



Université Abdelmalek Essaadi  
**FACULTÉ POLYDISCIPLINAIRE  
À LARACHE**



**Département Informatique**  
**Master DevOps et Cloud Computing**

**Rapport de Projet**  
**Application e-commerce (mini)**

**Réalisé par**

**Hajar El Bachiri**  
**Fatima Zahra Boutefah Khatib**  
**Anass Eraoui**  
**Khalid Lazrag**

**Encadré par**

**Pr. Mohamed EL Mahjouby**

**Année universitaire 2025–2026**

# Dédicaces

Au nom d'Allah, le Clément, le Miséricordieux.

Nous dédions humblement ce travail à toutes les personnes qui nous ont soutenus et encouragés tout au long de la réalisation de ce mini-projet.

À nos chères familles, pour leur soutien constant, leurs encouragements et leur confiance. Leur appui moral a été une source de motivation essentielle dans notre parcours universitaire.

À notre encadrant, pour son accompagnement, ses conseils avisés et sa disponibilité durant la réalisation de ce projet. Son encadrement nous a permis de progresser et d'approfondir nos connaissances.

À nos camarades et amis, avec qui nous partageons cette expérience universitaire, pour leur esprit de collaboration et leur soutien.

Merci à tous.

# Remerciements

Nous tenons à exprimer nos sincères remerciements à notre encadrant, pour son accompagnement, sa disponibilité et ses conseils pertinents tout au long de la réalisation de ce mini-projet. Son encadrement nous a permis de mieux comprendre les concepts abordés dans le cadre de ce module et d'améliorer nos compétences techniques.

Nous remercions également l'ensemble des responsables du Master DevOps et Cloud Computing pour la qualité de la formation dispensée, ainsi que pour l'environnement pédagogique favorable à l'apprentissage et au développement de nos compétences.

Enfin, nous adressons nos remerciements à toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail.

# Résumé

Ce rapport présente le travail réalisé dans le cadre du mini-projet du module « Applications Distribuées » au sein de la Faculté Polydisciplinaire à Larache.

Dans un contexte où le commerce électronique connaît une croissance rapide, les plateformes en ligne permettent aux utilisateurs d'acheter des produits de manière simple, rapide et sécurisée. Ces applications reposent sur des architectures distribuées assurant la gestion efficace des données, la communication entre les différentes couches du système et la sécurisation des transactions.

L'objectif principal de ce projet est de concevoir et développer une application e-commerce permettant la gestion des produits, des utilisateurs et des commandes, tout en mettant en œuvre une architecture backend structurée et sécurisée.

Pour ce faire, nous avons utilisé Spring Boot et Spring MVC pour le développement du backend, Thymeleaf pour la couche de présentation, ainsi qu'une base de données MySQL pour la persistance des données. La sécurisation de l'application a été assurée par l'utilisation de JSON Web Token (JWT).

**Mots clés :** application web, e-commerce, applications distribuées, Spring Boot, Thymeleaf, MySQL, JWT.

# Liste des abréviations

Abréviation	Signification
API	Application Programming Interface (Interface de Programmation d'Application)
CSS	Cascading Style Sheets (Feuilles de Style en Cascade)
CRUD	Create, Read, Update, Delete (Créer, Lire, Mettre à jour, Supprimer)
DTO	Data Transfer Object (Objet de Transfert de Données)
HTML	HyperText Markup Language (Langage de Balises Hyper-texte)
HTTP	HyperText Transfer Protocol (Protocole de Transfert Hyper-texte)
JPA	Java Persistence API (API de Persistance Java)
JSON	JavaScript Object Notation (Notation d'Objet JavaScript)
JSP	JavaServer Pages (Pages Serveur Java)
JWT	JSON Web Token (Jeton Web JSON)
MVC	Model-View-Controller (Modèle-Vue-Contrôleur)
MySQL	Structured Query Language – Système de base de données relationnelle
ORM	Object-Relational Mapping (Mapping Objet-Relationnel)
PFE	Projet de Fin d'Études
REST	Representational State Transfer
SGBD	Système de Gestion de Base de Données
SMI	Sciences Mathématiques et Informatique
SQL	Structured Query Language (Langage de Requête Structuré)
UML	Unified Modeling Language (Langage de Modélisation Unifié)
UI	User Interface (Interface Utilisateur)
URL	Uniform Resource Locator (Localisateur de Ressource Uniforme)
XML	eXtensible Markup Language (Langage de Balisage Extensible)

