



المدرسة العليا  
للتكنولوجيا - الصويرة  
L'ÉCOLE SUPÉRIEURE DE  
TECHNOLOGIE – ESSAOUIRA

**Université Cadi Ayyad**  
**École Supérieure de Technologie Essaouira**  
**Département Génie Informatique et Mathématique**

---

---

**Filière : “Génie Informatique”**

*présentée et soutenue par*

**ELAIMA IKRAM & ID-MOULLAY IMANE**

le 2024

**SOUS LE THÈME :**

---

**Développement d'une Application web pour la gestion de  
projet.**

---

sous la direction de : **CHOUHADE NADIA**

# Table des matières

<b>Remerciements</b>	<b>5</b>
<b>Introduction générale</b>	<b>6</b>
<b>1 Contexte Général du Projet</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Contexte général . . . . .	7
1.3 Problématique . . . . .	7
1.4 Objectifs du projet . . . . .	7
1.5 Objectifs Personnels . . . . .	8
1.6 Conclusion . . . . .	8
<b>2 Analyse et Conception du Projet :</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Langage UML . . . . .	9
2.2.1 Présentation du langage UML . . . . .	9
2.2.2 Intérêt de la modélisation . . . . .	9
2.2.3 Les avantages d'UML . . . . .	9
2.3 Conception avec UML . . . . .	10
2.3.1 Le diagramme de cas d'utilisation . . . . .	10
2.3.2 Le Diagramme de Séquence . . . . .	11
2.3.3 Le Diagramme de classe . . . . .	13
2.4 Les outils utilisés . . . . .	15
2.4.1 React . . . . .	15
2.4.2 Node.js . . . . .	16
2.4.3 MySQL . . . . .	16
2.4.4 XAMPP . . . . .	17
2.4.5 Visual Studio . . . . .	18
2.4.6 Github . . . . .	18
2.4.7 Latex . . . . .	19
2.5 Conclusion . . . . .	19
<b>3 Réalisation du projet :</b>	<b>20</b>
3.1 Introduction . . . . .	20
3.2 Page Signup . . . . .	20
3.3 Page Login . . . . .	20
3.4 Interface d'administrateur . . . . .	21
3.4.1 Page dashboard . . . . .	21
3.4.2 Page Projects . . . . .	22
3.4.3 Page Tasks . . . . .	22
3.4.4 Page Users . . . . .	23
3.4.5 Page Documents . . . . .	23
3.5 Interface de membre d'équipe . . . . .	24

3.5.1	Page dashboard . . . . .	24
3.5.2	Page Tasks . . . . .	25
3.5.3	Page Documents . . . . .	25
3.6	Conclusion . . . . .	26
<b>Conclusion générale</b>		<b>27</b>
<b>Bibliographie</b>		<b>28</b>
*		

# Table des figures

2.1	Le diagramme de cas d'utilisation . . . . .	11
2.2	Le diagramme de Séquence . . . . .	12
2.3	Le diagramme de classe . . . . .	13
2.4	Logo de React . . . . .	15
2.5	Logo de Node.js . . . . .	16
2.6	Logo de MySQL . . . . .	16
2.7	Logo de Xampp . . . . .	17
2.8	Logo de Visual Studio . . . . .	18
2.9	Logo de Github . . . . .	18
2.10	Logo de Latex . . . . .	19
3.1	Page Signup . . . . .	20
3.2	Page Signup . . . . .	21
3.3	Page dashboard . . . . .	21
3.4	Page dashboard . . . . .	22
3.5	Page Projects . . . . .	22
3.6	Page Tasks . . . . .	23
3.7	Page Users . . . . .	23
3.8	Page Documents . . . . .	24
3.9	Page dashboard-utilisateur . . . . .	24
3.10	Page dashboard-utilisateur . . . . .	25
3.11	Page Tasks-utilisateur . . . . .	25
3.12	Page Documents-utilisateur . . . . .	26

\*

# Liste des tableaux

2.1	Classe Utilisateur. . . . .	13
2.2	Classe Projet. . . . .	14
2.3	Classe Etape. . . . .	14
2.4	Classe Tache. . . . .	14
2.5	Classe Document. . . . .	15

\*

# REMERCIEMENTS :

Avant toute chose, nous tenons à exprimer notre profonde gratitude à Dieu le Tout-Puissant, qui nous a accompagnés tout au long de ces années d'études et nous a donné la force, le courage et la patience nécessaires pour mener à bien ce projet.

Nos remerciements les plus sincères s'adressent à **Madame CHOUHAD Nadia**, notre encadrante, pour son soutien indéfectible, sa disponibilité, ses précieux conseils et ses critiques constructives qui ont contribué de manière significative à l'amélioration de notre travail. Sa rigueur et son expertise nous ont permis d'approfondir notre réflexion et d'affiner notre méthodologie.

Nous sommes également reconnaissants envers l'ensemble de nos enseignants pour leurs précieux enseignements, leurs conseils avisés et leur générosité dans le partage de leurs connaissances. Ils ont joué un rôle crucial dans notre formation et nous ont permis d'acquérir les compétences nécessaires pour réaliser ce projet.

Nos sincères remerciements s'adressent également à toute l'équipe pédagogique de l'École Supérieure de Technologie d'Essaouira pour son soutien et son encouragement tout au long de notre cursus.

Nous tenons à remercier les membres du jury pour l'honneur qu'ils nous font en acceptant de présider la soutenance de notre projet et pour leurs remarques constructives qui nous aideront à le perfectionner.

Enfin, nous exprimons notre gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet, que ce soit par leurs conseils, leur soutien moral ou leur aide matérielle.

# Introduction générale :

Dans un environnement professionnel en constante mutation, la gestion de projet devient cruciale pour assurer le succès des initiatives collaboratives au sein des organisations. C'est dans ce contexte dynamique que notre projet de fin d'études prend place, avec pour ambition de développer une application web de gestion de projet collaborative. L'objectif central est de proposer une solution complète facilitant la planification, la coordination et le suivi des activités au sein des équipes de travail, simplifiant ainsi la gestion des tâches, des échéanciers et des ressources.

En mettant l'accent sur des fonctionnalités essentielles telles que la création, la modification et la suppression de projets, la gestion d'étapes et de tâches, ainsi que l'attribution de responsabilités et de dates limites, notre application vise à répondre aux besoins diversifiés des équipes collaboratives. Les piliers de notre approche reposent sur des tableaux de bord personnalisables, une collaboration transparente via des commentaires, une messagerie instantanée dédiée, des notifications en temps réel, et une gestion avancée des utilisateurs avec des niveaux d'autorisation différenciés.

Le rapport détaillera les aspects techniques, le processus de développement et les résultats attendus, structurés autour de trois axes majeurs : le contexte général et les choix méthodologiques, l'analyse et la conception, et enfin, la réalisation concrète du projet. Chacune de ces phases apportera une contribution significative à la compréhension complète de notre démarche, témoignant de notre engagement envers la qualité et la rigueur dans la réalisation de ce projet de fin d'études.

# Chapitre 1

## Contexte Général du Projet

### 1.1 Introduction

Dans ce chapitre, nous aborderons le contexte général du projet, sa présentation et sa description, la problématique ainsi que ses objectifs et ses avantages.

### 1.2 Contexte général

Le monde professionnel est en constante évolution. Les organisations doivent faire face à une concurrence accrue et à des changements rapides dans leur environnement. Dans ce contexte, la gestion de projet devient un élément crucial du succès.

Les projets collaboratifs sont de plus en plus fréquents. Ils permettent de combiner les compétences et les expertises de différentes personnes pour atteindre des objectifs communs. Cependant, la gestion de ces projets peut être complexe. Il est nécessaire de coordonner le travail des équipes, de suivre l'avancement des tâches et de gérer les ressources de manière efficace.

C'est dans ce contexte que s'inscrit notre projet de fin d'études. Nous avons pour objectif de développer une application web de gestion de projet collaborative.

### 1.3 Problématique

Les équipes de travail collaboratives sont souvent confrontées à des défis importants en matière de gestion de projet. La communication, la coordination et le suivi des tâches peuvent être difficiles, surtout lorsque les équipes sont dispersées géographiquement ou travaillent sur des fuseaux horaires différents. De plus, les outils de gestion de projet traditionnels ne sont pas toujours adaptés aux besoins des équipes collaboratives.

