
Rapport de Projet

Développement d'un Site E-commerce.

Realisé par:

- Achraf Abdelfadel
- Zakaria Ennaqui
- Mohammed Amsguine
- Mohammed El Ouafir
- Abdrahmane Chiguer
- Saad Ben Aboudi

Encadrant:

- Mr.Haddi

Table of Contents

I. Introduction.....	4
A. Contexte du projet.....	4
B. Objectifs du projet	5
C. Méthodologie adoptée	6
II. Analyse des besoins.	7
A. Identification des parties prenantes.....	7
B. Besoins fonctionnels :	8
C. Besoins non fonctionnels.....	9
D. Analyse des contraintes	10
III. Spécifications Techniques	11
A. Technologies utilisées	11
B. Architecture du projet.	12
1. Architecture générale	12
C. Schéma d'architecture	13
D. Diagramme de l'architecture	14
IV. Analyse et conception	15
A. Les diagrammes des cas d'utilisation.....	15
1. Définition	15
2. Diagramme de cas d'utilisation de notre site web.	16
B. Le diagramme de séquence	16
1. Définition	16
2. Diagrammes de séquences de notre site web.....	17
C. Le diagramme de classes	18
1. Définition	18
2. Diagrammes de classes de notre site web.....	19
D. Le diagramme de composants	19
1. Définition	19
2. Diagrammes de composants de notre site web.....	20
E. Le diagrammes de déploiement	20
1. Définition	20
2. Diagramme de déploiement de notre site web.....	21
V. Conception de l'Interface Utilisateur.....	21

A.	Mockups et Wireframes.....	21
B.	Design UI/UX	22
VI.	Planification et Gestion du Projet.....	23
A.	Répartition des tâches et responsabilités.....	23
B.	Planning et diagramme de Gantt	24
C.	Gestion de version avec Git et GitHub	25
VII.	Implémentation.....	26
A.	Fonctionnalités principales développées.....	26
B.	Code source (description générale des modules).....	26
VIII.	Tests et Validation	27
A.	Plan de test unitaire	27
B.	Plan de test d'intégration.....	29
IX.	Déploiement	30
A.	Environnement de production.....	30
B.	Processus de déploiement	31
X.	Conclusion et Perspectives	33
A.	Résumé des réalisations	33
B.	Limitations du projet.....	33
C.	Améliorations futures	34
XI.	Annexes.	34
A.	Documentation utilisateur.....	34
B.	Codes sources supplémentaires	38
C.	Références	39

I. Introduction.

A. Contexte du projet.

Dans un monde de plus en plus digitalisé, le commerce en ligne est devenu un pilier essentiel pour les entreprises souhaitant étendre leur portée et offrir une expérience d'achat pratique à leurs clients. Ce projet s'inscrit dans le cadre de la réalisation d'un site web e-commerce permettant de mettre en relation des commerçants et des consommateurs, tout en offrant des fonctionnalités intuitives pour la gestion des produits, des commandes et des paiements.

Le projet a été choisi en raison de son applicabilité pratique et de son alignement avec les compétences techniques nécessaires dans le domaine de l'informatique. Il constitue également une opportunité d'appliquer les concepts étudiés, notamment dans la conception logicielle, le développement web et la gestion de projet.

B. Objectifs du projet

L'objectif principal de ce projet est de développer une plateforme e-commerce fonctionnelle, robuste et évolutive. Ce site web devra répondre aux attentes des utilisateurs finaux en termes d'ergonomie et de fonctionnalités. Les objectifs spécifiques incluent :

- ✓ Permettre aux utilisateurs (administrateurs, vendeurs et clients) d'interagir avec le système selon leurs rôles respectifs.
- ✓ Proposer une interface utilisateur attrayante et facile à utiliser.
- ✓ Intégrer des technologies modernes (HTML, CSS, JavaScript, Node.js) pour assurer une expérience fluide et performante.
- ✓ Garantir la sécurité des données sensibles, notamment les informations personnelles et financières des utilisateurs.

C. Méthodologie adoptée

La méthodologie utilisée pour le développement du projet repose sur les étapes suivantes :

- ✓ **Analyse des besoins** : Une étude approfondie des exigences fonctionnelles et non fonctionnelles.
- ✓ **Conception UML** : Création de diagrammes pour visualiser la structure et les interactions du système.
- ✓ **Développement** : Implémentation des fonctionnalités principales en suivant une approche modulaire.
- ✓ **Tests et validation** : Vérification des performances, de la sécurité et de l'ergonomie.
- ✓ **Déploiement** : Mise en ligne de la plateforme dans un environnement de production.
- ✓ L'ensemble de ces étapes est structuré pour garantir une progression logique et une qualité optimale tout au long du projet.

II. Analyse des besoins.

A. Identification des parties prenantes

Pour garantir que le site e-commerce réponde aux attentes des utilisateurs, il est essentiel de bien définir les parties prenantes et leurs rôles. Ces parties prenantes incluent :

- **Administrateurs :**

- ✓ **Description :** Responsables de la gestion globale du site.
- ✓ **Besoins principaux :**
 - ❖ Gestion des comptes (création, modification, suppression des utilisateurs).
 - ❖ Supervision des activités, incluant les commandes et transactions.
 - ❖ Accès aux rapports statistiques pour suivre les performances du site.
 - ❖ Application des politiques de sécurité pour protéger les données des utilisateurs.

- **Vendeurs :**

- ✓ **Description :** Fournissent les produits ou services disponibles sur le site.
- ✓ **Besoins principaux :**
 - ❖ Ajout, modification et suppression de produits avec des informations détaillées (nom, description, prix, images).
 - ❖ Suivi et gestion des commandes passées par les clients.
 - ❖ Consultation de statistiques de ventes et d'avis clients.

- ❖ Notifications sur les commandes et autres événements importants.

- **Clients :**

- ✓ **Description :** Utilisateurs finaux qui consultent et achètent les produits sur le site.

- ✓ **Besoins principaux :**

- ❖ Création de compte utilisateur et gestion de leur profil.
 - ❖ Navigation intuitive pour explorer et rechercher des produits.
 - ❖ Gestion du panier d'achat et suivi des commandes (statut, livraison).
 - ❖ Possibilité de laisser des avis et d'évaluer les produits.
 - ❖ Accès à un support client pour résoudre d'éventuels problèmes

B. Besoins fonctionnels :

Les besoins fonctionnels définissent les fonctionnalités essentielles du site pour satisfaire les attentes des parties prenantes :

- Gestion des utilisateurs :Création, authentification et gestion des rôles des utilisateurs (administrateurs, vendeurs, clients).
- Catalogue de produits :Présentation des produits avec des options de filtrage et de tri (par catégorie, prix, popularité).
- Recherche avancée avec des mots-clés et des filtres dynamiques.
- Gestion des commandes :Ajout au panier, validation des commandes et suivi des livraisons.
- Gestion des retours et remboursements.

- Statistiques et rapports :Accès aux données analytiques pour les administrateurs et vendeurs (ventes, interactions des clients).
- Communication :Notifications par e-mail ou SMS pour informer des mises à jour ou promotions.
- Intégration d'un support client (chat en ligne ou formulaire).

