



Institut de Formation Aux Normes et Technologies de  
l'Informatique  
-Sokodé-



## Rapport De Stage de Fin d'Études

En vue de l'obtention du diplôme

# LICENCE INFORMATIQUE

Filière : Génie Informatique

---

## Projet de développement d'une chaîne de restauration

---

Effectué à : Sirius Digital du 1er Août 2024 au 1er Février 2025

*Réalisé par :*  
M. KOUTEMA Ditoma

*Encadré par :*  
M. KPABOU Isidore

Année universitaire  
2023/2024

# Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude envers Dieu, le Tout-Puissant, pour m'avoir accordé la force, le courage et la persévérance nécessaires à la réalisation de ce travail.

Je souhaite adresser mes remerciements les plus sincères à **M. KPABOU Isidor** et **M. NGADEAU Hassan** mes tuteurs de stage, pour leurs accompagnement exceptionnel, leurs suivi rigoureux et les précieux conseils qu'il m'a prodigués tout au long de cette expérience. Son professionnalisme et sa bienveillance ont été des atouts majeurs dans l'accomplissement de ce projet.

Je tiens également à remercier chaleureusement **M. OURO BODI**, Directeur général (**Sirius Digital! (Sirius Digital!)**), pour m'avoir offert l'opportunité d'intégrer son institution et pour son soutien constant durant cette période.

Je tiens à témoigner ma reconnaissance aux membres du jury pour l'honneur qu'ils me font en acceptant d'évaluer ce travail. Leur expertise et leurs retours seront d'une grande valeur pour moi.

Enfin, je souhaite exprimer ma gratitude à l'ensemble de l'équipe pédagogique et administrative de l'**Institut de Formation aux Normes et Technologies de l'Informatique (IFNTI)** pour leur engagement et leur dévouement à offrir une formation de qualité, ainsi que pour leur soutien tout au long de ce parcours académique.

À toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail, je témoigne ici ma plus profonde reconnaissance. Leur soutien a été un pilier essentiel dans l'aboutissement de ce projet.

# **Annexe**

# Table des matières

<b>Remerciements</b>	<b>I</b>
<b>Annexe</b>	<b>II</b>
<b>Sigles, Acronymes et Abréviations</b>	<b>VIII</b>
<b>Introduction Générale</b>	<b>1</b>
<b>1 Présentation de l'organisme d'accueil Sirius Digital</b>	<b>3</b>
1.1 Historique de Sirius Digital . . . . .	4
1.1.1 Historique . . . . .	4
1.2 Les organes de SIRIUS DIGITAL . . . . .	5
<b>2 Déroulement du stage</b>	<b>6</b>
2.1 Prise de fonction . . . . .	7
2.2 Tâches effectuées . . . . .	7
2.2.1 Assistance technique générale . . . . .	8
2.3 Remarques et suggestions . . . . .	9
2.4 Bilan et conclusion . . . . .	9
<b>3 Expression des besoins</b>	<b>10</b>
3.1 introduction . . . . .	11
3.1.1 Spécifications techniques . . . . .	11
3.1.2 Exigences non fonctionnelles . . . . .	11
3.2 Définition des acteurs système . . . . .	12
3.2.1 Client . . . . .	12
3.2.2 Serveur . . . . .	12
3.2.3 Caissier . . . . .	12
3.2.4 ChefSection . . . . .	13
3.2.5 DG . . . . .	13
3.3 Définition des cas d'utilisation . . . . .	13
3.3.1 Listes des cas d'utilisation . . . . .	13
3.4 Illustration des fonctionnalités à travers les diagrammes de cas d'utilisation . . . . .	14

## TABLE DES MATIÈRES

---

3.5 En résumé . . . . .	16
<b>4 Phase de réalisation du projet</b>	<b>17</b>
4.1 Introduction . . . . .	18
4.2 Objectifs . . . . .	18
4.3 Modélisation et diagrammes . . . . .	18
4.3.1 Dictionnaire de données . . . . .	18
4.3.2 Diagramme de classes . . . . .	20
4.4 Description des entités et relations . . . . .	20
4.4.1 Entités principales . . . . .	21
4.5 Technologies et outils . . . . .	24
4.6 Conclusion . . . . .	24
<b>5 Réalisation</b>	<b>25</b>
5.1 Introduction . . . . .	26
5.2 Architecture du projet . . . . .	26
5.2.1 Présentation de l'architecture globale . . . . .	26
5.2.2 Choix des technologies et outils . . . . .	26
5.3 Développement et implémentation . . . . .	28
5.3.1 Mise en place de l'environnement de développement . . . . .	28
5.3.2 Développement des fonctionnalités principales . . . . .	31
5.4 Problèmes rencontrés et solutions adoptées . . . . .	31
5.4.1 Difficultés techniques . . . . .	31
5.4.2 Optimisation et amélioration du code . . . . .	31
5.5 Tests et validation . . . . .	31
5.5.1 Stratégie de test . . . . .	31
5.5.2 Résultats des tests . . . . .	31
5.6 Conclusion . . . . .	31
<b>6 Bilan</b>	<b>32</b>
6.1 Introduction . . . . .	33
6.2 Retour sur les objectifs initiaux . . . . .	33
6.2.1 Rappel des objectifs du projet . . . . .	33
6.2.2 Évaluation de leur atteinte . . . . .	33
6.3 Analyse des résultats obtenus . . . . .	33
6.3.1 Synthèse des résultats . . . . .	33
6.3.2 Difficultés rencontrées et solutions adoptées . . . . .	33
6.4 Compétences acquises et développées . . . . .	33
6.4.1 Compétences techniques . . . . .	33
6.4.2 Compétences organisationnelles . . . . .	33
6.4.3 Compétences personnelles . . . . .	33
6.5 Limites et axes d'amélioration . . . . .	33
6.5.1 Contraintes techniques et organisationnelles . . . . .	33

6.5.2	Améliorations possibles et perspectives d'évolution	33
6.6	Conclusion	33
<b>7</b>	<b>Conclusion Générale</b>	<b>34</b>

# Table des figures

2.1	Page d'accueil de la <b>PASSE!</b> ( <b>PASSE!</b> ) . . . . .	7
2.2	Interface de connexion de la <b>PASSE!</b> . . . . .	8
2.3	Interface de connexion de la <b>PASSE!</b> . . . . .	8
3.1	Diagramme de cas d'utilisation Gestion des Congés et Absences . . . . .	14
3.2	Diagramme de cas d'utilisation Gestion des Congés et Absences . . . . .	15
3.3	Diagramme de cas d'utilisation Gestion des Congés et Absences . . . . .	15
4.1	Diagramme de classe - Gestion du projet OptiHR . . . . .	20

