

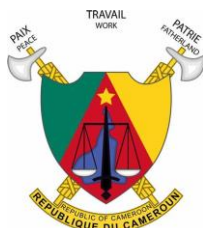
REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

MINISTERE DE L'EMPLOI ET DE LA FORMATION
PROFESSIONNELLE

CENTRE DE FORMATIONS NUMERIQUES

FABRIQUE INCLUSIVE INCH'CLASS



REPUBLIC OF CAMEROON

Peace – Work – Fatherland

MINISTRY OF EMPLOYMENT AND VOCATIONAL
TRAINING

CENTRE DE FORMATIONS NUMERIQUES

FABRIQUE INCLUSIVE INCH'CLASS



RAPPORT DE PROJET CHEF-D'OEUVRE

THEME :

DEVELOPPEMENT D'UNE PLATEFORME WEB
POUR RECHERCHE MINI-JOBS ETUDIANTS

Rapport rédigé en vue de l'obtention d'une certification en :

DEVELOPPEMENT WEB ET WEB MOBILE

Réalise par :

AICHATOU KANGNE SIRADJOU

(Apprenante **Inch'Class**)

Sous l'encadrement de :

M. CYPRIEN DONTSA :
responsable des formations

Promotion </> | **LE DCLIC DU CODE**

SESSION Jan-Mai 2025

SESSION Jan-Mai 2025

SOMMAIRE

REMERCIEMENTS	3
Introduction	4
Première Partie :	5
COMPREHENSION, ANALYSE ET CHOIX TECHNIQUES	5
I. Compréhension du besoin client	5
II. Etat de l'art	8
➤ Positionnement et originalité de notre solution	8
III. Traduction technique et choix technique du projet	9
1. Back end – PHP avec le Framework Laravel	9
2. Front end – JavaScript avec React.js	9
DEUXIEME PARTIE :	11
I. Gestion de projet	11
a) Méthodologie de travail adoptée	11
b) Découpage du projet en phases	11
.....	12
II. Conception des digrammes	12
1. Diagramme de cas d'utilisation (Use Case)	12
2. Diagramme de classes	13
III. Retours d'expériences sur les outils et les techniques	14
TROISIEME PARTIE	15
I. BILAN GENERAL	15
II. AMELIORATIONS POSSIBLES	15
CONCLUSION GENERALE	16
.....	16

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui m'ont accompagné tout au long de ce projet. Je remercie tout particulièrement Mme Chantal NGOUBEYOU directrice du centre pour l'opportunité offerte en l'intégrant, mon encadrant, M. Cyprien, pour ses conseils avisés, sa disponibilité et son soutien constant. Merci également à ma famille, mes amis et collègues pour leur encouragement et leur compréhension durant cette période. Enfin, je souhaite remercier l'ensemble des intervenants et formateurs d'inch'class pour les connaissances transmises qui m'ont permis de mener à bien ce projet.



Introduction

Trouver un emploi ponctuel représente aujourd'hui un véritable défi pour de nombreux étudiants. Entre les contraintes liées aux études, le manque de flexibilité des employeurs traditionnels, et l'absence d'outils centralisés adaptés à leur réalité, les étudiants peinent à concilier vie académique et besoins financiers. Pourtant, la demande pour des missions courtes ou occasionnelles existe bel et bien, tant du côté des étudiants que des particuliers ou entreprises recherchant une aide ponctuelle.

Face à ce constat, une problématique centrale se pose : **comment faciliter et encadrer la mise en relation entre étudiants disponibles et offreurs de petits jobs, tout en assurant simplicité, fiabilité et efficacité pour les deux parties ?**

Ce projet répond à cette problématique par la conception et le développement d'une **application web et mobile** dédiée, qui permettra aux étudiants de consulter des offres de missions, de postuler facilement, et de gérer leur profil de manière autonome. Quant aux offreurs, ils pourront publier leurs annonces, entrer en contact avec des profils pertinents, et suivre les candidatures selon des critères de délai et de rémunération.

