

Documentation technique EcoRide

1. Réflexions initiales et choix technologiques

- **Front-end** : Vanilla HTML, CSS, JavaScript — Typographie : Google Font Jua
- **Backend** : PHP 8.2 (Slim Framework)
- **Base de données** : MySQL (Railway) — initialisée via *1-schema.sql* et *2-seed.sql*
- **Déploiement** :
 - Front-end : Netlify
 - Back-end : Railway (Dockerfile + variables d'environnement)
- **Authentification** : JWT, CORS configuré (domaine Netlify autorisé)
- **Gestion de projet** : Trello Kanban

2. Configuration de l'environnement

Backend :

1. Cloner le dépôt : git clone ...
2. Installer les dépendances PHP :
docker run --rm -v "\${PWD}/backend:/app" -w /app composer:2 install --no-dev
3. Lancer les conteneurs :
docker compose up -d --build
4. Vérifier :
 - API : <http://localhost:8080> → doit afficher ■ API EcoRide en ligne !
 - Adminer : <http://localhost:8081> (login : ecoride / ecoride)

Frontend :

1. Se placer dans frontend/Projet_ecoride/Html
2. Lancer : python -m http.server 8000
3. Ouvrir : <http://localhost:8000>

3. Modèle conceptuel de données (MCD)

Entité	Description
users	email, password_hash, role, credits, created_at
rides	origin, destination, date_time, price, seats, seats_left
vehicles	brand, model, plate, energy
reviews	auteur, note, commentaire (si implémenté)

4. Déploiement (Production)

Front-end : Déployé sur Netlify (publish directory = frontend/Projet_ecoride/Html)

Back-end : Déployé sur Railway (Railpack). Variables d'environnement :

- DB_HOST, DB_NAME, DB_USER, DB_PASS
- CORS_ALLOW_ORIGIN=<https://golden-medovik-8f81e4.netlify.app>

Test en prod :

<https://ecoride-production-0838.up.railway.app/ping> → doit renvoyer pong