# Liste des tableaux

4.2	Tableau du dictionnaire des données . . . . .	20
4.3	Tableau des technologies et outils utilisés pour la conception . . . . .	24
5.2	Tableau des technologies utilisées pour la réalisation . . . . .	28

# Sigles, Acronymes et Abréviations

<b>SIRIUS DIGITAL</b>	Autorité de Régulation de la Commande Publique
<b>IFNTI</b>	Institut de Formation aux Normes et Technologies de l'Informatique
<b>API</b>	Application Programming Interface
<b>CSS</b>	Cascading Style Sheets
<b>DBMS</b>	Database Management System
<b>HTML</b>	HyperText Markup Language
<b>JS</b>	JavaScript
<b>JSON</b>	JavaScript Object Notation
<b>PHP</b>	Hypertext Preprocessor
<b>SQL</b>	Structured Query Language

# Introduction Générale

## **contexte**

Dans le cadre de leur formation, les étudiants en troisième année de l’Institut de Formation aux Normes et Technologies de l’Informatique doivent effectuer un stage de fin d’études au sein d’une organisation avant l’obtention de leur diplôme. Cette immersion professionnelle vise à leur permettre d’acquérir des compétences pratiques et à mieux appréhender le fonctionnement du monde du travail.

C'est dans cette perspective que le stage a été réalisé au sein de SIRIUS DIGITAL. Sirius Digital est une startup créée en 2022 et a pour but de répondre aux enjeux grandissants des nouvelles technologies en proposant des solutions à travers la conception des applications web et mobile.

Le présent rapport a pour objectif de présenter les différentes missions effectuées au sein de Sirius Digital, ainsi que les réalisations techniques développées au cours du stage. Il met en lumière les compétences acquises et les défis rencontrés, tout en illustrant l’apport de cette expérience dans le cadre de la formation académique.

## **Objectifs**

L’objectif principal de ce stage de fin d’études est de permettre à l’étudiant de mettre en pratique ce qu’il aura acquis tout au long de sa formation en licence d’informatique des organisations à l’IFNTI, que ce soit en termes de compétences techniques comme de compétences humaines et professionnelles. L’étudiant devra être capable de s’adapter rapidement aux besoins de l’organisation afin de s’intégrer aux projets sur lesquels il travaillera. La prise en compte des démarches qualités et des méthodes associées au sein de l’organisation est primordiale. Enfin, ce stage doit permettre à l’étudiant d’acquérir une certaine autonomie technique.

Tout au long du stage, l’étudiant doit être accompagné d’un maître de stage ayant entre autres pour rôle de l’accompagner dans son intégration au fonctionnement de l’organisation, mais aussi de le suivre en se rendant disponible.

# **Chapitre 1**

## **Présentation de l'organisme d'accueil Sirius Digital**

## 1.1 Historique de Sirius Digital

### 1.1.1 Historique

Fondée en 2022 par OURO BODI, Sirius Digital est une startup informatique basée à Lomé, dans le quartier de Kegué, en face du complexe scolaire Dino Golo. Dès ses débuts, l'entreprise s'est donnée pour mission de concevoir des solutions numériques innovantes répondant aux défis technologiques des entreprises, qu'elles soient petites ou grandes.

Face à l'**essor!** (**essor!**) rapide des nouvelles technologies et à l'évolution croissante des besoins dans tous les secteurs, Sirius Digital s'impose progressivement comme un acteur clé dans le développement d'applications web et mobiles. Grâce à une approche axée sur l'innovation et la qualité, la startup accompagne les entreprises dans leur transformation digitale, leur offrant des outils performants et adaptés à leurs réalités.

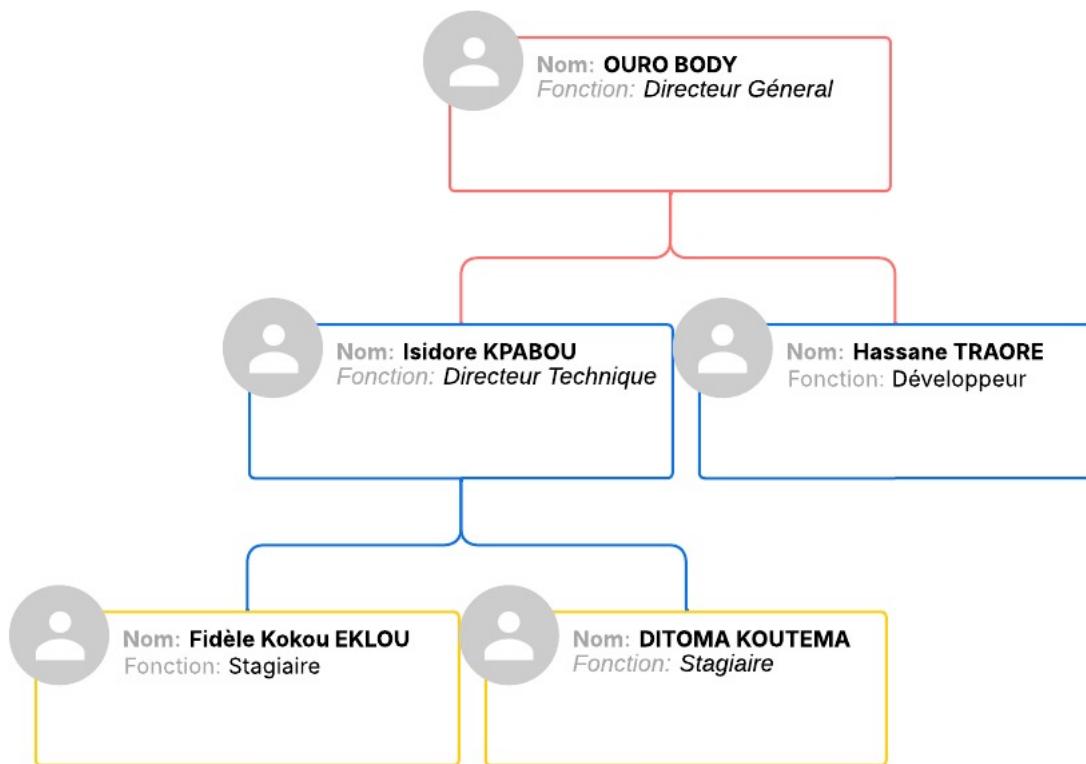
Aujourd'hui, Sirius Digital continue d'évoluer en mettant la technologie au service du développement, avec pour ambition de devenir une référence incontournable dans le domaine des solutions numériques en Afrique et au-delà.