# Table des matières

Dédicaces	i
Remerciements	ii
Résumé	iii
Liste des abréviations	iv
Liste des tableaux	viii
Introduction générale	1
Contexte général du projet	3
1.1 Introduction . . . . .	4
1.2 Présentation du projet . . . . .	4
1.3 Contexte du projet . . . . .	4
1.4 Problématique . . . . .	5
1.5 Objectifs du projet . . . . .	5
1.6 Exigences fonctionnelles . . . . .	5
1.7 Exigences non fonctionnelles . . . . .	6
1.8 Étude de l'existant . . . . .	7
1.9 Solution proposée . . . . .	7
1.10 Fonctionnalités principales . . . . .	7
1.11 Conclusion . . . . .	7
Déroulement du projet	9
2.1 Introduction . . . . .	10
2.2 Planification opérationnelle . . . . .	10
2.2.1 Tableau des tâches . . . . .	10
2.3 Processus de développement . . . . .	10
2.3.1 Cycle de vie du développement . . . . .	10
2.3.2 Phases du développement . . . . .	11
2.4 Conclusion . . . . .	11
Spécification et modélisation	13

---

3.1	Introduction . . . . .	14
3.2	Identification des acteurs . . . . .	14
3.3	Diagrammes de cas d'utilisation . . . . .	14
3.4	Modélisation des données . . . . .	14
3.5	Diagrammes de séquence . . . . .	15
3.6	Conclusion . . . . .	15
<b>Implémentation et réalisation</b>		<b>16</b>
4.1	Introduction . . . . .	17
4.2	Environnement et outils de développement . . . . .	17
4.2.1	Spring Boot . . . . .	17
4.2.2	Spring MVC . . . . .	17
4.2.3	Spring Data JPA . . . . .	17
4.2.4	MySQL . . . . .	18
4.2.5	Thymeleaf . . . . .	18
4.2.6	Lombok . . . . .	18
4.3	Architecture du projet . . . . .	18
4.4	Conclusion . . . . .	19
<b>Conclusion générale</b>		<b>20</b>

# Table des figures



# Liste des tableaux

# Introduction générale

L'évolution rapide des technologies numériques a profondément transformé les modes de consommation et les pratiques commerciales. Le commerce électronique, communément appelé e-commerce, s'est imposé comme un élément central de l'économie moderne. Il permet aux utilisateurs d'accéder à des services et à des produits à distance, tout en offrant aux entreprises de nouvelles opportunités de gestion, de visibilité et de croissance. Cette transformation repose en grande partie sur le développement d'applications informatiques distribuées capables de garantir performance, fiabilité et sécurité.

Les applications distribuées jouent aujourd'hui un rôle fondamental dans la conception des systèmes modernes. Elles permettent la séparation des responsabilités, la gestion efficace des données et la communication entre plusieurs composants logiciels. Dans le domaine du e-commerce, ces caractéristiques sont essentielles pour assurer la disponibilité des services, la protection des informations sensibles et une expérience utilisateur fluide.

Dans le cadre du module « Applications Distribuées », nous avons été amenés à concevoir et développer une mini-application e-commerce illustrant les principes étudiés durant le semestre. Ce projet vise à mettre en pratique les concepts théoriques liés aux architectures client-serveur, à la structuration d'une application web moderne, ainsi qu'à la sécurisation des échanges entre les différents composants du système.

L'application développée propose un ensemble de fonctionnalités essentielles à une plateforme de vente en ligne : gestion des produits, authentification des utilisateurs, manipulation du panier d'achat et traitement des commandes. Elle repose sur un backend construit avec Spring Boot et Spring MVC, une interface web utilisant Thymeleaf, ainsi qu'une base de données relationnelle MySQL. La sécurité est assurée par l'utilisation de JSON Web Tokens (JWT), garantissant un accès contrôlé aux ressources.