Pour cela on a pensé qu'on a besoin d'une application web de gestion de projet qui répond aux besoins spécifiques des équipes modernes. L'application doit être facile à utiliser, flexible et personnalisable. Elle doit également offrir des fonctionnalités avancées pour la communication, la collaboration et le suivi des projets.

### 1.4 Objectifs du projet

D'après le contexte général et la problématique définis précédemment, nous pouvons identifier les objectifs clairs et précis de notre projet :

- Ajout de projets et étapes : Le chef d'équipe peut ajouter des projets et définir les étapes associées à ces projets.



- Suivi de l'avancement : Permet de suivre l'avancement des projets avec un pourcentage de progression basé sur les tâches terminées.
- La création de tâches attribuées à différents membres de l'équipe.
- Possibilité de définir des détails spécifiques pour chaque tâche, tels que la description, la priorité et les dates limites.
- Attribution efficace des ressources : Permet l'attribution efficace des ressources nécessaires à chaque tâche ou étape du projet, en tenant compte des compétences et des disponibilités des membres de l'équipe.
- Accès pour les chefs d'équipe : Les chefs d'équipe ont accès aux documents et rapports envoyés par les membres de l'équipe pour assurer une collaboration transparente.
- Personnalisation des autorisations : Les administrateurs peuvent définir des niveaux d'autorisation personnalisés pour chaque utilisateur, leur permettant de changer le statut de leurs tâches une fois terminées.

## 1.5 Objectifs Personnels

- Maîtriser les langages de développement web ainsi que les frameworks et bibliothèques utilisés tels que React et Node.js, qui sont nouveaux pour nous.
- Savoir gérer un projet de manière professionnelle et structurée.
- Se confronter à des situations professionnelles.
- Faire preuve de créativité et avoir de nouvelles idées.
- Le travail en groupe sur un projet comme celui-ci nous permet d'échanger des idées et des informations tout en répartissant les tâches.
- Développer le sens du travail en équipe, le sens du relationnel et des responsabilités.
- Finaliser notre formation informatique en mettant en pratique les compétences acquises.

## 1.6 Conclusion

Dans ce chapitre, nous avons discuté du contexte général du projet, de sa présentation et de sa description, de la problématique ainsi que des objectifs de notre application, en incluant également les objectifs personnels.

# Chapitre 2

## Analyse et Conception du Projet :

### 2.1 Introduction

Dans cette section, nous traiterons de la conception de notre application web en utilisant le langage UML (Unified Modeling Language), notamment les principaux diagrammes qui décrivent la conception de notre application et toutes ses fonctionnalités. De plus, nous examinerons les langages et les outils utilisés tout au long du développement de notre projet.

### 2.2 Langage UML

#### 2.2.1 Présentation du langage UML

UML (en anglais Unified Modeling Language, ■ langage de modélisation unifié ■) est un langage graphique de modélisation des données et des traitements. C'est une formalisation nonpropriétaire de la modélisation objet utilisée en génie logiciel. UML spécifie plusieurs objectifs qui font un outil exact de communication : Comprendre et décrire les besoins. Spécifier un système. Établir l'architecture logicielle.

#### 2.2.2 Intérêt de la modélisation

L'utilisation de la modélisation conceptuelle dans le développement des systèmes d'information permet de prendre en compte les besoins des applications d'une façon plus adéquats et de présenter d'une manière abstraite certains aspects des systèmes physiques et humains.

#### 2.2.3 Les avantages d'UML

UML (Unified Modeling Language) est un langage formel et standardisé. Il offre plusieurs avantages :

- ▶ Gain de précision.
- ▶ Motivation à l'utilisation d'outils.
- ▶ Gagne de stabilité et de fixité.

UML est un support de communication adéquat et compétitif. Il éclaire et facilite la compréhension de représentations abstraites complexes. Sa plurivalence et sa souplesse en font un langage universel. UML a pour objectif de spécifier, édifier et documenter les systèmes à base de logiciel. Il n'est pas une méthode, mais une notation qui laisse la liberté de la conception. UML permet de modéliser tous les types de systèmes informatiques, mais nécessite une méthodologie de conception.

UML normalise les concepts objet. Sa notation graphique permet d'exprimer une solution objet, simplifiant la comparaison et l'appréciation des solutions. Il cadre l'analyse objet, sous-entendant

une démarche d'analyse permettant de reproduire une solution objet de manière itérative grâce aux diagrammes, qui supportent l'abstraction.

Un diagramme UML est une représentation graphique, avec différentes vues statiques et dynamiques :

**Statique :**

- Cas d'utilisation
- Classes
- Composants
- Objets
- Déploiement

**Dynamique (comportemental) :**

- Séquences
- Activité
- État-transition
- Collaboration

**Fonctionnel :**

- Cas d'utilisation
- Collaboration

Ces diagrammes ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utilisables sont les diagrammes d'activités, de classes, de cas d'utilisation, d'objets, d'états transitions et de séquence. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'étude à qu'ils permettent de formaliser les contraintes de la réalisation et les solutions.

## **2.3 Conception avec UML**

### **2.3.1 Le diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation permet de déterminer les possibilités d'interférence entre le système et les acteurs, c'est-à-dire déterminer toutes les fonctionnalités que doit fournir le système. Il permet aussi de délimiter ce dernier.

- Chaque usage effectué par les acteurs est représenté par un cas d'utilisation.
- Chaque cas d'utilisation symbolise une fonctionnalité qui leur est offerte afin d'engendrer le résultat attendu.
- Le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur.

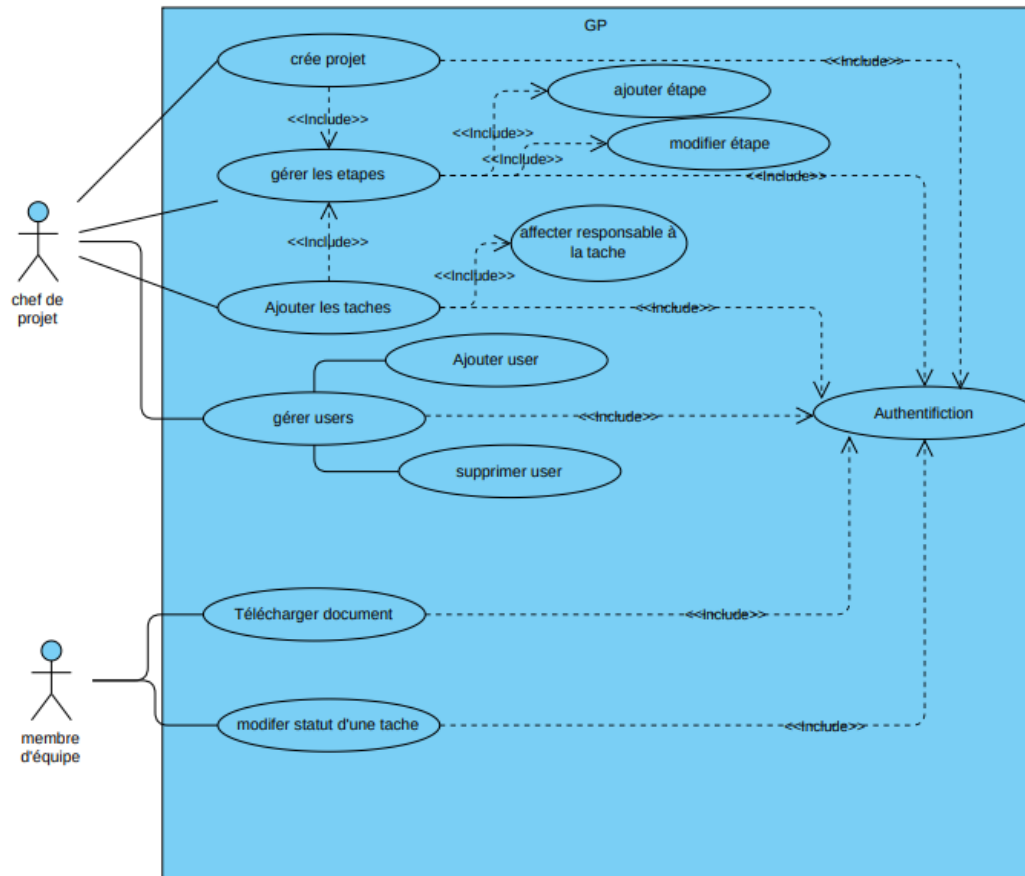


FIGURE 2.1 – représente le diagramme de cas d'utilisation global de notre projet, avec les acteurs suivants : Chef de projet et Membre d'équipe.

