

1. Introduction

Le projet **Symfony Ecommerce** vise à développer une plateforme de commerce en ligne complète et moderne. Cette application permet aux utilisateurs d'acheter des produits en ligne tout en offrant aux administrateurs des outils pour gérer efficacement les produits, commandes, utilisateurs et paiements.

2. Objectifs

- Fournir une boutique en ligne complète avec un système de panier et de commande.
- Offrir une interface d'administration permettant la gestion des produits, commandes et utilisateurs.
- Assurer la sécurité des transactions et la protection des données utilisateurs.

3. Fonctionnalités principales

3.1 Gestion des Produits

- Ajout, modification et suppression de produits.
- Catégorisation des produits.
- Ajout d'images et de descriptions détaillées.

3.3 Système de Paiement

- Intégration du mode de paiement **Stripe** pour les transactions en ligne.

3.4 Gestion des Utilisateurs

- Inscription et authentification des utilisateurs.
- Gestion des rôles (utilisateur, administrateur, banni).
- Interface de gestion des clients pour les administrateurs.
- Système de demande de changement de mot de passe via email.

3.5 Interface d'Administration

- Tableau de bord affichant les statistiques générales.
- Gestion des produits, commandes, utilisateurs et catégories.
- Gestion des avis et commentaires des clients.
- Gestion des factures, tags et adresses.

3.6 Système de Recherche et Filtres

- Recherche avancée de produits par nom, catégorie, prix, popularité.
- Filtres dynamiques pour affiner la recherche.

3.7 Avis et Notations

- Les utilisateurs peuvent laisser des avis sur les produits.
- Affichage des notes moyennes et commentaires des clients.

3.8 Gestion des Adresses

- Possibilité d'ajouter plusieurs adresses de livraison.
- Sélection d'une adresse par défaut pour les commandes.

4. Architecture Technique

- **Framework** : Symfony (PHP)
- **Base de données** : MySQL / PostgreSQL
- **Front-end** : Twig
- **API** : API REST pour communication

5. Gestion des Utilisateurs et Rôles

- **Utilisateur standard** : Peut acheter des produits, gérer son profil et ses commandes.
- **Administrateur** : Peut gérer les produits, commandes, avis et utilisateurs.
- **Utilisateur banni** : Accès refusé à la plateforme.

6. Modélisation des Données

Relations ManyToMany

- **Product ⇔ Category** : Un produit peut appartenir à plusieurs catégories, et une catégorie peut contenir plusieurs produits.
- **Product ⇔ Tag** : Un produit peut avoir plusieurs tags, et un tag peut être associé à plusieurs produits.

Relations OneToMany

- **User ⇔ Review** : Un utilisateur peut avoir plusieurs avis, mais chaque avis appartient à un seul utilisateur.
- **Product ⇔ Review** : Un produit peut avoir plusieurs avis, mais chaque avis appartient à un seul produit.
- **Order ⇔ OrderItem** : Une commande peut contenir plusieurs articles, mais un article appartient à une seule commande.
- **Cart ⇔ CartItem** : Un panier peut contenir plusieurs articles, mais un article appartient à un seul panier.
- **User ⇔ Order** : Un utilisateur peut passer plusieurs commandes, mais chaque commande appartient à un seul utilisateur.

- **User ⇔ Cart** : Un utilisateur peut avoir plusieurs paniers, mais chaque panier appartient à un seul utilisateur.
- **User ⇔ Invoice** : Un utilisateur peut avoir plusieurs factures, mais chaque facture est associée à un seul utilisateur.
- **User ⇔ Address** : Un utilisateur peut avoir plusieurs adresses, mais chaque adresse appartient à un seul utilisateur.
- **Invoice ⇔ Orders** : Une facture est liée à une commande unique.

Sous-classes de Product

- **DigitalProduct** : Représente les produits numériques (e-books, logiciels, etc.).
- **PhysicalProduct** : Représente les produits physiques (vêtements, gadgets, etc.).

7. Planning

Phase 1 : Analyse et conception

- Définition des besoins fonctionnels et techniques.
- Conception du schéma de base de données.

Phase 2 : Développement

- Implémentation des entités et relations.
- Développement du back-end et des fonctionnalités principales.
- Création de l'interface utilisateur.

Phase 3 : Tests et validation

- Tests unitaires et fonctionnels.
- Correction des bugs et optimisation des performances.