Dans le premier chapitre, nous présentons le cadre général du projet ainsi que les notions fondamentales liées aux applications distribuées et au commerce électronique. Nous y introduisons les concepts théoriques nécessaires à la compréhension de l'architecture adoptée, notamment le modèle client-serveur, la séparation des couches applicatives et les principes de communication entre les composants d'un système distribué.

Le deuxième chapitre est consacré à l'analyse et à la conception du système. Il décrit les besoins fonctionnels et non fonctionnels de l'application, les différents acteurs intervenant dans le système, ainsi que la modélisation UML comprenant les diagrammes de cas d'utilisation et le diagramme de classes. Cette phase permet de structurer l'application avant son implémentation et d'assurer une vision claire de son organisation

interne.

Le troisième chapitre détaille l'architecture technique et l'environnement de développement. Il présente les technologies utilisées, notamment Spring Boot, Spring MVC, Thymeleaf et MySQL, ainsi que le mécanisme d'authentification basé sur JWT. Ce chapitre met en évidence la structuration en couches (contrôleurs, services, accès aux données) et explique le rôle de chaque composant dans le fonctionnement global de l'application.

Enfin, le quatrième chapitre expose la phase d'implémentation et les principaux résultats obtenus. Il décrit les fonctionnalités réalisées, les interfaces développées et les interactions entre le frontend et le backend. Les difficultés rencontrées ainsi que les solutions adoptées y sont également présentées.

Ce rapport se termine par une conclusion générale qui synthétise les apports pédagogiques du projet et met en lumière les perspectives d'amélioration envisageables.

# CHAPITRE 1

---

Contexte général du projet

---

## 1.1 Introduction

Dans un contexte marqué par la digitalisation croissante des activités commerciales, les entreprises, qu'elles soient petites, moyennes ou grandes, ont désormais besoin de solutions informatiques leur permettant de gérer efficacement leurs opérations. La gestion des produits, des commandes, des utilisateurs et des transactions constitue aujourd'hui un élément central de la compétitivité.

Ce chapitre présente le cadre général de notre projet. Il expose le contexte, la problématique, les objectifs, ainsi qu'une étude de l'existant afin de justifier la solution proposée.

## 1.2 Présentation du projet

Le projet réalisé consiste en la conception et le développement d'un système de gestion e-commerce permettant d'administrer une plateforme de vente en ligne.

Un système e-commerce peut être défini comme : Une application web permettant la gestion des produits, des commandes, des utilisateurs et des transactions via Internet.

Selon International Business Machines (IBM), le commerce électronique désigne l'ensemble des transactions commerciales effectuées électroniquement via des réseaux numériques, principalement Internet [1].

Notre projet vise donc à développer une application web permettant :

- La gestion des produits
- La gestion des utilisateurs
- La gestion du panier
- Le traitement des commandes
- La gestion des codes promos

## 1.3 Contexte du projet

Le commerce électronique connaît une croissance mondiale continue. D'après Statista, le chiffre d'affaires mondial du e-commerce dépasse plusieurs milliers de milliards de dollars par an, avec une croissance soutenue.

Au Maroc également, le secteur connaît une expansion importante grâce à :

- La démocratisation d'Internet
- L'augmentation des paiements en ligne
- L'évolution des habitudes d'achat

Cependant, de nombreuses petites structures ne disposent pas de systèmes de gestion adaptés. Certaines utilisent :

- Des outils manuels
- Des solutions non intégrées

- Des plateformes complexes difficiles à personnaliser

Notre projet s'inscrit dans une démarche de conception d'un système e-commerce structuré, pédagogique et techniquement cohérent.

## 1.4 Problématique

Le développement d'un système de gestion e-commerce implique bien plus que la simple création d'une interface de vente en ligne. Il nécessite la mise en place d'une architecture cohérente permettant de gérer efficacement les produits, les catégories, les utilisateurs et les commandes, tout en garantissant la sécurité et la fiabilité des données.

