# 📄 Documentation du Projet E-commerce

## 🔍 Présentation du projet

Ce projet consiste à développer une application web e-commerce simple en 3 jours,   
en utilisant Node.js (Express) pour le backend et React.js (avec Vite et TailwindCSS) pour le frontend.  
L’objectif est double : apprendre à collaborer efficacement en équipe avec Git, et construire une application web fonctionnelle avec une architecture claire.  
Git est utilisé rigoureusement avec une branche par fonctionnalité, des commits fréquents et explicites, et des pull requests bien définies.

## 👥 Répartition de l’équipe

Dev A - Backend – Auth : Authentification JWT, cryptage des mots de passe, middlewares de protection des routes  
Dev B - Backend – App : Création des routes/API pour les utilisateurs, produits et commandes (CRUD complet)  
Dev C - Frontend – User : Interfaces utilisateurs (login, inscription, catalogue produits, commandes)  
Dev D - Frontend – Admin : Tableau de bord d’administration (gestion des utilisateurs, produits et commandes)

## 🛠️ Technologies utilisées

Backend : Node.js, Express, MongoDB Atlas, Mongoose  
Frontend : React, Vite, TailwindCSS, Axios  
Outils : Postman, MongoDB Atlas, GitHub, ChatGPT  
Versioning : Git, GitHub avec branches par fonctionnalité

## 🎯 Fonctionnalités de l’application

👤 Gestion des utilisateurs :  
- Inscription / Connexion  
- Rôles : admin et user  
- L’admin peut gérer les comptes utilisateurs  
  
🛍️ Gestion des produits :  
- Affichage du catalogue produit  
- Ajout / modification / suppression des produits (admin)  
- Visualisation des détails d’un produit  
  
📦 Commandes :  
- Un utilisateur peut passer une commande  
- Historique de ses commandes  
- Admin peut visualiser toutes les commandes

## 🧱 Architecture technique

## /e-commerce\_projet-git

## ├── backend/

## │ ├── controllers/ # Logique métier (User, Product, Order)

## │ ├── models/ # Schémas Mongoose (User.js, Product.js, Order.js)

## │ ├── routes/ # Routes Express (users.js, products.js, orders.js)

## │ ├── index.js # Point d’entrée principal (Express + MongoDB)

## │ ├── .env # Variables d’environnement (MongoDB URI, port, etc.)

## │ └── package.json # Dépendances backend

## │

## ├── frontend/

## │ ├── src/

## │ │ ├── pages/ # Pages React (Home, ProductList, UserList, OrderList, etc.)

## │ │ ├── components/ # Composants réutilisables (Navbar, etc.)

## │ │ ├── services/ # Services Axios (userService.js, productService.js, orderService.js)

## │ │ ├── App.jsx # Application principale (routage)

## │ │ ├── index.js # Entrée React

## │ │ └── index.css # Styles globaux (import Tailwind)

## │ └── package.json # Dépendances frontend

## │

## └── README.md # Documentation du projet (optionnel)

## 🔄 Gestion de version (Git)

- Une branche par fonctionnalité : feature/login, feature/products, feature/api, etc.  
- Commits clairs et réguliers  
- Pull requests validées avant merge  
- README maintenu à jour

## ✅ Modalités de test

- L’API fonctionne avec MongoDB Atlas  
- Les routes ont été testées avec Postman  
- Le frontend communique via Axios avec l’API  
- L’authentification sécurise les accès avec JWT (gérée par Dev A)

## 📄 Pour lancer le projet

Backend :  
cd backend  
npm install  
npm run dev  
  
Frontend :  
cd frontend  
npm install  
npm run dev

## 📚 IA utilisée

- Pour générer des descriptions, des documents techniques et optimiser la structure du projet  
- Documentation générée en partie à l’aide de ChatGPT

**🧩 Améliorations frontend**

Une barre de navigation (Navbar) a été ajoutée pour permettre aux utilisateurs de naviguer facilement entre les pages principales (Accueil, Produits, Commandes, etc.).  
Une page d’accueil (Home) simple a été mise en place pour accueillir l’utilisateur et présenter l’application.