



ÉCOLE MAROCAINE DES SCIENCES DE L'INGÉNIEUR (EMSI)

RAPPORT DE PROJET DE FIN D'ANNÉE

Nutstree

Application e-commerce de vente de fruits secs

Réalisé par : Ghazaouni Ryad et Mouhlal Akram

Encadré par : Ahmed Azouaoui

Année universitaire 2024-2025

Remerciements

Nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de fin d'études.

Nous tenons tout particulièrement à remercier notre encadrant, Monsieur **Ahmed Azouaoui**, pour son accompagnement, ses conseils avisés et sa disponibilité tout au long de ce travail. Son expertise et sa rigueur ont grandement enrichi notre réflexion et notre démarche.

Nos remerciements vont également à l'ensemble du corps professoral et administratif de l'**École Marocaine des Sciences de l'Ingénieur (EMSI)**, pour la qualité de l'enseignement dispensé et le cadre de travail stimulant offert durant notre formation.

Nous exprimons aussi notre reconnaissance envers nos familles et nos proches, pour leur soutien moral constant, leur patience et leurs encouragements durant ces mois de travail intensif.

Enfin, nous remercions tous ceux qui ont, d'une manière ou d'une autre, contribué à la réussite de ce projet.

Ghazaouni Ryad et Mouhlal Akram

Résumé

Ce rapport présente le développement du projet **Nutstree**, une plateforme e-commerce spécialisée dans la vente en ligne de fruits secs. L'objectif principal de ce projet est de concevoir une application web moderne, intuitive et performante, permettant aux clients de consulter les produits, de passer commande et de bénéficier d'un service de livraison avec paiement sécurisé à la réception des produits.

Dans un contexte où le commerce électronique connaît une croissance rapide, notamment au Maroc, ce projet s'inscrit dans une logique d'innovation numérique en répondant aux besoins spécifiques des consommateurs en matière d'alimentation saine et de praticité d'achat.

Le travail réalisé couvre l'ensemble des étapes de développement logiciel, depuis l'analyse des besoins jusqu'au déploiement, en passant par la conception technique, l'implémentation backend et frontend, les tests et la validation. La solution a été développée à l'aide du framework **Laravel** pour la partie serveur, de **MySQL** pour la gestion de la base de données, et de **Tailwind CSS** et **JavaScript** pour l'interface utilisateur.

En plus des fonctionnalités classiques d'un site e-commerce (authentification, panier, gestion des commandes, suivi de livraison), Nutstree intègre également un espace d'administration complet permettant de gérer les produits, les utilisateurs, les commandes, ainsi que les livraisons. Une attention particulière a été portée à la simplicité du processus de paiement, effectué exclusivement en espèces à la livraison, afin de garantir la sécurité et la confiance des clients.

Ce rapport reflète notre démarche d'ingénieurs en formation et témoigne des compétences acquises tout au long de notre cursus à l'EMSI, aussi bien en gestion de projet informatique qu'en développement d'applications web.

Ghazaouni Ryad et Mouhlal Akram

Table des matières

Remerciements	1
Résumé	2
1 Introduction Générale	7
1.1 Contexte et motivation	7
1.2 Présentation du projet Nutstree	7
1.3 Objectifs du projet	7
1.4 Technologies utilisées	8
1.4.1 Backend	8
1.4.2 Base de données	8
1.4.3 Frontend	9
1.5 Organisation du rapport	9
2 État de l'art	10
2.1 Le e-commerce aujourd'hui	10
2.2 Solutions existantes	11
2.3 Laravel, Tailwind CSS et paiement en espèces	12
2.3.1 Laravel	12
2.3.2 Tailwind CSS	12
2.3.3 Paiement en espèces à la livraison	12
2.4 Comparaison entre Frameworks et CMS	13
2.4.1 Qu'est-ce qu'un Framework ?	13
2.4.2 Qu'est-ce qu'un CMS ?	13
2.4.3 Comparaison des deux approches	14
2.4.4 Choix pour le projet Nutstree	14
3 Analyse des besoins	15
3.1 Besoins fonctionnels	15

3.1.1	Besoins fonctionnels côté client	15
3.1.2	Besoins fonctionnels côté administration	16
3.2	Besoins non fonctionnels	16
3.2.1	Performance	16
3.2.2	Sécurité	17
3.2.3	Disponibilité	17
3.2.4	Scalabilité	17
3.2.5	Compatibilité	17
3.2.6	Ergonomie et expérience utilisateur	17
3.2.7	Maintenabilité	17
3.2.8	Conformité légale	18
3.3	Modélisation UML	18
3.3.1	Diagramme de cas d'utilisation (Use Case)	19
3.3.2	Diagramme de séquence	20
3.3.3	Diagramme de classes	21
3.4	Architecture du système	21
3.4.1	Vue d'ensemble	21
3.4.2	Composants principaux	22
3.5	Conception des interfaces	22
3.5.1	Principes de conception	22
3.5.2	Interfaces principales	23
3.5.3	Maquettes des interfaces	23
3.6	Choix techniques	24
3.6.1	Langage et Framework Backend	24
3.6.2	Base de données	24
3.6.3	Technologies Frontend	25
3.6.4	Gestion des dépendances et outils de développement	25
3.6.5	Choix d'hébergement	25
4	Implémentation	26
4.1	Mise en place du backend	26
4.1.1	Installation et configuration de Laravel	26
4.1.2	Structure MVC	26
4.1.3	Création des entités principales	26
4.1.4	Gestion des relations	27
4.1.5	Authentification et rôles	27

4.1.6	API interne (facultatif)	27
4.2	Développement des fonctionnalités	27
4.3	Développement des fonctionnalités	27
4.3.1	Inscription et authentification des utilisateurs	27
4.3.2	Gestion des produits	28
4.3.3	Panier et commande	28
4.3.4	Paiement à la livraison (Cash on Delivery)	28
4.3.5	Suivi des commandes	28
4.3.6	Commentaires et avis sur les produits	28
4.3.7	Gestion des utilisateurs, commandes et livraisons (Admin) . . .	29
4.3.8	Gestion du blog (optionnelle)	29
4.4	Sécurité	29
4.4.1	Protection des données utilisateurs	29
4.4.2	Validation des entrées	30
4.4.3	Gestion des rôles et des autorisations	30
4.4.4	Sécurisation des routes	30
4.4.5	Sécurité des fichiers et du serveur	30
5	Tests et Validation	31
5.1	Méthodologie de test	31
5.1.1	Types de tests réalisés	31
5.1.2	Environnement de test	32
5.1.3	Outils et bonnes pratiques	32
5.2	Résultats des tests	32
5.2.1	Tests unitaires	32
5.2.2	Tests fonctionnels	32
5.2.3	Tests d'intégration	33
5.2.4	Tests manuels	33
5.2.5	Conclusion	33
6	Déploiement et maintenance	34
6.1	Environnement de production	34
6.1.1	Serveur d'hébergement	34
6.1.2	Déploiement de l'application	34
6.1.3	Nom de domaine et accès HTTPS	35
6.1.4	Surveillance et logs	35

6.2	Procédure de déploiement	35
6.2.1	Préparation de l'environnement	35
6.2.2	Transfert des fichiers	35
6.2.3	Installation des dépendances	36
6.2.4	Configuration de l'application	36
6.2.5	Migrations et seeders	36
6.2.6	Cache et optimisation	36
6.2.7	Gestion des permissions	37
6.2.8	Tests post-déploiement	37
6.3	Maintenance continue	37
6.3.1	Maintenance corrective	37
6.3.2	Maintenance évolutive	37
6.3.3	Maintenance préventive	38
6.3.4	Documentation et support	38
7	Conclusion et perspectives	39
A	Annexes	41
A.1	Captures d'écran de l'application	41
Bibliographie		43

Chapitre 1

Introduction Générale

1.1 Contexte et motivation

Avec l'essor fulgurant du commerce électronique à l'échelle mondiale, de plus en plus de consommateurs adoptent l'achat en ligne pour sa praticité, sa rapidité et la diversité des offres disponibles. Au Maroc, ce phénomène connaît également une croissance notable, favorisée par l'amélioration des infrastructures numériques et l'évolution des comportements d'achat.