C. Besoins non fonctionnels

Les besoins non fonctionnels garantissent que le système soit performant, sécurisé et évolutif :

- **Performance** :Temps de réponse rapide des pages pour améliorer l'expérience utilisateur.
- **Capacité à gérer plusieurs utilisateurs simultanément sans dégradation des performances.**
- **Sécurité** :Implémentation d'un système de chiffrement pour protéger les données sensibles.
- **Contrôle d'accès basé sur les rôles (RBAC).**
- **Compatibilité** :Assurer le fonctionnement sur différents appareils et navigateurs.
- **Évolutivité** :Prévoir la possibilité d'ajouter de nouvelles fonctionnalités et de gérer une base d'utilisateurs en expansion.
- **Accessibilité** :Respect des normes d'accessibilité (WCAG) pour garantir l'inclusivité.

D. Analyse des contraintes

Le projet est également soumis à certaines contraintes :

- **Contraintes techniques** : Utilisation des technologies choisies (HTML, CSS, JavaScript, Node.js, etc.).
- **Intégration avec des APIs tierces pour le traitement des paiements.**
- **Contraintes temporelles** : Respect du calendrier établi pour chaque phase du projet.
- **Contraintes légales** : Conformité aux réglementations sur la protection des données personnelles (RGPD).
- **Contraintes budgétaires** : Budget limité pour l'hébergement, les outils de développement et les ressources externes.

III. Spécifications Techniques

A. Technologies utilisées

Le développement du site e-commerce repose sur un ensemble de technologies modernes permettant de garantir une interface utilisateur intuitive, une expérience performante, et une maintenance facilitée :

- **Front-end :**
 - ❖ **HTML5** : Utilisé pour la structure des pages web, garantissant une compatibilité multi-plateformes et une sémantique claire.
 - ❖ **CSS3** : Appliqué pour le stylisme et la mise en page des pages, avec des animations et des transitions pour améliorer l'interactivité.
 - ❖ **JavaScript** : Langage principal pour l'interactivité et la logique côté client (validation des formulaires, navigation dynamique).
 - ❖ **Tailwinds** : Framework CSS utilisé pour accélérer le développement d'une interface utilisateur responsive et moderne.
- **Back-end :**
 - ❖ **Node.js** : Plateforme JavaScript pour gérer la logique serveur, les requêtes des utilisateurs, et la gestion des données.
 - ❖ **Express.js** : Framework léger pour la création d'API RESTful et la gestion des routes du serveur.
- **Base de données :**
 - ❖ **MongoDB** : Base de données NoSQL utilisée pour stocker et gérer les informations relatives aux utilisateurs, produits, commandes, et autres données.
- **Autres technologies :**

- ❖ **Git & GitHub** : Versionnement et collaboration pour maintenir le code source.
- ❖ **Postman** : Test des API développées avant leur intégration.
- ❖ **VS Code** : Environnement de développement intégré (IDE) pour écrire et déboguer le code.

B. Architecture du projet.

Le projet suit une architecture modulaire qui sépare les responsabilités en différentes couches afin de garantir la maintenabilité, la scalabilité et la clarté du code.

1. Architecture générale

L'architecture du projet est basée sur le modèle Client-Serveur et inclut les composants suivants :

- **Client (Front-end) :**
 - ✓ Interagit avec les utilisateurs finaux via une interface graphique.
 - ✓ Envoie des requêtes au serveur pour récupérer ou envoyer des données.
 - ✓ Gère les interactions dynamiques grâce à JavaScript et aux frameworks front-end.
- **Serveur (Back-end) :**
 - ✓ Traite les requêtes HTTP envoyées par le client.
 - ✓ Gère la logique métier, l'authentification, et la validation des données.
 - ✓ Fournit des données au format JSON via des API RESTful.
- **Base de données :**

- ✓ Stocke les données structurées et semi-structurées liées aux produits, utilisateurs, commandes, etc.
- ✓ Permet des opérations CRUD (**Create, Read, Update, Delete**) via des requêtes.

C. Schéma d'architecture

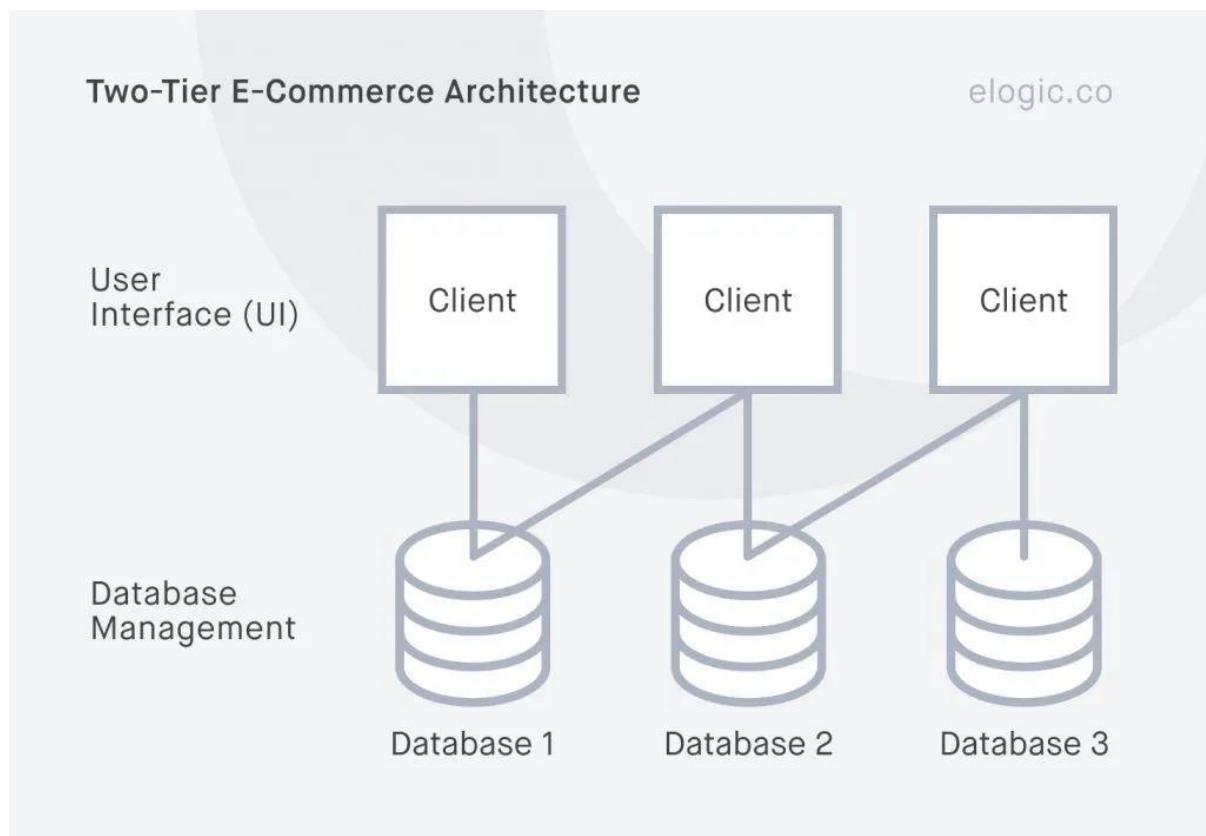
Voici une description des principales couches de l'architecture :

- **Côté client :**
 - ✓ Interfaces utilisateur développées en HTML, CSS, et JavaScript avec Bootstrap pour assurer un design responsive.
 - ✓ Requêtes envoyées via AJAX/Fetch API pour un chargement dynamique des données.
- **Côté serveur :**
 - ✓ **Routeur** : Gère les endpoints pour les différentes fonctionnalités (connexion, gestion des produits, commandes).
 - ✓ **Contrôleurs** : Contiennent la logique métier pour traiter les données et les envoyer au client.
 - ✓ **Services** : Classes ou modules pour la gestion des tâches complexes comme l'envoi d'e-mails ou l'intégration avec des API tierces.
 - ✓ **Modèles** : Définissent les schémas pour les données dans MongoDB.
- **Base de données :**
 - ✓ Stockage des collections pour chaque entité (utilisateurs, produits, commandes).
 - ✓ Indexation pour accélérer les recherches fréquentes.