### 2.3.2 Le Diagramme de Séquence

C'est un diagramme qui désigne ce qu'il se passe durant un cas d'utilisation, en représentant les interactions entre les acteurs et le système selon un ordre chronologique par des messages synchrones et asynchrones.

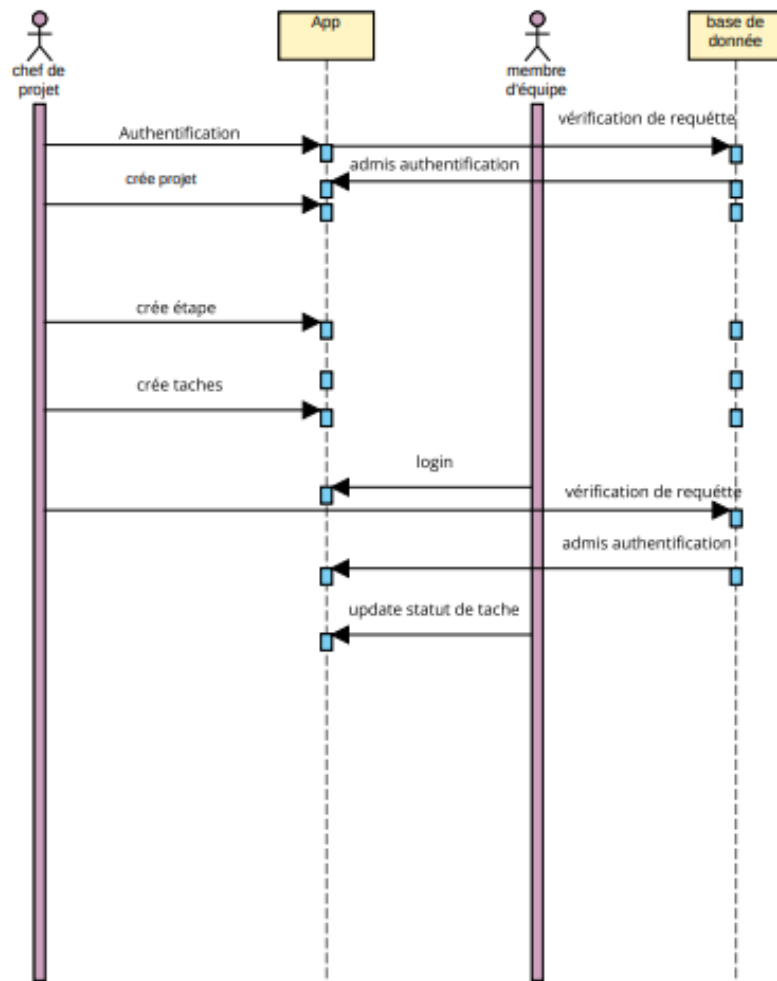


FIGURE 2.2 – Le diagramme de séquence représente le processus de création d’une tâche dans notre application de gestion de projet.”

Le diagramme de séquence montre les interactions entre les différents acteurs et composants du système lors de la création d’une tâche. Les étapes se déroulent comme suit :

- Authentification : L'utilisateur saisit ses identifiants (login et mot de passe) dans l'interface. puis L'application envoie une requête d'authentification au serveur.et Le serveur vérifie les identifiants et renvoie un jeton d'authentification si la connexion est réussie.
- Création du projet et étapes : Le chef de projet crée le projet et crée les étapes composant ce projet.
- Création de la tâche :Le chef de projet ajoute une tâche. L'application envoie une requête au serveur pour créer une nouvelle tâche.
- Affectation de la tâche :Le chef de projet sélectionne un membre d'équipe et l'affecte à la tâche. L'application envoie une requête au serveur pour affecter la tâche au membre d'équipe.
- Mise à jour du statut de la tâche : L'utilisateur met à jour le statut de la tâche (en cours, terminée). L'application envoie une requête au serveur pour mettre à jour le statut de la tâche.

### 2.3.3 Le Diagramme de classe

Un diagramme de classe est un diagramme UML qui contient des classes, des interfaces, des packages et leurs relations, et qui fournit une vue logique de tout ou partie d'un système informatique. On construit un diagramme de classes pour simplifier l'interaction des objets d'un système qu'on est en train de modéliser. Ces diagrammes expriment la structure statique d'un système en termes de classes et de relations entre eux. Une classe décrit un ensemble d'objets et une association décrit un ensemble de liens. Un diagramme de classe n'exprime rien de spécifique concernant les liens d'un objet particulier, mais il décrit, le lien potentiel entre un objet et d'autres objets.

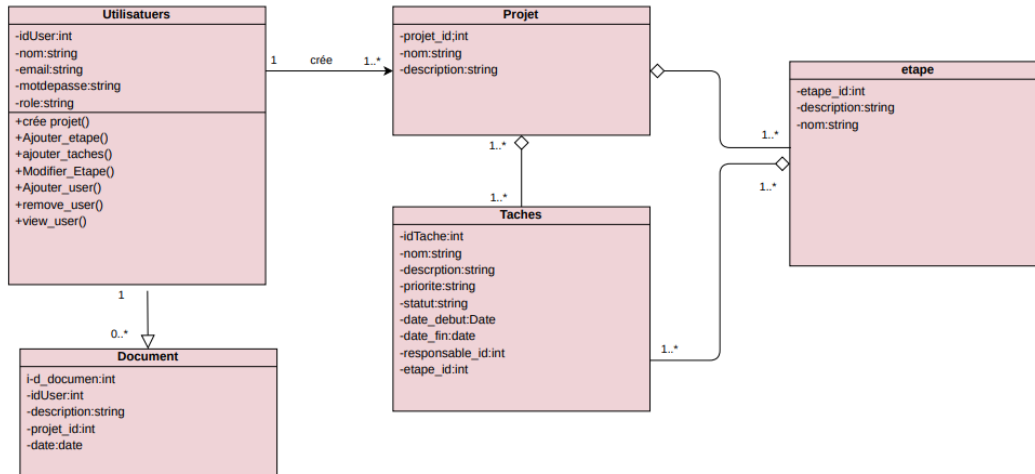


FIGURE 2.3 – décrit le diagramme de classes de notre application web.

- **Présentation des classes :**

Notre application comporte les classes suivantes :

- Utilisateur : Cette table contient des informations relatives aux utilisateurs.
- Projet : Cette table contient des informations relatives aux projets.
- Etape : Cette table contient les informations sur les différentes étapes du projet.
- Tache : Cette table contient les informations sur les différentes tâches du projet.
- Document : Cette table les documents du projet.

- **Les classes, les attributs :**

Classe: Utilisateur		
Attributs		
Nom	Type	Description
Id	int	Identifiant unique de l'utilisateur
nom	String	Nom de l'utilisateur
email	String	Adresse e-mail de l'utilisateur
motdepasse	String	Mot de passe de l'utilisateur
role	String	Rôle de l'utilisateur dans le projet (ex: chef de projet, Membre d'équipe)

TABLE 2.1 – Classe Utilisateur.

Classe: Projet		
Attributs		
Nom	Type	Description
Id	int	Identifiant unique du projet
nom	String	Nom du projet
description	String	Description du projet

TABLE 2.2 – Classe Projet.

Classe: Etape		
Attributs		
Nom	Type	Description
Id	int	Identifiant unique de l'étape
nom	String	Nom de l'étape
description	String	Description de l'étape

TABLE 2.3 – Classe Etape.