Le secteur agroalimentaire, et plus particulièrement la commercialisation des fruits secs, constitue un marché porteur où l'innovation digitale peut apporter une réelle valeur ajoutée. En effet, les consommateurs recherchent des produits de qualité, accessibles facilement et avec un service de livraison fiable.

1.2 Présentation du projet Nutstree

Dans ce contexte, le projet **Nutstree** a pour ambition de développer une plateforme e-commerce spécialisée dans la vente de fruits secs, permettant aux clients de naviguer, choisir et commander en ligne, tout en offrant un mode de paiement simple et sécurisé, exclusivement en espèces à la livraison. Cette approche vise à répondre aux contraintes spécifiques du marché local et à instaurer une relation de confiance avec les clients.

1.3 Objectifs du projet

Les objectifs principaux du projet Nutstree sont les suivants :

- Développer une plateforme web moderne, intuitive et responsive.

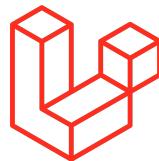
- Permettre aux clients de consulter facilement les produits et leurs fiches détaillées.
- Faciliter la gestion du panier et le passage de commande.
- Assurer un système de paiement sécurisé en cash à la livraison.
- Offrir un espace d'administration complet pour gérer les utilisateurs, produits, commandes et livraisons.
- Garantir un suivi efficace des commandes et des livraisons.

1.4 Technologies utilisées

Pour atteindre les objectifs du projet Nutstree, un ensemble de technologies robustes et performantes a été sélectionné. Ces choix permettent d'assurer une architecture cohérente et évolutive ainsi qu'une expérience utilisateur optimale.

1.4.1 Backend

- **Laravel** : Framework PHP moderne, élégant et puissant, facilitant le développement d'applications web robustes et sécurisées.



1.4.2 Base de données

- **MySQL** : Système de gestion de base de données relationnelle, performant et largement utilisé pour la gestion fiable des données.



1.4.3 Frontend

- **Tailwind CSS** : Framework CSS utilitaire permettant de créer des interfaces modernes, responsives et personnalisables rapidement.
- **JavaScript** : Langage de programmation utilisé pour rendre la plateforme interactive et dynamique.



1.5 Organisation du rapport

Ce rapport est organisé de la manière suivante :

- **État de l'art** : présentation des solutions existantes et technologies utilisées.
- **Analyse des besoins** : identification des fonctionnalités et contraintes du système.
- **Conception** : description de l'architecture et des modèles de données.
- **Implémentation** : détails du développement backend et frontend.
- **Tests et validation** : présentation des méthodes de test et résultats obtenus.
- **Conclusion et perspectives** : bilan du projet et axes d'amélioration futurs.

Chapitre 2

État de l'art

2.1 Le e-commerce aujourd’hui

Le commerce électronique, ou e-commerce, désigne l’ensemble des transactions commerciales effectuées via des réseaux électroniques, principalement Internet. Depuis son émergence dans les années 1990, le e-commerce a connu une croissance exponentielle à l’échelle mondiale, transformant profondément les modes de consommation et les stratégies commerciales des entreprises.

Aujourd’hui, le e-commerce représente un secteur clé de l’économie numérique, avec des milliards de dollars de ventes réalisées chaque année. Cette croissance est portée par plusieurs facteurs, notamment la démocratisation de l'accès à Internet, la généralisation des smartphones, ainsi que l'évolution des habitudes des consommateurs, de plus en plus enclins à effectuer leurs achats en ligne pour gagner du temps et accéder à une offre diversifiée.

Parmi les tendances actuelles, on observe une montée en puissance des plateformes mobiles, l'intégration des réseaux sociaux dans les parcours d'achat, ainsi que le développement de solutions de paiement sécurisées et diversifiées. En parallèle, les attentes des consommateurs en matière d'expérience utilisateur se sont accrues, avec une exigence forte pour des interfaces intuitives, des processus d'achat simplifiés et un service client réactif.

Cependant, le e-commerce doit également faire face à des défis majeurs tels que la sécurisation des données personnelles, la gestion logistique des livraisons, et la fidélisation des clients dans un environnement très concurrentiel.

Dans le contexte marocain, le e-commerce est en pleine expansion, soutenu par les politiques gouvernementales visant à encourager la transformation digitale. Néanmoins,

certains freins subsistent, notamment en ce qui concerne la confiance dans les paiements en ligne, ce qui rend le paiement à la livraison particulièrement populaire.

Ce constat oriente le développement de projets comme Nutstree, qui cherchent à adapter les solutions e-commerce aux spécificités locales tout en tirant parti des technologies modernes pour offrir une expérience utilisateur de qualité.

2.2 Solutions existantes

Dans le domaine du e-commerce, de nombreuses plateformes ont vu le jour afin de répondre aux besoins variés des consommateurs et des commerçants. Parmi les solutions les plus populaires à l'échelle mondiale, on retrouve des géants tels qu'Amazon, eBay, et Alibaba, qui proposent une large gamme de produits avec des systèmes sophistiqués de gestion des stocks, des paiements et des livraisons.

Ces plateformes se caractérisent par des interfaces utilisateurs bien conçues, une grande variété de moyens de paiement, ainsi que des mécanismes avancés de recommandation et de gestion des avis clients. Elles investissent également dans des technologies telles que l'intelligence artificielle pour améliorer l'expérience client et optimiser la logistique.

Au niveau local, notamment au Maroc, plusieurs solutions e-commerce sont en développement pour s'adapter aux spécificités du marché. Des plateformes comme Jumia, Hmizate ou encore des sites spécialisés dans l'agroalimentaire proposent des services similaires, mais rencontrent parfois des limitations liées à la gestion des paiements en ligne, la fiabilité des livraisons ou encore la diversification des catalogues produits.

Concernant la vente de produits alimentaires, et plus spécifiquement de fruits secs, les offres restent souvent fragmentées, avec une prédominance de commerces physiques ou de petites boutiques en ligne ne disposant pas toujours d'une interface utilisateur performante ou de fonctionnalités avancées comme le suivi de commande ou la gestion des avis.

Le projet Nutstree vise à combler ces lacunes en proposant une plateforme spécialisée, intuitive et adaptée aux attentes des clients marocains, avec une attention particulière portée à la gestion simple des commandes et au paiement en espèces à la livraison, répondant ainsi aux contraintes du marché local.

Cette étude des solutions existantes nous a permis d'identifier les bonnes pratiques à adopter ainsi que les aspects à améliorer, qui ont été intégrés dans la conception de Nutstree.

2.3 Laravel, Tailwind CSS et paiement en espèces

Pour le développement de la plateforme Nutstree, plusieurs choix technologiques ont été retenus afin d'assurer robustesse, performance et une expérience utilisateur optimale.

2.3.1 Laravel

Laravel est un framework PHP open source reconnu pour sa simplicité d'utilisation et sa puissance. Il facilite le développement d'applications web grâce à une architecture MVC (Modèle-Vue-Contrôleur) claire et à une multitude de fonctionnalités intégrées, telles que la gestion des routes, l'authentification, la gestion des bases de données via Eloquent ORM, et un système de migrations pour le schéma de la base de données.

Laravel offre également un écosystème riche avec des outils de tests automatisés, de gestion des tâches et de sécurité renforcée, ce qui en fait un choix privilégié pour les projets nécessitant une maintenance évolutive et une forte fiabilité.

2.3.2 Tailwind CSS

Pour le design frontend, Nutstree utilise Tailwind CSS, un framework CSS utilitaire qui permet de construire rapidement des interfaces modernes et responsives. Contrairement aux frameworks traditionnels basés sur des composants prédéfinis, Tailwind offre une grande flexibilité en permettant de composer les styles directement dans les classes HTML.

Cette approche favorise la personnalisation fine du design tout en optimisant la taille finale des feuilles de styles grâce à un processus de purge des classes inutilisées. Ainsi, l'interface utilisateur est à la fois esthétique, cohérente et performante, s'adaptant parfaitement aux différents types d'écrans (ordinateurs, tablettes, smartphones).

2.3.3 Paiement en espèces à la livraison

Contrairement à de nombreuses plateformes e-commerce qui intègrent des solutions de paiement en ligne, Nutstree a choisi de privilégier le paiement en espèces à la livraison. Ce choix répond aux particularités du marché marocain, où la confiance envers les paiements en ligne reste encore limitée pour une partie des consommateurs.

