

DOCUMENTATION TECHNIQUE

ECORIDE

Plateforme de Covoiturage Écologique

Taieb MIMOUNI

Juillet 2025

ECF - Développeur Web et Web Mobile

ARCHITECTURE DU PROJET

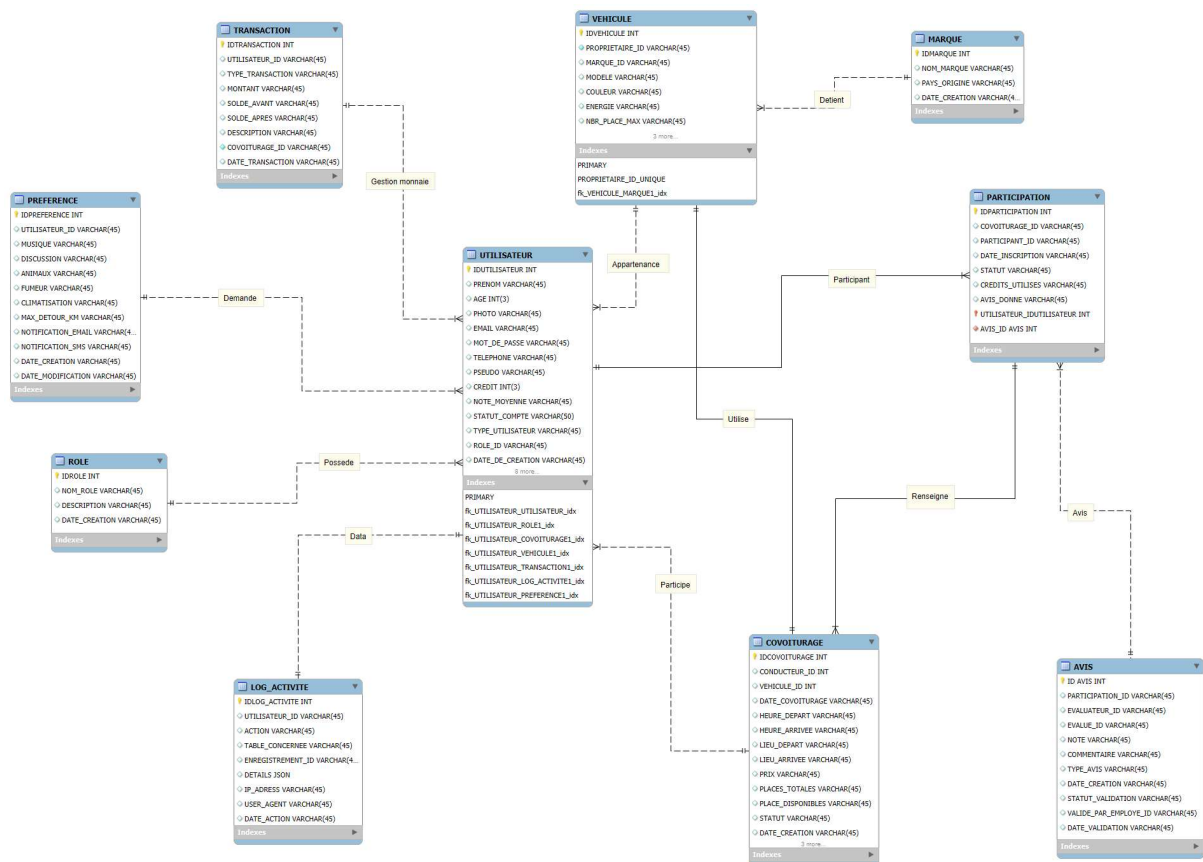
STRUCTURE GENERALE :

- Frontend : HTML5, CSS3, JavaScript
- Backend : PHP 8 avec POO
- Base de données : MySQL + MongoDB (hybride)
- Conteneurisation : Docker + Docker Compose
- Serveur web : Apache

ARCHITECTURE 3-TIERS :

- Couche Présentation : Templates HTML/CSS responsive
- Couche Métier : Classes PHP (User, Covoiturage, Reservation, MongoDB)
- Couche Données : MySQL (relationnelle) + MongoDB (NoSQL)

MODÈLE CONCEPTUEL DE DONNÉES



TABLES MYSQL :

- utilisateur (id, nom, prenom, email, mdp, credits, role...)
- covoiturage (id, conducteur_id, ville_depart, ville_arrivee, date_depart, prix...)
- reservation (id, covoiturage_id, passager_id, statut, date_reservation...)

COLLECTIONS MONGODB :

- voitures (modèle, marque, énergie, places...)
- avis (note, commentaire, date...)
- préférences (musique, température, animaux...)

APIS REST DÉVELOPPÉES

ENDPOINTS UTILISATEUR (/api/users.php) :

- POST /api/users.php?endpoint=create - Créer un utilisateur
- GET /api/users.php?endpoint=by-id&id=X - Récupérer un utilisateur
- POST /api/users.php?endpoint=login - Authentification

ENDPOINTS COVOITURAGE (/api/covoiturations.php) :

- POST /api/covoiturations.php?endpoint=create - Créer un trajet
- GET /api/covoiturations.php?endpoint=all - Lister tous les trajets
- GET /api/covoiturations.php?endpoint=by-user&id=X - Trajets d'un utilisateur

ENDPOINTS RESERVATION (/api/reservations.php) :

- POST /api/reservations.php?endpoint=create - Créer une réservation
- GET /api/reservations.php?endpoint=by-user&id=X - Réservations d'un utilisateur
- DELETE /api/reservations.php?endpoint=delete&id=X - Annuler une réservation

ENDPOINTS MONGODB (/api/mongodb.php) :

- POST /api/mongodb.php?endpoint=create-car - Ajouter une voiture
- GET /api/mongodb.php?endpoint=get-cars&user_id=X
- Voitures d'un utilisateur

MÉCANISMES DE SÉCURITÉ

AUTHENTIFICATION :

- Hachage des mots de passe avec password_hash() PHP
- Sessions PHP sécurisées
- Vérification des rôles utilisateur

PROTECTION DES DONNÉES :

- Requêtes préparées (PDO) contre les injections SQL
- Validation des données côté serveur
- Échappement des données en sortie

SÉCURITÉ API :

- Vérification des sessions pour les endpoints sensibles
- Validation des paramètres d'entrée
- Gestion des erreurs sans exposition de données sensibles

FRONTEND :

- Validation JavaScript côté client
- Protection contre les XSS
- Gestion sécurisée des formulaires

DÉPLOIEMENT ET ENVIRONNEMENT

ENVIRONNEMENT DE DÉVELOPPEMENT :

- Docker Compose pour l'orchestration
- Conteneurs : PHP, MySQL, MongoDB, Apache
- Volumes persistants pour les données

ENVIRONNEMENT DE PRODUCTION :

- [À compléter après déploiement]
- Hébergement : [Plateforme choisie]
- URL de production : [Lien à ajouter]

INSTALLATION :

- Cloner le repository GitHub
- Exécuter docker-compose up
- Importer les scripts SQL
- Configurer les variables d'environnement