Classe: Tache		
Attributs		
Nom	Type	Description
Id	int	Identifiant unique de la tâche
nom	String	Nom de la tâche
description	String	Description de la tâche
priorite	énuméré	Priorité de la tâche
statut	énuméré	Statut de la tâche
Responsable_id	int	id Utilisateur responsable de la tâche
dateDebut	Date	Date de début de la tâche
dateFin	Date	Date de fin de la tâche
Etape_id	int	Identifiant unique de l'étape du projet à laquelle appartient la tâche

TABLE 2.4 – Classe Tache.

Classe: Document		
Attributs		
Nom	Type	Description
Id	int	Identifiant unique du document.
iduser	id	Identifiant unique de l'utilisateur qui a créé ou téléchargé le document.
description	String	Description du document. Cela pourrait inclure le titre.
idprojet	id	Identifiant unique du projet auquel le document est associé.
Date	Date	Date de création du document

TABLE 2.5 – Classe Document.

## 2.4 Les outils utilisés

### 2.4.1 React

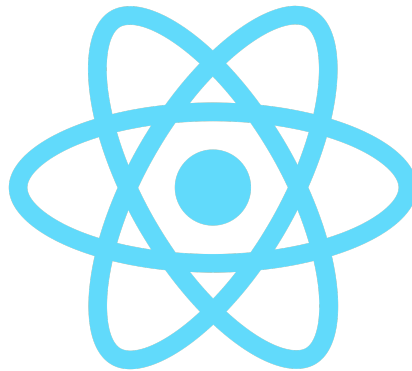


FIGURE 2.4 – Logo de React

React, également connu sous le nom de React.js ou ReactJS, est une bibliothèque JavaScript open source. Lancée en 2013, elle est maintenue par Meta (anciennement Facebook) ainsi que par une communauté de développeurs individuels et d'entreprises.

Son objectif principal est de simplifier la création d'applications web monopages en permettant la création de composants qui dépendent d'un état et génèrent une portion de page HTML à chaque modification de cet état.

React se concentre exclusivement sur l'interface de l'application, considérée comme la vue dans le modèle MVC (Modèle-Vue-Contrôleur). Elle peut être intégrée avec d'autres bibliothèques ou frameworks MVC tels qu'AngularJS. Ce qui la distingue de ses concurrents, c'est sa flexibilité et ses performances. Elle utilise un DOM virtuel et ne met à jour le rendu dans le navigateur que lorsque nécessaire, ce qui améliore l'efficacité et les performances de l'application.



### 2.4.2 Node.js



FIGURE 2.5 – Logo de Nodejs

Node.js est une plateforme logicielle en JavaScript, open source, conçue pour les applications réseau événementielles hautement concurrentes nécessitant une montée en charge efficace. Elle tire parti de la machine virtuelle V8 et de la bibliothèque libuv pour la gestion des événements, et elle met en œuvre les spécifications CommonJS sous licence MIT.

Parmi ses modules natifs, on compte notamment http, qui facilite le développement de serveurs HTTP. Ainsi, lors du déploiement de sites web et d'applications développés avec Node.js, il est possible de se passer de serveurs web comme Nginx ou Apache.

En somme, Node.js offre un environnement côté serveur bas niveau pour exécuter du JavaScript.

### 2.4.3 MySQL



FIGURE 2.6 – Logo de MySQL

Comme son nom l'indique, MySQL est un système de gestion de base de données (ou SGBD) relationnelles basé sur SQL. Ce logiciel open source est aujourd'hui géré par Oracle, l'entreprise qui a également développé le langage de programmation Java.

MySQL stocke, gère et affiche des données dans des tableaux en fonctionnant comme un système client-serveur. Si l'on considère que la base de données fait office de serveur, sur lequel sont stockées toutes les informations pertinentes, le logiciel peut alors se voir comme un client. Il permet aux utilisateurs de la base de données relationnelle de formuler différentes requêtes (en anglais queries) pour les envoyer au système de base de données dans le langage de requête SQL. Celles-ci sont ensuite traitées par MySQL, raison pour laquelle l'accès aux données est également un élément important de MySQL.

MySQL se caractérise par sa grande indépendance vis-à-vis des plateformes. Il est ainsi possible d'utiliser MySQL sur plus de 20 plateformes différentes, parmi lesquelles les systèmes d'exploitation les plus courants comme Windows, macOS et Linux. De plus, chacun peut installer MySQL facilement.

## 2.4.4 XAMPP



FIGURE 2.7 – Logo de Xampp

Le noyau principal de XAMPP est son option Apache, qui crée un serveur Apache sur votre PC. Cette fonctionnalité est essentielle car elle vous permet d'exécuter et d'éditer des sites Web directement sur votre ordinateur à l'aide de navigateurs Web, tout comme vous le feriez pour un site Web accessible en ligne. Cependant, dans ce cas, le site que vous développez fonctionne localement, c'est-à-dire sur votre propre PC, et n'est pas accessible via Internet.

En d'autres termes, le serveur Apache fourni par XAMPP agit comme un environnement de développement Web local, vous permettant de tester et de travailler sur des sites Web avant de les déployer sur un serveur en ligne. Cela offre un moyen pratique et efficace de développer des sites Web sans avoir besoin d'une connexion Internet constante et sans risque de perturber des sites Web en production.

### 2.4.5 Visual Studio

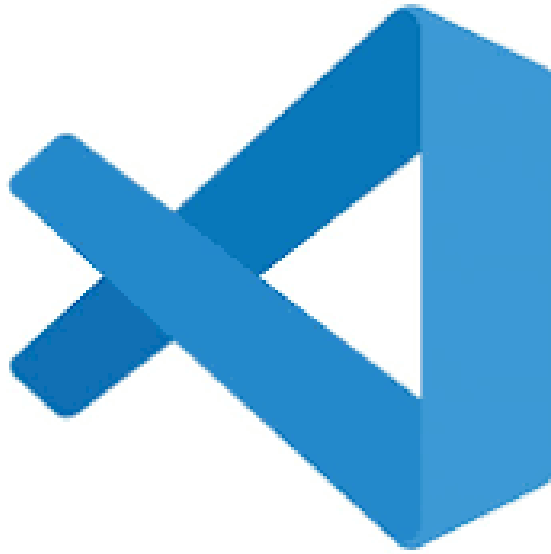


FIGURE 2.8 – Logo de Visual Studio

Visual Studio Code est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l'édition, à la construction, jusqu'au débogage.

### 2.4.6 Github



FIGURE 2.9 – Logo de Github

GitHub est une plateforme open source de gestion de versions et de collaboration destinée aux développeurs de logiciels. Livrée en tant que logiciel à la demande (SaaS, Software as a Service), la solution GitHub a été lancée en 2008. Elle repose sur Git, un système de gestion de code open source

créé par Linus Torvalds dans le but d'accélérer le développement logiciel.

Git permet de stocker le code source d'un projet et de suivre l'historique complet de toutes les modifications apportées à ce code. Grâce aux outils qu'elle fournit pour gérer les conflits éventuels résultant des changements apportés par plusieurs développeurs, il est possible de collaborer efficacement sur un même projet. GitHub facilite la programmation collaborative en mettant une interface Web à disposition du référentiel de code de Git, ainsi que des outils d'administration favorisant la collaboration.

### 2.4.7 Latex



FIGURE 2.10 – Logo de Latex

LaTeX est un langage et un système de composition de documents. Il s'agit d'une collection de macrocommandes destinées à faciliter l'utilisation du ■ processeur de texte ■ TeX de Donald Knuth.

LaTeX permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de LaTeX est son mode mathématique, qui permet de composer des formules complexes.

LaTeX est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne (tels que des articles) ou importante (thèses ou livres, par exemple). Néanmoins, il peut être employé pour générer des documents de types très variés (lettres ou transparents, par exemple). Enfin, de nombreux sites Internet — dont le texte est typiquement mis en forme par d'autres moyens — emploient un sous-ensemble de LaTeX pour composer notamment leurs formules mathématiques.

## 2.5 Conclusion

En résumé, ce chapitre a abordé la conception de notre application en utilisant le langage UML. Nous nous sommes concentrés sur l'utilisation de différents diagrammes de cas d'utilisation, de classe et de séquence afin de comprendre les interactions entre les acteurs et le système de l'application web. De plus, nous avons mentionné les langages de programmation et les logiciels utilisés pendant la réalisation de notre application mobile.