D. Diagramme de l'architecture

Le diagramme de l'architecture met en évidence les interactions entre les différentes couches :

- Le client envoie une requête HTTP au serveur via une API.
- Le serveur traite la requête (via le routeur et les contrôleurs).
- Les données sont récupérées ou modifiées dans MongoDB.
- Une réponse est renvoyée au client, qui met à jour dynamiquement l'interface utilisateur.



1D. Diagramme de l'architecture

IV. Analyse et conception

A. Les diagrammes des cas d'utilisation

1. Définition

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système. Le diagramme de cas se compose de trois éléments principaux :

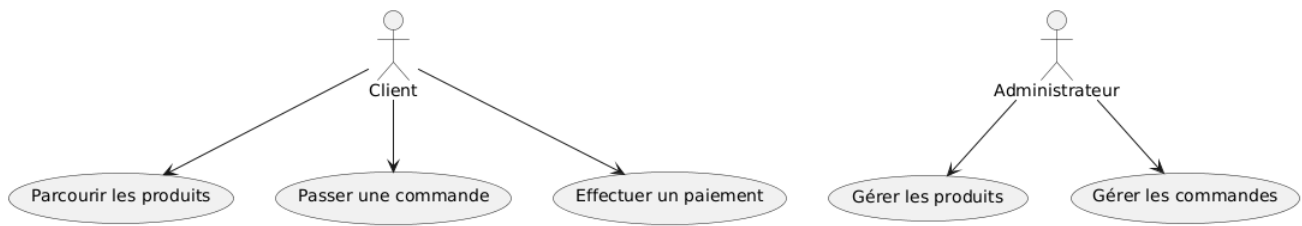
Acteurs principaux :

- **Client** : Parcourt les produits, passe des commandes, effectue des paiements.
- **Administrateur** : Gère les produits, les commandes et les utilisateurs.

Cas d'utilisation typiques :

- **Parcourir les produits** : Le client consulte les catégories et les fiches produits.
- **Passer une commande** : Le client ajoute des produits au panier et confirme l'achat.
- **Effectuer un paiement** : Le client paie la commande via une passerelle de paiement sécurisée.
- **Gérer les produits** : L'administrateur ajoute, modifie ou supprime des produits du catalogue.
- **Gérer les commandes** : L'administrateur consulte et met à jour le statut des commandes.

2. Diagramme de cas d'utilisation de notre site web.



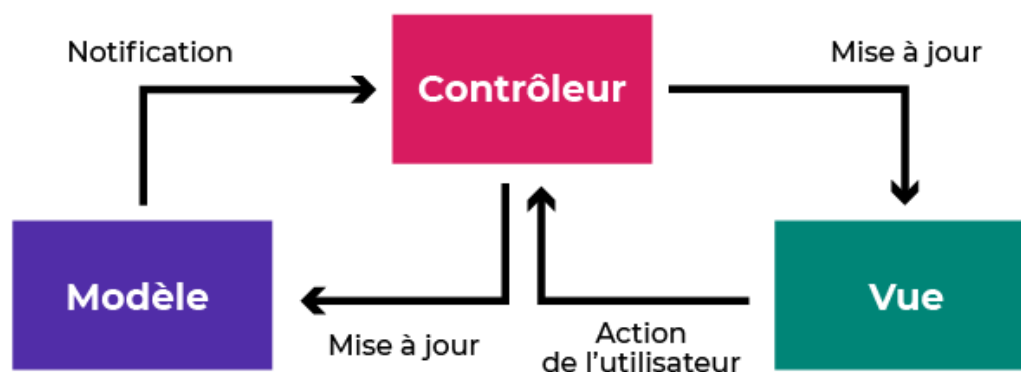
B. Le diagramme de séquence

1. Définition

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont

En ce qui suit, nous présenterons quelques diagrammes de séquences relatifs aux cas d'utilisations présentés.

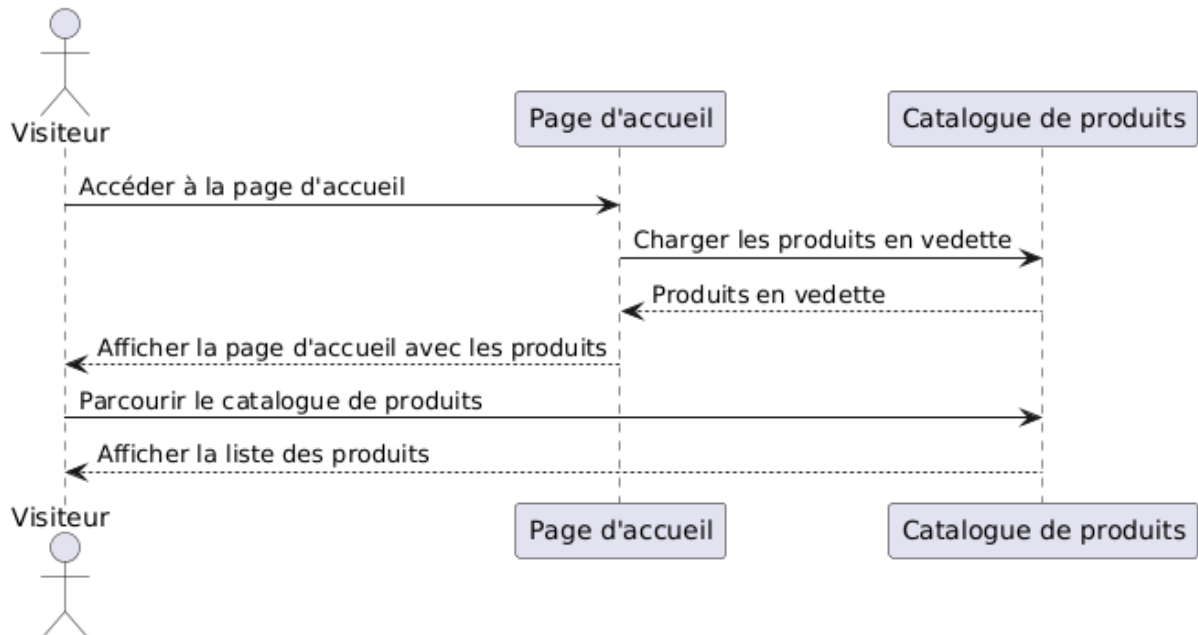
Les diagrammes de séquences sont basées sur le model MVC



