

Filière :

Informatique et Management des Systèmes

Encadré par :

M. Boukhdir Khalid

par :

DRISS DKHISSI

SALMA SABIR

Rapport de Projet

Smart Inventory & Sales Management System

Février 2026

Table des matières

1	Introduction	4
2	Structure Générale du Projet	4
3	Partie Métier : Classes et Logique	4
3.1	Classes Principales	4
3.2	Règles Métier	4
4	Base de Données	4
4.1	Tables Principales	5
5	Interface Web avec Django	5
5.1	Architecture	5
5.2	Fonctionnalités Web	5
6	Analyse des Données	5
6.1	Outils Utilisés	5
6.2	Résultats	5
7	Exemple de Code	6
8	Conclusion	6

1 Introduction

Ce rapport présente le projet **Smart Inventory & Sales Management System**, disponible sur GitHub à l'adresse suivante :

<https://github.com/Driss278/projet-de-Smart-Inventory-Sales-Management-System>

L'objectif de ce projet est de concevoir une application permettant la gestion intelligente des stocks, des ventes et des clients, tout en intégrant une interface web et des outils d'analyse de données.

Ce système vise à remplacer les méthodes manuelles par une solution numérique fiable, évolutive et facile à utiliser.

2 Structure Générale du Projet

Le projet est organisé en quatre parties principales :

- La logique métier et les classes Python
- La base de données relationnelle
- L'interface web développée avec Django
- L'analyse des données avec Pandas, NumPy et Matplotlib

Cette séparation permet une meilleure organisation du code et facilite la maintenance du système.

3 Partie Métier : Classes et Logique

La logique métier est implémentée à l'aide de classes Python représentant les entités principales du système.

3.1 Classes Principales

- **Product** : gère les informations liées aux produits (nom, prix, quantité en stock).
- **Customer** : représente les clients et leurs informations.
- **Order** : représente une commande effectuée par un client.
- **OrderItem** : représente les lignes de commande avec les quantités.

3.2 Règles Métier

Des règles métier sont appliquées afin de garantir la cohérence du système, notamment :

- interdiction des quantités négatives,
- vérification du stock avant validation d'une commande,
- gestion des erreurs à l'aide d'exceptions personnalisées.

4 Base de Données

Le stockage des données est assuré par une base de données relationnelle **MySQL**.

4.1 Tables Principales

Les principales tables sont :

- **products**
- **customers**
- **orders**
- **order_items**

Les relations entre ces tables sont gérées à l'aide de clés étrangères afin d'assurer l'intégrité des données.

5 Interface Web avec Django

Le framework **Django** est utilisé pour développer l'interface web de l'application.

5.1 Architecture

L'application suit l'architecture MVC :

- **Models** : représentent les données et les tables.
- **Views** : contiennent la logique de traitement.
- **Templates** : assurent l'affichage des pages HTML.

5.2 Fonctionnalités Web

- gestion des produits,
- gestion des clients,
- création et suivi des commandes,
- interaction simple et intuitive avec l'utilisateur.

6 Analyse des Données

Une partie du projet est dédiée à l'analyse des données de vente.

6.1 Outils Utilisés

- **Pandas** : manipulation et analyse des données.
- **NumPy** : calculs numériques.
- **Matplotlib** : visualisation graphique.

6.2 Résultats

Cette analyse permet d'obtenir :

- le chiffre d'affaires,
- les produits les plus vendus,
- l'évolution des ventes,
- la valeur totale du stock.

7 Exemple de Code

```
1 import pandas as pd
2
3 df = pd.read_csv("sales_data.csv")
4 print(df.head())
```

Listing 1 – Chargement et affichage des données avec Pandas

8 Conclusion

Le projet **Smart Inventory & Sales Management System** propose une solution complète de gestion des stocks et des ventes.

Il combine une logique métier claire, une base de données fiable, une interface web moderne et des outils d'analyse performants.

Ce projet constitue une base solide pouvant être enrichie par des fonctionnalités futures telles que la facturation, les alertes de stock ou une application mobile.