UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

"EcommerceWebsite"

Submitted by:

Suhas B	**PES1UG21CS632**
Swaraj M K	**PES1UG21CS652**
Tharun M	**PES1UG21CS673**
Tilak Matagunde	**PES1UG21CS675**

6th Semester ***k*** *Section*

Team Number :

Prof. Bhargavi Mokashi
Assistant Professor

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING

PES UNIVERSITY

1. Introduction:

1.1. Problem Statement:

The rapid growth of e-commerce has introduced challenges in managing online stores effectively. Traditional shopping systems often lack the agility, scalability, and security required to meet the evolving demands of customers and businesses. Moreover, building a seamless integration between frontend and backend components poses a significant technical hurdle for developers. Our project aims to address these challenges by offering a streamlined solution that integrates a feature-rich frontend with a powerful backend, ensuring a smooth shopping experience for users while providing robust management capabilities for administrators.

1.2. Proposed Idea:

In the dynamic landscape of e-commerce, the need for efficient and reliable systems to facilitate online transactions is paramount. Our project, an E-commerce Shopping System, addresses this need by providing a comprehensive solution that encompasses both frontend and backend functionalities. Leveraging Object-Oriented Analysis and Design principles along with robust technologies like Spring Boot, Hibernate, Spring Security, and JavaFX, we have created a scalable, secure, and user-friendly application.

In summary, our E-commerce Shopping System represents a comprehensive solution to the challenges faced in the e-commerce domain. By combining cutting-edge technologies with sound design principles, we aim to revolutionize the online shopping experience for both customers and businesses.

2. Features:

- User authentication and authorization using Spring Security.
- RESTful APIs for managing products, orders, and user information.
- JavaFX interface to demonstrate the APIs

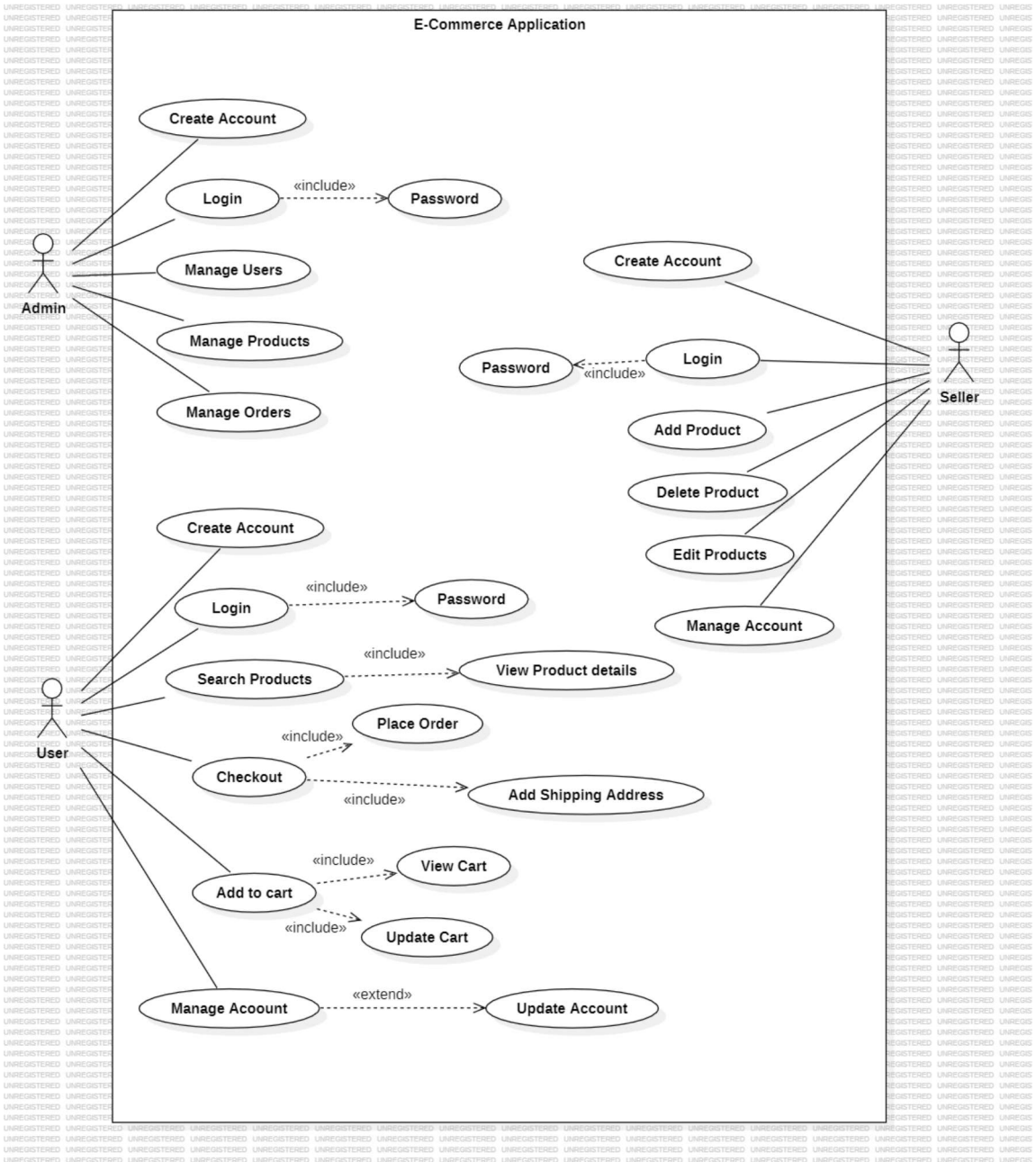
3. Technologies Used:

