Documentation technique – EcoRide

Auteur : Seifeddine Maachaoui
Projet : EcoRide – Plateforme de covoiturage écoresponsable
Type : Front-end (HTML / CSS / JavaScript / localStorage)
Version : 0.1

1. Introduction

Je présente ici la documentation technique de mon application EcoRide. Ce document détaille mes choix technologiques, la configuration de mon environnement de travail, la structure logique du code, le modèle de données ainsi que le processus de déploiement de l'application.

2. Réflexion technologique initiale

Dès le début du projet, j'ai choisi de concevoir EcoRide en 100 % front-end, afin d'approfondir mes compétences en HTML, CSS et JavaScript tout en simulant un fonctionnement complet sans serveur. Je voulais prouver qu'il est possible de créer une expérience utilisateur complète – connexion, profil, trajets, réservations – sans recourir à une base de données ni à un back-end.

Pour cela, j'ai opté pour : HTML5 pour la structure, CSS3 pour le design responsive, JavaScript (ES6) pour la logique métier, localStorage pour simuler la base de données, et PHP uniquement pour l'inclusion des éléments communs.

3. Configuration de mon environnement de travail

Pour le développement, j'ai utilisé VS Code comme éditeur principal, l'extension Live Server pour tester directement mes pages en local, et Chrome pour le débogage via la console. Aucun serveur back-end ni base de données SQL n'a été nécessaire : le code est entièrement exécutable depuis le navigateur.

4. Structure du projet et architecture logicielle

L'application repose sur une structure inspirée du modèle MVC (Modèle – Vue – Contrôleur). Les models contiennent les classes de données (UserModel.js, TrajetModel.js, etc.), les controllers gèrent la logique métier, et les views s'occupent de l'affichage. Le dossier utils contient des fonctions transversales comme la gestion de session.

5. Modèle conceptuel de données (MCD)

Mon modèle repose sur trois entités principales : Utilisateur, Trajet et Réservation. L'utilisateur peut être chauffeur ou passager. Un trajet appartient à un chauffeur et peut être réservé par plusieurs passagers. Chaque réservation relie un utilisateur à un trajet.

Toutes ces données sont stockées sous forme d’objets JSON dans le localStorage.

6. Diagrammes d’utilisation et de séquence (description)

Le diagramme de cas d’utilisation illustre les interactions entre les acteurs : visiteur, passager, chauffeur et administrateur. Le diagramme de séquence montre la chronologie d’une réservation, depuis la recherche d’un trajet jusqu’à l’ajout de la réservation dans le localStorage et l’affichage d’une notification de confirmation.

7. Déploiement de l’application

J’ai organisé les fichiers (assets, js, css, views) en respectant une structure claire. L’application a été testée en local via Live Server. Elle peut ensuite être hébergée sur un service statique comme GitHub Pages ou Netlify sans configuration supplémentaire. Le simple chargement des fichiers suffit pour exécuter EcoRide.

8. Conclusion

Ce projet m’a permis d’appliquer concrètement la logique MVC en JavaScript et de structurer une application front-end complète sans dépendance serveur. Grâce à une architecture claire, EcoRide est une application stable, évolutive et facile à maintenir, démontrant ma capacité à concevoir et documenter un projet web complet.