De nombreuses plateformes existantes comme Shopify ou Magento proposent des solutions complètes, mais leur utilisation ne permet pas toujours de comprendre les mécanismes internes d'un système e-commerce. Dans un cadre académique, il est essentiel de concevoir et développer une solution personnalisée afin de maîtriser les concepts fondamentaux d'architecture logicielle et de gestion de base de données.

Ainsi, la problématique de ce projet consiste à concevoir un système de gestion e-commerce structuré, sécurisé et maintenable, capable d'administrer efficacement les opérations commerciales tout en respectant les bonnes pratiques du développement web.

## 1.5 Objectifs du projet

L'objectif principal de ce projet est de concevoir et développer un système de gestion e-commerce permettant d'administrer efficacement une plateforme de vente en ligne. Il s'agit de mettre en place une application web structurée, sécurisée et maintenable, capable de gérer les produits, les utilisateurs et les commandes. Ce projet vise également à appliquer les concepts étudiés durant la formation, notamment la conception orientée objet, la modélisation UML, la gestion des bases de données et l'architecture des applications web modernes.

## 1.6 Exigences fonctionnelles

Les exigences fonctionnelles définissent les services et les fonctionnalités que le système doit fournir aux utilisateurs. Elles décrivent précisément les tâches que l'application doit accomplir afin de répondre aux besoins identifiés.

### a. Authentification et gestion des utilisateurs

- Les utilisateurs doivent pouvoir créer un compte et s'authentifier à l'aide d'identifiants sécurisés.
- Le système doit permettre la gestion des rôles (administrateur, utilisateur, etc.).

- L'administrateur doit pouvoir consulter, modifier ou supprimer les comptes utilisateurs.
- b. Gestion des produits**
  - L'administrateur doit pouvoir ajouter un nouveau produit.
  - Le système doit permettre la modification et la suppression des produits existants.
  - Les produits doivent être affichés sous forme de liste organisée.
- c. Gestion des catégories**
  - L'administrateur doit pouvoir créer, modifier et supprimer des catégories.
  - Chaque produit doit être associé à une catégorie.
  - Le système doit permettre l'affichage des produits par catégorie.
- d. Gestion des commandes**
  - Les utilisateurs doivent pouvoir passer une commande.
  - Le système doit enregistrer automatiquement les informations relatives à chaque commande.
  - L'administrateur doit pouvoir consulter l'historique des commandes.
- e. Consultation et suivi des opérations**
  - Le système doit permettre l'affichage des détails d'une commande.
  - Les informations doivent être présentées de manière claire.
  - Le système doit conserver un historique des opérations.

## 1.7 Exigences non fonctionnelles

Les exigences non fonctionnelles décrivent les qualités attendues du système. Elles ne concernent pas directement les fonctionnalités, mais plutôt les critères de performance, de sécurité, de fiabilité et de maintenabilité que l'application doit respecter.

### a. Sécurité

Le système doit garantir la protection des données et sécuriser l'authentification des utilisateurs.

### b. Performance

L'application doit répondre rapidement aux requêtes et gérer efficacement les interactions avec la base de données.

### c. Fiabilité

Les données doivent être correctement enregistrées et rester cohérentes.

### d. Maintenabilité

Le code doit être structuré, organisé et documenté afin de faciliter les évolutions futures.

e. **Évolutivité**

L'architecture doit permettre l'ajout de nouvelles fonctionnalités sans modification majeure du système existant.

f. **Utilisabilité**

L'interface doit être claire, intuitive et facile à utiliser.

## 1.8 Étude de l'existant

De nombreuses plateformes e-commerce existent aujourd'hui, allant des solutions complètes destinées aux grandes entreprises aux applications simplifiées pour les petites structures. Ces solutions offrent souvent des fonctionnalités avancées mais peuvent présenter une complexité élevée ou des contraintes techniques.

Dans un cadre pédagogique, développer une application dédiée permet de mieux comprendre les mécanismes internes des systèmes distribués sans dépendre d'outils préconfigurés.