- . **Adresse :** Kegué, Lomé-TOGO, 16BP :20
- . **Téléphone :** (+228) 91 09 16 56 / 22 61 78 84
- . **E-mail :** siriusdigital@app-siriusdigital.com
- . **Site web :** app-siriusdigital.com

## CHAPITRE 1. PRÉSENTATION DE L'ORGANISME D'ACCEUIL SIRIUS DIGITAL

---

### 1.2 Les organes de SIRIUS DIGITAL



## **Chapitre 2**

### **Déroulement du stage**

## CHAPITRE 2. DÉROULEMENT DU STAGE

---

### 2.1 Prise de fonction

J'avais commencé officiellement le stage le 1er Août 2024 dans les locaux de la Société Sirius Digital. J'ai bien été accueillis dans le cadre du stage. Le stage commence tous les jours de 7h à 12h et de 14h à 17h de lundi à Vendredi marquée par une pose de 12h à 14h ou l'on est libre de ses actions. L'entreprise m'a été présenté par un des membres de l'équipe déjà en place suivit d'une explication de DG qui, avait bien compris ma mission au sein de l'entreprise et m'expliquer les choses étapes par étapes pour l'avancement du projet.

- . Découvert l'environnement de travail et les outils utilisés.
- . Échangé avec les membres de l'équipe et le DG pour mieux comprendre les besoins et le cadre du stage.

### 2.2 Tâches effectuées

Pendant ce stage, j'ai eu à effectuer plusieurs tâches notamment

- . Modélisation des différents diagrammes
- . Développement Web
- . Rédaction du support technique
- . Rédaction du manuel d'utilisation

#### Illustration

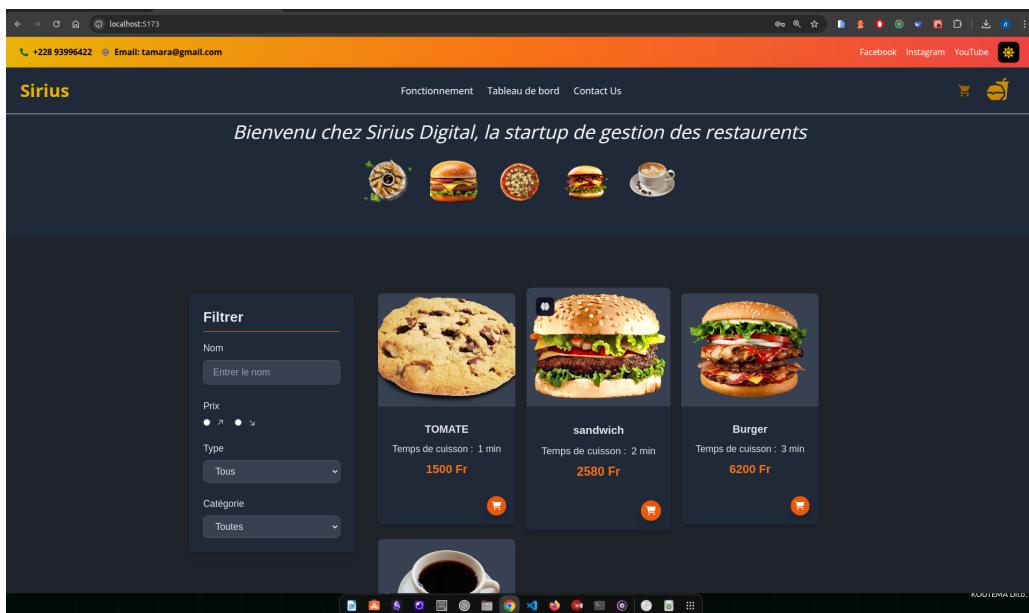


FIGURE 2.1 – Page d'accueil de la PASSE!

## CHAPITRE 2. DÉROULEMENT DU STAGE

---

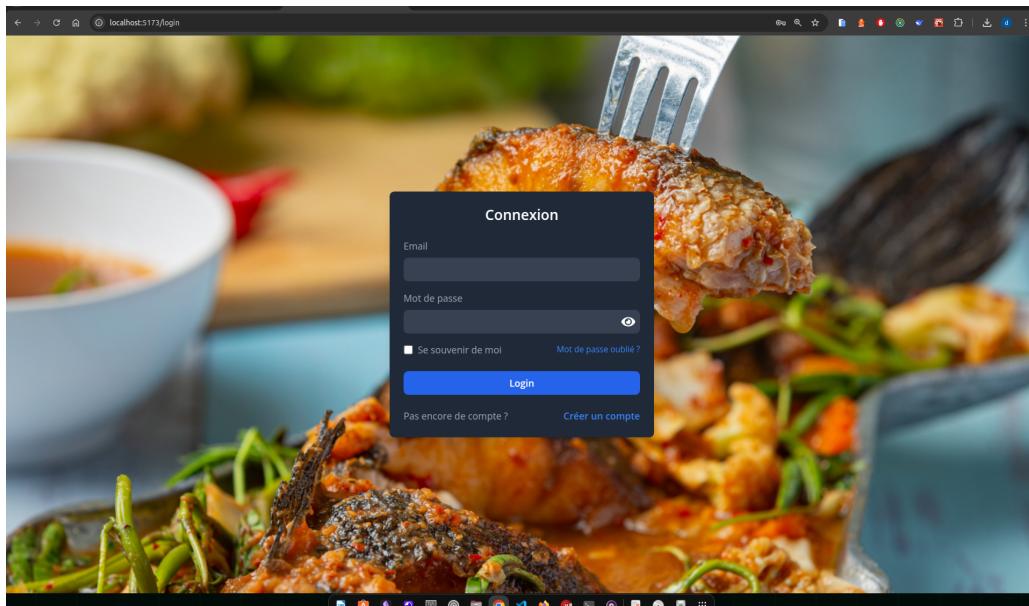


FIGURE 2.2 – Interface de connexion de la PASSE!