Afin de rendre compte de cette démarche, le présent rapport est structuré en trois grandes parties :

- Une **première partie** consacrée à la compréhension du besoin client, à l'état de l'art existant, ainsi qu'aux choix techniques adoptés pour la mise en œuvre de la solution ;
- Une **seconde partie** portant sur la méthode de gestion de projet appliquée et les retours d'expérience sur les outils utilisés ;
- Enfin, une **troisième partie** qui dresse un **bilan global du projet**, ses points forts, ses limites, ainsi que les pistes d'améliorations envisagées.

Première Partie :

COMPREHENSION, ANALYSE ET CHOIX TECHNIQUES

Le but de cette partie est de présenter le contexte général du projet. Nous commencerons tout d'abord par la phase de spécification pour bien organiser, clarifier les tâches du projet et suivre une bonne méthodologie c'est-à-dire la compréhension du besoin client. Puis, nous présenterons l'état de l'art. Enfin nous exposerons l'étude effectuée sur les notions technologiques, les solutions et principes utilisés dans le cadre de la réalisation de ce projet

I. Compréhension du besoin client

Pour comprendre le besoin client, nous avons utilisé plusieurs méthodes de recueil des besoins adaptées aux caractéristiques du projet et du public. Nous avons ainsi combiné des méthodes quantitatives et qualitatives afin de recueillir des données objectives et subjectives ainsi que les motivations des clients potentiels. Ces méthodes sont entre autre

a) Identification des acteurs

L'application repose sur l'interaction de trois types d'utilisateurs, chacun disposant de rôles et de niveaux d'accès spécifiques. Cette répartition permet d'assurer une gestion fluide et sécurisée de la plateforme, en fonction des besoins et des responsabilités de chaque acteur.

- **L'administrateur :**
Il s'agit de l'utilisateur doté du plus haut niveau de privilèges. L'administrateur est responsable de la gestion globale de la plateforme. Il peut superviser les activités, gérer les comptes utilisateurs (étudiants et offreurs), modérer les contenus publiés, suspendre des profils en cas d'abus, et assurer le bon fonctionnement de l'ensemble du système.
- **L'offreur de job :**
Ce profil représente les personnes physiques ou morales (particuliers, entreprises, associations) désireuses de proposer des missions ponctuelles. L'offreur peut publier des annonces détaillées (description du job, durée, rémunération, etc.), consulter les candidatures reçues, entrer en contact avec des étudiants intéressés, et suivre l'évolution de ses offres.
- **L'étudiant :**
Utilisateur principal visé par la plateforme, l'étudiant peut consulter les offres disponibles, postuler à celles qui correspondent à ses disponibilités et compétences, et gérer son profil personnel. Il a également la possibilité de modifier ses informations, de suivre ses candidatures, et de recevoir des notifications liées à ses activités.

b) Les besoins fonctionnels

Les besoins fonctionnels correspondent à l'ensemble des fonctionnalités que l'application doit proposer pour répondre aux attentes des utilisateurs. Ils sont définis en fonction des rôles identifiés (administrateur, offreur, étudiant) et couvrent les interactions principales avec le système. Voici les besoins fonctionnels identifiés pour chaque type d'utilisateur:

➤ Pour l'étudiant

- **Création de compte et authentification** (inscription, connexion).
- **Consultation des offres de job disponibles** via un tableau de bord ou un moteur de recherche.
- **Postulation à une offre** via un formulaire de candidature.
- **Suivi des candidatures envoyées** et de leur statut (en attente, acceptée, refusée).
- **Accès à un espace personnel** permettant :
 - La modification de ses informations personnelles et professionnelles.
 - La mise à jour de ses compétences, disponibilités ou expériences.
- **Réception de notifications** concernant les candidatures ou les nouveaux jobs disponibles.