## 1.9 Solution proposée

La solution développée repose sur une architecture web moderne intégrant un backend basé sur Spring Boot et Spring MVC, une interface utilisateur dynamique avec Thymeleaf, et une base de données MySQL.

L'application met en œuvre un système d'authentification sécurisé utilisant JWT, garantissant un contrôle d'accès aux différentes fonctionnalités.

## 1.10 Fonctionnalités principales

L'application propose les fonctionnalités suivantes :

- Gestion des produits
- Gestion des utilisateurs
- Gestion du panier
- Traitement des commandes
- Authentification sécurisée
- Consultation des données en temps réel

## 1.11 Conclusion

Ce premier chapitre a permis de présenter le cadre général du projet, son contexte, sa problématique et ses objectifs. L'étude de l'existant a montré l'importance des systèmes



e-commerce modernes et a justifié le développement d'une solution personnalisée dans un cadre académique.

Le chapitre suivant décrira le déroulement du projet, la planification des tâches et le processus de développement adopté.

# CHAPITRE 2

---

Déroulement du proje

---

## 2.1 Introduction

Ce chapitre présente le déroulement du projet depuis sa planification initiale jusqu'à sa mise en œuvre finale. Il décrit l'organisation des tâches, la méthodologie adoptée et les différentes phases du développement. L'objectif est de montrer comment le projet a été structuré afin d'assurer une progression logique, maîtrisée et conforme aux objectifs pédagogiques du module « Applications Distribuées ».

## 2.2 Planification opérationnelle

Afin de garantir une réalisation efficace du projet, une planification opérationnelle a été mise en place dès le début du développement. Le travail a été découpé en tâches principales correspondant aux différentes fonctionnalités de l'application ainsi qu'aux aspects techniques liés à l'architecture distribuée.

Chaque tâche a été organisée selon son niveau de priorité, sa dépendance vis-à-vis des autres composants et sa complexité. Cette organisation a permis de suivre une progression cohérente tout en facilitant la validation intermédiaire des fonctionnalités développées.

### 2.2.1 Tableau des tâches

Le tableau suivant présente une vue synthétique des principales tâches du projet, leur description et leur objectif.

(Espace réservé pour le tableau des tâches ou diagramme de Gantt — à compléter ultérieurement)

## 2.3 Processus de développement

### 2.3.1 Cycle de vie du développement

Le développement de l'application a suivi une approche itérative et incrémentale. Chaque phase du projet a été réalisée progressivement, permettant de valider les fonctionnalités au fur et à mesure de leur implémentation.

Cette méthode favorise une meilleure compréhension des interactions entre les différentes couches de l'application, tout en réduisant les risques d'erreurs majeures lors de l'intégration finale.

Le choix de cette approche s'explique par :

— la modularité de l'architecture de l'application,

- la possibilité de tester chaque composant indépendamment,
- la facilité d'intégration progressive des fonctionnalités.

### 2.3.2 Phases du développement

Le processus de développement a été structuré en plusieurs phases successives :

#### **Phase 1 : Analyse des besoins**

Cette phase a consisté à identifier les fonctionnalités essentielles d'une application e-commerce distribuée. Les besoins fonctionnels et non fonctionnels ont été définis afin d'établir une base claire pour le développement.

#### **Phase 2 : Conception de l'architecture**

Une architecture client-serveur a été conçue en séparant la couche présentation, la logique métier et la gestion des données. La sécurité de l'application a été intégrée dès cette étape.

#### **Phase 3 : Mise en place de l'environnement**

Les outils de développement ont été configurés, incluant le framework Spring Boot, la base de données MySQL et l'environnement de test. Cette étape a permis de préparer une base stable pour le codage.

#### **Phase 4 : Développement des fonctionnalités**

Les modules principaux ont été implémentés progressivement :

- gestion des produits,
- gestion des utilisateurs,
- panier et commandes,
- authentification sécurisée.

Chaque fonctionnalité a été testée avant son intégration avec le reste du système.

#### **Phase 5 : Intégration de la sécurité**

Un mécanisme d'authentification basé sur JWT a été mis en place afin de contrôler l'accès aux différentes ressources et garantir la protection des données.