- **Frontend**
    1. JavaFX
- **Backend**
    1. Spring Boot
    2. Hibernate

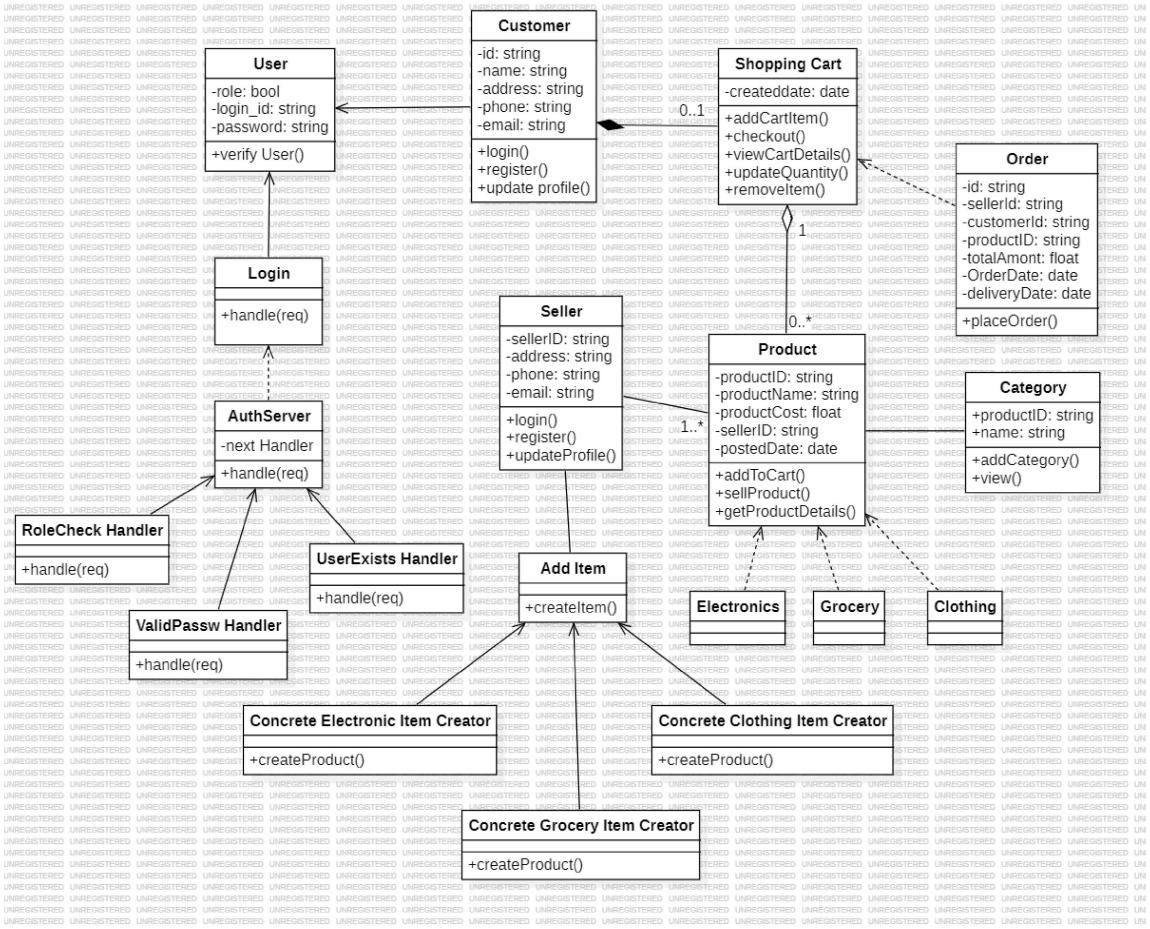
3. Spring Security
4.