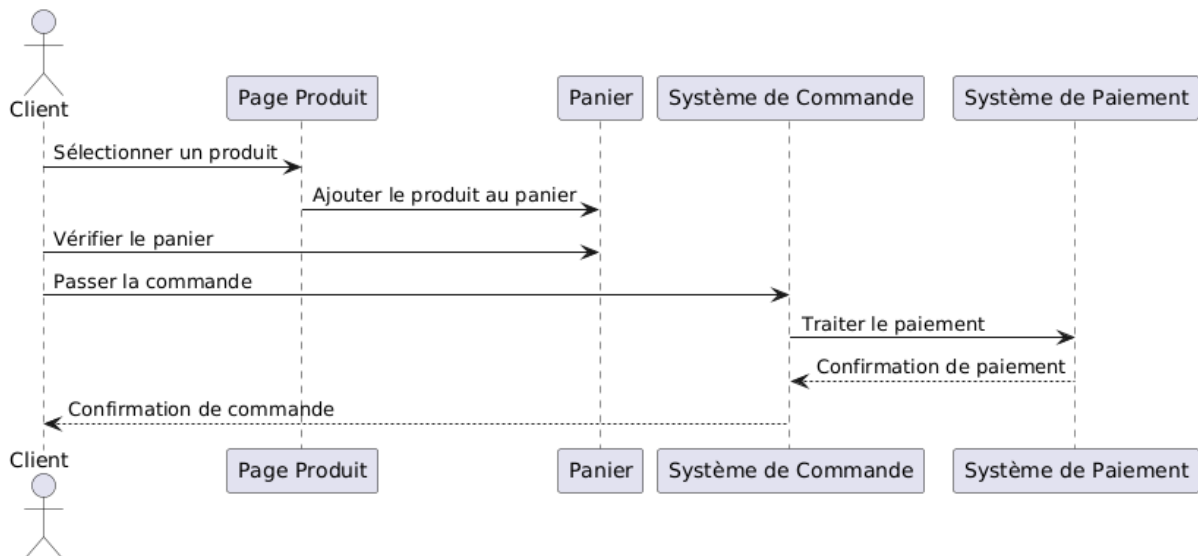
2 :L'architecture MVC

2. Diagrammes de séquences de notre site web.

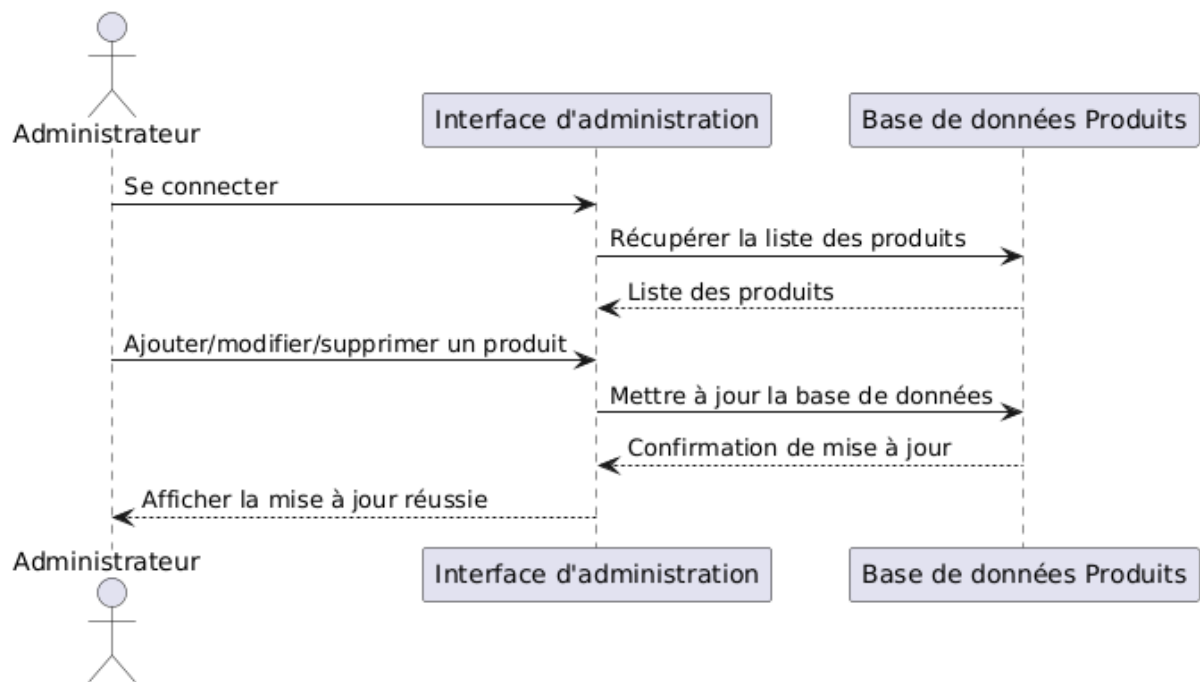
❖ Le visiteur.



❖ Client.



❖ L'administrateur.

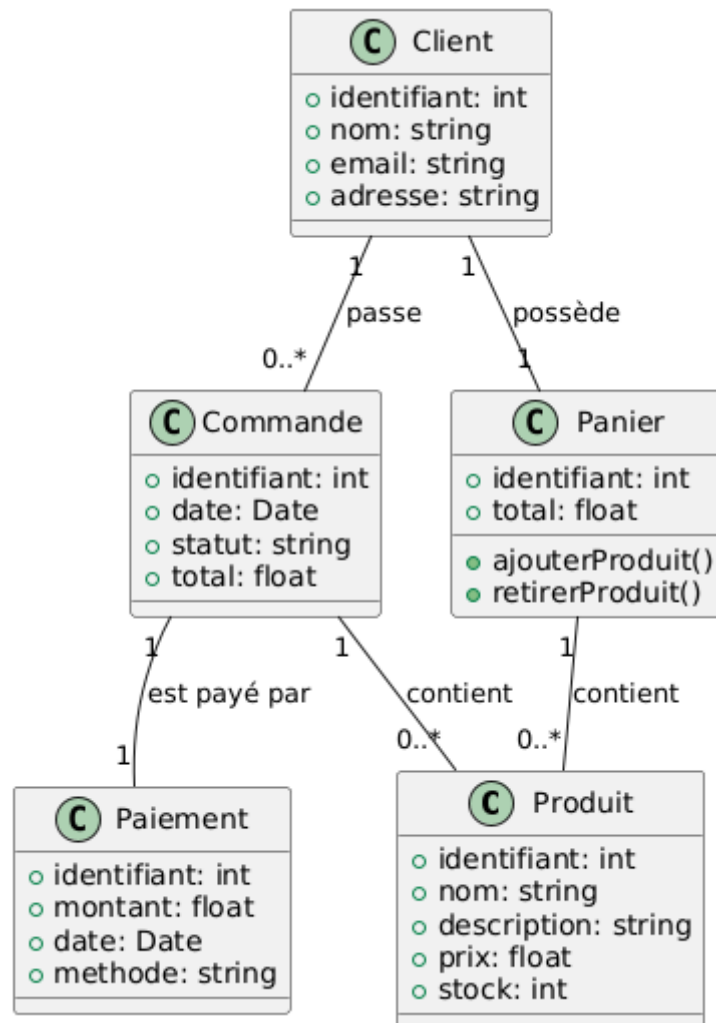


C. Le diagramme de classes

1. Définition

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées sur notre site web, à la fois en interne et en communication avec ses utilisateurs. Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données

