DOCUMENTATION TECHNIQUE

ECORIDE

Plateforme de Covoiturage Écologique

Taieb MIMOUNI

Juillet 2025

ECF - Développeur Web et Web Mobile

ARCHITECTURE DU PROJET

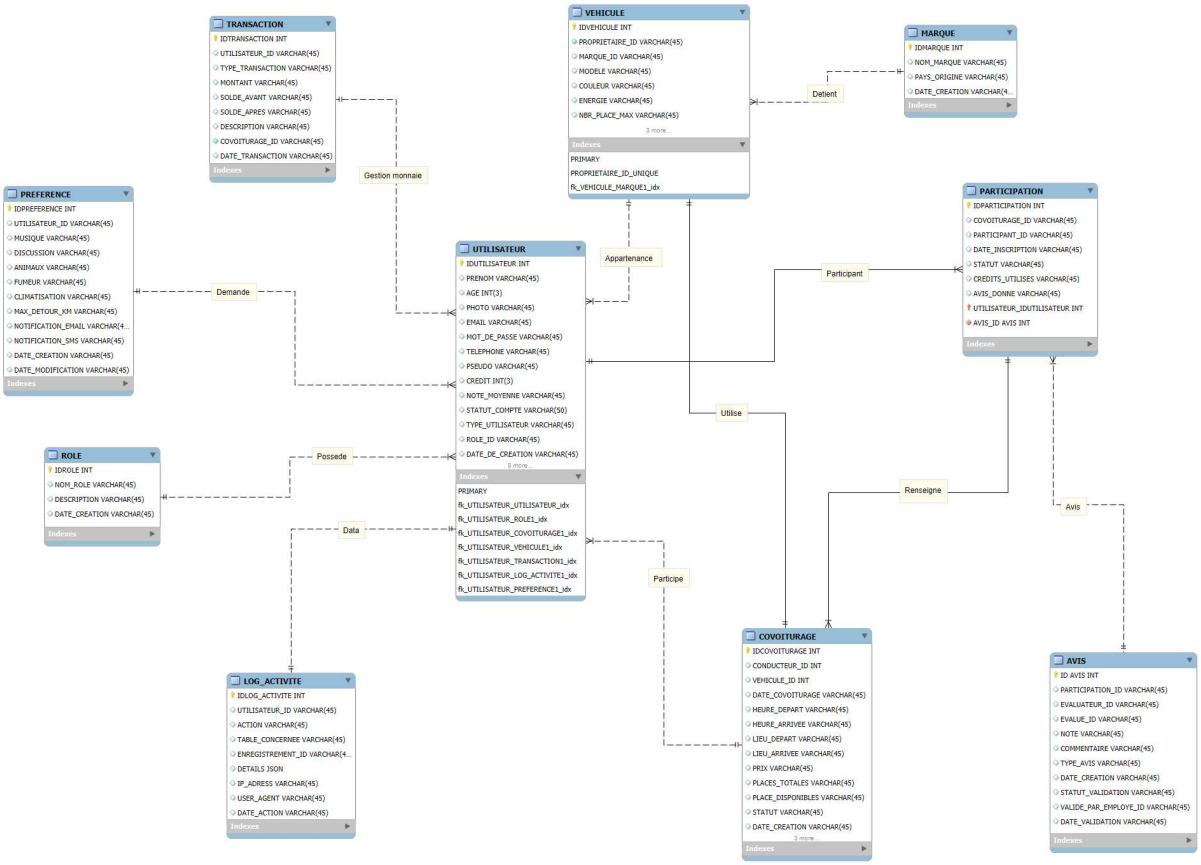
STRUCTURE GENERALE :

- 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/covoiturages.php) :

- POST /api/covoiturages.php?endpoint=create - Créer un trajet

- GET /api/covoiturages.php?endpoint=all - Lister tous les trajets

- GET /api/covoiturages.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 8.4, MySQL, MongoDB, Apache/Nginx
- Volumes persistants pour les données
- PhpMyAdmin (localhost:8081) et Mongo Express (localhost:8083)

ENVIRONNEMENT DE PRODUCTION :

- Hébergement : Heroku (PaaS)
- Base de données : JawsDB MySQL (add-on Heroku)
- URL de production : https://ecf-ecoride-mimounitaieb-87e15deda74d.herokuapp.com/templates/home/index.php
- Configuration automatique des variables d'environnement

- Détection automatique local/production via JAWSDB_URL

INSTALLATION :

DÉVELOPPEMENT LOCAL :

1. Cloner le repository GitHub : git clone https://github.com/REDEVILS24/ECF_ECORIDE

2. Exécuter docker-compose up -d

3. Accéder au site : http://localhost:8082

4. PhpMyAdmin : http://localhost:8081 (user/password)

5. Mongo Express : http://localhost:8083 (admin/password)

DÉPLOIEMENT PRODUCTION :

1. Connexion GitHub → Heroku

2. Ajout add-on JawsDB MySQL

3. Configuration automatique des variables d'environnement

4. Déploiement automatique via Git push

COMPTES DE TEST DISPONIBLES :

- Admin : admin@ecoride.com / Admin123!

- Conducteur : conducteur@test.com / Conduc123!

- Passager : passager@test.com / Pass123!

OUTILS DE GESTION :

- GitHub : https://github.com/REDEVILS24/ECF_ECORIDE
- Trello : https://trello.com/b/nBO2xQlU/ecoride