4. Models:

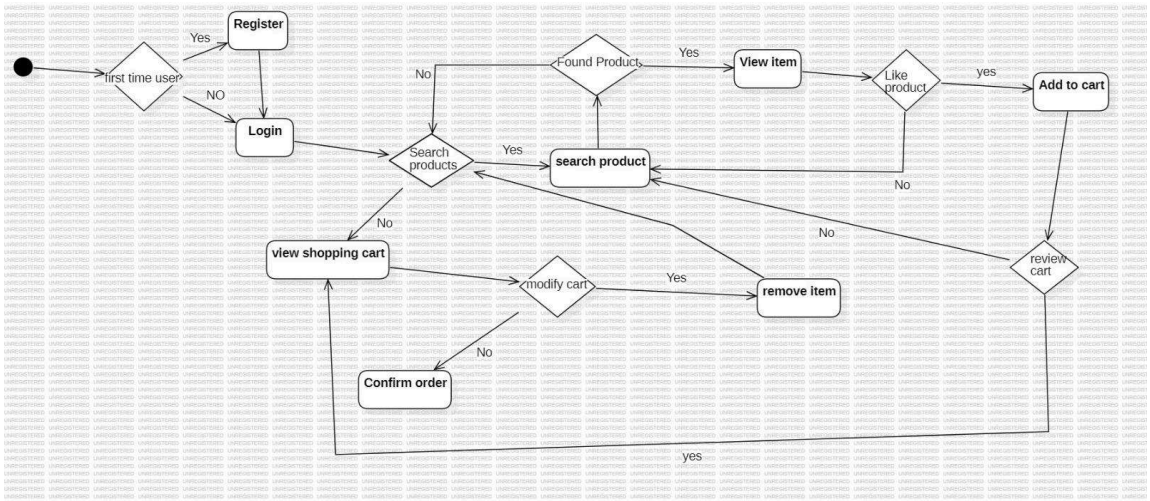
4.1. Use Case Diagram



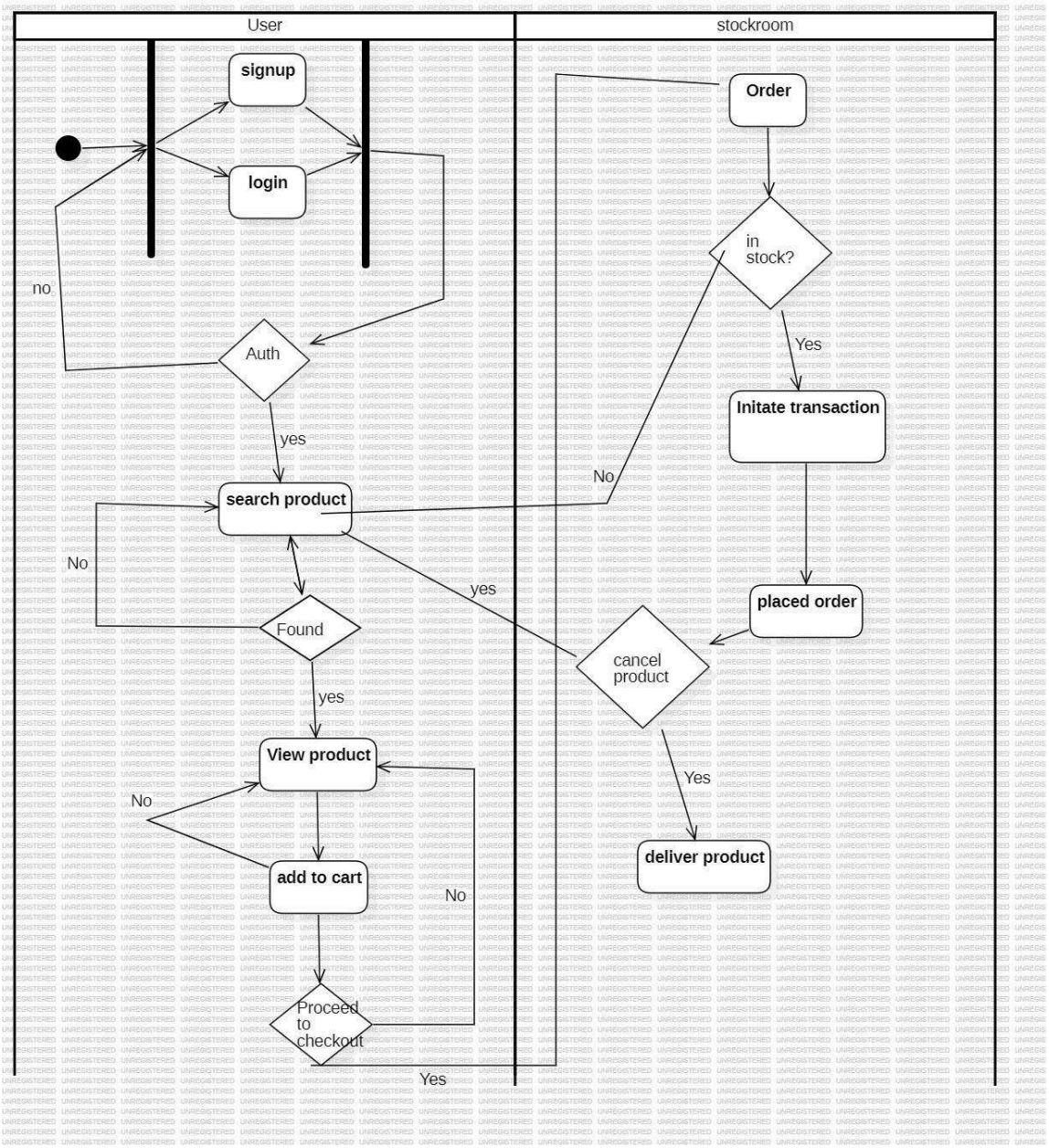
4.2. Class Diagram



4.3. State Diagram



4.4. Activity Diagram



5.Architecture Patterns, Design Principles, and Design Patterns

Based on the implementation of the project, the following architectural patterns, design principles, and design patterns are applied in the e-commerce system:

- **Architectural Patterns:**

1. Layered Architecture: The system follows a layered architecture pattern, where different layers handle specific concerns. For example, the `User`, `Customer`, and `Seller` classes represent the domain layer, while `AuthServer` and its handlers represent the authentication layer, and `ShoppingCart`, `Order`, and `Product` classes represent the business logic layer.

2. Client-Server Architecture: The system is designed to have a client-side (e.g., web or mobile application) and a server-side (backend) architecture. The client-side interacts with the server-side via RESTful APIs or other communication protocols.

- **Design Principles:**

1. Single Responsibility Principle (SRP): Each class has a specific responsibility. For example, the `User` class is responsible for user-related operations, `ShoppingCart` handles cart operations, and `Order` manages orders.