#### **Phase 6 : Tests et validation**

Des tests fonctionnels ont été réalisés pour vérifier le bon fonctionnement de l'application, la cohérence des données et la stabilité globale du système.

#### **Phase 7 : Documentation**

La dernière étape a consisté à rédiger le rapport du projet afin de documenter les choix techniques, la méthodologie adoptée et les résultats obtenus.

## 2.4 Conclusion

Ce chapitre a présenté l'organisation et la méthodologie suivies pour mener à bien le projet. La planification structurée et l'approche incrémentale ont permis de garantir

une progression cohérente et une intégration efficace des fonctionnalités. Ces éléments constituent une base solide pour aborder les aspects de conception et de modélisation présentés dans le chapitre suivant.

# CHAPITRE 3

---

Spécification et modélisation

---

## 3.1 Introduction

Après l'analyse des besoins et la définition du cadre du projet, cette étape vise à formaliser les spécifications fonctionnelles et techniques de l'application. La modélisation permet de traduire les exigences en représentations structurées afin de faciliter la conception et le développement.

Dans le cadre de notre application e-commerce de gestion commerciale, cette modélisation décrit les acteurs impliqués, les interactions avec le système, ainsi que l'organisation des données.

## 3.2 Identification des acteurs

L'application repose sur plusieurs profils utilisateurs disposant de droits spécifiques :

- **Administrateur** : gère les utilisateurs, les produits, les commandes et supervise l'ensemble du système.
- **Gestionnaire** : responsable des ventes, du stock et des commandes.
- **Client** : consulte les produits, passe des commandes et suit ses achats.

Cette séparation garantit une gestion sécurisée et adaptée aux responsabilités de chaque utilisateur.

## 3.3 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation illustrent les interactions entre les acteurs et le système :

- Authentification et gestion des comptes
- Gestion des produits
- Gestion du stock
- Gestion des commandes
- Consultation de l'historique

Ces diagrammes permettent de visualiser les fonctionnalités accessibles selon chaque profil utilisateur.

## 3.4 Modélisation des données

Le diagramme de classes définit les principales entités :

- Utilisateur
- Produit
- Commande

- Client
- Paiement

Chaque entité contient des attributs spécifiques permettant une gestion cohérente des informations commerciales.

## 3.5 Diagrammes de séquence

Les diagrammes de séquence décrivent le déroulement des interactions pour des scénarios clés :

- Connexion utilisateur
- Passage de commande
- Mise à jour du stock

Ils illustrent la communication entre les composants du système.

## 3.6 Conclusion

La modélisation fournit une vision claire de l'architecture fonctionnelle du système. Elle constitue une base solide pour l'implémentation en assurant cohérence, organisation et évolutivité.



# CHAPITRE 4

---

Implémentation et réalisation

---

## 4.1 Introduction

Ce chapitre présente la phase d'implémentation du système de gestion e-commerce. Après avoir étudié les besoins et conçu l'architecture du projet, nous avons procédé au développement concret de l'application en utilisant un ensemble de technologies modernes adaptées aux applications web professionnelles.

Nous détaillerons dans un premier temps l'environnement de développement et les outils utilisés, puis nous présenterons les différentes fonctionnalités implémentées à travers des captures d'écran commentées.

## 4.2 Environnement et outils de développement

### 4.2.1 Spring Boot

Spring Boot est un framework open source basé sur l'écosystème Spring, destiné à simplifier le développement d'applications Java. Il permet de créer rapidement des applications autonomes, prêtes à être déployées, en réduisant la configuration manuelle.

Il repose sur le principe de la configuration automatique (Auto-Configuration) et intègre un serveur embarqué (Tomcat par défaut), ce qui permet d'exécuter l'application sans configuration complexe.

Dans notre projet, Spring Boot constitue le framework principal du backend. Il assure la gestion des requêtes HTTP, l'intégration avec la base de données et la structuration du projet selon une architecture en couches [2].

### 4.2.2 Spring MVC

Spring MVC (Model-View-Controller) est un module du framework Spring permettant la gestion des requêtes web selon le modèle architectural MVC.

