Cahier des Charges : Application E-Commerce

Module : Application Informatique Encadrée

Sous-Groupe :

- ➢ BRAHIMI Rayan (G2)
- ➢ MECHEKOUR Billal (G3)
- ➢ MESSAOUDENE Said (G3)
- ➢ MERSEL Lyes (G3)

---

1. Introduction

1.1 Contexte du projet

Avec la croissance rapide du commerce en ligne, ce projet vise à développer une application e-commerce permettant aux utilisateurs de vendre, acheter et gérer des produits facilement. Conçue dans un cadre pédagogique, elle inclut une simulation de paiement pour reproduire un processus d'achat réaliste. L'application propose des fonctionnalités adaptées aux visiteurs, clients, vendeurs, et administrateurs, tout en garantissant une expérience fluide et sécurisée.

1.2 Objectifs généraux

1. Faciliter les achats et ventes en ligne via une plateforme intuitive.
2. Simplifier la gestion des produits, commandes et comptes pour les administrateurs.
3. Intégrer une simulation de paiement conforme aux exigences pédagogiques.

---

2. Besoins Fonctionnels

L'application e-commerce doit offrir des fonctionnalités adaptées à chaque type d'utilisateur, garantissant une navigation fluide, une gestion simplifiée des produits et des commandes, tout en assurant la sécurité des transactions.

2.1 Fonctionnalités par acteur

- **Visiteur**
  - Consulter le catalogue : Parcourir les produits disponibles.
  - Chercher un produit : Rechercher un produit via des critères (nom, catégorie, couleurs,...).
  - Consulter les évaluations: Voir les avis et notes sur les produits.
  - Gérer le panier : Ajouter, modifier ou supprimer des produits.
  - Créer un compte : S'inscrire en tant que client.

- **Client**
  En plus des options offertes aux visiteurs, le client pourra :
    - Passer une commande : Finaliser l'achat et effectuer le paiement.
    - Évaluer un produit : Laisser une note et un commentaire.
    - Signaler un produit : Informer d'un produit inapproprié ou erroné.
    - Gérer les notifications : Consulter et supprimer.
    - Gérer son compte : Modifier ses informations personnelles.
    - Demander à devenir vendeur : Soumettre une demande pour obtenir un statut de vendeur.
- **Vendeur**
  En plus des options offertes aux clients, le vendeur pourra :
    - Gérer ses produits : Ajouter, modifier ou supprimer des produits.
    - Répondre aux évaluations : Réagir aux commentaires clients.
    - Demander son paiement : Réclamer les gains générés par les ventes.
- **Administrateur**
    - Gérer le catalogue global (ajouter, modifier, supprimer des produits).
    - Valider, suivre et annuler les commandes.
    - Superviser les comptes utilisateurs (création, modification, suppression).
    - Modérer les avis et traiter les signalements.
    - Gérer les paiements des vendeurs.
    - Consulter une synthèse des gains.

2.2 Fonctionnalités transversales

- Implémenter un système d'authentification sécurisé pour tous les utilisateurs.
- Assurer un suivi des activités via un système de notifications (confirmation de commande, validation de paiement, etc.).
- Garantir la traçabilité des actions (historique des commandes et des ventes).
- Chatbot d'assistance : Répondre aux questions fréquentes et guider les utilisateurs.

---

3. Besoins Non Fonctionnels

L’application doit respecter des standards de performance, de sécurité et d’ergonomie pour garantir une expérience optimale et fiable pour chaque utilisateur.

3.1 Sécurité

- Mise en place d’un chiffrement des mots de passe avec un algorithme sécurisé (BCrypt).

- Gestion des sessions : Utilisation de la stratégie JWT avec rafraîchissement automatique des tokens.
- Sécurisation des routes API : Mise en place de restrictions d'accès et de contrôles d'authentification.
- Protection contre les attaques courantes (injections SQL, CSRF).
- Limitation des tentatives de connexion pour prévenir les attaques par force brute.
- Sécurité des images : Limiter les formats, analyser les fichiers et les stocker dans un répertoire sécurisé.

3.2 Performance

- Optimisation des images (compression, formats modernes)
- Mise en place de la pagination pour améliorer la vitesse de chargement des listes de produits.
- Adopter une stratégie équilibrée entre Server Side Rendering (SSR) et Client Side Rendering (CSR) pour optimiser les performances.

3.3 Ergonomie et Accessibilité

- Interface utilisateur intuitive et réactive, compatible avec les principaux navigateurs.
- Accessibilité optimisée pour les utilisateurs avec des besoins spécifiques (navigation clavier).

3.4 Possibilités d'Évolution et d'Extension

- Intégrer un système de messagerie permettant la communication directe entre clients, vendeurs et administrateurs.
- Étudier l'intégration future de services de paiement réels, notamment Stripe pour les paiements internationaux et Slick Pay pour les paiements locaux en Algérie.
- Assurer une API REST complète et documentée pour permettre une éventuelle extension vers des applications mobiles.
- Structure modulaire facilitant l'ajout de nouvelles fonctionnalités.
- Capacité à gérer une augmentation progressive du nombre d'utilisateurs et de produits.

---

4. Architecture Technique

L'architecture repose sur Next.js (full-stack) pour gérer l'interface et les API, avec Prisma pour l'interaction avec MySQL.

Technologies principales :

- Conception: UML (via Draw.io)
- Langage de programmation : TypeScript
- Front-end & Back-end : Next.js (React)
- Base de données : MySQL (via l'ORM Prisma)
- Authentification : NextAuth.js + JWT
- Stockage : Local ou Cloud (images produits)
- Paiements : Simulation (extensible vers Stripe/Slick Pay)
- Outils et gestion du projet : Git, GitHub, Postman, Discord.

Déploiement :

- **Local :** Node.js + MySQL
- **Production :** Vercel (Next.js) + Railway (MySQL)

---

5. Gestion de Projet

5.1 Répartition des tâches :

Les membres du groupe collaborent sur toutes les tâches : conception, design, développement front-end, back-end, gestion de la base de données et documentation.

5.2 Planning prévisionnel : Le projet est divisé en phases :

- Phase 1 : Conception (UML, cahier des charges)
- Phase 2 : Développement (Next.js, Prisma)
- Phase 3 : Tests (fonctionnels, API)
- Phase 4 : Finalisation et présentation

5.1 Répartition des tâches :

- Documentation : Cahier des charges, diagrammes UML, Rapport.
- Code source : Référentiel centralisé et versionné sur GitHub.
- Démo : Présentation fonctionnelle de l'application lors de la soutenance.

---

6. Conclusion

Ce projet e-commerce combine technologie moderne et travail collaboratif pour offrir une plateforme performante et évolutive. Il nous permet d'appliquer nos compétences tout en préparant le terrain pour des fonctionnalités avancées à l'avenir.