Le paiement en cash facilite ainsi l'accès au service pour un plus grand nombre de clients, tout en assurant une transaction sécurisée et transparente. Cette

méthode impose cependant une gestion rigoureuse des commandes et des livraisons, afin de garantir la satisfaction client et la bonne tenue du suivi logistique.

En combinant ces technologies et ce mode de paiement, Nutstree vise à proposer une solution adaptée aux besoins locaux, tout en garantissant qualité, simplicité et fiabilité.

2.4 Comparaison entre Frameworks et CMS

Dans le développement de sites web, deux approches principales s'offrent aux développeurs : l'utilisation d'un **framework** ou d'un **Système de Gestion de Contenu (CMS)**. Chacune de ces solutions présente des avantages et des limites en fonction des besoins du projet.

2.4.1 Qu'est-ce qu'un Framework ?

Un framework est un ensemble d'outils et de bibliothèques fournissant une structure de base pour développer une application web. Il propose une architecture définie (souvent le modèle MVC — Modèle-Vue-Contrôleur), des fonctionnalités réutilisables (gestion des routes, bases de données, authentification, etc.), et facilite la mise en place d'une application personnalisée.

Les frameworks, comme Laravel (PHP), Django (Python) ou Ruby on Rails (Ruby), offrent une grande flexibilité et permettent de construire des solutions sur mesure, adaptées précisément aux besoins fonctionnels et techniques du projet. Cependant, leur utilisation nécessite des compétences en programmation plus avancées et un temps de développement souvent plus long.

2.4.2 Qu'est-ce qu'un CMS ?

Un CMS (Content Management System) est une application prête à l'emploi qui permet de créer, gérer et publier du contenu sur un site web sans nécessiter de connaissances approfondies en programmation. Des exemples populaires incluent WordPress, Joomla ou Drupal.

Les CMS offrent une interface utilisateur intuitive, une gestion simplifiée des contenus, ainsi qu'un grand nombre de thèmes et plugins permettant d'étendre rapidement les fonctionnalités du site (boutique en ligne, blog, forum, etc.). Ils sont particulièrement adaptés aux projets où la rapidité de mise en place et la gestion autonome du contenu sont prioritaires.

2.4.3 Comparaison des deux approches

- **Flexibilité** : Les frameworks offrent une plus grande liberté pour créer des fonctionnalités sur mesure, alors que les CMS sont limités aux options proposées par leurs thèmes et plugins.
- **Facilité d'utilisation** : Les CMS sont généralement plus accessibles aux utilisateurs non techniques, tandis que les frameworks nécessitent des compétences en développement.
- **Performances** : Une application développée avec un framework peut être optimisée plus finement, ce qui peut améliorer les performances par rapport à un CMS souvent plus générique.
- **Maintenance** : Les CMS bénéficient d'une large communauté et de mises à jour régulières facilitant la maintenance, tandis que les frameworks demandent une gestion plus technique.
- **Sécurité** : Les deux solutions nécessitent une attention particulière à la sécurité. Les CMS sont souvent des cibles fréquentes d'attaques en raison de leur popularité, mais les frameworks peuvent aussi être vulnérables si le code n'est pas bien écrit.

2.4.4 Choix pour le projet Nutstree

Pour Nutstree, le choix s'est porté sur l'utilisation du framework Laravel afin de bénéficier d'une grande flexibilité dans la conception d'une plateforme e-commerce spécifique aux besoins locaux, notamment pour gérer le paiement en espèces et le suivi des livraisons. Cette approche permet de développer une application plus adaptée, modulaire et évolutive que ce qu'un CMS pourrait offrir.

Chapitre 3

Analyse des besoins

3.1 Besoins fonctionnels

L'analyse des besoins fonctionnels constitue une étape fondamentale dans la réussite du projet Nutstree. Elle permet de définir précisément les services que le système doit offrir afin de répondre aux attentes des utilisateurs et aux objectifs métier.

Pour Nutstree, les besoins fonctionnels ont été identifiés selon les deux profils principaux d'utilisateurs : les clients (utilisateurs finaux) et les administrateurs (gestionnaires de la plateforme).

3.1.1 Besoins fonctionnels côté client

- **Inscription et authentification** : Permettre aux clients de créer un compte sécurisé et de se connecter pour accéder à leur espace personnel.
- **Navigation et recherche de produits** : Offrir une interface intuitive permettant la consultation des différentes catégories de fruits secs et la recherche par critères (nom, prix, catégorie).
- **Consultation des fiches produits** : Fournir des fiches détaillées présentant les caractéristiques, le prix, la disponibilité et les avis des autres clients.
- **Gestion du panier** : Permettre l'ajout, la modification ou la suppression des produits dans le panier avant validation de la commande.
- **Passage de commande** : Proposer un processus clair pour finaliser la commande avec la possibilité de choisir le mode de paiement (paiement en espèces à la livraison).
- **Suivi des commandes** : Offrir un espace où le client peut consulter le statut de ses commandes (en attente, en cours, livré, annulé).

- **Gestion des avis** : Permettre aux clients de laisser des commentaires et des évaluations sur les produits achetés.

3.1.2 Besoins fonctionnels côté administration

- **Gestion des utilisateurs** : Créer, modifier, supprimer et gérer les profils des clients, administrateurs et livreurs.
- **Gestion des produits** : Ajouter, modifier, classer par catégorie, ou supprimer les produits proposés sur la plateforme.
- **Gestion des commandes** : Suivre et modifier le statut des commandes, gérer les annulations et les retours.
- **Gestion des livraisons** : Organiser les livraisons, attribuer des commandes aux livreurs et assurer le suivi logistique.
- **Gestion des avis** : Modérer les commentaires et évaluations pour garantir leur pertinence et leur conformité.
- **Gestion du contenu** : Publier et mettre à jour les articles de blog liés à la plateforme.

Cette définition claire des besoins fonctionnels garantit que le développement de Nutstree sera aligné avec les attentes des utilisateurs et les contraintes métier, facilitant ainsi la conception et la mise en œuvre d'une solution adaptée et efficace.

3.2 Besoins non fonctionnels

Les besoins non fonctionnels correspondent aux contraintes et exigences qualitatives auxquelles la plateforme Nutstree doit répondre pour assurer sa performance, sa fiabilité et sa convivialité. Ces besoins sont essentiels pour garantir une expérience utilisateur satisfaisante et la pérennité du système.

3.2.1 Performance

La plateforme doit être capable de gérer un grand nombre de connexions simultanées sans dégradation significative des temps de réponse, notamment lors des pics de fréquentation. Les temps de chargement des pages doivent rester faibles pour éviter la perte d'utilisateurs.

3.2.2 Sécurité

La sécurité des données personnelles des utilisateurs et des informations relatives aux commandes est primordiale. Le système doit intégrer des mécanismes robustes d'authentification, de gestion des sessions et de protection contre les attaques courantes (injections SQL, cross-site scripting, etc.). De plus, la confidentialité des informations liées aux paiements en espèces doit être assurée.

3.2.3 Disponibilité

Nutstree doit garantir une disponibilité maximale, avec un temps d'indisponibilité minimal, pour permettre aux clients d'accéder à la plateforme à tout moment. Des mécanismes de sauvegarde et de récupération des données doivent être mis en place pour prévenir toute perte d'information.

3.2.4 Scalabilité

La solution doit pouvoir évoluer facilement pour intégrer de nouvelles fonctionnalités ou supporter une augmentation du nombre d'utilisateurs sans nécessiter de refonte majeure.

3.2.5 Compatibilité

Le site doit être compatible avec les principaux navigateurs web modernes (Chrome, Firefox, Edge, Safari) ainsi qu'avec les différents types d'appareils (ordinateurs, tablettes, smartphones) grâce à un design responsive.

3.2.6 Ergonomie et expérience utilisateur

L'interface doit être intuitive et facile à utiliser, permettant une navigation fluide et un accès rapide aux fonctionnalités principales. Le design doit être attractif et cohérent avec l'image de marque de Nutstree.

