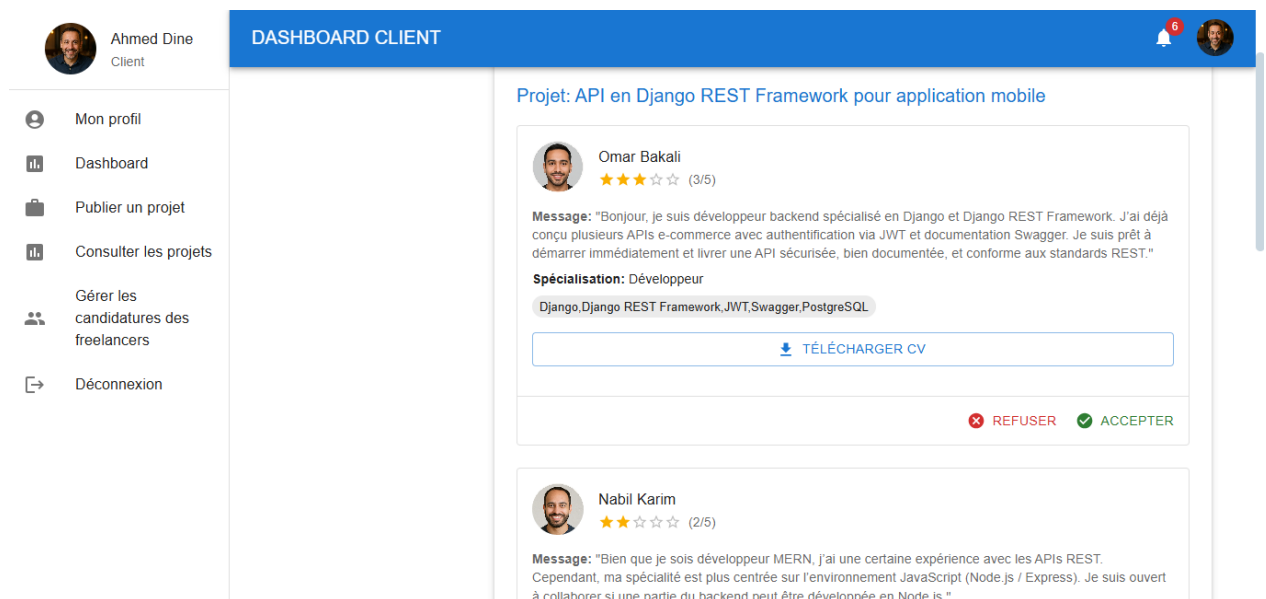


FIGURE 5.37 – Interface “Gérer les candidatures des freelancers”

*Dans le cas où le client Ahmed Dine a refusé que le freelancer Nabil Karim travaille sur son projet :*