2. Diagrammes de classes de notre site web.



D. Le diagramme de composants

1. Définition

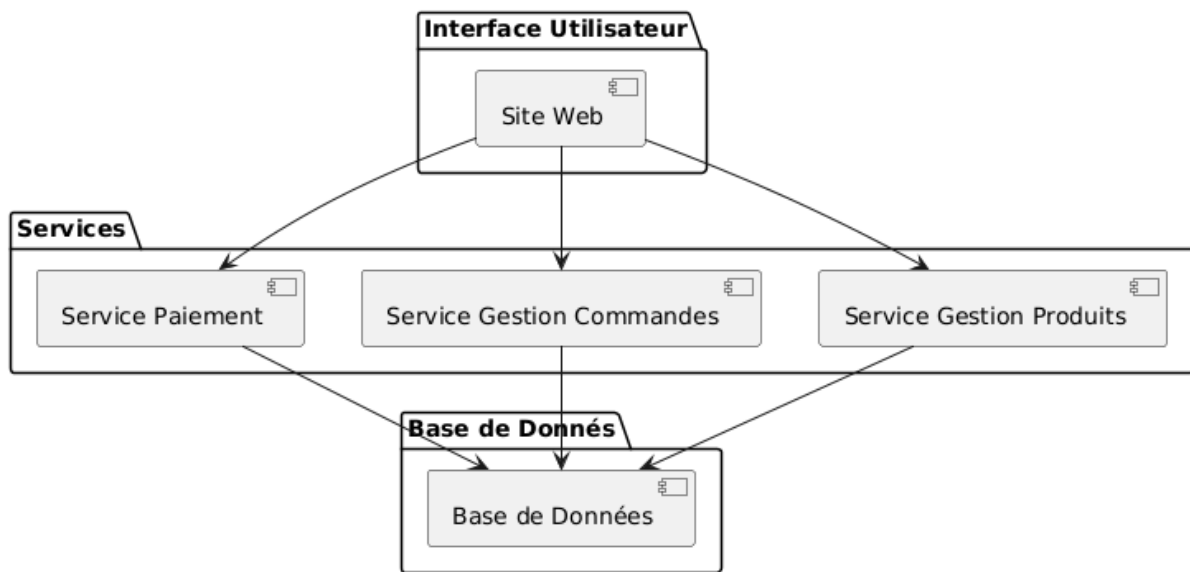
Le **diagramme de composants** est essentiel pour visualiser l'architecture modulaire d'un système e-commerce. Il illustre l'organisation et les dépendances entre les différents modules, facilitant ainsi la compréhension, la maintenance et l'évolution du système.

Composants principaux :

- **Interface utilisateur** : Gère l'affichage et les interactions avec le client.

- **Service de gestion des produits** : Gère les opérations liées aux produits.
- **Service de gestion des commandes** : Gère les processus de commande.
- **Service de paiement** : Gère les transactions financières.
- **Base de données** : Stocke les informations sur les clients, produits, commandes, etc.

2. Diagrammes de composants de notre site web.



E. Le diagrammes de déploiement

1. Définition

Le diagramme de déploiement décrit la configuration matérielle et logicielle du système, ainsi que la distribution des composants sur les nœuds du réseau.

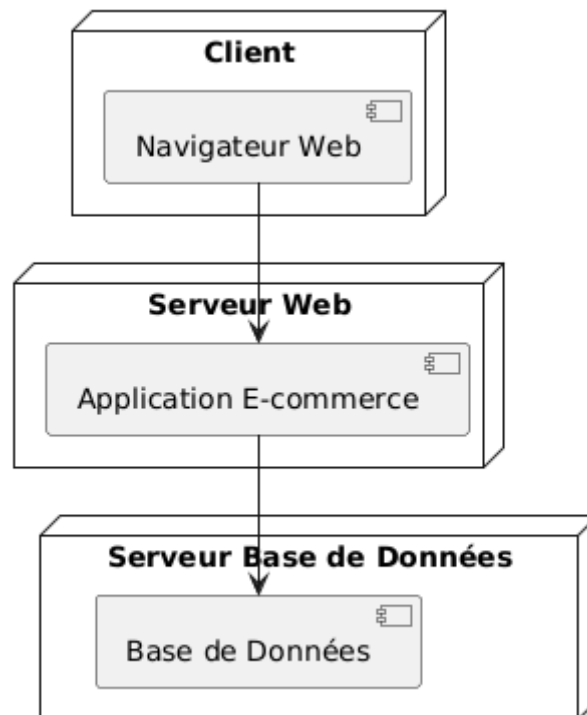
Nœuds principaux :

- **Serveur web** : Héberge l'application e-commerce.
- **Serveur de base de données** : Stocke les données persistantes.
- **Client** : Accède au site via un navigateur web.

Relations :

- Le Client envoie des requêtes HTTP au Serveur web.
- Le Serveur web interagit avec le Serveur de base de données pour les opérations CRUD.

2. Diagramme de déploiement de notre site web



V. Conception de l'Interface Utilisateur

A. Mockups et Wireframes

Les **mockups** et **wireframes** jouent un rôle crucial dans la conception de l'interface utilisateur (UI) en permettant de visualiser la structure de l'application avant de commencer à la développer. Ils sont utilisés pour esquisser les éléments clés du site, tels que les menus de navigation, les pages d'accueil, les fiches produits, et les formulaires de commande.

Wireframes : Les wireframes sont des croquis de base qui illustrent la disposition des éléments sur une page, comme les sections, les boutons, et les champs de texte. Ils n'ont pas de design visuel détaillé mais servent de guide pour la structure de la page.

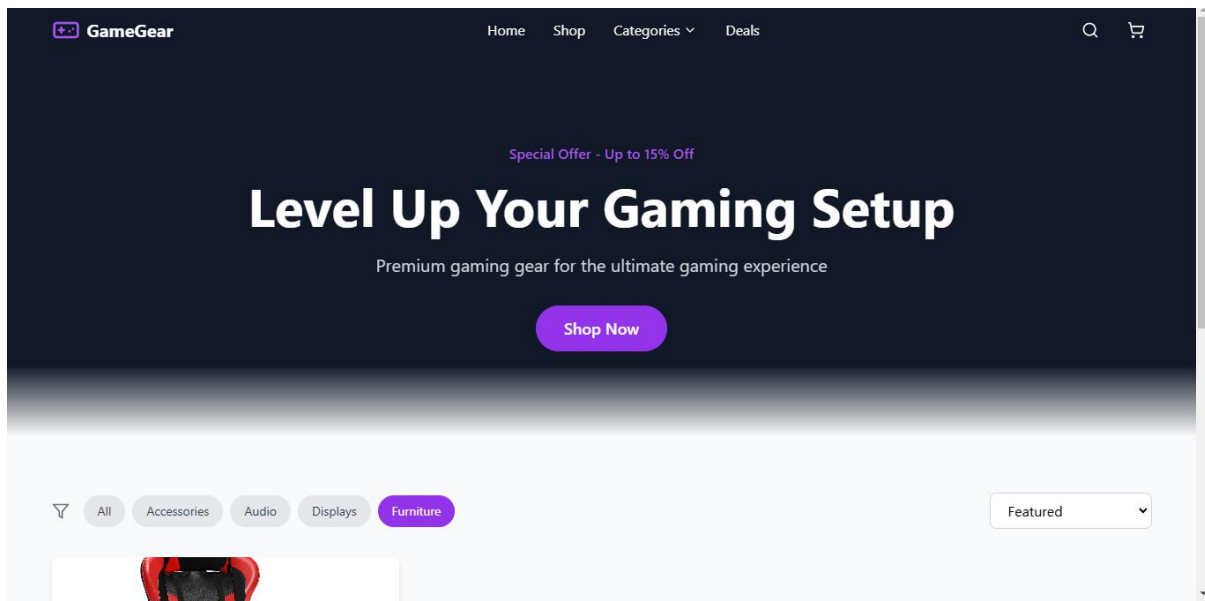


Figure 03 :Page d'accueil

B. Design UI/UX

Le **Design UI/UX** se concentre sur l'expérience utilisateur (UX) et l'interface utilisateur (UI) du site, visant à offrir une navigation fluide, intuitive et agréable. Il combine des principes de conception esthétique et de fonctionnalité.