3.2.7 Maintenabilité

Le code source doit être organisé, documenté et modulable afin de faciliter les mises à jour, la correction des bugs et l'évolution future de la plateforme.

3.2.8 Conformité légale

La plateforme doit respecter les réglementations en vigueur concernant la protection des données personnelles (par exemple, la loi marocaine sur la protection des données personnelles) ainsi que les normes relatives au commerce électronique.

En intégrant ces besoins non fonctionnels dans la conception et le développement de Nutstree, le projet pourra garantir une solution fiable, sécurisée et agréable à utiliser pour ses utilisateurs.

3.3 Modélisation UML

La modélisation UML (Unified Modeling Language) est une étape essentielle pour représenter de manière visuelle et structurée le fonctionnement et l'architecture du système Nutstree. Elle permet de clarifier les interactions entre les acteurs et le système ainsi que la structure des données.

3.3.1 Diagramme de cas d'utilisation (Use Case)

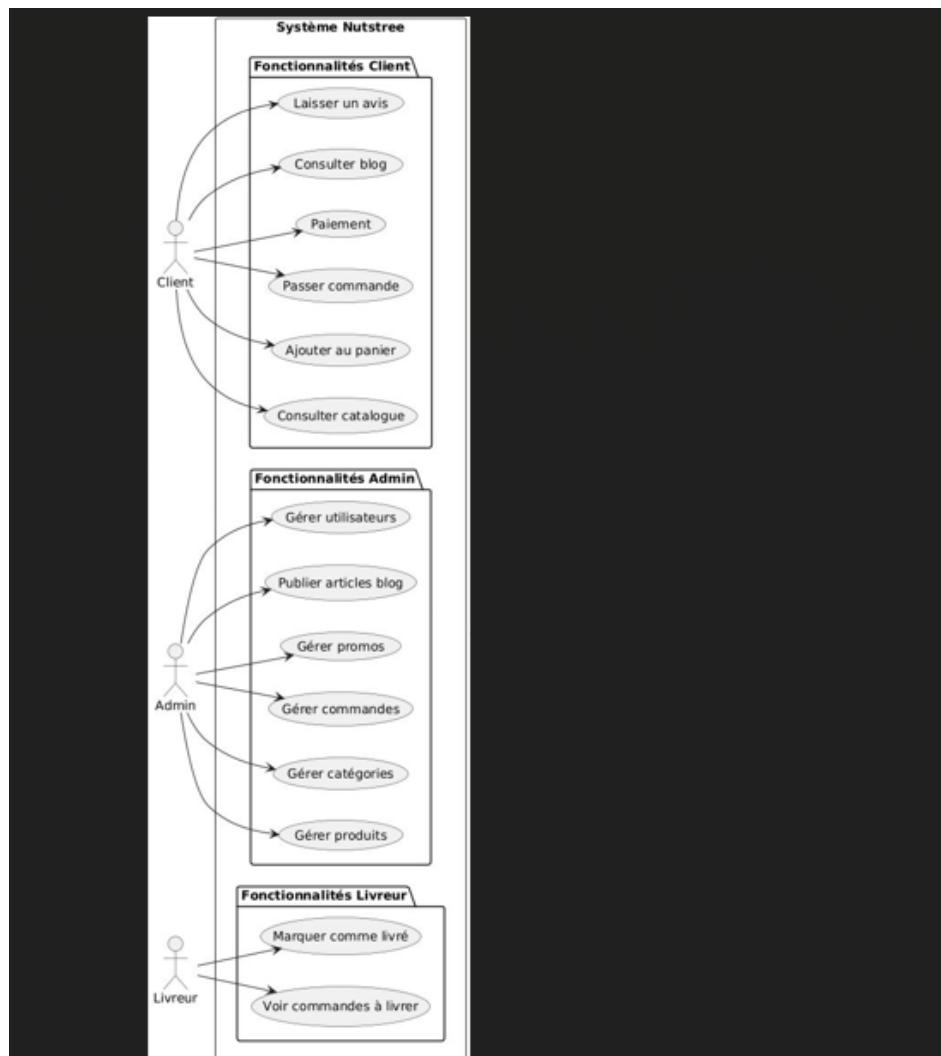


FIGURE 3.1 – Diagramme de cas d'utilisation

3.3.2 Diagramme de séquence

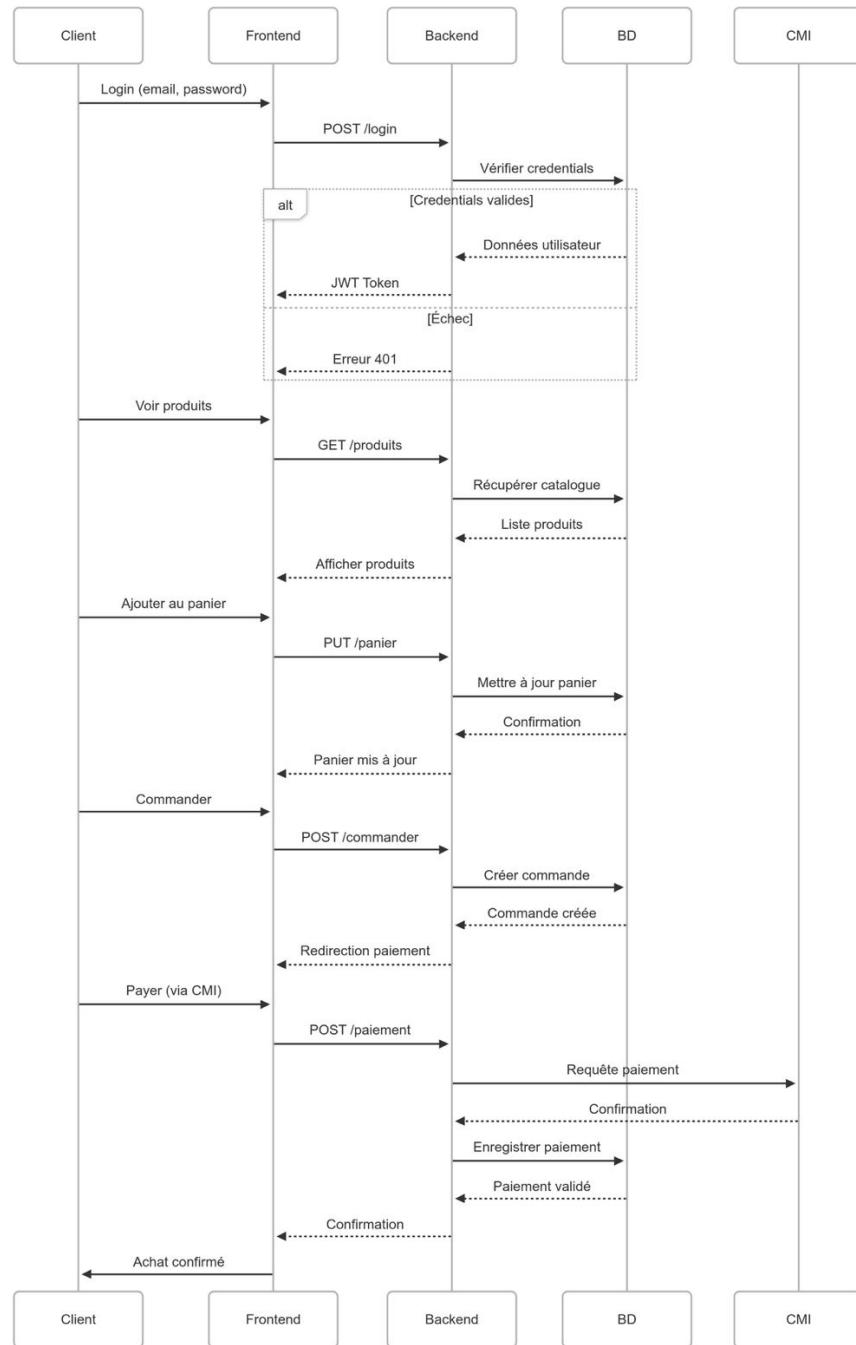


FIGURE 3.2 – Diagramme de séquence

3.3.3 Diagramme de classes

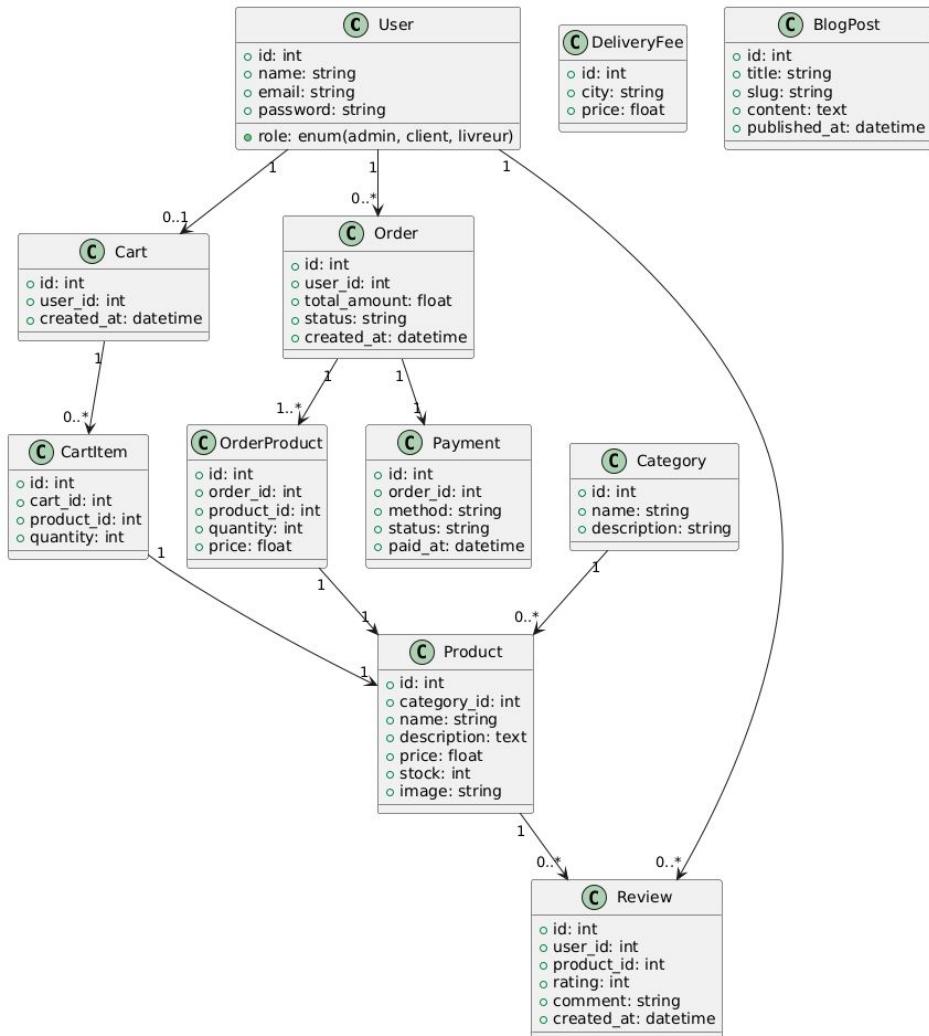


FIGURE 3.3 – Diagramme de classes

3.4 Architecture du système

L'architecture du système Nutstree repose sur une structure en couches, typique des applications web modernes. Cette approche permet de séparer clairement les responsabilités, de faciliter la maintenance, l'évolutivité du système, et d'assurer une meilleure organisation du code.

3.4.1 Vue d'ensemble

L'application est construite selon une architecture MVC (Modèle - Vue - Contrôleur), qui constitue la base du framework Laravel. Ce modèle permet de découpler les

différentes parties de l'application, à savoir :

- **Modèle** : Représente les données et la logique métier.
- **Vue** : Gère l'affichage et l'interface utilisateur via Blade et Tailwind CSS.
- **Contrôleur** : Fait le lien entre la vue et le modèle, en gérant les requêtes utilisateur.

3.4.2 Composants principaux

L'architecture comprend les composants suivants :