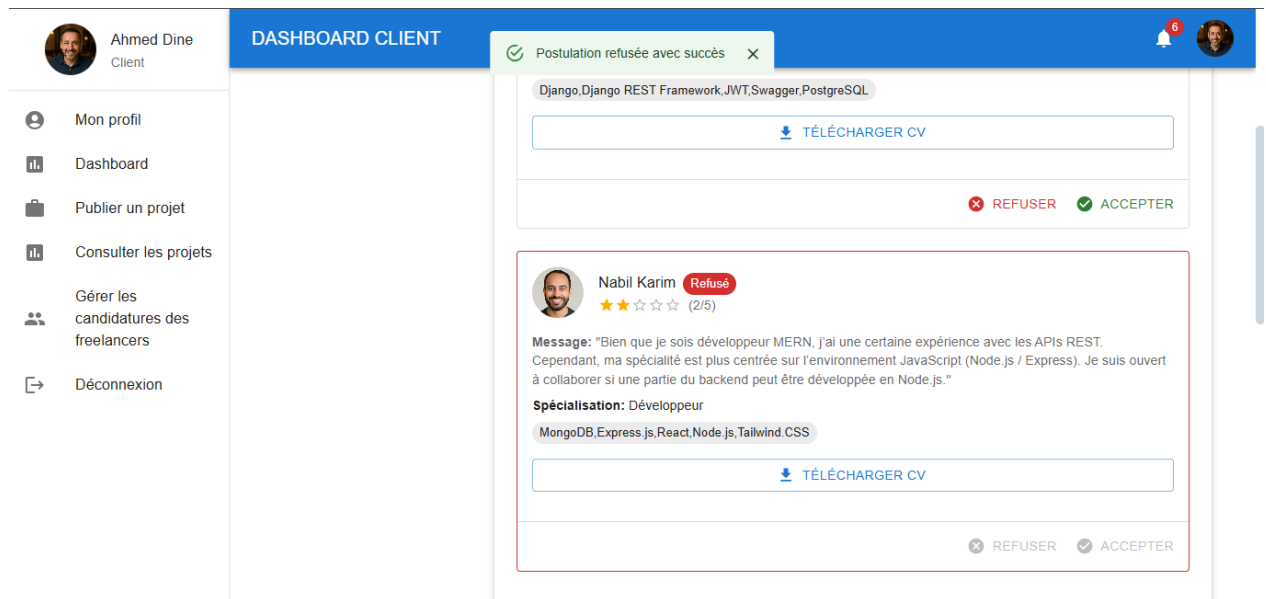


FIGURE 5.38 – Postulation refusée avec succès

*Le freelancer Nabil Karim reçoit une notification de refus de sa candidature pour ce projet :*

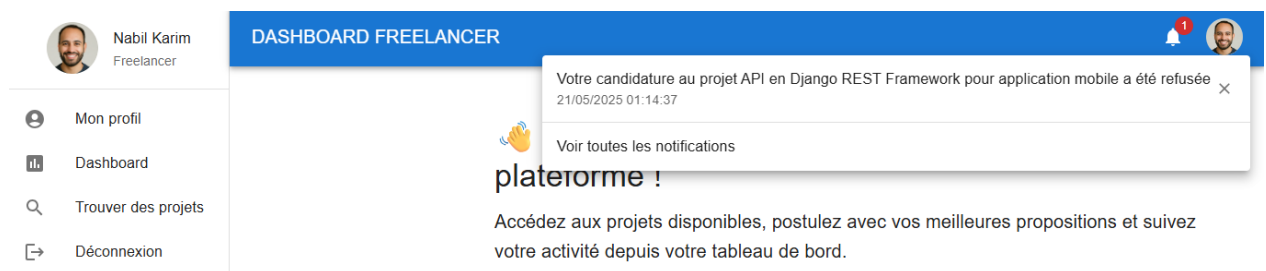


FIGURE 5.39 – Notification de refus de sa candidature

*Dans le cas où le client Ahmed Dine a accepté que le freelancer Omar Bakali travaille sur son projet :*

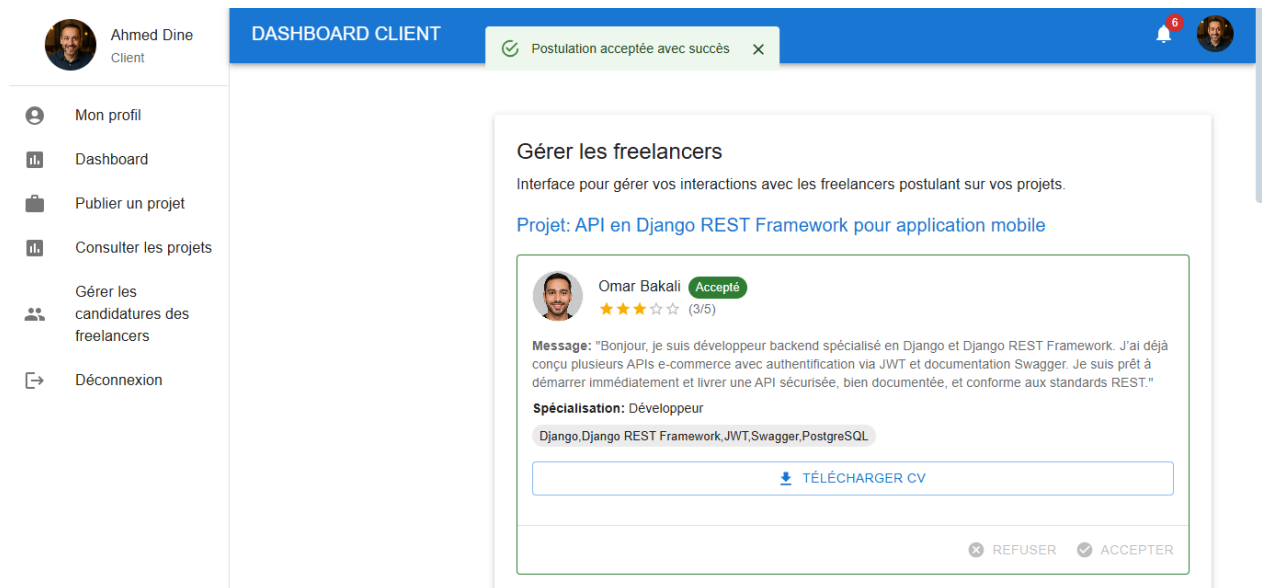


FIGURE 5.40 – Postulation acceptée avec succès

*Le freelancer Omar Bakali reçoit une notification d'acceptation de sa candidature pour ce projet, accompagnée d'un email :*