- **UI (Interface Utilisateur)** : L'interface utilisateur englobe tous les éléments visuels avec lesquels l'utilisateur interagit, comme les boutons, les menus, les icônes et les champs de formulaire. Une bonne interface utilisateur doit être attrayante, fonctionnelle et facile à utiliser.
- **UX (Expérience Utilisateur)** : L'UX concerne la manière dont l'utilisateur interagit avec le site et l'efficacité de cette interaction.

VI. Planification et Gestion du Projet

A. Répartition des tâches et responsabilités.

La **répartition des tâches** est une étape cruciale pour garantir le bon déroulement du projet et s'assurer que chaque membre de l'équipe travaille sur des aspects précis du projet en fonction de ses compétences et de ses responsabilités. Dans notre projet de création de site e-commerce, les tâches ont été divisées de manière équilibrée afin d'assurer une progression fluide et un respect des délais. Voici la répartition des principales responsabilités :

1. Chef de projet :

- Suivi global du projet.
- Gestion des ressources et des délais.
- Communication avec les parties prenantes.

2. Développeur Front-End :

- Conception de l'interface utilisateur (UI).
- Développement des pages web avec HTML, CSS, JavaScript.
- Intégration des maquettes et wireframes.

3. Développeur Back-End :

- Création des services de gestion des produits, des commandes et du paiement.
- Intégration de la base de données et gestion des transactions.
- Mise en place de l'API RESTful pour l'échange de données avec le front-end.

4. Designer UI/UX :

- Conception des maquettes visuelles (mockups).
- Développement du design graphique du site.

- Tests d'utilisabilité pour s'assurer de la fluidité de l'interface.

5. Testeur / QA (Quality Assurance) :

- Création des plans de tests unitaires et d'intégration.
- Exécution des tests pour valider la fonctionnalité des différentes parties du site.
- Vérification de la performance et de l'accessibilité.

B. Planning et diagramme de Gantt

La planification est essentielle pour gérer efficacement le projet et respecter les délais impartis. Le diagramme de Gantt est un outil visuel qui nous permet de suivre l'avancement des tâches du projet, de coordonner les efforts de l'équipe et de nous assurer que chaque étape est réalisée dans les temps.

Un diagramme de Gantt présente les différentes étapes du projet sous forme de barres chronologiques, avec la durée estimée pour chaque tâche. Il montre aussi les interdépendances entre les tâches et les dates de début et de fin.

Le projet a été planifié sur 6 semaines avec les étapes suivantes :

- **Analyse** : Collecte des besoins et faisabilité (5 jours).
- **Conception** : Création et validation des maquettes (7 jours).
- **Développement** : Front-end et back-end (25 jours).
- **Tests** : Tests fonctionnels et de performance (7 jours).
- **Lancement** : Mise en production et suivi post-lancement (7 jours).

Le diagramme de Gantt permet également de suivre la progression et de réajuster les tâches en cas de retard, garantissant ainsi que le projet respecte les délais prévus.

C. Gestion de version avec Git et GitHub

La **gestion de version** est un aspect clé du développement collaboratif, permettant à chaque membre de l'équipe de travailler sur des parties du projet sans interférer avec les autres. Pour ce projet, nous avons utilisé **Git** comme système de gestion de versions et **GitHub** comme plateforme de stockage et de collaboration.

Les principales étapes de la gestion de version incluent :

- **Création de branches** : Chaque fonctionnalité (comme la page d'accueil, la gestion des produits, etc.) a été développée dans des branches séparées afin d'éviter les conflits de code.
- **Commits réguliers** : Chaque modification a été enregistrée avec des messages de commit clairs pour suivre l'évolution du code.
- **Pull requests** : Lorsqu'une fonctionnalité est terminée, une pull request est ouverte pour intégrer la branche dans la branche principale (**master/main**). Les autres membres de l'équipe peuvent alors réviser le code avant son fusionnement.
- **Gestion des conflits** : Lorsqu'un conflit survient (par exemple, deux développeurs modifiant la même partie du code), il est résolu manuellement en combinant les modifications de manière cohérente.
- **Utilisation de GitHub pour les mises à jour** : GitHub permet de stocker tout le code source du projet, et grâce à ses outils, nous avons pu suivre les problèmes et les suggestions via des issues et des discussions.

Cette gestion de version garantit un flux de travail harmonieux et permet une collaboration efficace entre les membres de l'équipe.

VII. Implémentation.

A. Fonctionnalités principales développées

Dans cette section, nous citons brièvement les principales fonctionnalités du site e-commerce que nous avons développées.

- **Page d'accueil dynamique** : Affichage des produits populaires, offres spéciales, et options de recherche.
- **Gestion des produits** : Ajout, modification et suppression des produits par l'administrateur.
- **Panier d'achat** : Ajout et suppression d'articles, affichage du total, et possibilité de passer à la caisse.
- **Processus de commande** : Sélection du mode de paiement, validation des informations de livraison, et confirmation de la commande.
- **Espace client** : Création de compte, suivi des commandes, modification des informations personnelles.
- **Système de paiement sécurisé** : Intégration avec des API de paiement comme PayPal, Stripe, ou carte bancaire.
- **Notifications par email** : Envoi de confirmations de commande et notifications de mise à jour.

B. Code source (description générale des modules)

Modules Front-End :

- HTML / CSS / JavaScript : Utilisés pour la structure, le style et l'interactivité de la page.
- HTML : Définit la structure de la page (navigation, produits, panier).
- CSS : Responsable de la mise en forme des éléments (design responsive, thèmes).
- JavaScript : Gère les interactions (ajout au panier, affichage des détails produits).
- React.js : Framework utilisé pour construire l'interface utilisateur dynamique.

- Composants React pour afficher les produits, gérer l'état du panier, et afficher les résultats des recherches.

Modules Back-End :

- Node.js : Serveur pour gérer les requêtes HTTP, la logique métier et la communication avec la base de données.
- Express.js : Framework utilisé pour créer des routes API RESTful permettant de gérer les produits, les commandes et les utilisateurs.
- Base de données (MongoDB) : Stocke les informations des utilisateurs, produits, commandes et paiements.
- Modèles : Schémas définissant les données, tels que User, Product, Order.

Intégration des API :

- API de paiement (Stripe/PayPal) : Module qui gère les paiements sécurisés en intégrant des API tierces.
- Système d'authentification : Utilisation de JWT (JSON Web Tokens) pour la gestion des sessions et de l'authentification des utilisateurs.

Déploiement :

- Docker : Conteneurisation des services pour faciliter le déploiement en environnement de production.
- GitHub : Utilisation de Git pour le contrôle de version, hébergement du code source et gestion des branches.

VIII. Tests et Validation

A. Plan de test unitaire

Le **test unitaire** consiste à tester chaque composant du système de manière isolée pour s'assurer que chaque unité fonctionne comme prévu. Voici un exemple de plan de test pour certaines fonctionnalités clés du site e-commerce :

Modules à tester :

- **Module d'authentification (Login / Inscription)**
 - ❖ Tester la validation des formulaires (email, mot de passe).
 - ❖ Tester la gestion des erreurs (mots de passe incorrects, email non valide).
 - ❖ Tester le mécanisme de session utilisateur après connexion.
- **Module de gestion des produits**
 - ❖ Tester l'ajout, la modification et la suppression des produits.
 - ❖ Tester la gestion des erreurs, comme la tentative d'ajout de produits sans informations suffisantes.
 - ❖ Tester la validation des prix, descriptions, et images des produits.
- **Module du panier d'achat**
 - ❖ Tester l'ajout d'articles au panier.
 - ❖ Tester la suppression et la modification de la quantité des produits dans le panier.
 - ❖ Tester le calcul du total du panier (avec les remises, les frais de livraison).
- **Module de paiement**
 - ❖ Tester l'intégration avec l'API de paiement (ex. Stripe).
 - ❖ Tester le traitement des transactions avec des cartes valides et invalides.
 - ❖ Tester le comportement du site lors d'une transaction échouée.
- **Outils recommandés**
 - ❖ Jest : Pour l'écriture des tests unitaires JavaScript.
 - ❖ Mocha ou Chai : Pour les tests de logique back-end.