- **Interface utilisateur (Frontend)** : Développée avec Blade (moteur de templates Laravel), Tailwind CSS pour le design, et JavaScript pour les interactions dynamiques.
- **Serveur d'application (Backend)** : Développé en Laravel, il traite les requêtes, applique la logique métier, et interagit avec la base de données.
- **Base de données** : MySQL est utilisée pour stocker les informations sur les utilisateurs, produits, commandes, avis, etc.

3.5 Conception des interfaces

La conception des interfaces utilisateur constitue une étape essentielle dans la réalisation de la plateforme Nutstree. Elle vise à garantir une expérience utilisateur intuitive, fluide et agréable, tout en assurant la cohérence visuelle et fonctionnelle de l'application.

3.5.1 Principes de conception

La conception des interfaces a été guidée par les principes suivants :

- **Simplicité** : Réduction des éléments visuels superflus pour concentrer l'attention sur l'essentiel.
- **Accessibilité** : Interface claire et compréhensible pour tous les utilisateurs, y compris les novices.
- **Responsive design** : Adaptabilité à tous types d'écrans (PC, tablette, smartphone) grâce à Tailwind CSS.
- **Cohérence graphique** : Utilisation de composants réutilisables et d'un thème visuel harmonisé.

3.5.2 Interfaces principales

Les interfaces les plus importantes ont été maquettées afin d'anticiper leur ergonomie et leur navigation. Elles incluent :

- Page d'accueil
- Page de connexion / inscription
- Fiche produit
- Panier et validation de commande
- Tableau de bord administrateur

3.5.3 Maquettes des interfaces

Les maquettes des interfaces ont été réalisées à l'aide d'un outil de prototypage (comme Figma ou Adobe XD). Ces maquettes ont permis de valider les choix graphiques et fonctionnels avant le développement.

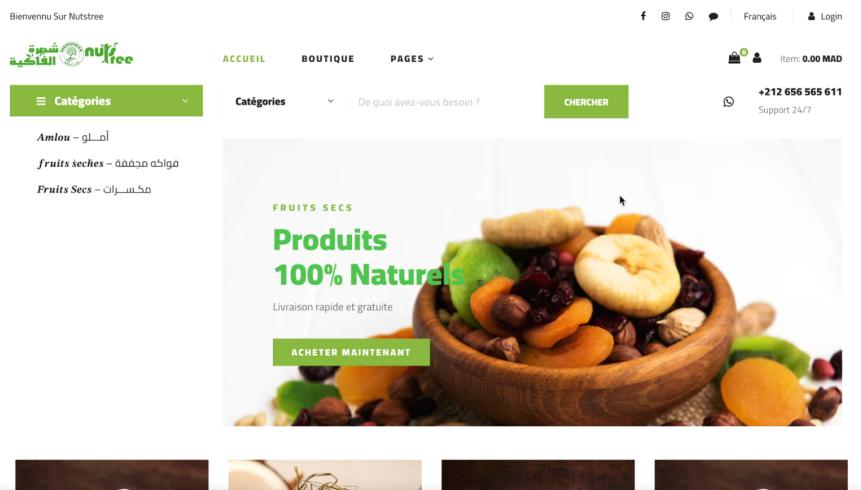


FIGURE 3.4 – Maquette de la page d'accueil

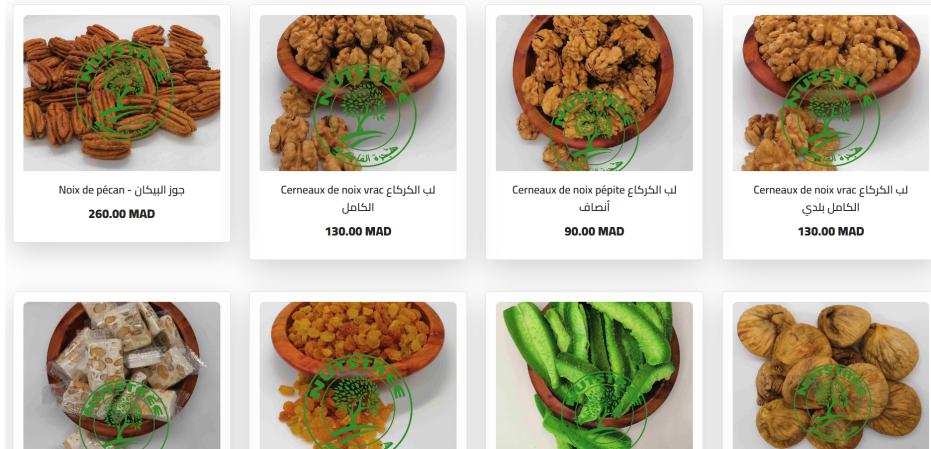


FIGURE 3.5 – Maquette d'une fiche produit

3.6 Choix techniques

Les choix techniques effectués pour le projet Nutstree ont été guidés par plusieurs critères essentiels : performance, évolutivité, simplicité de maintenance, documentation riche, et compatibilité avec les bonnes pratiques du développement web moderne.