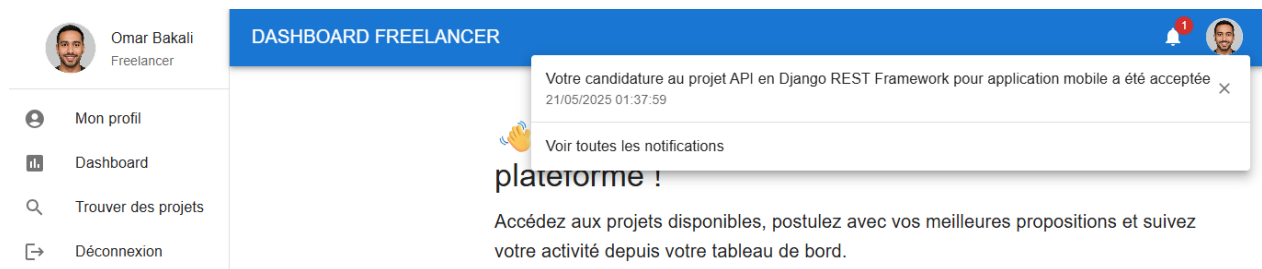


FIGURE 5.41 – Notification d'acceptation de sa candidature





FIGURE 5.42 – Email(freelancer)

*Le client Ahmed Dine reçoit également un e-mail :*



FIGURE 5.43 – Email(client)

*Interface d'evaluation de freelancer lorsque le client veut terminer un projet :*

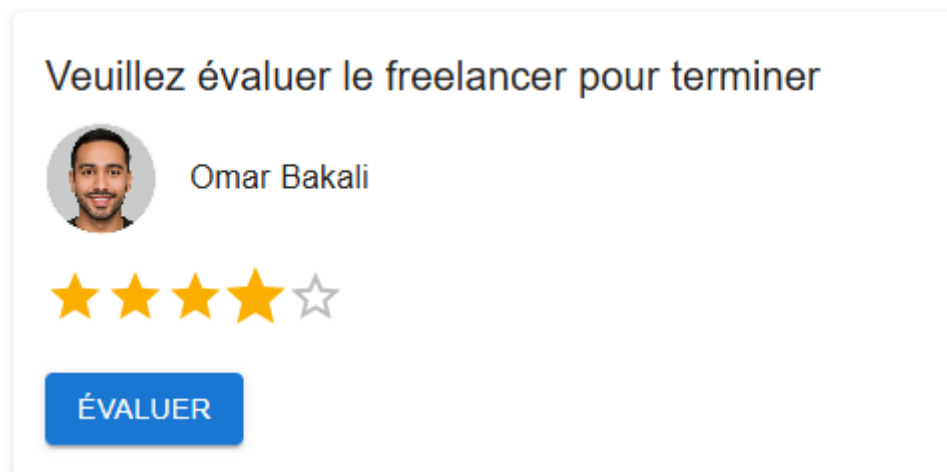



FIGURE 5.44 – Évaluer le freelancer

## Interfaces de l'espace Freelancer



FIGURE 5.45 – DASHBOARD FREELANCER


*Interface “Mon Profil” :*




## Nabil Karim

Freelancer

Informations professionnelles

 **Poste:**Développeur MERN

 **Spécialisation:**Développeur

Compétences:

MongoDB,Express.js,React,Node.js,Tailwind.CSS

CV:

TÉLÉCHARGER LE CV

Coordonnées

**Email:** nabil.karim.21397@gmail.com  
**Téléphone:** 23322897


 MODIFIER LE PROFIL

FIGURE 5.46 – Interface “Mon Profil”(freelancer)

*Interface “Modifier le profil ” :*

## Modifier le profil




Photo de profil  
 Aucun fichier choisi

Nom complet \*

Nabil Karim

Numéro de téléphone \*

23322897

Spécialisation \*

Développeur

Intitulé de poste \*

Développeur MERN

Compétences

MongoDB, Express.js, React, Node.js, Tailwind.CSS

CV

CV actuel: Nabil\_Karim.pdf

Aucun fichier choisi

FIGURE 5.47 – Modifier le profil (freelancer)

