

Cahier des Charges : Application Web Express Market

1. Présentation du projet

1.1 Contexte

Dans un contexte de digitalisation croissante du secteur agroalimentaire et d'évolution des habitudes de consommation, le projet Express Market vise à créer une plateforme permettant aux consommateurs d'acheter des produits alimentaires en ligne sans avoir à se déplacer.

1.2 Objectifs du projet

- Développer une application web e-commerce dans le secteur agroalimentaire
- Permettre aux utilisateurs de commander des produits alimentaires depuis leur domicile
- Offrir une expérience d'achat simple, intuitive et sécurisée
- Faciliter la livraison à domicile des produits commandés

1.3 Public cible

- Personnes actives avec peu de temps pour faire leurs courses
- Personnes à mobilité réduite
- Habitants des zones urbaines et péri-urbaines
- Consommateurs intéressés par des produits agroalimentaires de qualité

2. Spécifications fonctionnelles

2.1 Fonctionnalités utilisateurs

- Création et gestion de compte utilisateur
- Navigation dans le catalogue de produits par catégories
- Recherche de produits avec filtres (prix, catégorie, disponibilité)
- Ajout de produits au panier
- Gestion du panier d'achat (modification quantités, suppression d'articles)
- Processus de commande et paiement sécurisé
- Suivi des commandes en cours
- Accès à l'historique des commandes
- Système de notation et d'avis sur les produits
- Gestion des adresses de livraison

2.2 Fonctionnalités administrateur

- Tableau de bord d'administration
- Gestion du catalogue de produits (ajout, modification, suppression)
- Gestion des catégories
- Suivi et traitement des commandes
- Gestion des stocks
- Gestion des promotions et offres spéciales
- Visualisation des statistiques de vente
- Gestion des utilisateurs
- Configuration des frais de livraison

3. Spécifications techniques

3.1 Architecture technique

- **Frontend:** React.js, Bootstrap, CSS, Axios
- **Backend:** Laravel (PHP)
- **Base de données:** MySQL
- **API:** RESTful
- **Authentification:** Laravel Sanctum/JWT
- **Palements:** Intégration avec Stripe/PayPal
- **Hébergement:** AWS/DigitalOcean

3.2 Structure de la base de données

- Table Users (utilisateurs)
- Table Products (produits)
- Table Categories (catégories)
- Table Orders (commandes)
- Table Order_Items (éléments de commande)
- Table Addresses (adresses)
- Table Reviews (avis)
- Table Cart_Items (éléments du panier)
- Table Promotions (promotions)

3.3 Outils de développement

- **Gestion de versions:** Git (GitHub/GitLab)
- **Environnement de développement:** Docker

- **Tests:** PHPUnit, Jest
- **CI/CD:** GitHub Actions/Jenkins
- **Gestion de projet:** Jira/Trello

4. Interface utilisateur

4.1 Pages principales

- Page d'accueil avec produits en vedette et promotions
- Page de catalogue avec filtres
- Page détaillée de produit
- Page panier
- Pages de checkout (paiement)
- Page de profil utilisateur
- Page de suivi de commande
- Pages d'authentification (connexion/inscription)

4.2 Design et ergonomie

- Design responsive (compatible mobile, tablette, desktop)
- Interface intuitive et conviviale
- Temps de chargement rapide
- Palette de couleurs en accord avec l'univers alimentaire (vert, blanc, couleurs fraîches)
- Utilisation d'icônes cohérentes
- Photographies de haute qualité pour les produits

5. Sécurité

5.1 Sécurisation des données

- Chiffrement des mots de passe (bcrypt)
- Protection contre les injections SQL
- Validation des entrées utilisateur
- Protection CSRF
- Authentification sécurisée
- Chiffrement des données de paiement

5.2 Conformité

- Conformité RGPD

- Politique de confidentialité
- Conditions générales de vente
- Mentions légales

6. Processus de livraison

6.1 Options de livraison

- Livraison standard (24-48h)
- Livraison express (même jour)
- Sélection de créneaux horaires

6.2 Suivi de commande

- Notification par email des étapes de la commande
- Suivi en temps réel de l'état de la commande
- Historique des commandes dans l'espace client

7. Planning de développement

7.1 Phases du projet

- **Phase 1** (2 semaines): Analyse et conception
- **Phase 2** (4 semaines): Développement frontend et backend initial
- **Phase 3** (2 semaines): Intégration et tests
- **Phase 4** (1 semaine): Déploiement et finalisation

7.2 Méthodologie

- Approche Agile (Scrum)
- Sprints de 2 semaines
- Revues régulières avec le client

8. Tests et validation

8.1 Types de tests

- Tests unitaires
- Tests d'intégration
- Tests fonctionnels
- Tests de performance
- Tests de sécurité

- Tests d'utilisabilité

8.2 Critères d'acceptation

- Fonctionnalités conformes aux spécifications
- Interface responsive sur tous les appareils cibles
- Temps de réponse < 3 secondes
- Passage de tous les tests automatisés
- Sécurité des transactions validée

9. Maintenance et évolutions

9.1 Maintenance

- Mises à jour de sécurité
- Correction de bugs
- Surveillance des performances
- Sauvegarde régulière des données

9.2 Évolutions futures potentielles

- Application mobile native (iOS/Android)
- Intégration avec des producteurs locaux
- Système de fidélité
- Abonnements à des paniers récurrents
- Intelligence artificielle pour recommandations personnalisées

10. Budget et ressources

10.1 Ressources humaines

- 1 Chef de projet
- 2 Développeurs backend (Laravel)
- 2 Développeurs frontend (React)
- 1 Designer UI/UX
- 1 Testeur QA

10.2 Infrastructures techniques

- Serveur de développement
- Serveur de production
- Système de sauvegarde

- Outils de monitoring

11. Livrables

11.1 Documentation

- Documentation technique
- Documentation utilisateur
- Documentation administrateur
- Spécifications API

11.2 Code source

- Repository Git avec l'ensemble du code
- Scripts de déploiement
- Scripts de migration de base de données
- Fichiers de configuration

11.3 Application

- Application web fonctionnelle
- Interface d'administration
- Base de données initiale avec données de test