2. Open/Closed Principle (OCP): The use of inheritance and the Factory Method pattern allows for extending the system's functionality without modifying the existing code. For example, new product categories can be added by creating new concrete product creator classes without modifying the existing ones.

3. Liskov Substitution Principle (LSP): The inheritance relationships between `User`, `Customer`, and `Seller` classes follow the LSP, where objects of the derived classes (`Customer` and `Seller`) can be substituted for objects of the base class (`User`) without affecting the system's correctness.

4. Interface Segregation Principle (ISP): The use of specific handlers in the authentication process (`RoleCheckHandler`, `UserExistsHandler`, `ValidPasswHandler`) segregates the authentication concerns into separate interfaces, adhering to the ISP.

- **Design Patterns:**

1. Chain of Responsibility Pattern: The authentication process follows the Chain of Responsibility pattern, where the `AuthServer` class delegates the authentication request to a chain of handlers (`RoleCheckHandler`, `UserExistsHandler`, `ValidPasswHandler`).

2. Factory Method Pattern: The concrete product creator classes (`Concrete Electronic Item Creator`, `Concrete Clothing Item Creator`, `Concrete Grocery Item Creator`) implement the Factory Method pattern for creating products in their respective categories.

3. Observer Pattern: The use case diagram suggests the potential use of the Observer pattern, where the `ShoppingCart` class can observe changes in the `Product` class and update itself accordingly (e.g., when a product is added or removed from the cart).

4. State Pattern: The use case diagram shows different states or conditions that the system can be in, such as "Found Product" or "Like product," suggesting the potential use of the State Pattern to manage these states.

The architectural patterns, design principles, and design patterns used in the e-commerce system aim to promote modularity, maintainability, extensibility, and code reuse, ultimately contributing to the development of a robust and scalable e-commerce application.

6.Github link to the Codebase:

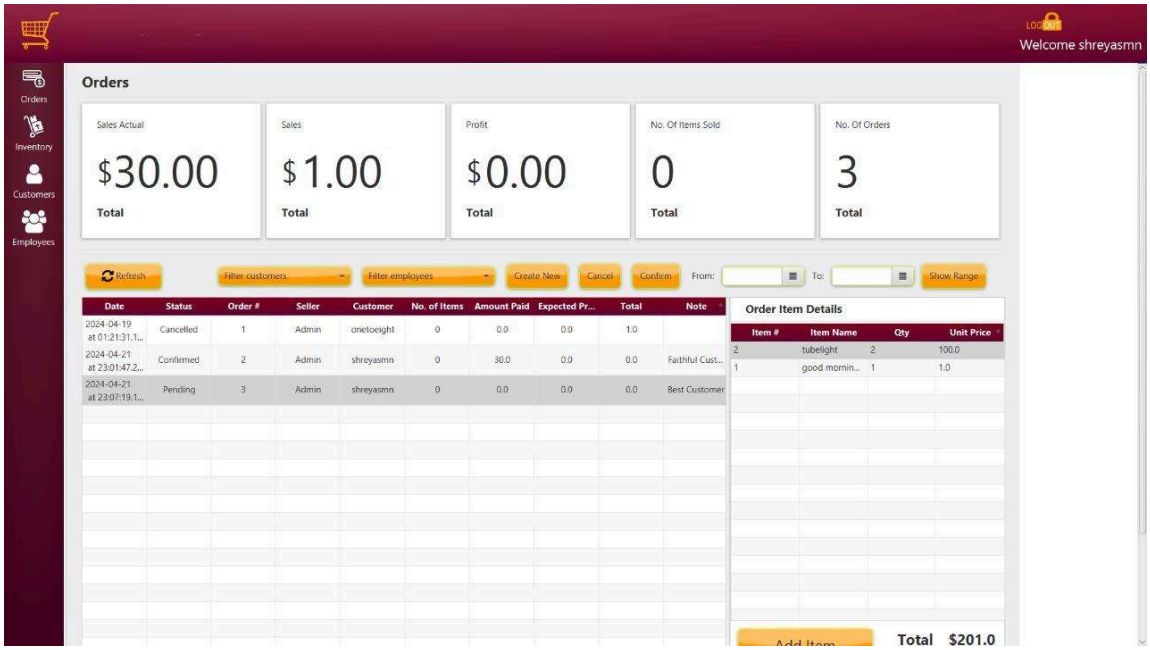
https://github.com/shreyas-m-n/OOADJEcommerce

7.Individual contributions of the team members.