**Interface “Trouver des projets” :**

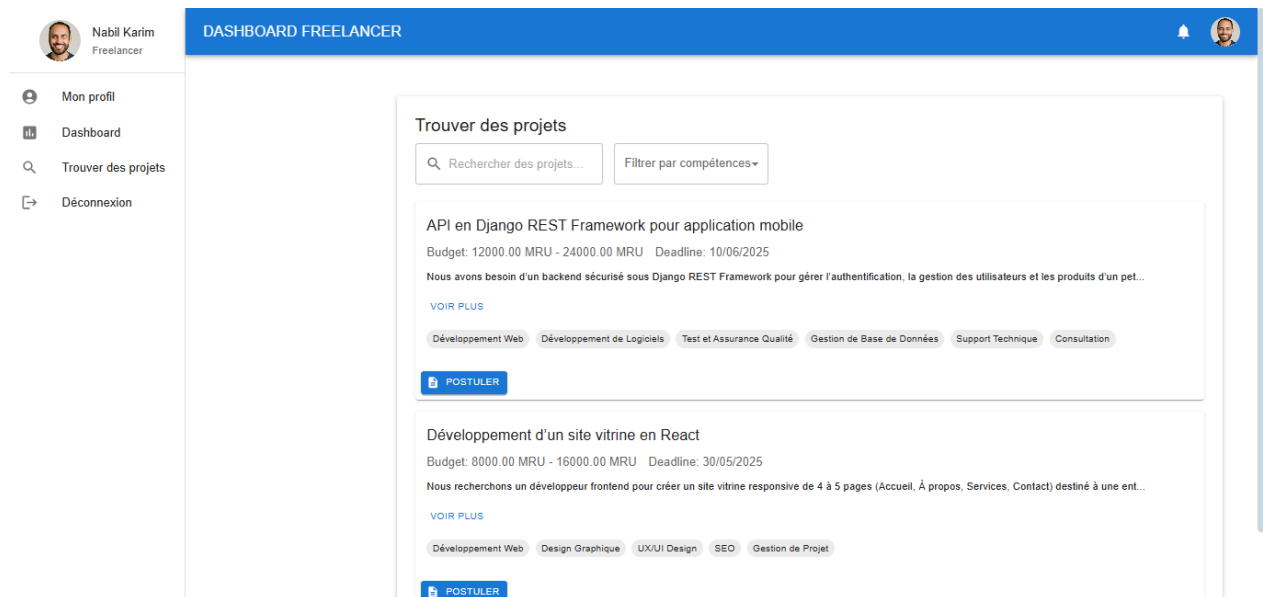


FIGURE 5.48 – Interface “Trouver des projets”

