

Documentation Technique – EcoRide

Plateforme de covoiturage écologique

Table des matières

1. Réflexions technologiques initiales
2. Configuration de l'environnement de travail
3. Modèle conceptuel de données
4. Diagrammes d'utilisation et de séquence
5. Architecture technique
6. Sécurité
7. Déploiement

1. Réflexions technologiques initiales

- **Objectif** : Créer une plateforme web de covoiturage écologique accessible sur desktop et mobile.
- **Technologies choisies** :
 - **Back-end** : PHP 8.x (facile à déployer et adapté à XAMPP)
 - **Base de données** : MySQL (relationnelle, intégrée à XAMPP)
 - **Front-end** : HTML5, CSS3, JavaScript, Bootstrap (pour le responsive)
- **Justification** : Ces technologies sont adaptées à un projet débutant, largement documentées et compatibles avec les outils gratuits.

2. Configuration de l'environnement de travail

- **Serveur local** : XAMPP (Apache, PHP, MySQL)
- **Éditeur de code** : Visual Studio Code
- **Gestion de versions** : GitHub (branches main, develop, feature)
- **Organisation des fichiers** :
 - ecoride/
 - └─ backend/ # Code PHP, logique serveur
 - └─ frontend/ # HTML, CSS, JS, images

- |— docker/ # Fichiers Docker (optionnel)
- |— docs/ # Documentation PDF (technique, charte, manuel, gestion projet)
- |— script/ # Scripts divers
- |— SQL/ # Fichiers SQL (création, intégration)
- └─ README.md # Guide d'installation

3. Modèle conceptuel de données

Tables principales :

- **utilisateur** : id, nom, prénom, email, mot_de_passe, date_naissance, rôle
- **trajet** : id, départ, arrivée, date, nombre_places, chauffeur_id
- **réservation** : id, id_utilisateur, id_trajet, date_reservation

Relations :

- Un utilisateur peut effectuer plusieurs réservations
- Un trajet peut être réservé par plusieurs utilisateurs

4. Diagrammes d'utilisation et de séquence

Cas d'utilisation principaux :

- S'inscrire et se connecter
- Rechercher un trajet
- Réserver un trajet
- Consulter ses réservations
- Gérer les trajets (admin)

Diagramme de séquence – Réservation d'un trajet :

1. L'utilisateur sélectionne un trajet et clique sur "Réserver"
2. Le serveur vérifie la disponibilité
3. Si une place est disponible, la réservation est enregistrée
4. L'utilisateur reçoit une confirmation

5. Architecture technique

- **Modèle** : Fonctions PHP pour la gestion des données
- **Vue** : Fichiers HTML/PHP avec Bootstrap pour le design
- **Contrôleur** : Logique PHP dans chaque page

6. Sécurité

- **Validation des formulaires** côté client et serveur
- **Protection contre les injections SQL** (requêtes préparées avec PDO)
- **Hashage des mots de passe** (password_hash et password_verify)
- **Contrôle d'accès** (sessions, vérification du rôle utilisateur)
- **Protection XSS** (htmlspecialchars sur les entrées utilisateur)

7. Déploiement

- **Hébergement** : 000webhost (hébergement gratuit compatible PHP/MySQL)
- **Transfert des fichiers** : via FileZilla (FTP)
- **Création de la base de données** : via l'interface 000webhost, import du fichier SQL
- **Configuration** : adaptation du fichier config.php avec les identifiants de la base distante
- **Tests** : vérification des fonctionnalités principales en ligne
- **Sécurisation** : activation du HTTPS et protection de l'espace administrateur

Auteur : Brandès Thibault

Date : 20 juillet 2025

Version : 2025