3.6.1 Langage et Framework Backend

Le framework **Laravel** a été retenu pour le développement du backend. Basé sur PHP, Laravel est reconnu pour sa structure MVC claire, ses outils intégrés (migration, validation, authentification, etc.), et sa communauté active.

- Architecture MVC bien définie
- Sécurité renforcée (protection CSRF, validation, middleware)
- Écosystème riche (Eloquent ORM, Blade, Artisan)

3.6.2 Base de données

Le choix s'est porté sur **MySQL**, une solution robuste et largement adoptée pour la gestion des bases de données relationnelles.

- Excellente compatibilité avec Laravel (via Eloquent)
- Performances fiables même avec un volume important de données
- Facilité d'intégration et d'administration

3.6.3 Technologies Frontend

Pour l'interface utilisateur, les technologies suivantes ont été choisies :

- **Blade** : moteur de templates de Laravel permettant une intégration fluide entre les vues et les données.
- **Tailwind CSS** : framework CSS utilitaire, permettant un design responsive et moderne avec une grande flexibilité.
- **JavaScript** : pour ajouter de l'interactivité (gestion du panier, menus dynamiques, etc.).

3.6.4 Gestion des dépendances et outils de développement

- **Composer** : gestionnaire de dépendances PHP pour Laravel.
- **npm** : utilisé pour les paquets JavaScript et Tailwind CSS.
- **Git** : pour le versionnement du code et la collaboration.

3.6.5 Choix d'hébergement

L'application est conçue pour être hébergée sur un serveur web compatible PHP/-MySQL. Une solution comme **Laravel Forge**, ou un hébergement mutualisé avec support PHP 8+ et accès SSH, est envisagée pour le déploiement.

Chapitre 4

Implémentation

4.1 Mise en place du backend

Le développement du backend de la plateforme Nutstree repose sur le framework Laravel, choisi pour sa robustesse, sa clarté architecturale et ses outils intégrés facilitant la création rapide d'applications web sécurisées et performantes.

4.1.1 Installation et configuration de Laravel

La première étape a consisté à installer Laravel via Composer. Une nouvelle application a été générée et la configuration initiale (connexion à la base de données, timezone, langue) a été ajustée dans le fichier `.env`.

```
composer create-project laravel/laravel nutstree
```

4.1.2 Structure MVC

Laravel adopte une architecture MVC (Modèle-Vue-Contrôleur). Les éléments principaux développés sont :

- **Modèles** : gestion des entités comme les produits, commandes, utilisateurs.
- **Contrôleurs** : logique métier (validation, traitement des requêtes).
- **Routes** : définition des URL et leur association avec les contrôleurs.

4.1.3 Création des entités principales

Les migrations ont été utilisées pour créer les tables dans la base de données. Voici quelques entités principales du système :

- **Utilisateurs** : Clients, administrateurs, livreurs.
- **Produits** : Informations, prix, quantité, image.
- **Commandes** : Statut, date, client associé.
- **Commentaires** : Avis et notes laissés par les clients.

4.1.4 Gestion des relations

Les relations entre les entités ont été gérées grâce à l'ORM Eloquent :

- Un utilisateur peut passer plusieurs commandes (`hasMany`).
- Une commande est liée à plusieurs produits via une table pivot (`belongsToMany`).
- Un produit peut avoir plusieurs commentaires.

4.1.5 Authentification et rôles

Laravel Breeze a été utilisé pour mettre en place l'authentification. Une gestion des rôles (client, admin, livreur) a été ajoutée pour adapter l'accès aux différentes parties du système.

4.1.6 API interne (facultatif)

Une API interne RESTful peut être développée pour gérer les actions côté frontend via JavaScript (ex. : ajout au panier, consultation produits, etc.).

4.2 Développement des fonctionnalités

4.3 Développement des fonctionnalités

Le développement des fonctionnalités de la plateforme Nutstree a été réalisé en suivant une approche itérative, en se basant sur les besoins exprimés dans le cahier des charges. Chaque fonctionnalité a été soigneusement implémentée afin d'assurer une expérience utilisateur fluide, intuitive et sécurisée.

4.3.1 Inscription et authentification des utilisateurs

Un système d'inscription et de connexion a été mis en place grâce aux outils intégrés de Laravel (Breeze ou Jetstream). Il permet :

- La création sécurisée de comptes clients.

- La connexion via email et mot de passe.
- La distinction entre les rôles (client, administrateur, livreur).
- La redirection vers des tableaux de bord adaptés à chaque rôle.

4.3.2 Gestion des produits

Les administrateurs peuvent gérer les produits via une interface dédiée :

- Ajout de nouveaux produits avec image, description, prix et quantité.
- Modification et suppression de produits.
- Organisation des produits par catégories.

4.3.3 Panier et commande

Les utilisateurs peuvent ajouter des produits au panier, modifier les quantités, puis passer commande. Le système :

- Gère dynamiquement le contenu du panier (via session ou localStorage).
- Calcule le total de la commande.
- Enregistre les commandes en base de données avec un statut initial "En attente".

4.3.4 Paiement à la livraison (Cash on Delivery)

Un système de paiement à la livraison a été mis en place :

- Aucun traitement de carte bancaire n'est requis.
- Lors de la validation de commande, le client choisit le paiement à la livraison.
- La commande est enregistrée et transmise à l'administrateur pour validation.

4.3.5 Suivi des commandes

Chaque utilisateur peut consulter l'état de ses commandes via un tableau de bord personnel :

- Affichage des commandes passées.
- Visualisation du statut de livraison (En attente, En cours, Livrée, Annulée).

4.3.6 Commentaires et avis sur les produits

Les clients peuvent laisser des avis sur les produits achetés :

- Note sous forme d'étoiles.
- Commentaire textuel.

- Affichage des avis sur la fiche produit.

4.3.7 Gestion des utilisateurs, commandes et livraisons (Admin)

L'administrateur dispose d'un tableau de bord lui permettant :

- De consulter la liste des utilisateurs et leur rôle.
- De valider, modifier ou annuler des commandes.
- D'attribuer un statut de livraison à chaque commande.
- De consulter les paiements et les transactions associées.

4.3.8 Gestion du blog (optionnelle)

Une section blog permet à l'administrateur de publier des articles liés à la nutrition ou aux produits Nutstree. Chaque article comprend :

- Un titre, une image, une date de publication.
- Un contenu textuel enrichi.

4.4 Sécurité

La sécurité est un aspect fondamental dans le développement de toute application web. Dans le cadre du projet Nutstree, plusieurs mesures ont été mises en œuvre pour protéger les données des utilisateurs, éviter les attaques courantes, et garantir l'intégrité du système.

4.4.1 Protection des données utilisateurs

Les données sensibles (comme les mots de passe) sont protégées selon les normes actuelles :

- Les mots de passe sont automatiquement hachés grâce au système d'authentification de Laravel (bcrypt).
- Aucune donnée sensible n'est stockée en clair dans la base de données.
- Les données utilisateurs ne sont accessibles qu'aux utilisateurs autorisés selon leur rôle.

4.4.2 Validation des entrées

Toutes les données entrées par l'utilisateur (formulaires, requêtes) sont systématiquement validées pour éviter :

- Les injections SQL : protégées par l'ORM Eloquent.
- Les failles XSS (Cross-site scripting) : échappement automatique des données dans Blade.
- Les failles CSRF (Cross-site Request Forgery) : protection automatique de Laravel via des jetons CSRF.

4.4.3 Gestion des rôles et des autorisations

L'application implémente un système de gestion des rôles pour contrôler l'accès aux différentes sections du site :

- **Client** : accès au catalogue, commandes, avis.
- **Administrateur** : gestion complète du système.
- **Livreur** : accès aux commandes en livraison.