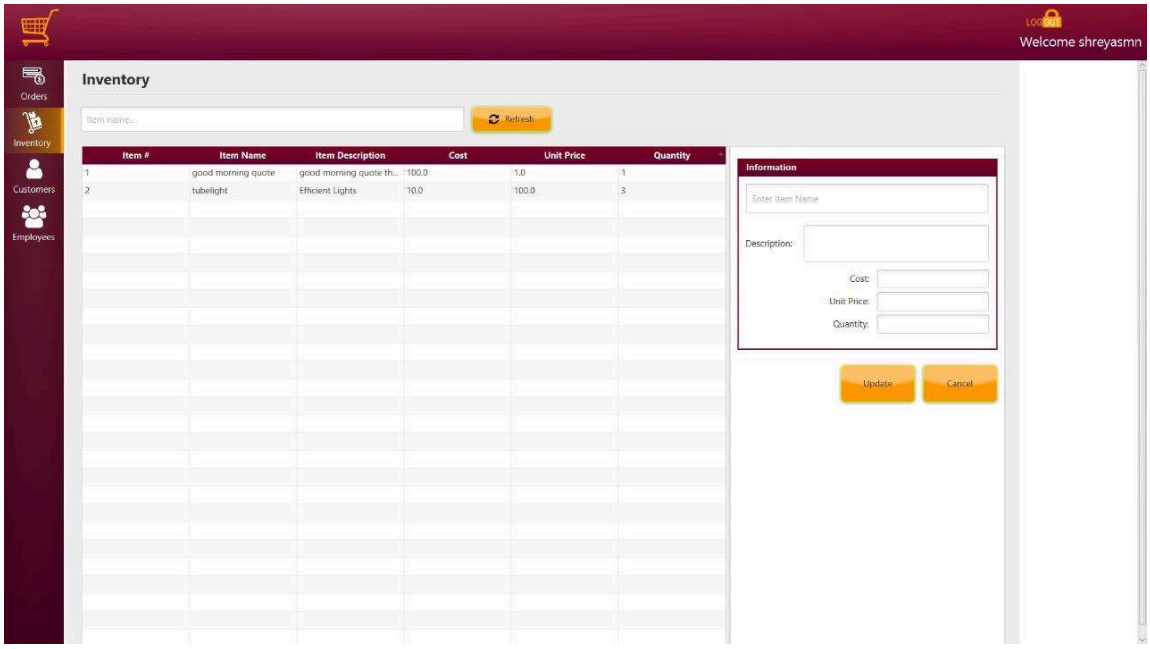
Sr.No	Name	Contribution
1.	Tharun M	Backend and Model Diagrams
2.	Tilak M	Backend and Models
3.	Suhas B	Frontend and Models
4.	Swaraj M K	Frontend and Models

8.Screenshots with input values populated and output shown (Use white background screens).