Il sépare l'application en trois composants :

- Modèle (Model) : gestion des données
- Vue (View) : affichage des informations
- Contrôleur (Controller) : traitement des requêtes et coordination entre le modèle et la vue

Dans notre application, Spring MVC permet de gérer les endpoints (URLs), traiter les requêtes des utilisateurs et retourner les réponses appropriées [3].

### 4.2.3 Spring Data JPA

Spring Data JPA est un module facilitant l'accès aux bases de données relationnelles en utilisant le concept d'ORM (Object Relational Mapping).

Il permet de manipuler les données via des interfaces Repository sans écrire manuellement les requêtes SQL complexes [4].

Dans notre projet, Spring Data JPA est utilisé pour :

- L’insertion des produits
- La récupération des catégories
- la gestion des commandes
- l’authentification des utilisateurs

#### 4.2.4 MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) open source. Il permet de stocker, organiser et gérer les données de manière structurée [5].

Dans notre application, MySQL est utilisé pour :

- Stocker les informations des utilisateurs
- Enregistrer les produits
- Gérer les catégories
- Conserver les commandes

#### 4.2.5 Thymeleaf

Thymeleaf est un moteur de templates Java utilisé pour générer des pages HTML dynamiques côté serveur.

Il permet d’intégrer des données provenant du backend directement dans les pages HTML, tout en gardant un code propre et lisible [6].

#### 4.2.6 Lombok

Lombok est une bibliothèque Java qui réduit le code répétitif (boilerplate code) en générant automatiquement des méthodes comme getters, setters, constructeurs, etc.

Dans notre projet, Lombok simplifie la définition des entités et améliore la lisibilité du code [7].

### 4.3 Architecture du projet

L’application est structurée selon une architecture en couches :

- Couche Controller : gestion des requêtes HTTP
- Couche Service : logique métier
- Couche Repository : accès aux données
- Couche Model (Entity) : représentation des tables de la base de données

Cette organisation permet une meilleure maintenabilité et séparation des responsabilités.

## 4.4 Conclusion

L'implémentation confirme la faisabilité technique du projet. Le système répond aux objectifs définis tout en restant extensible.

# Conclusion générale

Ce projet a permis de concevoir et développer une application e-commerce orientée gestion commerciale, répondant aux besoins des petites structures souhaitant digitaliser leurs activités.

À travers les différentes phases — analyse, modélisation, implémentation — nous avons mis en place une solution structurée, sécurisée et évolutive. L'application facilite la gestion des produits, des commandes et des transactions tout en offrant une interface simple et accessible.

Ce travail a également renforcé nos compétences en conception logicielle, architecture d'applications web et gestion de bases de données.

En perspective, plusieurs améliorations peuvent être envisagées :

- Ajout d'un module de paiement en ligne
- Version mobile de l'application
- Tableaux analytiques avancés
- Intégration cloud

Ce projet constitue ainsi une base solide pour une solution commerciale évolutive adaptée au contexte numérique actuel.

# Webographie

- [1] Qu'est-ce que le e-commerce ?  
[https ://www.ibm.com/fr-fr/think/topics/ecommerce](https://www.ibm.com/fr-fr/think/topics/ecommerce)
- [2] Spring Boot Documentation  
[https ://spring.io/projects/spring-boot](https://spring.io/projects/spring-boot)
- [3] Spring MVC Documentation  
[https ://docs.spring.io/spring-framework/reference/web/webmvc.html](https://docs.spring.io/spring-framework/reference/web/webmvc.html)
- [4] Spring Data JPA Documentation  
[https ://spring.io/projects/spring-data-jpa](https://spring.io/projects/spring-data-jpa)
- [5] MySQL Documentation  
[https ://dev.mysql.com/doc/](https://dev.mysql.com/doc/)
- [6] Thymeleaf Documentation  
[https ://www.thymeleaf.org/documentation.html](https://www.thymeleaf.org/documentation.html)
- [7] Lombok Documentation  
[https ://projectlombok.org/features/all](https://projectlombok.org/features/all)