# Chapitre 3

## Réalisation du projet :

### 3.1 Introduction

Cette partie, qui constitue le dernier volet de ce rapport, vise à présenter le travail accompli ainsi que les différentes interfaces de l'application web et leurs fonctionnalités, afin d'en faciliter l'utilisation pour l'utilisateur.

### 3.2 Page Signup

La première interface de notre application web est la page de inscription où l'utilisateur doit saisir son email, son nom d'utilisateur, son mot de passe, et cliquer sur ■ SignUp ■ pour terminer son authentification.

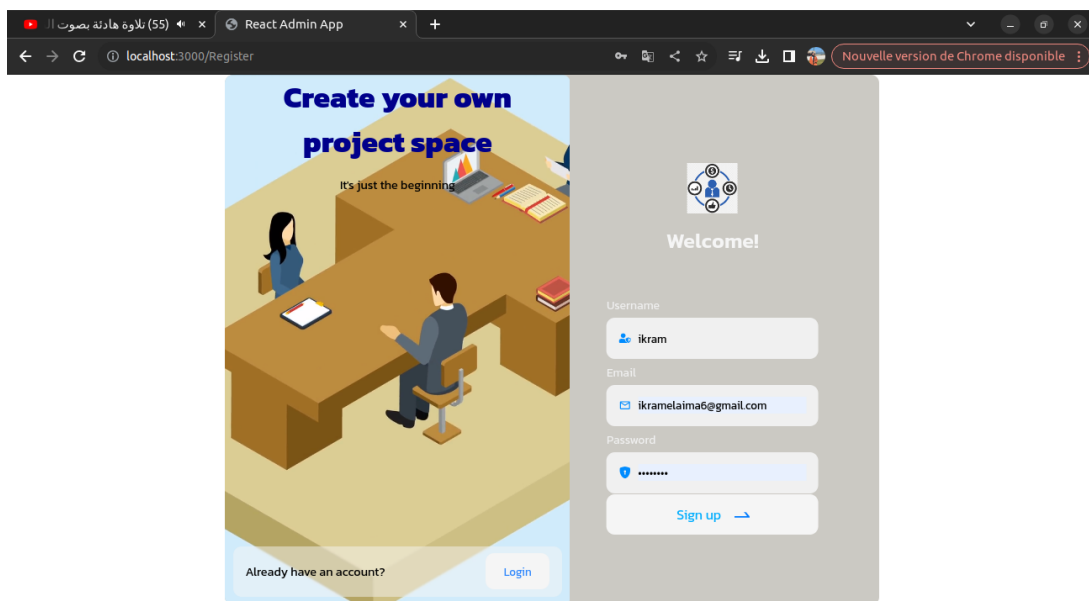


FIGURE 3.1 – Page Signup

### 3.3 Page Login

Après que l'utilisateur se soit inscrit, il est redirigé vers la page de connexion où il trouve deux champs à remplir : Email et Passeword. Une fois qu'il a saisi ses informations, il peut accéder à la page Dashboard.

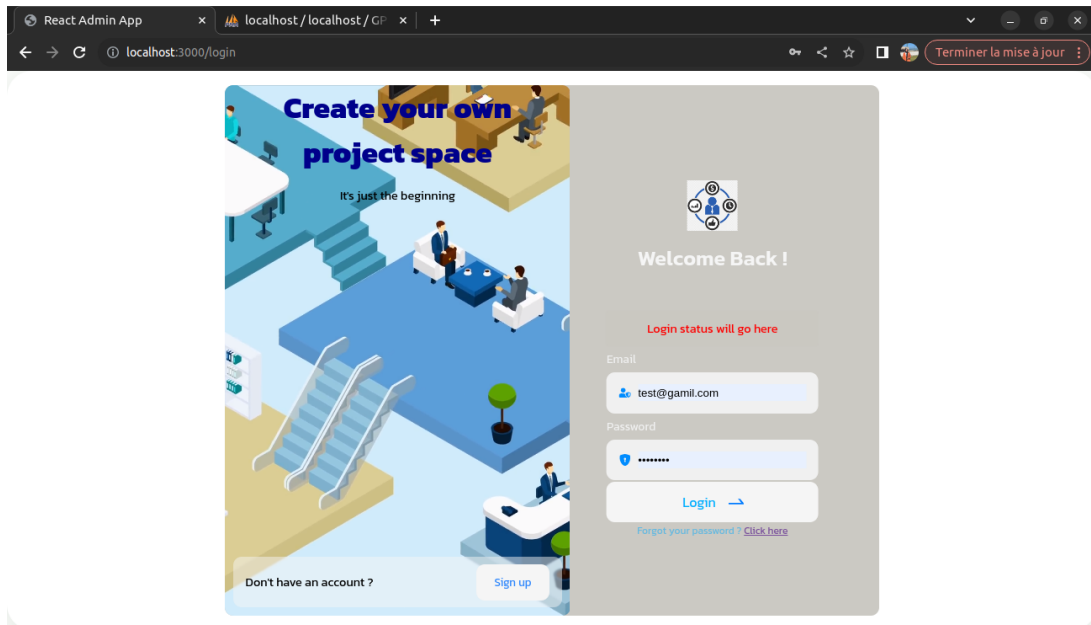


FIGURE 3.2 – Page Login

## 3.4 Interface d'administrateur

Lorsqu'un chef de projet se connecte, il accède à l'interface administrateur, qui comprend les pages suivantes :

### 3.4.1 Page dashboard

La page d'accueil ou "dashboard" est la première page affichée, et elle comprend les éléments suivants :

Une barre d'en-tête contenant le logo, la barre de recherche et un boutons pour ajouter des projets, gérer les utilisateurs et se déconnecter. des carte affichant les statistiques, notamment le nombre de projets, d'utilisateurs et de documents.

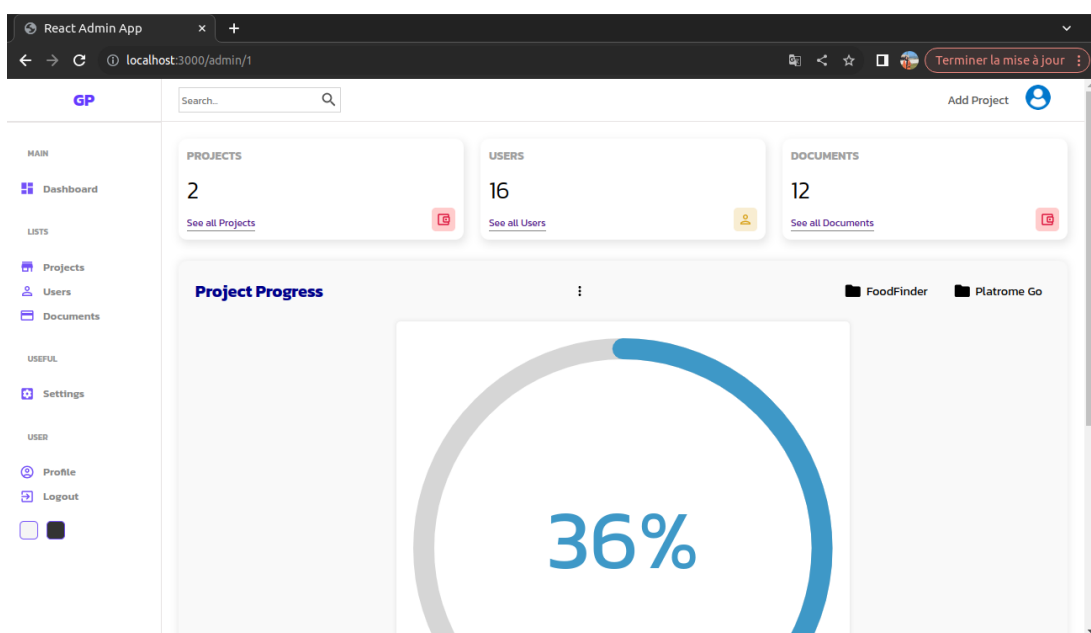


FIGURE 3.3 – Page dashboard

Une zone de contenu principale présentant un diagramme circulaire intitulé "Progression du projet", indiquant le pourcentage de chaque projet selon l'avancement des tâches. Cette zone affiche également le nombre de tâches terminées et celles en cours.