➤ Pour l'offreur de job

- **Création de compte et authentification.**
- **Publication d'une nouvelle offre de job** avec détails (titre, description, durée, rémunération, lieu, etc.).
- **Gestion des offres publiées** : édition, suppression, archivage.
- **Consultation des candidatures reçues** pour chaque offre.
- **Contact avec les candidats sélectionnés** via messagerie ou informations de contact.
- **Suivi de l'historique des missions proposées.**

➤ Pour l'administrateur

- **Accès à un tableau de bord administratif** centralisant l'activité de la plateforme.
- **Gestion des comptes utilisateurs** (consultation, suspension, suppression).
- **Modération des offres et des profils** en cas de contenu inapproprié.
- **Suivi statistique** de la plateforme (nombre d'inscriptions, offres publiées, candidatures...).
- **Gestion des paramètres globaux** du système (règles d'utilisation, messages automatiques, catégories de job...).

c) Les besoins non-fonctionnels

Les besoins non-fonctionnels regroupent l'ensemble des critères de qualité que doit respecter l'application afin de garantir une **expérience utilisateur optimale**, une **maintenabilité élevée** et une **robustesse technique**. Bien qu'ils ne décrivent pas des fonctionnalités directes, ces besoins sont essentiels pour assurer le bon fonctionnement, la pérennité et la sécurité du système. Voici les principaux besoins non-fonctionnels identifiés pour ce projet :

➤ Ergonomie de l'interface

L'application doit offrir une interface conviviale, intuitive et accessible, tant sur mobile que sur web.

Cela implique:

- Une navigation fluide et logique.
- Des formulaires clairs et faciles à remplir.
- Une expérience utilisateur homogène sur tous les supports (responsive design).
- Un design attractif adapté aux jeunes utilisateurs (étudiants).

➤ Fiabilité

Le système doit fonctionner correctement, sans interruption ni erreur critique, quelles que soient les conditions d'utilisation.

Cela comprend:

- La gestion des erreurs avec des messages explicites.
- La cohérence des données enregistrées (ex : candidatures valides).
- La sauvegarde régulière des données importantes.

➤ Disponibilité

L'application doit être accessible à tout moment, 7j/7 et 24h/24, avec un minimum de temps d'indisponibilité.

Cela peut être assuré par:

- Un hébergement stable et évolutif.
- La mise en place de surveillances automatiques et d'alertes en cas de panne.
- Une stratégie de mise à jour sans interruption majeure du service.

➤ Sécurité

La sécurité est cruciale pour protéger les données personnelles et professionnelles des utilisateurs.

Elle inclut:

- L'authentification sécurisée des comptes.
- Le chiffrement des mots de passe et des données sensibles.
- La protection contre les attaques courantes (injections SQL, XSS, CSRF, etc.).
- Une gestion rigoureuse des rôles et des permissions.

II. Etat de l'art

➤ Description de l'existant

Avant d'entamer le développement de notre solution, une étude comparative des plateformes existantes a été réalisée afin d'identifier leurs forces, leurs limites, et de positionner notre projet de manière pertinente sur le marché. Cette analyse a porté principalement sur trois services reconnus dans le domaine des petits jobs étudiants : **Student Pop**, **Side**, et **StaffMe**.

✓ Student Pop

Student Pop est une plateforme spécialisée dans la mise en relation entre étudiants et entreprises pour des missions ponctuelles, notamment dans l'événementiel et le commerce. Elle propose un processus de sélection exigeant, basé sur des entretiens, ce qui limite l'accès à certains profils. De plus, son rayonnement est principalement concentré dans les grandes villes.

✓ Side

Side se positionne comme une solution professionnelle pour des missions temporaires dans des secteurs variés (logistique, vente, support client...). Elle met l'accent sur la fiabilité et le professionnalisme des étudiants. Cependant, elle fonctionne également avec un système de présélection, ce qui peut être un frein pour les étudiants n'ayant pas encore d'expérience significative.