B. Plan de test d'intégration

Le **test d'intégration** s'assure que les différents modules du système fonctionnent ensemble de manière cohérente. Ce type de test permet de valider que les interactions entre le front-end et le back-end, ainsi que les communications avec la base de données et les API externes, se déroulent sans erreur.

Scénarios d'intégration :

- **Intégration front-end et back-end :**
 - ❖ Tester la communication entre le client et le serveur pour la création de commandes.
 - ❖ Vérifier la récupération et l'affichage des produits depuis le back-end vers l'interface utilisateur.
 - ❖ Tester l'ajout d'un article au panier, puis la validation de la commande.
- **Tests d'intégration avec la base de données**
 - ❖ Vérifier la bonne insertion, mise à jour et suppression des données dans la base de données (utilisateurs, produits, commandes).
 - ❖ Tester la gestion des erreurs de la base de données (connexion échouée, données invalides).
- **Tests d'intégration avec les API externes (ex. PayPal/Stripe)**
 - ❖ Tester la communication entre le site e-commerce et l'API de paiement.
 - ❖ Vérifier que les informations de paiement sont correctement envoyées et traitées.

- **Outils recommandés :**

- ❖ Supertest : Pour tester les routes API.
- ❖ Postman : Pour tester manuellement les appels d'API.

IX. Déploiement

A. Environnement de production

L'**environnement de production** est la configuration où le site e-commerce sera déployé et accessible par les utilisateurs finaux. Il doit être stable, sécurisé, et optimisé pour supporter un grand nombre de visiteurs simultanés. Voici les principales caractéristiques de l'environnement de production :

Infrastructure serveur :

- Hébergement : Le site sera hébergé sur un serveur cloud, comme Amazon Web Services (AWS), Microsoft Azure, ou DigitalOcean.
- Serveur web : Le serveur web utilisé sera Nginx ou Apache pour gérer les requêtes HTTP.
- Base de données : Utilisation de MySQL ou PostgreSQL pour stocker les données des utilisateurs, produits, et commandes.
- Cache : Un système de cache comme Redis peut être utilisé pour améliorer les performances des requêtes fréquemment demandées.

Sécurité

- SSL/TLS : Le site doit être accessible via HTTPS pour assurer la sécurité des données échangées.
- Firewall : Des pare-feux doivent être configurés pour protéger le serveur contre les attaques.

- Authentification : L'utilisation de mécanismes d'authentification sécurisés (ex. JWT ou OAuth) pour protéger l'accès aux données sensibles.
- Sauvegarde : Des stratégies de sauvegarde régulières pour éviter la perte de données.

Performance

- CDN (Content Delivery Network) : Utilisation d'un CDN pour distribuer les ressources statiques (images, CSS, JS) afin d'améliorer les temps de réponse.
- Compression : Compression des fichiers CSS, JavaScript et images pour réduire la latence.
- Monitoring : Des outils comme Prometheus ou Datadog pour surveiller la performance et la disponibilité du site en production.

Version du logiciel

- Node.js : Version stable et compatible avec le projet (par exemple, la version LTS).
- Nginx/Apache : Versions récentes avec les dernières mises à jour de sécurité.

B. Processus de déploiement

Le **processus de déploiement** décrit les étapes permettant de déployer une nouvelle version du site en production, depuis la préparation du code jusqu'à la mise en ligne. Il peut inclure l'automatisation pour garantir une mise en production sans erreur. Voici le processus de déploiement typique pour le site e-commerce :

Préparation :

- Vérification du code : Assurez-vous que le code est prêt pour la mise en production. Cela inclut la révision de tous les modules, tests unitaires et d'intégration.
- Versioning : Taggez la version stable dans le système de gestion de version (Git).
- Configuration : Vérifiez les paramètres de production (base de données, API de paiement, etc.).

Automatisation du déploiement :

- CI/CD (Intégration continue / Déploiement continu) : Utilisation d'une plateforme comme GitHub Actions, Jenkins, ou GitLab CI/CD pour automatiser les tests, la compilation, et le déploiement du site.
- Build de l'application : Compilez et minimisez les fichiers JavaScript et CSS (si nécessaire) pour le déploiement.

Transfert des fichiers :

- Déploiement du front-end : Les fichiers statiques (HTML, CSS, JS) sont transférés vers le serveur de production via FTP/SFTP ou un service comme AWS S3 ou Netlify.
- Déploiement du back-end : Le code serveur est mis à jour sur le serveur de production, en utilisant un service comme Docker pour la conteneurisation, ou via un simple transfert via SSH.

Mise à jour de la base de données :

- Migration des données : Si des changements dans la structure de la base de données sont nécessaires, effectuez des migrations en toute sécurité.

Utilisez des outils comme Flyway ou Liquibase pour gérer les migrations SQL.

Tests en production

- Tests fonctionnels : Testez les fonctionnalités principales sur le serveur de production pour vous assurer que tout fonctionne comme prévu.
- Tests de performance : Vérifiez que le site peut supporter un grand nombre de connexions simultanées sans affecter la performance.

X. Conclusion et Perspectives

A. Résumé des réalisations

Le projet de création du site e-commerce a été mené à bien en respectant les différentes étapes du développement, depuis l'analyse des besoins jusqu'à la mise en production. Les principales fonctionnalités du site ont été développées, notamment la gestion des produits, des commandes et des paiements. L'interface utilisateur a été conçue de manière intuitive, avec une expérience utilisateur optimisée. Le processus de déploiement a été réalisé avec succès, et le site est désormais accessible aux utilisateurs, offrant une plateforme fiable et fonctionnelle.

B. Limitations du projet

Malgré la réussite globale du projet, certaines limitations sont à prendre en compte. Tout d'abord, certaines fonctionnalités avancées, telles que la gestion des promotions et des recommandations personnalisées, n'ont pas été mises en œuvre dans cette version. De plus, la plateforme nécessite encore des améliorations en termes de performance, notamment pour gérer un grand nombre d'utilisateurs simultanés. Le processus de paiement, bien que

fonctionnel, pourrait également bénéficier de l'intégration de méthodes de paiement supplémentaires.

C. Améliorations futures

Pour les futures versions du site e-commerce, plusieurs améliorations peuvent être envisagées :

- **Optimisation des performances** : Améliorer la vitesse et la réactivité du site, surtout en cas de fort trafic.
- **Fonctionnalités avancées** : Intégrer des fonctionnalités supplémentaires comme la gestion des promotions, des recommandations personnalisées, ainsi que des options de filtrage des produits plus détaillées.
- **Sécurité renforcée** : Mettre en œuvre des mesures de sécurité plus strictes, notamment pour la gestion des données sensibles des utilisateurs et des transactions financières.
- **Amélioration de l'UI/UX** : Améliorer l'interface utilisateur en fonction des retours des utilisateurs et ajouter des éléments interactifs pour rendre l'expérience encore plus fluide.
- **Extensions à la gestion mobile** : Développer une version mobile du site, ou une application dédiée, pour répondre à la demande croissante des utilisateurs sur smartphone.