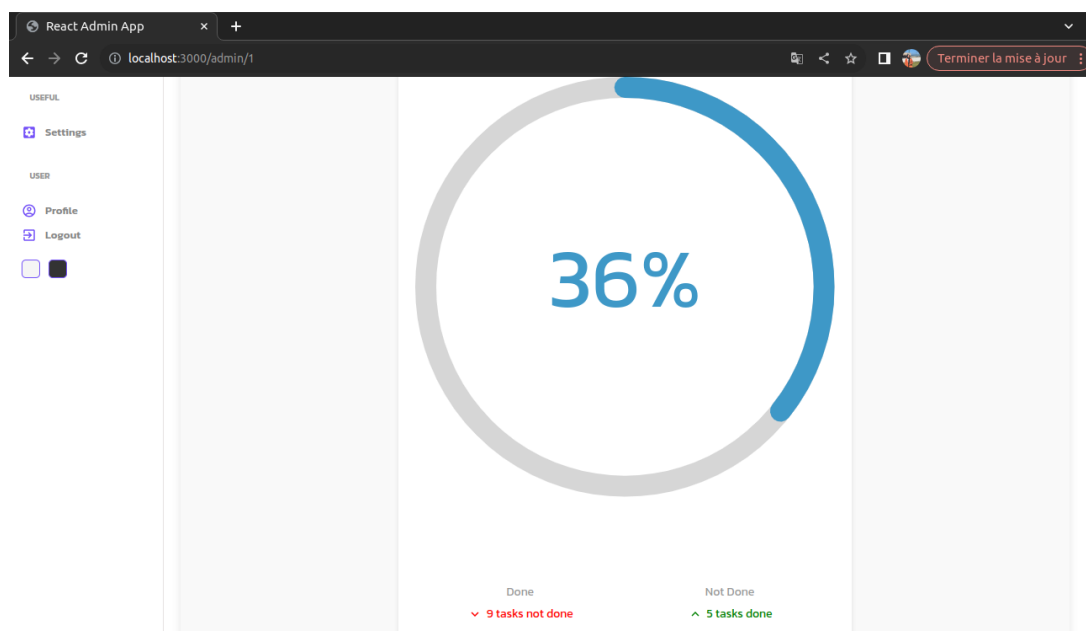


FIGURE 3.4 – Page dashboard

### 3.4.2 Page Projects

Lorsque vous cliquez sur "Projects", la page affiche une liste des projets créés. En cliquant sur un projet, vous accédez à une page qui affiche le nom et la description du projet, ainsi qu'une liste des étapes sous forme de cartes. De plus, un bouton "Add step" est disponible pour permettre l'ajout de nouvelles étapes.

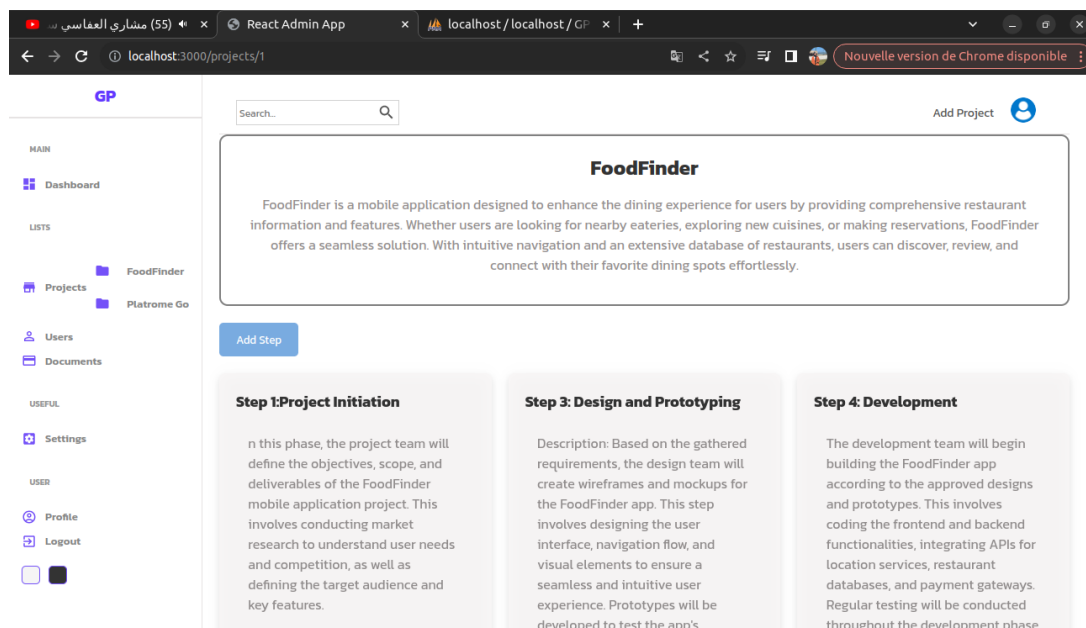


FIGURE 3.5 – Page Projects

### 3.4.3 Page Tasks

Si l'on clique sur l'une des cartes d'étape, cela redirige vers une page qui contient un tableau de tâches composé des descriptions, titres, responsables, dates de début et de fin, priorités. En bas de

la page, il y a un formulaire pour ajouter une nouvelle tâche.

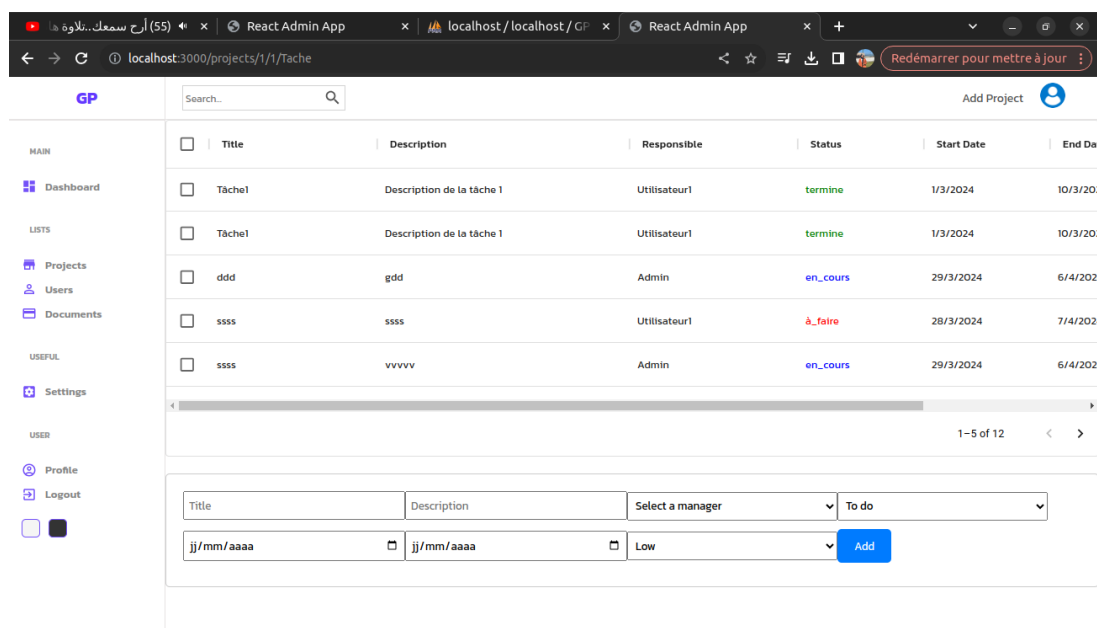


FIGURE 3.6 – Page Tasks

### 3.4.4 Page Users

Cette page contient tous les membres de l'équipe avec les actions "View" et "Delete", ainsi qu'un bouton "Add new".