### Interface “Postulation à un projet” :

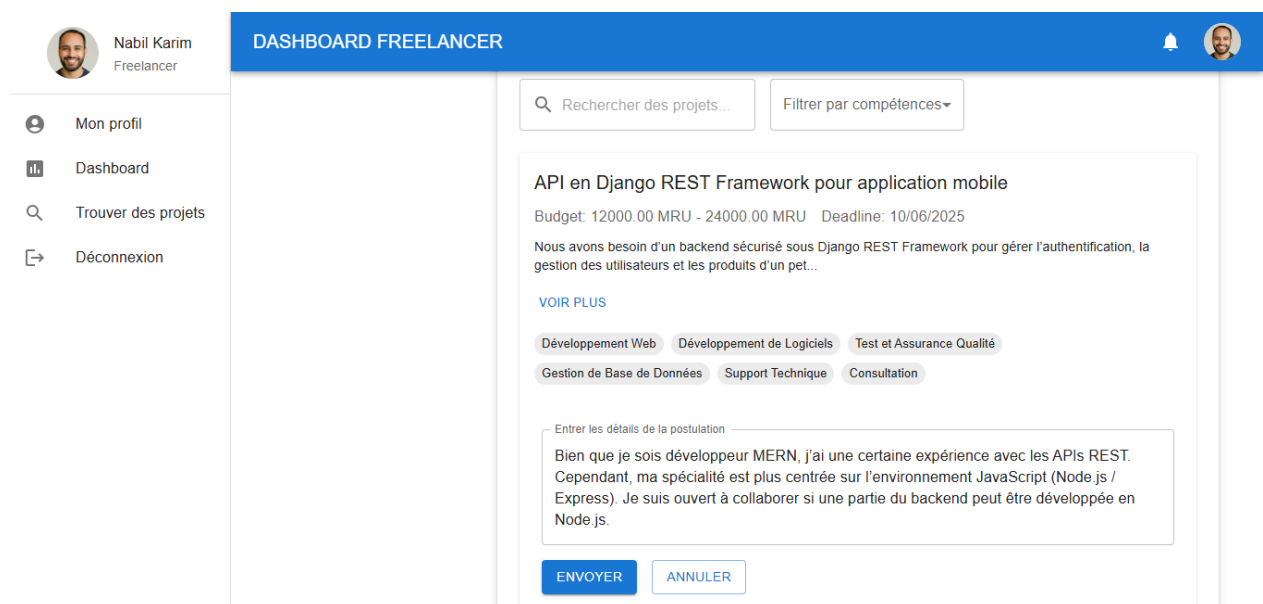


FIGURE 5.49 – Interface “Postulation à un projet”