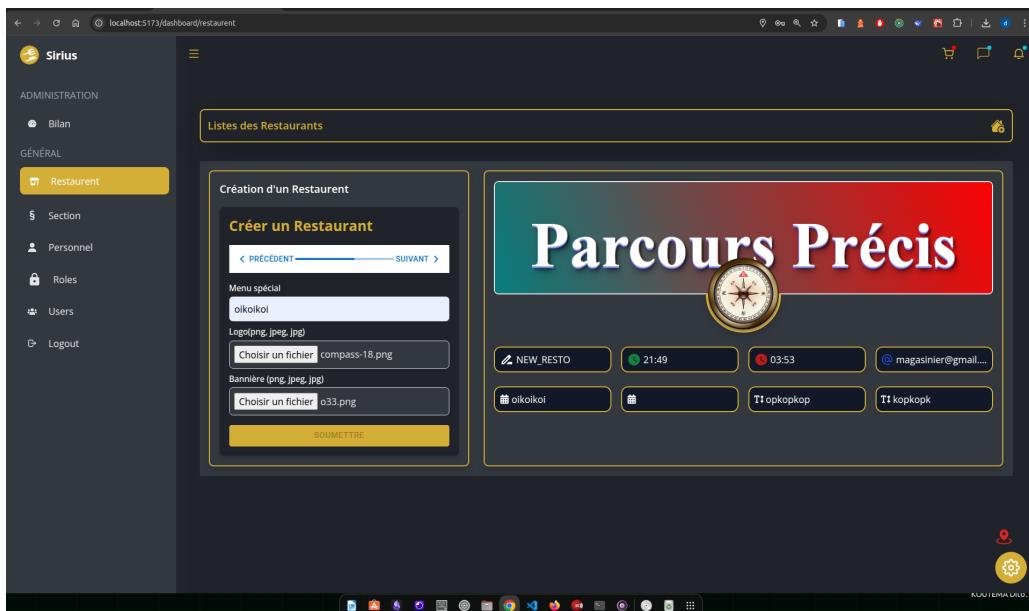


FIGURE 2.3 – Interface de connexion de la PASSE!

### 2.2.1 Assistance technique générale

En complément des missions spécifiques, un support technique global a été assuré au sein de l'organisation :

- . Dépannage et maintenance des outils informatiques utilisés par le personnel.

## **CHAPITRE 2. DÉROULEMENT DU STAGE**

---

- . Configuration et optimisation des postes de travail selon les besoins des employés.
- . Sensibilisation aux bonnes pratiques pour garantir la pérennité des équipements et la sécurité des données.

### **2.3 Remarques et suggestions**

### **2.4 Bilan et conclusion**

Cette expérience a permis de mieux appréhender les enjeux liés à la gestion d'un service informatique, alliant assistance aux utilisateurs, maintenance technique et développement d'outils numériques.

# **Chapitre 3**

## **Expression des besoins**

## 3.1 introduction

Ce projet vise à créer une marketplace dédiée à la restauration, où chaque utilisateur aura la possibilité de lancer et de gérer son propre restaurant en ligne. Grâce à une interface intuitive, il pourra personnaliser son espace, gérer son menu, suivre les commandes en temps réel et administrer son personnel selon ses besoins.

L'objectif est d'offrir une solution flexible et évolutive qui permettra aux restaurateurs, qu'ils soient indépendants ou en réseau, d'optimiser leur gestion et d'atteindre une clientèle plus large. Cette plateforme mettra également à disposition des outils de suivi des performances, des options de paiement sécurisées et des fonctionnalités facilitant l'interaction avec les clients.

En intégrant des fonctionnalités modernes et une expérience utilisateur fluide, cette marketplace ambitionne de révolutionner la manière dont les restaurants gèrent leurs activités en ligne.

### 3.1.1 Spécifications techniques

#### Langages et Technologies

- . **Framework** : Laravel (passport) pour une gestion efficace de la logique serveur et du backend (développement d'api).
- . **Base de données** : MySQL ou PostgreSQL pour la gestion des données avec Eloquent ORM.
- . **Frontend** : React js pour une interface utilisateur dynamique et fluide.

#### Sécurité

- . Chiffrement des données sensibles.
- . Gestion des accès par rôles.

#### Hébergement

- . Déploiement sur les serveurs internes de Sirius Digital.

### 3.1.2 Exigences non fonctionnelles

#### Utilisabilité

- . L'interface utilisateur doit être intuitive et facile à utiliser, avec une courbe d'apprentissage minimale.
- . Une documentation utilisateur complète doit être fournie, incluant des guides et des tutoriels.

### Compatibilité

- . **Navigateurs** : Le logiciel doit être compatible avec les dernières versions de Chrome, Firefox, Safari, et Edge.
- . **Mobile** : L'application web doit être responsive et accessible depuis les navigateurs mobiles.

### Scalabilité

Le logiciel doit pouvoir gérer un nombre croissant d'utilisateurs et de données sans dégradation des performances.

### Nom du logiciel

Le logiciel portera donc le nom S

## 3.2 Définition des acteurs système

Le logiciel est destiné à tous les utilisateurs souhaitant créer et gérer leur propre restaurant.

- . Client
- . Serveur
- . Caissier
- . ChefSection
- . DG

### 3.2.1 Client

L'utilisateur **Client** est simplement un utilisateur quelconque qui peut utiliser le logiciel pour commander un plat. De même après s'être inscrit, a aussi le choix de créer et gérer son propre restaurent.

### 3.2.2 Serveur

Le **Serveur** est un personnel appartenant à un restaurent donné qui inter-agit avec le système pour passer des commandes en interne pour les client sur place.

### 3.2.3 Caissier

Le **Casier** est un personnel appartenant à un restaurent donné qui enregistrer les factures et gères beaucoup d'autres choses.

### 3.2.4 ChefSection

Le **DG** est un personnel appartenant à un restaurant donné, il est en quelque sorte le sous directeur du restaurant auquel il appartient. De ce fait il a beaucoup de responsabilité.

### 3.2.5 DG

Le **DG** est le propriétaire du restaurant, de ce fait il a le contrôle absolu du système de gestion du restaurant à travers l'application.

## 3.3 Définition des cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique des interactions entre les acteurs d'un système et ses fonctionnalités. Il permet d'identifier et de structurer les besoins fonctionnels du logiciel en décrivant les actions réalisées par chaque acteur. Ce type de diagramme est particulièrement utile pour comprendre les interactions des utilisateurs avec le système et pour définir clairement les responsabilités de chaque acteur.

Dans ce contexte, nous définissons les cas d'utilisation pour chaque fonctionnalités énoncés précédemment.

### 3.3.1 Listes des cas d'utilisation

#### Description de chaque fonctionnalité

Cette application en son sein regroupe beaucoup de fonctionnalités telles que :

#### Gestion de restaurant

- . **Créer un restaurant** : Une fois sur l'application tout utilisateur doit visiter le guide d'utilisation de l'application et, créer autant de restaurant comme il le souhaite et gérer son personnel.
- . **Créer une section** : Ici l'utilisateur pour passer en mode DG du restaurant doit payer son abonnement et en suite, il pourra donc : Créer des sections pour chaque restaurant, gérer son personnel en fonction de leurs rôles.
- . **Gestion des rôles** : En tant que DG de son restaurant, lui seul a le droit de gérer les rôles et permissions de ses employés à sa guise.