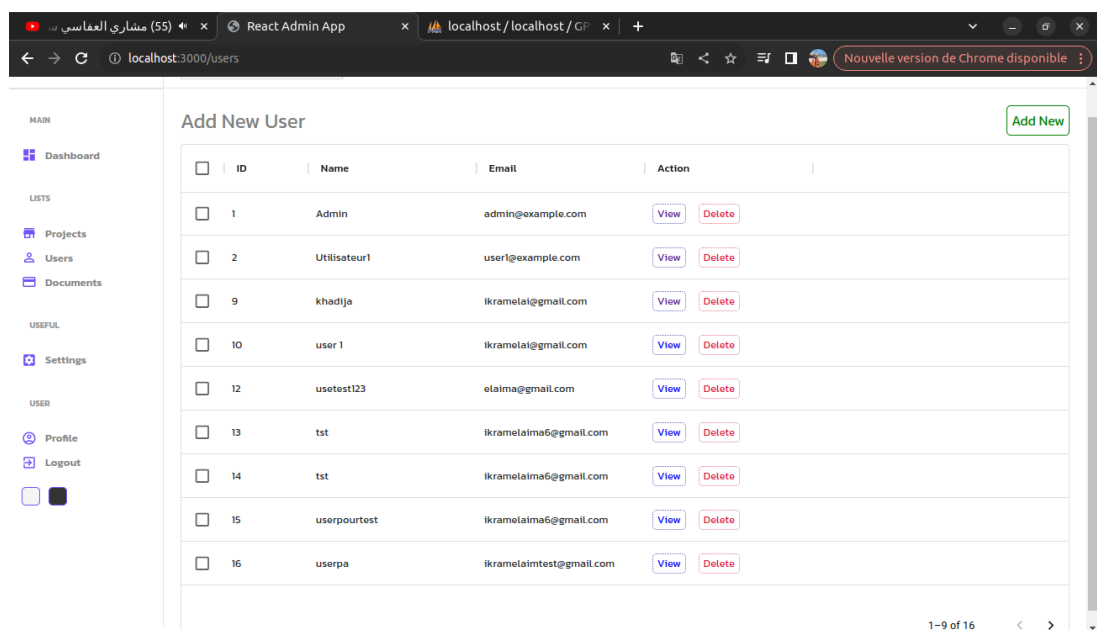


FIGURE 3.7 – Page Users

### 3.4.5 Page Documents

La page des documents affiche les documents et les rapports envoyés par les membres de l'équipe avec des détails.



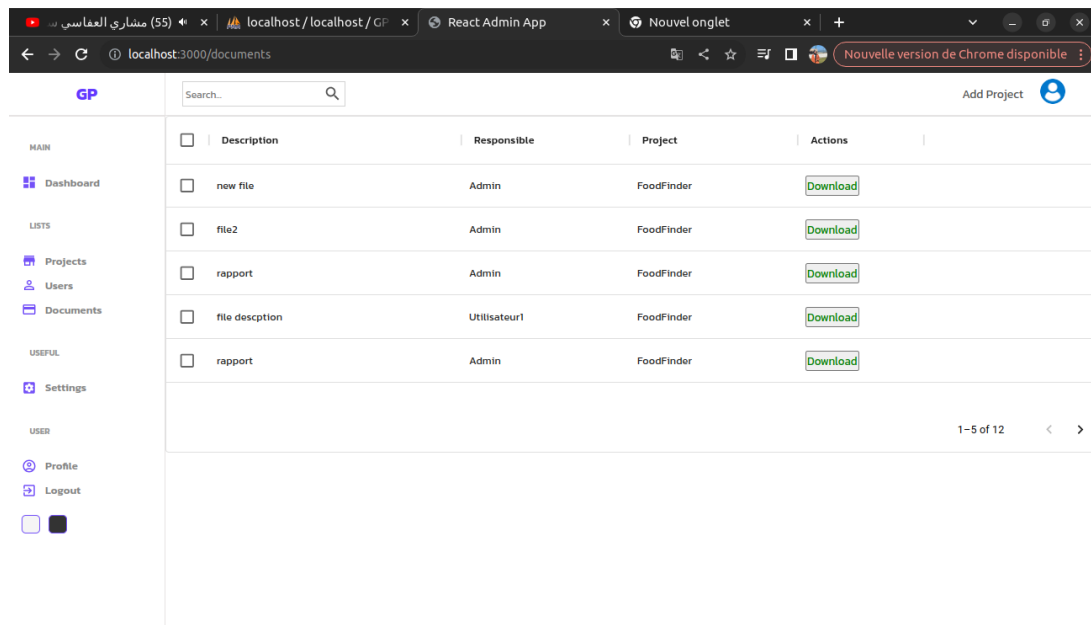


FIGURE 3.8 – Page Documents

## 3.5 Interface de membre d'équipe

Lorsqu'un membre de l'équipe de projet se connecte, il accède à l'interface utilisateur, qui comprend les pages suivantes :

### 3.5.1 Page dashboard

La page "dashboard" comprend les éléments suivants : Une barre d'en-tête contenant le logo, la barre de recherche et des cartes affichant les statistiques, notamment le nombre de tâches et de documents. Une zone de contenu principale présentant un diagramme circulaire intitulé "Progression des tâches", indiquant le pourcentage de tâches selon leur avancement. Cette zone affiche également le nombre de tâches terminées et celles en cours.