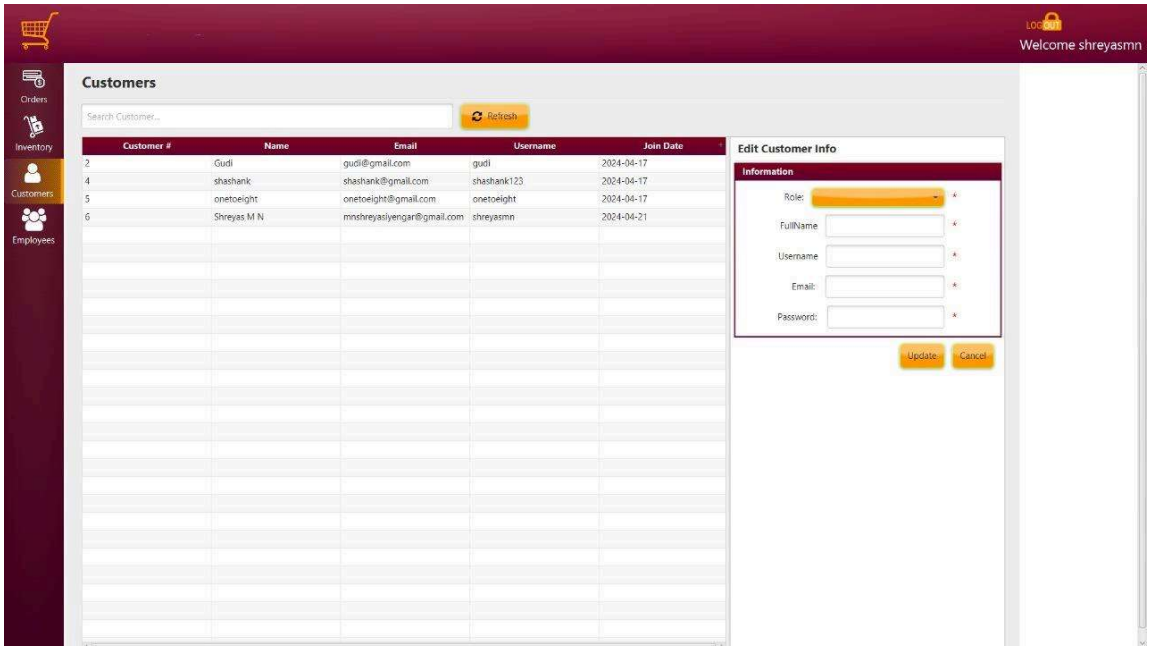
1. **Dashboard/Cart Page**



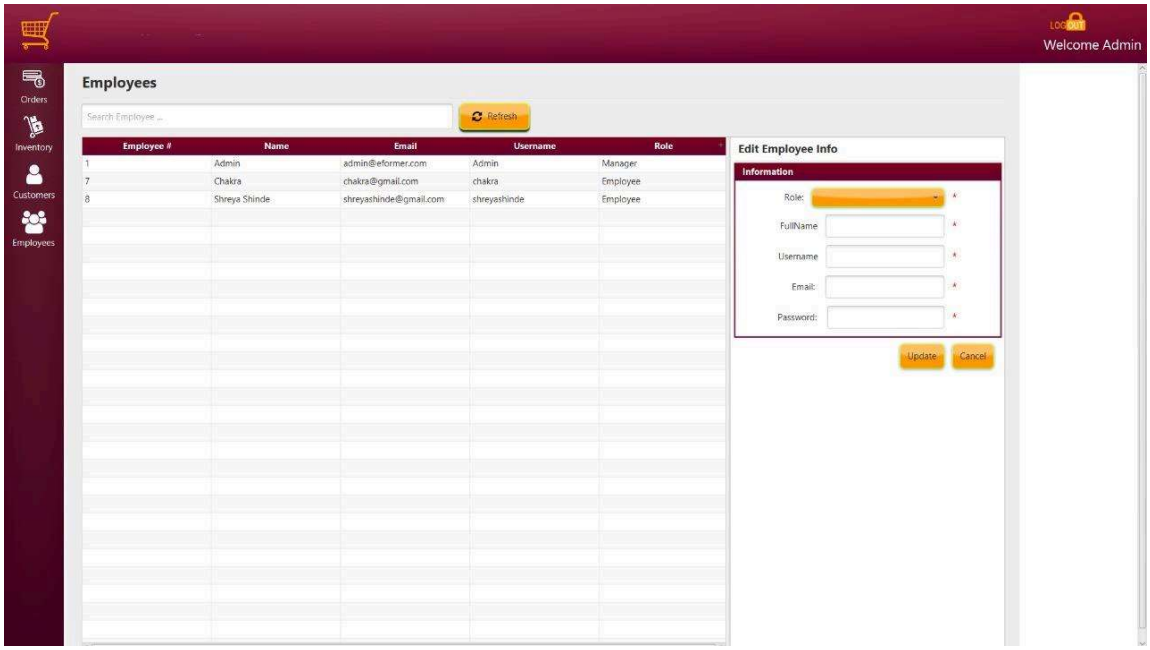
2. **Inventory Page**



3. Customers(Admin Side)



4. Employees.



5. Database

```
mysql> show tables;
+---------------------+
| Tables_in_eformer |
+---------------------+
| items               |
| order_items         |
| orders              |
| users               |
+---------------------+
4 rows in set (0.03 sec)

mysql> select * from items;
+---------+------+-----------------------------------------+----------------------------+--------------------+----------+------------+
| item_id | cost | description                             | introduction_date          | name               | quantity | unit_price |
+---------+------+-----------------------------------------+----------------------------+--------------------+----------+------------+
|       1 |  100 | good morning quote that says good night | 2024-04-17 09:41:46.526000 | good morning quote |        1 |          1 |
|       2 |   10 | Efficient Lights                        | 2024-04-19 10:18:26.123000 | tubelight          |        3 |        100 |
+---------+------+-----------------------------------------+----------------------------+--------------------+----------+------------+
2 rows in set (0.00 sec)

mysql> select * from order_items;
+---------+----------+----------+
| item_id | order_id | quantity |
+---------+----------+----------+
|       1 |        1 |        1 |
+---------+----------+----------+
1 row in set (0.01 sec)

mysql> select * from orders;
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
| order_id | amount_paid | creation_date              | note | number_of_items | profit | status  | total | customer_user_id | employee_user_id |
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
|        1 |           0 | 2024-04-19 01:21:31.124000 |      |               1 |    -99 | Pending |     1 |                5 |                1 |
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
1 row in set (0.00 sec)

mysql> select * from users;
+---------+----------+----------------------------+---------------------+-----------+--------------------------------------------------------------+-------