#### Gestion Séparée

- . **Rôle ChefSection** : On est conscient que pour un restaurant pratiquement impossible de gérer sa seule. Bien que toutes les actions que font les autres employés, le DG est au courant, il se doit de partager les rôles. Ainsi le chefSection gère les fonctionnalités suivantes :

Gestion Catégorie

Gestion Promotion

Gestion Plat

Gestion Réservation

Gestion des Tables

Gestion des Commandes

Gestion des Événement

- . **Rôle Serveur :** Le serveur agit sur le système pour passer des commandes pour des clients en internes

### 3.4 Illustration des fonctionnalités à travers les diagrammes de cas d'utilisation

Diagramme de cas d'utilisation par acteur du système

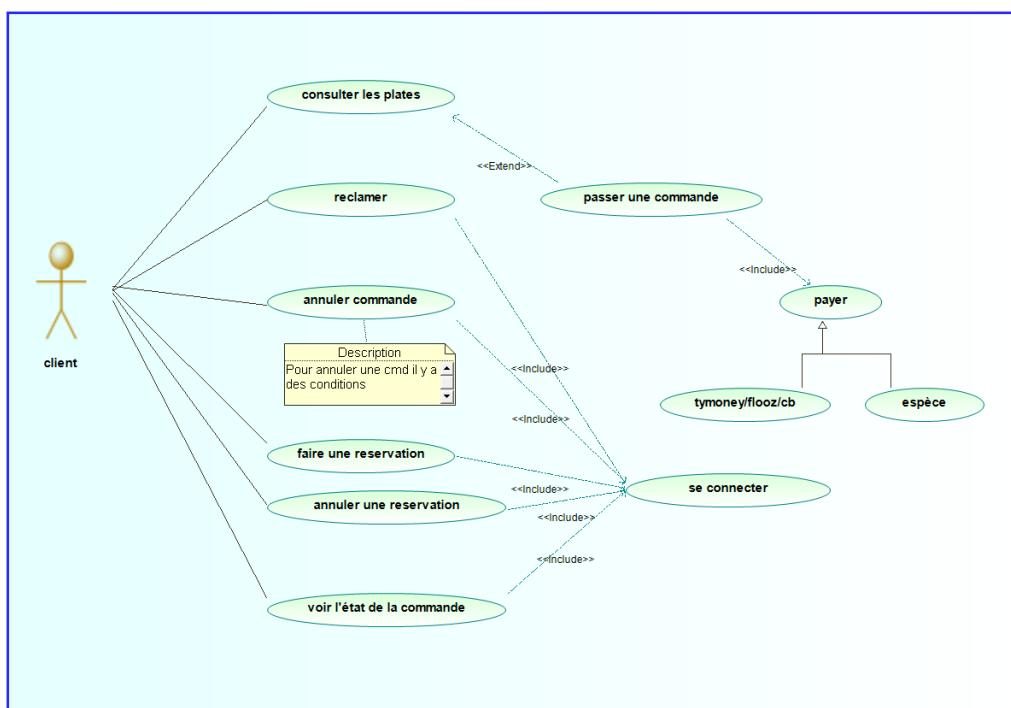


FIGURE 3.1 – Diagramme de cas d'utilisation Gestion des Congés et Absences

## CHAPITRE 3. EXPRESSION DES BESOINS

---

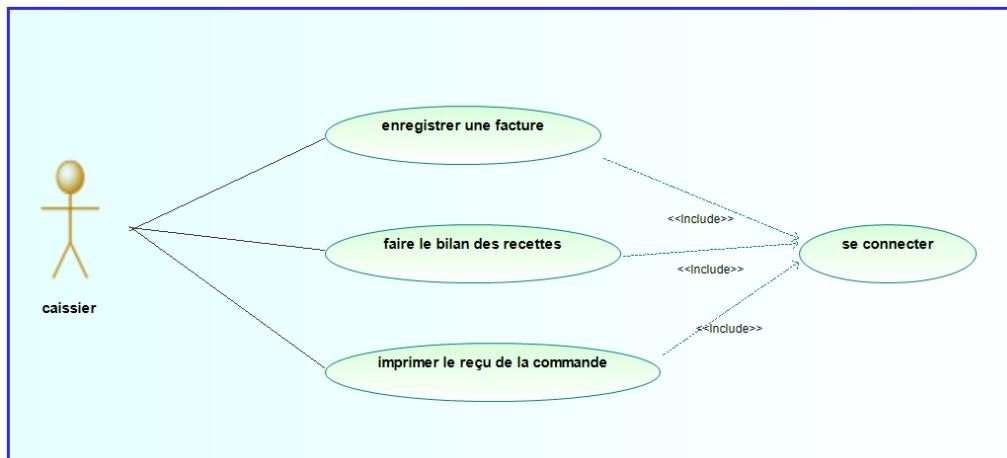


FIGURE 3.2 – Diagramme de cas d'utilisation Gestions des Congés et Absences

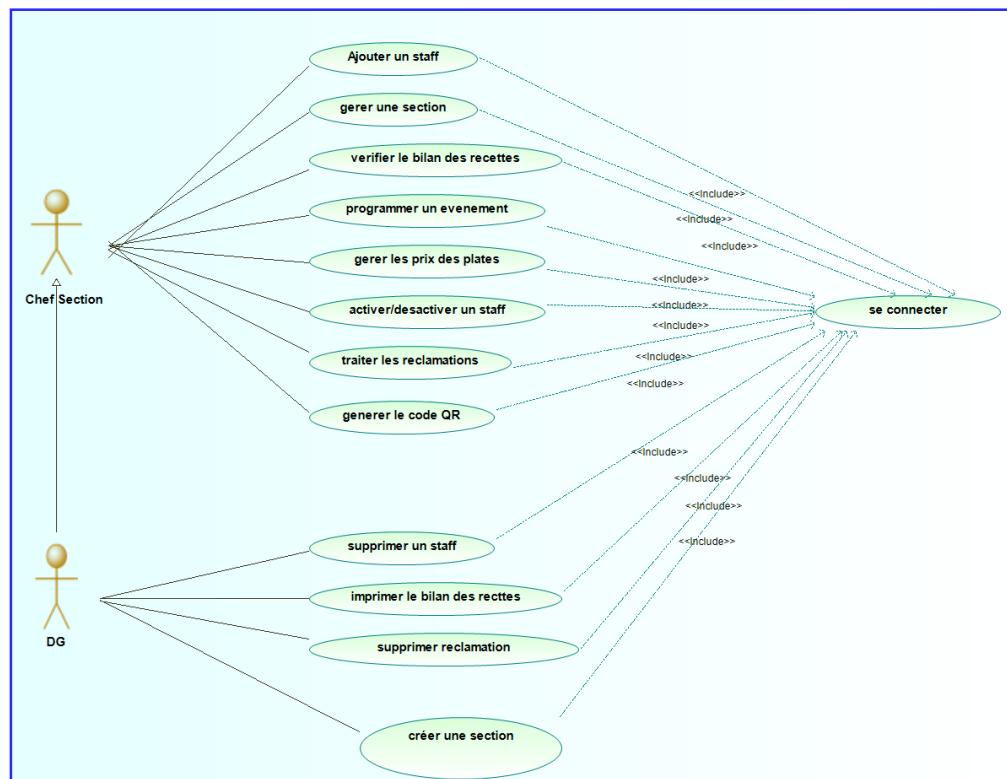


FIGURE 3.3 – Diagramme de cas d'utilisation Gestions des Congés et Absences

### **3.5 En résumé**

Le cahier des charges définit les fonctionnalités et les exigences techniques de ce logiciel faisant office d'un marché de restauration en ligne pour tous.