➤ Positionnement et originalité de notre solution

En analysant les limites des plateformes existantes, notre application a été conçue pour répondre à une demande différente, centrée sur l'**accessibilité**, la **simplicité**, et la **proximité**. Voici ce qui fait l'originalité et la valeur ajoutée de notre solution :

- **Simplicité d'utilisation** : l'inscription, la consultation des offres et la candidature sont conçues pour être rapides, intuitives et accessibles à tout étudiant, sans barrière à l'entrée.
- **Gestion décentralisée** : contrairement à d'autres plateformes où un intermédiaire valide les profils ou les missions, notre système repose sur l'autonomie des deux parties. L'étudiant comme l'offreur gère leurs interactions de manière directe et fluide.
- **Ancrage local fort** : l'application met l'accent sur les **petites missions de proximité**, souvent ignorées par les grandes plateformes (livraisons ponctuelles, aide à domicile, petits services, babysitting, etc.). Cela favorise un lien direct entre les habitants d'un même quartier ou ville.

III. Traduction technique et choix technique du projet

❖ Les langages de programmation

Le choix des langages de programmation utilisés pour ce projet a été guidé par plusieurs critères : la **performance**, la **modularité**, la **facilité de développement**, et la **disponibilité de la documentation et de la communauté**. Le projet repose sur une architecture **front end/back end** découplée (approche "API-centric"), avec les technologies suivantes :

1. Back end – PHP avec le Framework Laravel

Le back end de l'application a été développé en **PHP**, à l'aide du Framework **Laravel**, dans sa version la plus récente au moment du développement (Laravel 12). Ce choix s'explique par plusieurs avantages:

- **Simplicité et expressivité du code** grâce à une syntaxe claire et moderne.
- **Structure MVC (Modèle-Vue-Contrôleur)** facilitant la séparation des responsabilités et la maintenabilité du code.
- **Système de routage robuste** et sécurisation automatique des requêtes HTTP.
- **ORM Eloquent** pour une interaction fluide avec la base de données relationnelle (MySQL).
- **Outils intégrés** comme la validation, les migrations, les middlewares, les notifications, et le système d'authentification (Laravel Sanctum).
- **Facilité à créer une API RESTful** consommable par le front end React.



Laravel a donc permis de construire un back end performant, sécurisé et facilement extensible.

2. Front end – JavaScript avec React.js

L'interface utilisateur a été développée avec **React.js**, l'une des bibliothèques JavaScript les plus utilisées pour construire des interfaces interactives. Ce choix a été motivé par:

- **Composants réutilisables** facilitant le développement modulaire et la maintenance.
- **Réactivité de l'interface** grâce à un DOM virtuel performant.
- **Gestion fluide de l'état de l'application** (via use State, useEffect, ou avec des solutions comme Redux ou Context API si besoin).
- **Intégration facile avec Axios** pour consommer l'API Laravel.
- **Adaptabilité mobile (responsive)** via des librairies CSS modernes ou des Framework comme Tailwind CSS.



❖ Les outils utilisés

Le développement de l'application s'est appuyé sur un ensemble d'outils modernes, choisis pour leur efficacité, leur compatibilité avec les technologies Laravel et React, ainsi que leur facilité de prise en main. Voici les principaux outils utilisés :

1. VS Code (Visual Studio Code)

Environnement de développement léger, puissant et extensible, utilisé pour l'édition du code frontend (React) et backend (Laravel), avec des extensions utiles comme **Laravel Blade Snippets**, **ESLint**, et **Prettier**.



2. Postman

Outil indispensable pour tester et documenter les requêtes API REST entre le back end Laravel et le frontend React. Il a permis de simuler les requêtes (GET, POST, PUT, DELETE) et de valider la structure des réponses.



3. Git & GitHub

Git a été utilisé pour le versionnement du code, permettant de suivre l'évolution du projet, de collaborer efficacement et de revenir à des versions stables en cas de besoin. Git Hub a servi de plateforme de centralisation et de gestion du code source.



4. XAMPP

Utilisé pour exécuter l'environnement de développement local (PHP, MySQL, Apache) nécessaire au fonctionnement de Laravel en local.



