Rapport de Projet

# ProRide – Plateforme de covoiturage entre salariés

Nom : Aminata Ba

Établissement : Dakar Institute of Technology (DIT)

Année académique : 2024 - 2025

Encadrant : [Nom de l'encadrant]

# Table des matières

1. Remerciements

2. Résumé

3. Introduction

4. Étude préalable

5. Méthodologie de travail

6. Conception de l’application

7. Développement

8. Fonctionnalités clés

9. Captures d’écran

10. Difficultés rencontrées

11. Perspectives d’avenir

12. Conclusion

13. Annexes

14. Sources et bibliographie

# 1. Remerciements

Je tiens à remercier chaleureusement mon encadrant, mes enseignants, ainsi que toute l'équipe pédagogique du Dakar Institute of Technology pour leur accompagnement tout au long de ce projet. Mes remerciements vont également à mon binôme pour sa collaboration et son implication, et à mes proches pour leur soutien moral.

# 2. Résumé

Le projet ProRide est une plateforme de covoiturage destinée exclusivement aux salariés. Il permet de mettre en relation les conducteurs et les passagers d’une même entreprise ou secteur pour des trajets domicile-travail. L’objectif est de réduire les coûts de transport, favoriser l'entraide entre collègues et contribuer à la réduction des émissions de carbone. L’application a été développée avec Flask, HTML, CSS et Tailwind CSS, sans base de données externe, mais avec des listes Python simulant les tables.

# 3. Introduction

Dans un monde de plus en plus connecté, les entreprises cherchent à améliorer les conditions de déplacement de leurs employés. Le covoiturage entre salariés s’impose comme une solution pratique, économique et écologique. Ce projet a pour but de concevoir une application web simple, accessible et intuitive permettant aux salariés de s’organiser pour se déplacer ensemble.

# 4. Étude préalable

Avant le développement, une analyse des besoins a été effectuée : l’application devait permettre aux salariés de s’inscrire, publier ou rechercher un trajet, et réserver une place. L’analyse a aussi montré la nécessité d’avoir des fonctionnalités différentes selon le rôle de l’utilisateur (conducteur ou passager).

# 5. Méthodologie de travail

La méthodologie adoptée est basée sur une approche agile, en développant progressivement les fonctionnalités et en les testant au fur et à mesure. L’organisation du travail s’est faite à l’aide de Trello, et le code a été versionné localement avec une structure de fichiers bien définie.

# 6. Conception de l’application

- Architecture : application Flask avec structure MVC simplifiée

- Données : stockées dans des listes Python (`users`, `trajets`, `reservations`)

- Interfaces : pages HTML avec rendu Jinja, stylisées avec Tailwind CSS

- Rôles : distinction claire entre conducteurs et passagers avec `session['role']`

- Maquettes : conception directe via HTML (sans outils externes comme Figma)

# 7. Développement

Chaque fonctionnalité a été développée avec une route Flask spécifique. Voici quelques exemples :

- `/register` pour l’inscription

- `/login` pour la connexion

- `/conducteur/ajouter\_trajet` pour la publication d’un trajet

- `/recherche` pour la recherche par les passagers

- `/reserver/<int:index>` pour effectuer une réservation

Le système de session permet de gérer les connexions utilisateur et restreindre l’accès à certaines pages.

# 8. Fonctionnalités clés

- Inscription avec rôle (conducteur ou passager)

- Connexion sécurisée

- Tableau de bord conducteur avec liste des réservations

- Ajout de trajets

- Recherche et réservation de trajets

- Affichage dynamique des trajets avec disponibilité mise à jour

# 9. Captures d’écran

Des captures d’écran sont jointes en annexe pour illustrer :

- La page d’accueil

- La page d’inscription

- Le tableau de bord conducteur

- Le formulaire d’ajout de trajet

- La page de recherche

- Le tableau de bord passager

# 10. Difficultés rencontrées

- Problèmes d'installation de Tailwind CSS sur Windows

- Gestion conditionnelle des vues selon le rôle

- Stylisation responsive sans JavaScript

- Intégration de flash messages et gestion des sessions

# 11. Perspectives d’avenir

- Intégration d'une vraie base de données (MongoDB, SQLite...)

- Système d’authentification sécurisé (hashing, token...)

- Restriction aux emails professionnels uniquement

- Upload d’image pour les voitures des conducteurs

- Intégration d’un système de notation / avis

- Déploiement de l’application (Render, Vercel...)

- Version mobile ou PWA

# 12. Conclusion

Le projet ProRide a permis de mettre en œuvre les compétences acquises en développement web avec Flask. Il répond à une problématique réelle et actuelle. Cette première version fonctionnelle pose une bonne base pour un développement futur plus complet, avec une meilleure sécurité et un vrai système de persistance des données.

# 13. Annexes

- Structure du projet

- Arborescence des fichiers

- Exemple de code (routes, templates)

- Liens utiles

- Extraits du fichier CSS Tailwind personnalisé

# 14. Sources et bibliographie

- Flask by Example (Gareth Dwyer)

- Stack Overflow

- OpenAI ChatGPT

- Documentation officielle Flask et Tailwind CSS

- Google / Youtube