## **Chapitre 4**

### **Phase de réalisation du projet**

## 4.1 Introduction

Dans le cycle de développement d'un projet, la conception joue un rôle fondamental. Elle sert à établir l'architecture globale, à définir les principaux composants et à organiser les interactions entre les modules du système. Cette étape implique aussi l'identification des acteurs du système, qu'il s'agisse des utilisateurs ou des entités avec lesquelles il interagit, afin d'adapter les fonctionnalités aux besoins concrets. Une conception bien pensée assure une organisation efficace du projet, tout en simplifiant sa mise en œuvre et sa maintenance.

## 4.2 Objectifs

Les objectifs de la phase de conception sont les suivants :

- . Identifier et structurer les différents composants du système.
- . Définir l'architecture logicielle en fonction des besoins du projet.
- . Spécifier les interactions entre les modules pour assurer une cohérence globale.
- . Concevoir les diagrammes UML, y compris les diagrammes de classes et de processus, afin de modéliser clairement la structure et le fonctionnement du système.
- . Optimiser la conception en tenant compte des performances et de l'évolutivité.
- . Assurer la conformité aux standards et aux bonnes pratiques du développement logiciel.

## 4.3 Modélisation et diagrammes

### 4.3.1 Dictionnaire de données

Le dictionnaire de données décrit en détail les attributs des entités de la base de données, y compris leurs types, leurs rôles et leurs relations. Il permet d'assurer la cohérence et la structuration correcte des informations stockées.

Nom de la table	Nom de l'attribut	Type de données	Description
User	username	string	Nom d'utilisateur
User	password	string	Mot de passe
User	active	string	Statut actif/inactif
Role	name	string	Nom du rôle
Permission	name	string	Nom de la permission
Employee	first_name	string	Prénom de l'employé
Employee	last_name	string	Nom de famille de l'employé

## CHAPITRE 4. PHASE DE RÉALISATION DU PROJET

---

<b>Nom de la table</b>	<b>Nom de l'attribut</b>	<b>Type de données</b>	<b>Description</b>
Employee	phone_number	string	Numéro de téléphone
Employee	email	string	Adresse email
Employee	address1	string	Adresse principale
Employee	address2	string	Adresse secondaire
Employee	city	string	Ville
Employee	country	string	Pays
Employee	state	string	Région ou état
Employee	bank_name	string	Nom de la banque
Employee	rib	string	Relevé d'identité bancaire
Department	name	string	Nom du département
File	name	string	Nom du fichier
File	url	string	URL du fichier
File	mime_type	string	Type MIME
File	path	string	Chemin d'accès
File	upload_date	string	Date d'upload
Duty	duration	string	Durée de la mission
Duty	begin_date	string	Date de début
Duty	type	string	Type de mission
Duty	etat	string	État de la mission
Job	title	string	Intitulé du poste
Absence	day_requested	string	Jour demandé
Absence	start_date	string	Date de début
Absence	end_date	string	Date de fin
Absence	address	string	Adresse
Absence	date_of_application	string	Date de la demande
Absence	status	string	Statut de l'absence
Absence	date_of_approval	string	Date d'approbation
Absence	type_of_absence	string	Type d'absence
Absence	reasons	string	Raisons de l'absence
Absence	proof	string	Justificatif

## CHAPITRE 4. PHASE DE RÉALISATION DU PROJET

---

Nom de la table	Nom de l'attribut	Type de données	Description
Absence	comment	string	Commentaire

TABLE 4.2 – Tableau du dictionnaire des données

### 4.3.2 Diagramme de classes

Le diagramme de classes permet de représenter les différentes entités du système et leurs relations. Il offre une vision claire de la structure et facilite la compréhension des interactions entre les objets.

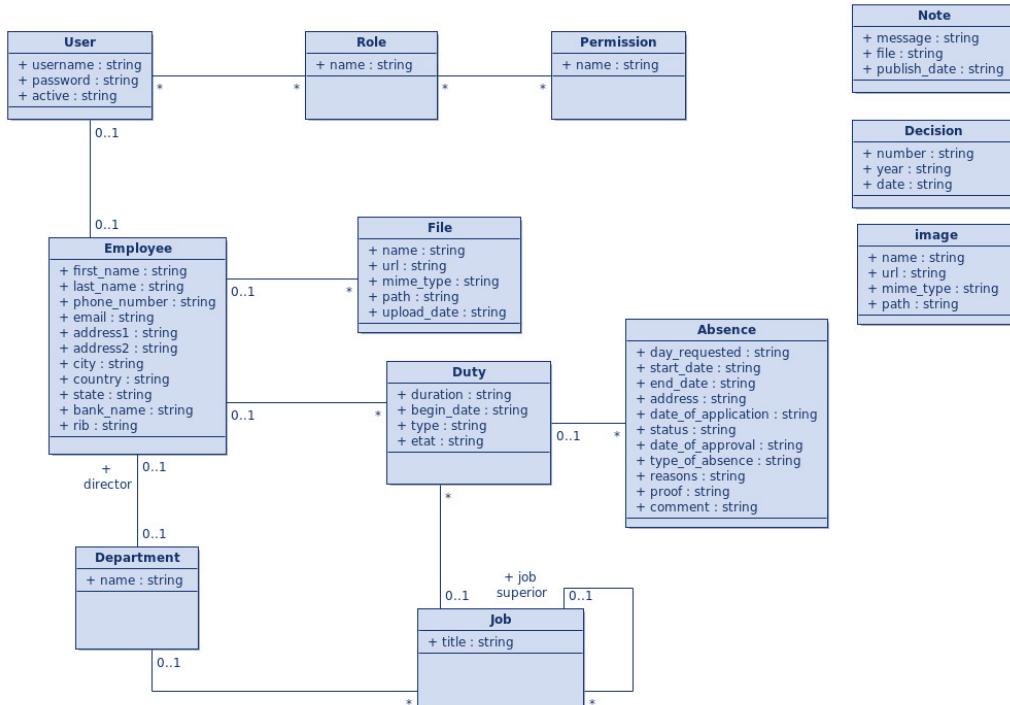


FIGURE 4.1 – Diagramme de classe - Gestion du projet OptiHR

## 4.4 Description des entités et relations

Le diagramme de classes représente un système de gestion des employés et de leurs activités professionnelles. Voici une description des entités principales et de leurs relations :