L'accès à chaque route ou fonctionnalité est protégé à l'aide de middlewares personnalisés.

4.4.4 Sécurisation des routes

Les routes sensibles (accès à l'administration, gestion des commandes, etc.) sont protégées par :

- Des middlewares d'authentification (`auth`).
- Des vérifications de rôle (middleware personnalisés ou politiques d'accès).

4.4.5 Sécurité des fichiers et du serveur

- Les fichiers envoyés sont vérifiés (type, taille, extension).
- L'accès aux fichiers de configuration est restreint côté serveur.
- Des sauvegardes régulières de la base de données sont prévues.
- Le code source est versionné sur Git pour assurer un suivi des modifications.

Chapitre 5

Tests et Validation

5.1 Méthodologie de test

Afin de garantir la fiabilité et la qualité du système Nutstree, une méthodologie de test rigoureuse a été adoptée durant les phases de développement et de déploiement. L'objectif est de détecter les éventuelles erreurs, valider le bon fonctionnement des fonctionnalités et s'assurer que le système réponde aux attentes fonctionnelles et non fonctionnelles.

5.1.1 Types de tests réalisés

- **Tests unitaires** : Ils permettent de tester individuellement les fonctions et méthodes des différentes classes (modèles, services, contrôleurs). Laravel fournit un environnement adapté avec PHPUnit pour l'exécution de ces tests.
- **Tests fonctionnels** : Ils vérifient que chaque fonctionnalité implémentée (inscription, ajout au panier, création de commande, etc.) fonctionne correctement selon les scénarios définis.
- **Tests d'intégration** : Ces tests ont pour but de valider l'interaction entre les différentes composantes du système : base de données, logique métier, vues, contrôleurs.
- **Tests manuels** : Réalisés via la navigation sur l'interface, ces tests permettent de simuler des comportements utilisateurs, détecter des anomalies visuelles ou des incohérences dans l'expérience utilisateur.

5.1.2 Environnement de test

Les tests ont été exécutés dans un environnement local identique à celui de production :

- Serveur local avec Laravel (sous XAMPP ou Laravel Sail).
- Base de données MySQL test.
- Navigateur web (Chrome, Firefox) pour les tests d'interface.
- Utilisation de Postman pour tester les routes backend (API, sécurité, validation).

5.1.3 Outils et bonnes pratiques

- Utilisation de `php artisan test` pour lancer les tests unitaires.
- Nettoyage de la base de données entre chaque test automatisé avec les traits `RefreshDatabase`.
- Écriture de tests clairs et documentés pour assurer la maintenabilité du code.

5.2 Résultats des tests

Les différents tests menés tout au long du développement du projet Nutstree ont permis de valider la robustesse, la fiabilité et la conformité du système par rapport aux spécifications fonctionnelles. Cette section présente un résumé des résultats obtenus pour chaque type de test.

5.2.1 Tests unitaires

Les tests unitaires ont été exécutés avec succès sur les principales classes du projet :

- Les modèles Eloquent (Produit, Commande, Utilisateur) ont été testés pour vérifier la validité des relations et des règles de validation.
- Les services métiers (gestion de panier, traitement de commande) ont passé les scénarios définis sans erreur.
- Aucun test unitaire n'a échoué au cours des itérations.

5.2.2 Tests fonctionnels

Les fonctionnalités principales de l'application ont été validées avec les résultats suivants :

- **Inscription/Connexion** : Fonctionnement correct avec gestion des erreurs (mot de passe incorrect, email déjà utilisé, etc.).
- **Navigation et recherche** : Affichage correct des produits, résultats cohérents lors de la recherche par mot-clé.
- **Gestion du panier** : Ajout, modification et suppression d'articles fonctionnent parfaitement.
- **Commande et paiement** : Processus fluide jusqu'à la validation de la commande avec paiement en espèces.
- **Tableau de bord utilisateur** : Affichage des commandes et suivi du statut sans anomalies.
- **Zone d'administration** : Gestion efficace des utilisateurs, produits, commandes et livraisons.

5.2.3 Tests d'intégration

Les tests d'intégration ont permis de confirmer la bonne interaction entre les composants du système :

- Lien correct entre les interfaces frontend et les routes backend.
- Communication fluide entre le backend Laravel et la base de données MySQL.
- Affichage dynamique des données sur l'interface grâce à JavaScript.

5.2.4 Tests manuels

Des tests manuels ont été réalisés pour simuler des scénarios réels :

- Navigation fluide sur desktop et mobile.
- Vérification de la réactivité des interfaces avec Tailwind CSS.
- Tests avec différents navigateurs (Chrome, Firefox, Edge) sans problème d'affichage.
- Absence de crash ou de comportements inattendus constatés lors de l'utilisation normale.

5.2.5 Conclusion

Les résultats des tests sont satisfaisants. L'ensemble des fonctionnalités prévues a été validé sans anomalies majeures. Le système est stable, sécurisé et conforme aux attentes définies dans le cahier des charges.

Chapitre 6

Déploiement et maintenance

6.1 Environnement de production

Le déploiement de l'application Nutstree a été réalisé dans un environnement de production stable et sécurisé, garantissant la disponibilité, la performance et la sécurité du service en ligne.

6.1.1 Serveur d'hébergement

Pour assurer une accessibilité continue de la plateforme, l'application a été hébergée sur un serveur compatible avec les technologies utilisées :

- **Type d'hébergement** : Serveur mutualisé / VPS selon les ressources nécessaires.
- **Système d'exploitation** : Linux (Ubuntu Server).
- **Serveur web** : Apache 2 ou Nginx.
- **Version de PHP** : PHP 8.x compatible avec Laravel 10.
- **Base de données** : MySQL 8.

6.1.2 Déploiement de l'application

Le déploiement de Laravel a suivi les étapes standards :

- Transfert des fichiers via FTP ou Git (GitHub).
- Configuration du fichier `.env` pour l'environnement de production (base de données, mail, etc.).
- Exécution des commandes `composer install`, `php artisan migrate` et `php artisan config:cache`.

- Attribution des permissions aux répertoires `storage` et `bootstrap/cache`.

6.1.3 Nom de domaine et accès HTTPS

Le site est accessible via un nom de domaine personnalisé, sécurisé par un certificat SSL :

- Enregistrement du domaine chez un registrar.
- Installation d'un certificat SSL (Let's Encrypt ou commercial).
- Redirection automatique HTTP → HTTPS assurée dans la configuration du serveur.

6.1.4 Surveillance et logs

Afin d'assurer la stabilité du système, les éléments suivants sont mis en place :

- Accès aux journaux d'erreurs Laravel via `storage/logs`.
- Surveillance des ressources du serveur (CPU, RAM, espace disque).
- Configuration des alertes email en cas de dysfonctionnement majeur.

6.2 Procédure de déploiement

Le déploiement de la plateforme Nutstree suit une procédure méthodique afin de garantir une mise en production sécurisée et sans interruption de service.

6.2.1 Préparation de l'environnement

Avant toute mise en production, l'environnement de destination est configuré avec :

- Installation des dépendances requises (PHP, extensions PHP, MySQL, serveur web).
- Configuration du serveur web (Apache/Nginx) avec les règles de redirection nécessaires.
- Mise en place du nom de domaine et du certificat SSL pour la sécurisation HTTPS.

6.2.2 Transfert des fichiers

Le code source est transféré vers le serveur de production via l'une des méthodes suivantes :

- Utilisation de Git pour cloner ou mettre à jour le dépôt distant.
- Transfert FTP/SFTP des fichiers si Git n'est pas disponible.

6.2.3 Installation des dépendances

Sur le serveur de production, les dépendances PHP sont installées avec Composer :

```
composer install --no-dev --optimize-autoloader
```

Cette commande installe uniquement les dépendances nécessaires en production et optimise le chargement automatique des classes.

6.2.4 Configuration de l'application

Le fichier `.env` est configuré avec les paramètres spécifiques à l'environnement de production, notamment :

- Les informations de connexion à la base de données.
- Les clés d'API ou autres secrets.
- Les configurations liées au mail, cache et sessions.