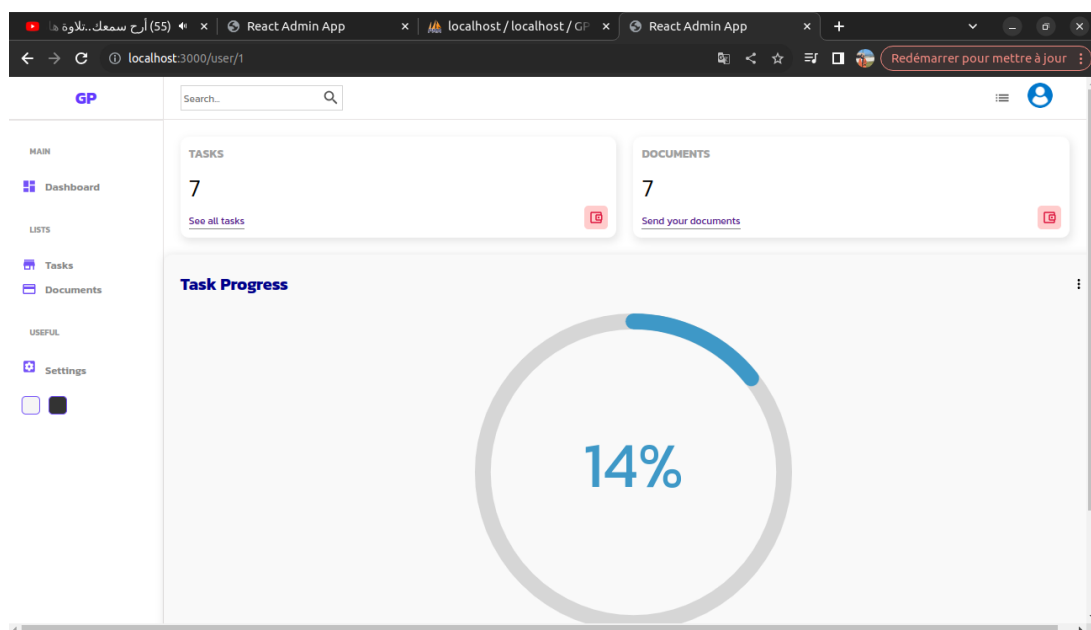


FIGURE 3.9 – Page dashboard-utilisateur

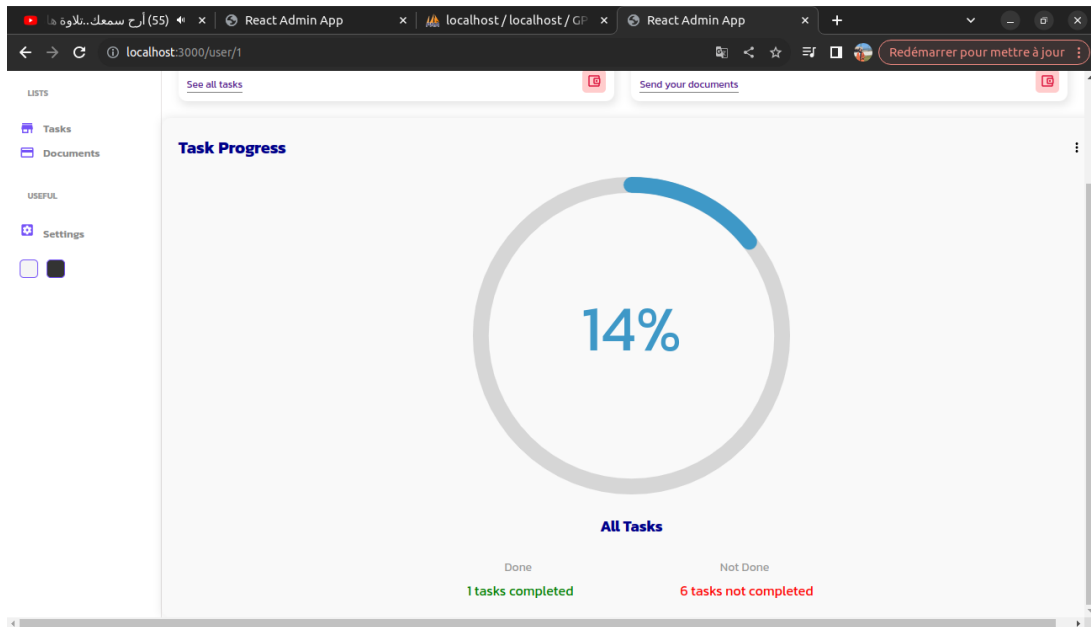


FIGURE 3.10 – Page dashboard-utilisateur

### 3.5.2 Page Tasks

La page "Tasks" de notre projet présente de manière organisée toutes les tâches assignées à l'utilisateur par l'administrateur. Chaque tâche est affichée dans un tableau clair, avec des détails tels que le titre, la description, l'état actuel et la priorité. Les utilisateurs ont la possibilité de modifier le statut des tâches une fois qu'elles sont terminées, ce qui garantit une gestion efficace et transparente des responsabilités. Les dates de début et de fin sont également affichées pour une meilleure planification et un suivi des échéances.

The screenshot displays the 'Taches List' page in the React Admin App. It features a search bar at the top and a table with the following data:

Titre	Description	Statut	Priorité	Date début	Date fin
Tâche1	Description de la tâche 1	Terminé	faible	2024-02-29T23:00:00.000Z	2024-03-09T23:00:00.000Z
Tâche2	Description de la tâche 2	En Cours	moyenne	2024-03-04T23:00:00.000Z	2024-03-14T23:00:00.000Z
Tache3	description	En Cours	moyenne	2024-02-29T23:00:00.000Z	2024-03-01T23:00:00.000Z
Tache4	description1	Terminé	elevation	2024-03-11T23:00:00.000Z	2024-03-28T23:00:00.000Z
Tache2	etape2	Terminé	elevation	2024-03-15T23:00:00.000Z	2024-03-17T23:00:00.000Z
tache3	etape2	Terminé	faible	2024-03-27T23:00:00.000Z	2024-03-29T23:00:00.000Z

FIGURE 3.11 – Page Tasks-utilisateur

### 3.5.3 Page Documents

La page "Documents" de notre projet propose une interface conviviale pour télécharger facilement des fichiers. Avec un design simple, les utilisateurs peuvent naviguer rapidement vers la zone de téléchargement et suivre leurs téléchargements en cours. Cette page vise à simplifier le processus de téléchargement de documents.

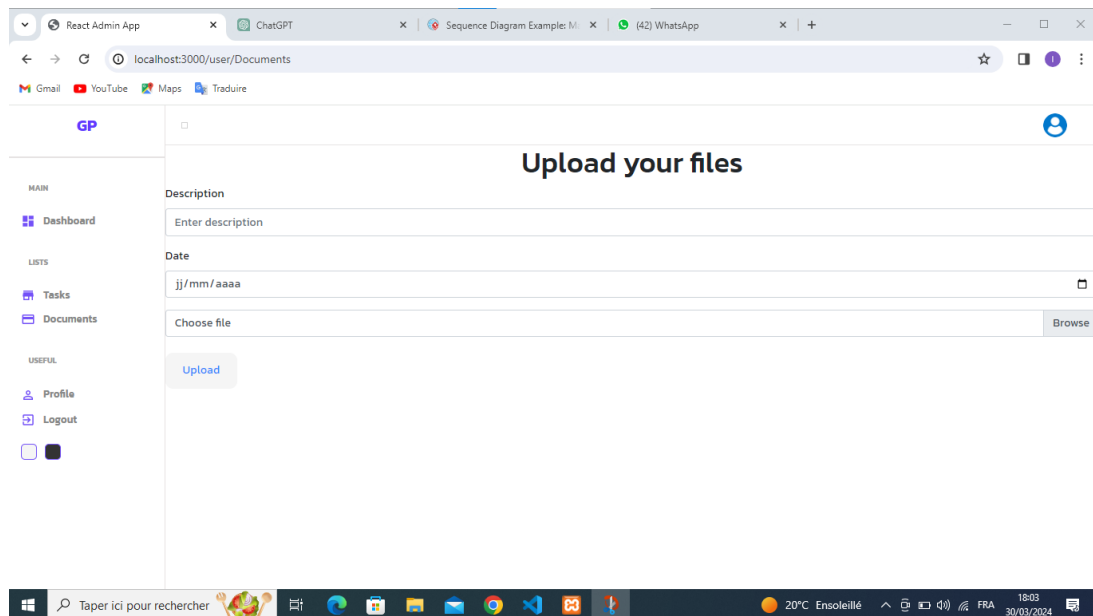


FIGURE 3.12 – Page Documents-utilisateur

## 3.6 Conclusion

En fait, dans ce chapitre, nous avons exposé les différentes interfaces du projet ainsi que nous avons fourni une description très détaillée pour faciliter l'utilisation de notre application web de gestion de projet développée avec React et Node.js. L'objectif principal de notre application est de faciliter la planification, la coordination et le suivi des tâches au sein des équipes de projet, tout en offrant une interface conviviale et intuitive pour les utilisateurs.

# Conclusion générale :

Notre projet d'application web de gestion de projet a été développé avec succès en utilisant des technologies modernes telles que React et Node.js. L'objectif principal de notre application est de simplifier la planification, la coordination et le suivi des tâches au sein des équipes de projet. Nous avons créé une interface conviviale et des fonctionnalités avancées telles que la création de projets, la gestion des tâches et la collaboration transparente entre les membres de l'équipe pour offrir une solution complète aux besoins de gestion de projet dans un environnement professionnel en constante évolution.

Nous avons utilisé des principes de conception avancés, notamment l'UML (Unified Modeling Language), pour modéliser notre système et créer des diagrammes décrivant son architecture et ses fonctionnalités. Ces diagrammes ont été cruciaux pour guider le processus de développement et assurer la cohérence de notre application.

Malgré les contraintes rencontrées, telles que les problèmes de compréhension et de maîtrise des technologies telles que Node.js et React, nous avons déployé des efforts supplémentaires pour les surmonter. Grâce à notre détermination, nous avons réussi à résoudre ces défis et à développer une application robuste et fonctionnelle.

Nous reconnaissons que la création d'une application de gestion de projet complète représente un défi de taille. Cependant, ce projet de fin d'études a été une opportunité précieuse pour acquérir de nouvelles compétences et techniques. Nous avons développé notre sens du travail en équipe et notre sens des responsabilités tout en nous initiant au développement web avec de nouvelles technologies.

En conclusion, nous sommes fiers du travail accompli et convaincus que notre application apportera une valeur ajoutée significative aux organisations en améliorant leur efficacité dans la gestion de leurs projets. Ce projet représente une étape importante dans notre parcours académique et professionnel, et nous sommes impatients de poursuivre notre développement dans ce domaine passionnant.

# Bibliographie

- [1] <https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-que-mysql/>
- [2] <https://fr.wikipedia.org/wiki/Wikip>
- [3] <https://www.lemagit.fr/definition/GitHub>
- [4] <https://online.visual-paradigm.com/fr/diagrams/features/uml-tool/>