-------+
| user_id | ad_level | create_time                | email               | full_name | password                                                     | usern
ame    |
```

```
mysql> select * from orders;
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
| order_id | amount_paid | creation_date              | note | number_of_items | profit | status  | total | customer_user_id | employee_user_id |
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
|        1 |           0 | 2024-04-19 01:21:31.124000 |      |               1 |    -99 | Pending |     1 |                5 |                1 |
+----------+-------------+----------------------------+------+-----------------+--------+---------+-------+------------------+------------------+
1 row in set (0.00 sec)

mysql> select * from users;
+---------+----------+----------------------------+---------------------+-----------+--------------------------------------------------------------+-------
-------+
| user_id | ad_level | create_time                | email               | full_name | password                                                     | usern
ame    |
+---------+----------+----------------------------+---------------------+-----------+--------------------------------------------------------------+-------
-------+
|       1 |        2 | 2024-04-16 15:25:06.938000 | admin@eformer.com   | Admin     | $2a$10$yDOrkzYNBWbADeEh0U8.7.4I76frFpu5ieX2auWKS4Djj9V1DZJyu | Admin
      |
|       2 |        0 | 2024-04-17 10:50:41.304000 | gudi@gmail.com      | Gudi      | $2a$10$Gqcaq.Puxzw4HRVe.QRhweiz6Y45lin/rPLt1tz/U5jUzsMZdFfGy | gudi
      |
|       4 |        0 | 2024-04-17 12:44:26.743000 | shashank@gmail.com  | shashank  | $2a$10$WXhxJ2y3j5oEBH1uN.AeEeOJ6d0NXKtNQWBJyTxIRBtdoHFGjMYVC | shash
ank123 |
|       5 |        0 | 2024-04-17 12:47:17.316000 | onetoeight@gmail.com | onetoeight | $2a$10$.ANpn4Ts2AfiQjSLfX83feFRS8Co/th/uEO4OwBtx98RHRZx/HuNm | oneto
eight  |
+---------+----------+----------------------------+---------------------+-----------+--------------------------------------------------------------+-------
-------+
4 rows in set (0.00 sec)

mysql>
```

Thank You.