6.2.5 Migrations et seeders

Pour mettre à jour la base de données et insérer les données initiales, les commandes suivantes sont exécutées :

```
php artisan migrate --force  
php artisan db:seed --force
```

L'option `-force` permet d'exécuter les commandes en production sans confirmation interactive.

6.2.6 Cache et optimisation

Pour améliorer les performances, les caches sont générés avec les commandes suivantes :

```
php artisan config:cache  
php artisan route:cache  
php artisan view:cache
```

6.2.7 Gestion des permissions

Les permissions des dossiers critiques sont ajustées pour permettre l'écriture par le serveur web :

```
chmod -R 775 storage bootstrap/cache
```

6.2.8 Tests post-déploiement

Enfin, des tests manuels sont effectués pour vérifier que l'application fonctionne correctement dans l'environnement de production, notamment :

- Accès au site via le domaine HTTPS.
- Connexion et navigation dans les différentes sections.
- Passage de commande test.
- Vérification des logs pour s'assurer de l'absence d'erreurs.

6.3 Maintenance continue

La maintenance continue est essentielle pour assurer la pérennité, la sécurité et l'évolution de la plateforme Nutstree après son déploiement. Elle comprend plusieurs axes complémentaires :

6.3.1 Maintenance corrective

Cette maintenance vise à corriger les bugs et dysfonctionnements détectés en production ou signalés par les utilisateurs. Elle comprend :

- La surveillance régulière des logs d'erreurs pour identifier rapidement les anomalies.
- La correction des failles de sécurité ou des bugs critiques.
- Le déploiement rapide de correctifs via des mises à jour planifiées.

6.3.2 Maintenance évolutive

Pour s'adapter aux besoins changeants des utilisateurs et aux évolutions technologiques, des améliorations et nouvelles fonctionnalités peuvent être intégrées :

- Analyse des retours utilisateurs et propositions d'amélioration.
- Ajout de nouvelles fonctionnalités ou optimisation des fonctionnalités existantes.

- Mise à jour régulière des dépendances et des frameworks (Laravel, bibliothèques JS, etc.).

6.3.3 Maintenance préventive

Elle vise à anticiper les risques afin d'éviter les interruptions de service ou pertes de données :

- Sauvegardes régulières des bases de données et des fichiers importants.
- Mise à jour des systèmes et correctifs de sécurité.
- Tests périodiques de performance et montée en charge.

6.3.4 Documentation et support

Un volet important de la maintenance est la gestion documentaire et l'assistance aux utilisateurs :

- Mise à jour de la documentation technique et utilisateur.
- Support technique pour les utilisateurs via e-mail ou système de tickets.
- Formation continue de l'équipe en charge du projet.

Grâce à cette approche globale, la plateforme Nutstree reste performante, sécurisée et adaptée aux attentes de ses utilisateurs sur le long terme.

Chapitre 7

Conclusion et perspectives

Perspectives

Plusieurs pistes d'amélioration peuvent être envisagées pour enrichir et faire évoluer Nutstree :

- **Intégration d'un système de paiement en ligne sécurisé** (ex : CMI, Pay-Pal) afin de diversifier les modes de paiement et faciliter les transactions.
- **Optimisation SEO** pour améliorer la visibilité de la plateforme sur les moteurs de recherche et attirer davantage de clients.
- **Développement d'une application mobile** native ou hybride pour élargir l'accès au service.
- **Ajout de fonctionnalités avancées** telles que la recommandation personnalisée de produits, la gestion de promotions ou de programmes de fidélité.
- **Amélioration continue de la sécurité** pour garantir la protection des données personnelles et la conformité aux normes en vigueur.
- **Mise en place d'un système de support client** plus interactif, avec chat en ligne ou chatbot.

Conclusion

Le projet Nutstree a permis de développer une plateforme e-commerce efficace et adaptée aux besoins du marché local, notamment dans la vente de fruits secs. En s'appuyant sur des technologies modernes et éprouvées, le système offre une expérience utilisateur fluide, une gestion complète des commandes et une interface d'administration performante.

Les différentes phases du projet, de l'analyse à la conception, puis à l'implémentation et au déploiement, ont été réalisées avec rigueur, ce qui a permis d'obtenir un produit stable et fonctionnel. La mise en place d'une maintenance continue garantit la pérennité de la solution et sa capacité à évoluer.

Ainsi, Nutstree constitue une base solide pour le développement futur du commerce en ligne dans ce secteur, tout en offrant un service de qualité aux clients et aux administrateurs.

Chapitre A

Annexes

A.1 Captures d'écran de l'application

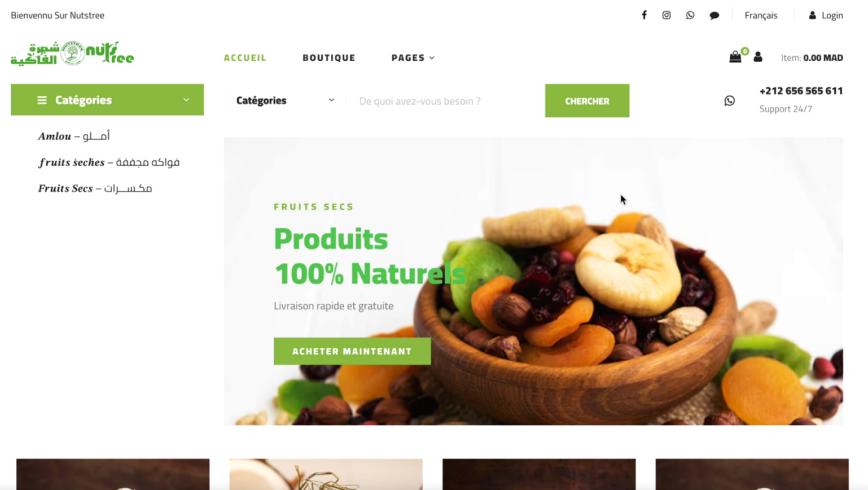


FIGURE A.1 – Page d'accueil de la plateforme Nutstree

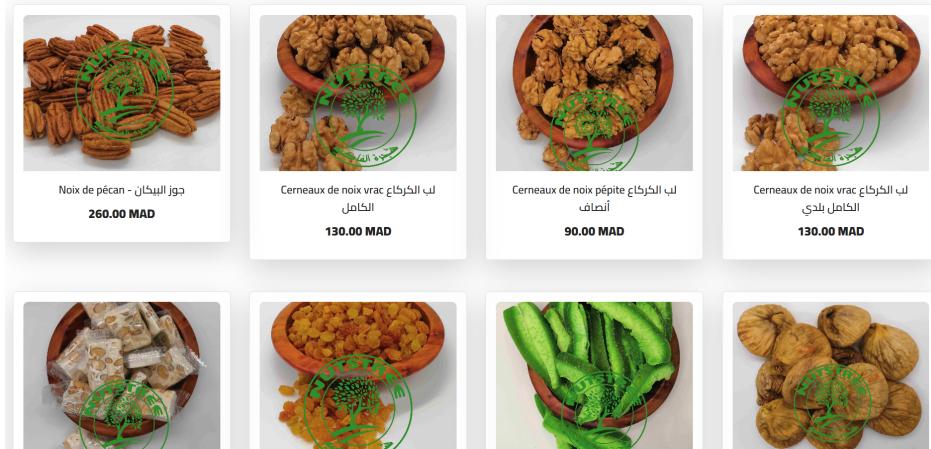


FIGURE A.2 – Liste des produits avec filtres et recherche

Produits	Prix	Quantité	Total
Votre panier est vide !			

Codes de réduction

Entrez votre code promo **APPLIQUER LE CODE**

Total du Panier

Sous-total	0.00 MAD
Frais de Livraison	0.00 MAD
Total	0.00 MAD

FIGURE A.3 – Gestion du panier avant validation de la commande

PRODUIT	NOM PRODUIT	PRIX	CATÉGORIE	QUANTITÉ
	0	0.00 MAD	Fruits Secs - مكسرات	0

FIGURE A.4 – Interface d'administration : gestion des produits

Bibliographie

- Laravel Documentation : <https://laravel.com/docs>
- Tailwind CSS : <https://tailwindcss.com/docs>
- Documentation CMI Maroc
- Articles sur le e-commerce