5. Figma

Outil de maquettage utilisé au début du projet pour concevoir les interfaces et planifier l'expérience utilisateur avant de démarrer le développement.



DEUXIEME PARTIE : REALISATION DU PROJET

Il est question pour nous dans cette partie de présenter les différents outils utilisés dans la gestion de notre projet avant et pendant sa phase de développement.

I. Gestion de projet

La gestion du projet a été un facteur déterminant dans sa réussite. Elle a permis de structurer les différentes phases de développement, de suivre l'avancement, et d'anticiper les risques.

a) Méthodologie de travail adoptée

Expliquer brièvement la méthode choisie pour organiser le projet.

Méthodologie recommandée : **Méthode agile / Scrum simplifié** (sprint court et livrables réguliers).

b) Découpage du projet en phases

Présenter les grandes étapes suivies du début à la fin :

- **Phase 1 : Analyse du besoin et rédaction des spécifications**
- **Phase 2 : Maquettage et conception de l'architecture technique**
- **Phase 3 : Développement du backend (API Laravel)**
- **Phase 4 : Développement du frontend (React)**
- **Phase 5 : Intégration frontend/backend**
- **Phase 6 : Tests et validation fonctionnelle**
- **Phase 7 : Déploiement local / hébergement test**
- **Phase 8 : Rédaction du rapport et mise en forme**

c) Suivi d'avancement et planification depuis trello

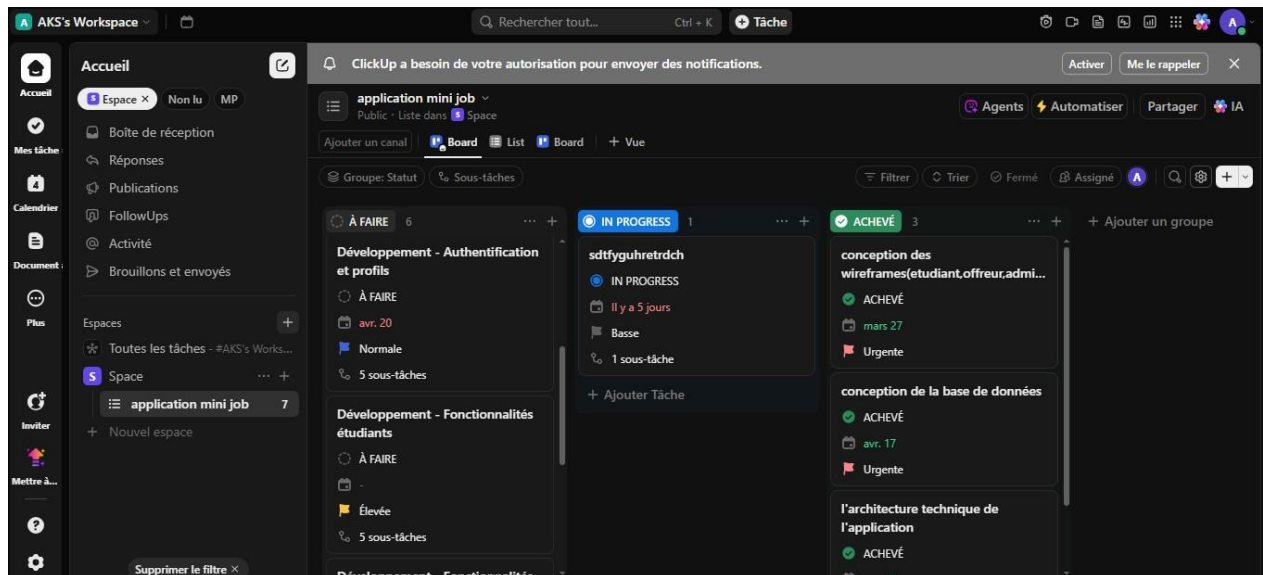


Figure 1: Tableau de bord TRELLO

II. Conception des digrammes

La phase de conception a constitué une étape essentielle dans la modélisation du système avant sa mise en œuvre. Elle a permis de visualiser l'organisation fonctionnelle de l'application ainsi que les interactions entre les différentes entités.