#### 4.4.1 Entités principales

##### User (Utilisateur)

**Attributs :**

- . username
- . password
- . active

**Relations :** Un utilisateur est associé à un employé.

##### Employee (Employé)

**Attributs :**

- . first\_name
- . last\_name
- . phone\_number
- . email
- . address1
- . address2
- . city
- . country
- . state
- . bank\_name
- . rib

**Relations :**

- . Un employé peut être un directeur d'un département.
- . Un employé appartient à un seul département.
- . Un employé peut avoir plusieurs fichiers associés.
- . Un employé peut être lié à plusieurs absences.

##### Department (Département)

**Attributs :**

- . name

**Relations :**

- . Un département peut avoir un seul directeur (Employee).
- . Un département peut contenir plusieurs employés.

## CHAPITRE 4. PHASE DE RÉALISATION DU PROJET

---

### Role (Rôle)

**Attributs :**

- . name

**Relations :** Un utilisateur peut avoir plusieurs rôles.

### Permission (Permission)

**Attributs :**

- . name

**Relations :** Un rôle peut avoir plusieurs permissions.

### File (Fichier)

**Attributs :**

- . name
- . url
- . mime\_type
- . path
- . upload\_date

**Relations :** Un employé peut avoir plusieurs fichiers.

### Duty (Tâche/Mission)

**Attributs :**

- . duration
- . begin\_date
- . type
- . etat

**Relations :** Un employé peut avoir plusieurs tâches.

### Job (Poste)

**Attributs :**

- . title

**Relations :** Un employé peut avoir un seul poste.

## CHAPITRE 4. PHASE DE RÉALISATION DU PROJET

---

### Absence (Absence)

**Attributs :**

- . day\_requested
- . start\_date
- . end\_date
- . address
- . date\_of\_application
- . status
- . date\_of\_approval
- . type\_of\_absence
- . reasons
- . proof
- . comment

**Relations :** Un employé peut avoir plusieurs absences.

### Note (Note)

**Attributs :**

- . message
- . file
- . publish\_date

### Decision (Décision)

**Attributs :**

- . number
- . year
- . date

### Image (Image)

**Attributs :**

- . name
- . url
- . mime\_type
- . path

## 4.5 Technologies et outils

Les outils et technologies suivants ont été utilisés pour la conception du projet **OptiHR**. Chaque outil est accompagné d'une image et d'une description détaillée.

Technologie	Description
	<b>UML</b> : Utilisé pour la modélisation des systèmes et la conception des structures du projet. Il permet de représenter graphiquement les différentes interactions et processus du système.
	<b>Modelio</b> : Outil de modélisation UML permettant de créer des diagrammes tels que les diagrammes de classes, de séquence et d'activités.

TABLE 4.3 – Tableau des technologies et outils utilisés pour la conception

## 4.6 Conclusion

La phase de conception pose les bases essentielles du projet en structurant l'architecture et en précisant les technologies et les modèles de données adoptés. Une conception rigoureuse garantit un développement fluide et efficace, tout en assurant la maintenance et l'évolutivité du système sur le long terme.

# **Chapitre 5**

## **Réalisation**

## 5.1 Introduction

Dans cette section, nous présentons le processus de développement et d'implémentation du système basé sur le modèle conceptuel défini précédemment. Nous détaillerons les choix technologiques adoptés, l'architecture logicielle mise en place, ainsi que les différentes étapes de développement, allant de la conception de la base de données à l'intégration des fonctionnalités essentielles.

L'implémentation suivra une approche modulaire afin de garantir la maintenabilité et l'évolutivité du système. Nous mettrons également en avant les bonnes pratiques de développement, telles que la structuration du code, la gestion des dépendances et l'optimisation des performances.

Enfin, nous testerons et validerons le bon fonctionnement du système à travers des scénarios réels d'utilisation, en nous assurant qu'il répond aux exigences fonctionnelles et non fonctionnelles définies.

## 5.2 Architecture du projet

### 5.2.1 Présentation de l'architecture globale

### 5.2.2 Choix des technologies et outils

Dans cette section, nous présentons les principales technologies utilisées pour le développement du projet. Chaque technologie a été sélectionnée pour ses performances, sa robustesse et sa facilité d'intégration dans notre stack.

Technologie	Description
 The Laravel logo consists of a red stylized 'L' icon followed by the word 'Laravel' in a bold, black, sans-serif font.  Laravel	<p><b>Laravel - Framework Backend :</b> Laravel est un framework PHP moderne basé sur l'architecture MVC (Modèle-Vue-Contrôleur). Il offre de nombreuses fonctionnalités telles que :</p> <ul style="list-style-type: none"><li>Un ORM puissant (Eloquent) pour la gestion de la base de données.</li><li>Un système de migration et de seeders pour faciliter le développement.</li><li>Un mécanisme de routage avancé et un middleware intégré pour la gestion des requêtes HTTP.</li></ul> <p>Grâce à Laravel, nous avons pu structurer le projet de manière efficace et évolutive.</p>

## CHAPITRE 5. RÉALISATION

---

Technologie	Description
	<p><b>Blade - Moteur de Templating :</b> Blade est le moteur de templates natif de Laravel. Il permet de créer des vues dynamiques avec une syntaxe claire et fluide. Ses principales caractéristiques sont :</p> <ul style="list-style-type: none"> <li>Une syntaxe simplifiée pour l'affichage des données et les structures de contrôle ('@if', '@foreach', etc.).</li> <li>La possibilité d'étendre des layouts grâce à l'héritage de templates.</li> <li>Une mise en cache automatique pour améliorer les performances.</li> </ul>
 PostgreSQL	<p><b>PostgreSQL - Base de Données Relationnelle :</b> PostgreSQL est un système de gestion de base de données relationnelle open-source reconnu pour sa stabilité et ses performances. Il a été choisi pour :</p> <ul style="list-style-type: none"> <li>Son support avancé des types de données et des transactions ACID.</li> <li>Sa capacité à gérer de gros volumes de données efficacement.</li> <li>Son intégration facile avec Laravel via l'ORM Eloquent.</li> </ul>
	<p><b>Bootstrap - Framework CSS :</b> Bootstrap est un framework CSS populaire utilisé pour concevoir une interface utilisateur réactive et attrayante. Il nous a permis de :</p> <ul style="list-style-type: none"> <li>Utiliser un système de grille pour une mise en page responsive.</li> <li>Accélérer le développement avec des composants préconçus (modals, boutons, alertes, etc.).</li> <li>Assurer une compatibilité avec tous les navigateurs modernes.</li> </ul>