### Interfaces de l'espace Administrateur

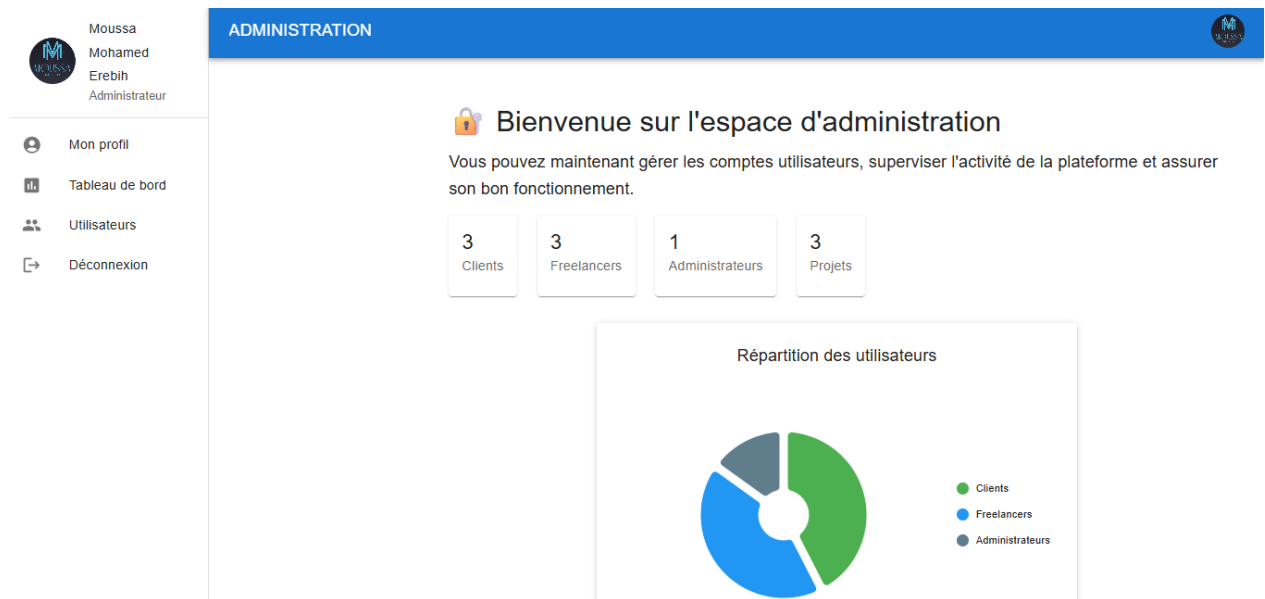


FIGURE 5.50 – DASHBOARD ADMINISTRATEUR

*Interface “Mon Profil” :*

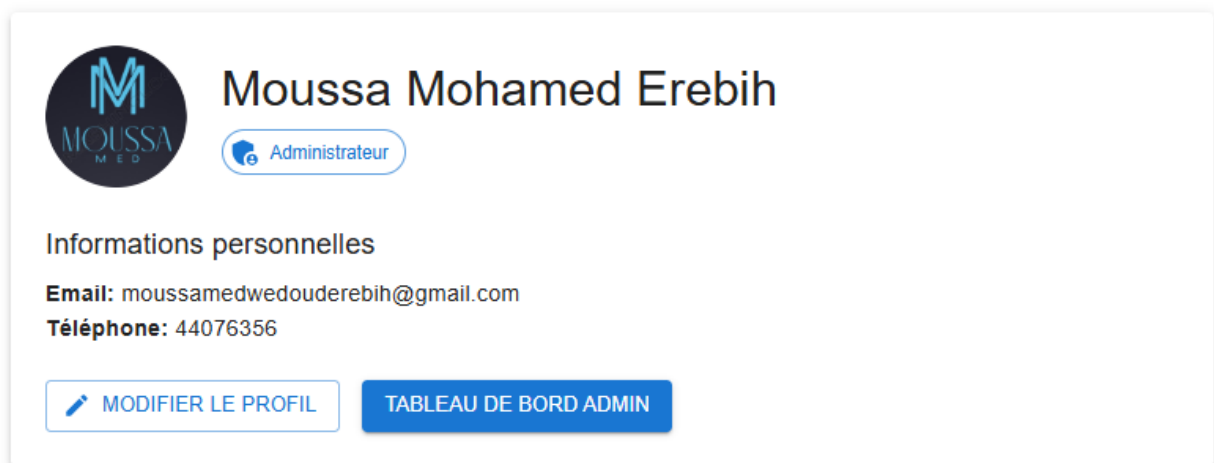


FIGURE 5.51 – Interface “Mon Profil”(admin)

*Interface “Modifier le profil” :*

# Modifier le profil administrateur




Photo de profil  
 Aucun fichier choisi  
 Formats supportés: JPG, PNG (max 5MB)

Nom complet \*  
 Moussa Mohamed Erebih


Email  
 moussamedwedouderebih@gmail.com

Numéro de téléphone \*  
 44076356













Membre depuis: Invalid Date

FIGURE 5.52 – Modifier le profil (admin)

## Interface de gestion des utilisateurs :


 Moussa  
Mohamed  
Erebih  
Administrateur

ADMINISTRATION

Photo	Nom Complet	Email	Type	Actions
	Nabil Karim	nabil.karim.21397@gmail.com	Freelancer	
	Omar Bakali	omar.bakali.21397@gmail.com	Freelancer	
	Ahmed Dine	ahmed.dine.21397@gmail.com	Client	
	Amira Sy	amira.sy.21397@gmail.com	Freelancer	
	Fatimata	fatimata.zahra.21397@gmail.com	Client	
	Yacine Elhassan	yacine.elhassan.21397@gmail.com	Client	

Mon profil  
 Tableau de bord  
 Utilisateurs  
 Déconnexion

FIGURE 5.53 – Interface de gestion des utilisateurs

# Conclusion Générale

Dans le cadre de notre projet de fin d'études en MIAE, nous avons eu l'opportunité de développer une plateforme web mettant en relation clients et freelancers. Ce projet a été l'occasion idéale d'appliquer les connaissances théoriques acquises durant notre formation, tout en relevant des défis techniques concrets.

Sur le plan professionnel comme personnel, cette expérience s'est révélée extrêmement formatrice. Nous avons dû adopter une méthodologie rigoureuse, approfondir nos compétences en développement et résoudre des problèmes complexes. Au final, non seulement les objectifs initiaux ont été atteints, mais nous avons aussi gagné en expertise et en maturité.

Contre toute attente, le projet a dépassé nos espérances, ce qui nous procure une grande fierté. Bien sûr, des améliorations restent possibles, et nous envisageons déjà des pistes d'évolution pour optimiser l'application.

Pour conclure, cette aventure a été à la fois exigeante et enrichissante. Elle a consolidé nos compétences et ouvert de nouvelles perspectives, tant sur le plan technique que sur celui de l'innovation. Nous sommes impatients de voir comment cette solution pourra évoluer à l'avenir.



# Webographie

- [1] <https://syskat.com/index.php/en/2013-02-20-08-59-23>
- [2] <https://www.lucidchart.com/pages/fr/exemple/schema-en-ligne>
- [3] <https://www.blogdumoderateur.com/tools/lucidchart/>
- [4] <https://openclassrooms.com/fr/courses/7192416-mettez-en-place-une-api-avec-django-rest-framework/7424482-decouvrez-django-rest-framework>
- [5] <https://www.postgresql.org/about/>
- [6] <https://fr.legacy.reactjs.org/tutorial/tutorial.html>
- [7] <https://mui.com/material-ui/getting-started/>
- [8] <https://jwt.io/introduction>
- [9] <https://www.blogdumoderateur.com/tools/github/>
- [10] <https://visualstudio.microsoft.com/fr/>
- [11] <https://www.postman.com/product/what-is-postman/>
- [12] [https://swagger.io/docs/specification/v3\\_0/about/](https://swagger.io/docs/specification/v3_0/about/)