Deux diagrammes principaux ont été élaborés dans cette phase : le **diagramme de cas d'utilisation** et le **diagramme de classes**.

1. Diagramme de cas d'utilisation (Use Case)

Le **diagramme de cas d'utilisation** permet de représenter les interactions entre les acteurs (utilisateurs) et les fonctionnalités principales du système. Il répond à la question : *"Qui fait quoi dans le système ?"*

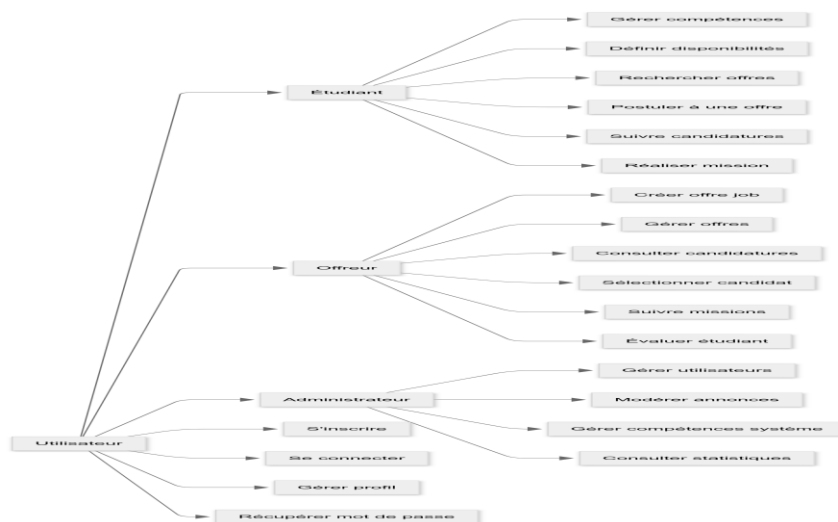


Figure 2: Diagramme de cas d'utilisation

2. Diagramme de classes

Le **diagramme de classes** modélise les entités du système ainsi que leurs relations. Il constitue un schéma de base pour la création de la base de données et des modèles dans le back end Laravel.

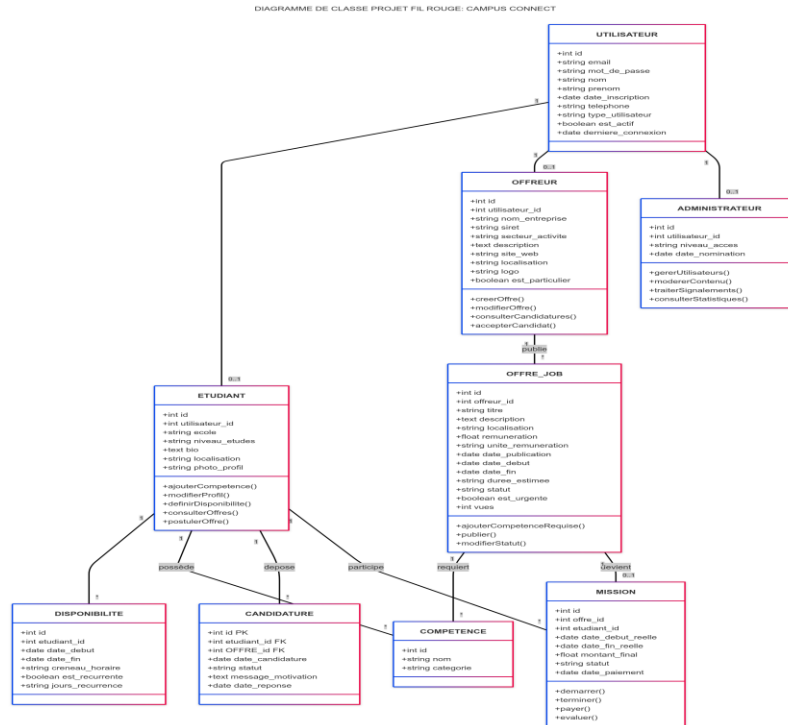




Figure 3: Diagramme de classe

III. Retours d'expériences sur les outils et les techniques

Au fil du développement de cette application, l'utilisation des outils techniques ainsi que l'implémentation des différentes fonctionnalités ont constitué une expérience enrichissante, marquée à la fois par des réussites et des défis. Cette section présente une synthèse critique de l'expérience vécue avec les principaux outils et techniques mobilisés.

