E-Commerce Management System – Project Report

1. Introduction

The E-Commerce Management System is a simple Python–SQL-based application designed to mimic the basic functionalities of an online shopping platform. It allows users to:

- Browse available products
- Add products to a cart
- Place an order

It also includes an Admin module used for:

- Adding new products
- Deleting products
- Viewing all orders

The project uses **Python**, **SQLite**, **HTML**, **CSS**, and **basic JavaScript** without advanced frameworks like Django or Flask. It is designed to be beginner-friendly and suitable for academic submission or a GitHub portfolio.

---

2. Technologies Used

- **Python** – Core backend logic
- **SQLite (SQL Database)** – Stores product and order data
- **HTML & CSS** – Basic UI pages
- **JavaScript** – Basic page interactivity

---

3. System Features

User Module

- View all available products
- Add selected products to cart
- Place an order
- View order summary

Admin Module

- Add new product (name, price, stock)
- Delete an existing product
- View all placed orders
- Update product stock

---

4. System Architecture

The project is divided into simple Python modules:

1 main.py

- Entry point of the application
- Shows user and admin menu options

2 database.py

- Creates the SQLite database file (ecommerce.db)
- Creates required tables:
    - products
    - orders

3 product_operations.py

Handles:

- Add product
- Delete product
- List all products

4 order_operations.py

Handles:

- Add items to cart
- Calculate bill
- Insert order into database

5 static/ (Folder)

- Contains basic HTML & CSS pages for UI
- Example: index.html, style.css

---

5. Database Design

Table 1: products

Column Name	Type	Description
id	INTEGER PRIMARY KEY	Product ID
name	TEXT	Product name
price	REAL	Product price
stock	INTEGER	Quantity in stock

Table 2: orders

Column Name	Type	Description
id	INTEGER PRIMARY KEY	Order ID
product_id	INTEGER	Product ordered
quantity	INTEGER	Number of items
total	REAL	Total bill

---

6. Workflow

User Workflow

1. User starts program → chooses "User"
2. Views product list
3. Selects product and quantity
4. Places order → Order stored in DB

Admin Workflow

1. Admin login
2. Adds or deletes products
3. Views available stock

4. Views all orders

---

7. Real-Life Applications

- Online shopping simulation
- Small business inventory management
- College mini-project demonstrating CRUD operations
- Training for Python + SQL beginners

---

8. Advantages

- Simple and easy to understand
- No complex frameworks needed
- Perfect for beginners
- Demonstrates real-world e-commerce logic
- Can be expanded into a bigger project

---

9. Limitations

- No authentication (optional improvement)
- Text-based Python interface (not web-based)
- Cart not stored permanently
- No payment integration

---

10. Conclusion

This project demonstrates how core e-commerce functionalities can be built using Python and SQL in a modular and beginner-friendly approach. It is ideal for academic submission, GitHub portfolio, or understanding CRUD operations.