## CHAPITRE 5. RÉALISATION

---

Technologie	Description
	<b>JavaScript - Langage de Programmation Frontend :</b> JavaScript est un langage de programmation utilisé pour ajouter des interactions dynamiques à l'interface utilisateur. Il a été employé pour : <ul style="list-style-type: none"><li>. Gérer les interactions utilisateur (événements, animations).</li><li>. Améliorer l'expérience avec des requêtes asynchrones (AJAX, Fetch API).</li><li>. Dynamiser le rendu des composants sans recharger la page.</li></ul>
	<b>Git et GitHub - Outils de Gestion de Version :</b> Pour la gestion du code source, nous avons utilisé **Git** et **GitHub** : <ul style="list-style-type: none"><li>. **Git** permet de suivre l'évolution du projet grâce à un système de versionnement performant.</li><li>. **GitHub** facilite la collaboration et l'hébergement du code en ligne, avec des fonctionnalités comme les pull requests et les issues.</li></ul> Ces outils nous ont permis de travailler efficacement en équipe et d'assurer la stabilité du code tout au long du développement.

TABLE 5.2 – Tableau des technologies utilisées pour la réalisation

L'utilisation de ces technologies nous a permis de construire une application performante, maintenable et évolutive. Le choix de Laravel avec Blade pour le backend, PostgreSQL pour la base de données et Bootstrap avec JavaScript pour le frontend a facilité l'implémentation et l'optimisation du projet. Enfin, l'utilisation de Git et GitHub a renforcé la gestion du code et le travail collaboratif.

### 5.3 Développement et implémentation

#### 5.3.1 Mise en place de l'environnement de développement

Pour assurer un développement fluide et efficace, un environnement de travail stable et bien configuré a été mis en place sous \*\*Ubuntu 24\*\*. Cette section détaille les étapes suivies pour

## CHAPITRE 5. RÉALISATION

---

l'installation et la configuration des outils nécessaires.

### Prérequis

Les outils suivants ont été utilisés :

- . Système d'exploitation : **Ubuntu 24**
- . Éditeur de code : **Visual Studio Code**
- . Serveur Web et PHP : **Apache, PHP 8.2**
- . Base de données : **PostgreSQL**
- . Gestionnaire de paquets : **Composer (pour PHP) et npm (pour JavaScript)**
- . Système de versionnement : **Git et GitHub**

### Installation des outils

#### Installation de PHP 8.2 et Apache

Ubuntu 24 ne propose pas PHP 8.2 par défaut. Pour l'installer avec Apache :

##### Installation de PHP 8.2 et Apache

```
sudo apt update && sudo apt upgrade -y  
sudo apt install apache2 php8.2 libapache2-mod-php8.2 php8.2-cli  
php8.2-mbstring php8.2-xml php8.2-curl php8.2-pgsql unzip -y
```

Une fois l'installation terminée, redémarrer Apache :

##### Redémarrage du serveur Apache

```
sudo systemctl restart apache2
```

Vérifier que PHP est bien installé :

##### Vérification de la version de PHP

```
php -v
```

### Installation de Composer

Composer est indispensable pour gérer les dépendances PHP :

##### Installation de Composer

```
curl -sS https://getcomposer.org/installer | php  
sudo mv composer.phar /usr/local/bin/composer  
composer -V
```

## CHAPITRE 5. RÉALISATION

---

### Installation et configuration de PostgreSQL

PostgreSQL est utilisé comme base de données :

#### Installation de PostgreSQL

```
sudo apt install postgresql postgresql-contrib -y
```

Démarrer PostgreSQL et vérifier son installation :

#### Démarrage et vérification

```
sudo systemctl start postgresql  
sudo systemctl enable postgresql  
psql -version
```

### Installation de Laravel

Créer un projet Laravel :

#### Création d'un projet Laravel

```
composer create-project --prefer-dist laravel/laravel:^10.0 nom_du_projet
```

Générer la clé d'application :

#### Génération de la clé d'application

```
php artisan key:generate
```

### Configuration de Git et clonage du projet

Vérifier si Git est installé :

#### Vérification de Git

```
git -version
```

Si Git n'est pas installé, l'ajouter avec :

#### Installation de Git

```
sudo apt install git -y
```

Cloner le projet depuis GitHub :

## CHAPITRE 5. RÉALISATION

---

### Clonage du projet

```
git clone https://github.com/utilisateur/nom_du_projet.git  
cd nom_du_projet
```

### Lancement du serveur de développement

Démarrer Laravel en mode développement :

### Lancement du serveur Laravel

```
php artisan serve
```

L'application est maintenant accessible à l'adresse :

### URL d'accès

```
http://127.0.0.1:8000/
```

### 5.3.2 Développement des fonctionnalités principales

Feature 1 : [Nom de la fonctionnalité]

Feature 2 : [Nom de la fonctionnalité]

## 5.4 Problèmes rencontrés et solutions adoptées

### 5.4.1 Difficultés techniques

### 5.4.2 Optimisation et amélioration du code

## 5.5 Tests et validation

### 5.5.1 Stratégie de test

### 5.5.2 Résultats des tests

## 5.6 Conclusion

# **Chapitre 6**

## **Bilan**

## 6.1 Introduction

Dans ce chapitre, nous faisons le bilan du projet en revenant sur les objectifs fixés, les résultats obtenus, les compétences acquises ainsi que les perspectives d'amélioration.

## 6.2 Retour sur les objectifs initiaux

### 6.2.1 Rappel des objectifs du projet

### 6.2.2 Évaluation de leur atteinte

## 6.3 Analyse des résultats obtenus

### 6.3.1 Synthèse des résultats

### 6.3.2 Difficultés rencontrées et solutions adoptées

## 6.4 Compétences acquises et développées

### 6.4.1 Compétences techniques

### 6.4.2 Compétences organisationnelles

### 6.4.3 Compétences personnelles

## 6.5 Limites et axes d'amélioration

### 6.5.1 Contraintes techniques et organisationnelles

### 6.5.2 Améliorations possibles et perspectives d'évolution

## 6.6 Conclusion

# **Chapitre 7**

## **Conclusion Générale**