Tableau 1: Outils de developpement

OUTILS DE DEVELOPPEMENT		
NOM DE L'OUTIL	POINTS POSITIFS	POINTS NEGATIFS
Laravel (backend) 	Laravel s'est révélé être un framework robuste, complet et bien documenté. Il facilite le développement rapide grâce à son système de routes, migrations, Eloquent ORM et ses fonctionnalités intégrées (authentification, validation, middleware...). <i>Points positifs : productivité, clarté de la structure MVC, facilité de gestion des relations de base de données.</i>	Difficultés rencontrées : configuration initiale, compréhension des politiques d'accès (Policies, Gates)
React (frontend) 	Points positifs : modularité, interactivité, bonne séparation des responsabilités.	intégration avec l'API Laravel, gestion de la sécurité côté client (authentification/token).

TROISIEME PARTIE:

BILAN DU PROJET

I. BILAN GENERAL

Le projet de développement d'une application web et mobile facilitant l'accès à des petits jobs pour les étudiants a permis d'atteindre les objectifs fixés. Une solution fonctionnelle a été mise en place, intégrant :

- Une interface pour les **étudiants** leur permettant de consulter les offres, postuler, et gérer leur profil.
- Une interface pour les **offreurs** afin de publier et gérer leurs missions.
- Un espace **administrateur** pour la modération et la supervision de la plateforme.
- Une communication efficace entre le backend (Laravel) et le frontend (React).
- Des fonctionnalités respectant les rôles et les permissions des différents utilisateurs.

II. AMELIORATIONS POSSIBLES

Pour enrichir le projet et améliorer son efficacité, plusieurs pistes sont envisageables :

1. **Ajout d'une messagerie interne** entre étudiants et offreurs.
2. **Mise en place de notifications en temps réel** (WebSockets ou Pusher).
3. **Version mobile native** avec React Native ou Flutter pour un meilleur confort d'utilisation sur smartphone.
4. **Ajout de filtres de recherche avancés** (par ville, type de mission, disponibilité).
5. **Évolution vers un modèle de recommandation intelligent**, basé sur les compétences et les précédentes candidatures.

CONCLUSION GENERALE

Ce projet de développement d'une application web et mobile dédiée à la mise en relation entre étudiants et offreurs de petits jobs répond à une problématique concrète et actuelle : faciliter l'accès à des opportunités ponctuelles de travail pour les jeunes en quête d'indépendance financière ou d'expérience professionnelle.

Tout au long de ce travail, nous avons suivi une démarche méthodique allant de l'analyse du besoin à la mise en œuvre technique. La combinaison des technologies modernes comme **Laravel** pour le backend et **React** pour le frontend a permis de concevoir une application performante, évolutive et centrée sur l'utilisateur.

En plus de renforcer les compétences techniques, ce projet a été une véritable leçon de gestion, d'adaptation et de rigueur. Il a mis en lumière les enjeux liés à la conception d'une solution numérique utile, ergonomique et sécurisée.

Bien que la version actuelle soit fonctionnelle, plusieurs pistes d'amélioration ont été identifiées pour aller encore plus loin dans la qualité de service offerte. Le potentiel de cette application est réel, notamment dans une perspective d'extension mobile, de notifications en temps réel, et d'algorithmes de recommandation.

En somme, ce projet constitue une base solide pour une solution innovante, ancrée dans le réel, avec un fort impact social et un avenir prometteur en termes de développement.