XI. Annexes.

A. Documentation utilisateur

1 : Création et gestion de compte

- Instructions pour la création d'un compte client :

- ✓ Allez sur la page d'inscription accessible depuis l'écran d'accueil ou via le menu de navigation.
- ✓ Entrez vos informations personnelles, notamment votre nom, prénom, adresse email, et mot de passe.
- ✓ Après avoir rempli le formulaire d'inscription, cliquez sur le bouton S'inscrire pour créer votre compte.
- ✓ Un email de confirmation vous sera envoyé pour valider votre adresse email. Cliquez sur le lien de validation dans l'email.
- Authentification :
 - ✓ Une fois inscrit, vous pouvez vous connecter en utilisant votre adresse email et votre mot de passe.
 - ✓ Accédez à la page de connexion via le menu en haut à droite de la page d'accueil.
 - ✓ Saisissez vos informations et cliquez sur Se connecter. Si vous avez oublié votre mot de passe, utilisez l'option de récupération via le lien Mot de passe oublié ?
- Gestion des informations personnelles :
 - ✓ Une fois connecté, allez dans la section Mon compte ou Profil.
 - ✓ Vous pourrez y mettre à jour vos informations personnelles, notamment votre adresse de livraison, votre email, et votre mot de passe.
 - ✓ Assurez-vous de sauvegarder les modifications après chaque mise à jour.

2. Navigation sur le site

- Explication des différentes sections du site :
 - ✓ **Page d'accueil** : La page d'accueil présente une vue d'ensemble du site, incluant les produits populaires, les promotions en cours, et les catégories principales de produits.
 - **Il y a un carrousel avec les meilleures offres et des liens rapides vers des sections comme Nouveautés et Promotions.**
 - **Un champ de recherche permet aux utilisateurs de trouver rapidement des produits en utilisant des mots-clés.**

- ✓ **Page des produits** : Cette page affiche une liste complète de produits. Vous pouvez filtrer par catégorie, prix, ou marque.
 - **Les produits sont affichés avec leur nom, prix, et image.**
 - **Chaque produit a un bouton Ajouter au panier et un lien vers sa page de détails où vous trouverez des informations supplémentaires telles que la description, les avis des clients, et les options de taille ou de couleur si disponibles.**
- ✓ **Gestion du panier** : Une fois des produits ajoutés au panier, un icône de panier dans la barre de navigation montre le nombre d'articles.
 - **Vous pouvez accéder à votre panier à tout moment en cliquant dessus. Cette page vous permet de visualiser les articles ajoutés, de modifier les quantités, ou de supprimer des produits.**
 - **Un bouton Passer à la caisse vous redirige vers le processus de commande.**

3. Passer une commande

- Guide étape par étape pour effectuer une commande :
 - ✓ Étape 1 : Choisir les produits :
 - **Naviguez sur le site et ajoutez les produits que vous souhaitez acheter à votre panier.**
 - **Allez dans le panier une fois que vous avez terminé vos sélections.**
 - ✓ Étape 2 : Vérification du panier :
 - **Vérifiez que les produits, quantités et prix sont corrects. Si nécessaire, modifiez les quantités ou supprimez des articles.**
 - **Si vous avez un code promo, entrez-le dans le champ prévu à cet effet avant de passer à la caisse.**
 - ✓ Étape 3 : Informations de livraison :
 - **Entrez votre adresse de livraison. Si vous avez déjà un compte, vos informations enregistrées seront automatiquement pré-remplies.**

- **Choisissez une méthode de livraison parmi les options proposées (standard, express, etc.).**
- ✓ Étape 4 : Choisir une méthode de paiement :
 - **Sélectionnez votre mode de paiement préféré (carte bancaire, PayPal, etc.).**
 - **Entrez les informations de paiement de manière sécurisée.**
- ✓ Étape 5 : Confirmation de la commande :
 - **Avant de finaliser l'achat, un récapitulatif de la commande est affiché avec le total des achats, les frais de livraison, et le mode de paiement choisi.**
 - **Cliquez sur Confirmer la commande pour la valider.**
- ✓ Étape 6 : Confirmation de commande :
 - **Après la validation de l'achat, vous recevrez une confirmation de commande par email avec les détails de votre achat et un numéro de suivi.**

4 :Suivi des commandes

Comment suivre le statut des commandes en cours et consulter l'historique des commandes passées :

- Accéder à l'historique des commandes :
 - ✓ **Allez dans votre Compte ou Mon profil pour consulter l'historique des commandes.**
 - ✓ **Toutes les commandes passées sont listées avec leur statut actuel (en traitement, expédiée, livrée).**
- Suivi de la commande :
 - ✓ Pour une commande en cours, un numéro de suivi vous sera fourni dès que l'expédition aura lieu. Vous pouvez suivre l'état de la livraison via ce numéro sur le site de l'entreprise de livraison.
- Modification ou annulation de commande :
 - ✓ Si vous avez effectué une erreur dans votre commande ou si vous souhaitez l'annuler avant expédition, contactez immédiatement le service client via la section *Contact* du site.

5 :Service client

- Contact avec le support pour toute question ou problème lié à l'utilisation du site :
- Assistance par chat :
 - ✓ Un support en ligne est disponible via un chat en direct pendant les heures d'ouverture. Un agent du service client pourra répondre à vos questions en temps réel.
- Email et téléphone
 - ✓ Vous pouvez également contacter le support par email à l'adresse *support@ecommerce.com* pour toute question ou problème. Le support est également disponible par téléphone au numéro suivant : +212 XXX XXX XXX.
- Retours et remboursements :
 - ✓ Si vous souhaitez retourner un produit, consultez la politique de retour sur le site. Vous devez remplir un formulaire de retour disponible dans votre compte sous la section *Mes commandes* et envoyer le produit dans son état original à l'adresse indiquée.

B. Codes sources supplémentaires

Le code source du projet est disponible dans le dépôt GitHub du projet. Vous y trouverez des exemples de code, des modules spécifiques, des scripts d'intégration ainsi que des tests unitaires. Ces ressources sont accessibles pour consultation ou pour une utilisation future dans des projets similaires.

- Lien vers le dépôt GitHub
- Modules spécifiques inclus :
 - ✓ Module de gestion des produits : Gestion des stocks, des informations sur les produits, et des catégories.
 - ✓ Module de gestion des commandes : Processus de commande, traitement des paiements, et gestion des livraisons.
 - ✓ Module de paiement sécurisé : Intégration avec des services de paiement tiers.
 - ✓ Tests unitaires : Scripts de test pour valider les fonctionnalités clés du site.

C. Références

Voici une liste des références utilisées pour la réalisation de ce projet :

- Documentation officielle de React.js - <https://reactjs.org/docs/getting-started.html> Fournit des informations sur l'utilisation de React pour le développement d'applications web interactives.
- Documentation de Node.js - <https://nodejs.org/en/docs>
Guide pour comprendre et utiliser Node.js dans le développement backend du site e-commerce.
- Bootstrap Documentation - <https://getbootstrap.com/docs>
- Documentation de Bootstrap utilisée pour le développement de l'interface responsive du site.
- Stripe API Documentation - <https://stripe.com/docs>
- Guide d'intégration pour les paiements en ligne via Stripe.
- MDN Web Docs - <https://developer.mozilla.org/fr>
- Source principale pour le développement web en JavaScript, HTML, et CSS.

Ces ressources ont été essentielles pour la réalisation technique du projet, de l'intégration des fonctionnalités à l'optimisation des performances.