***E-Commerce Project Documentation***

**Team Members :**

1- Farah Mamdouh (2100685)

2-Mahmoud Asem Mahmoud (2100929)

3-Kerolos Mokhtar Sedki (2100871)

4-Mahfouz Mohamed Mahfouz (2100905)

5-Haitham Abdelfattah Diab (2100756)

**Project Overview**

This project is an e-commerce application developed in Java using JDBC for database interactions and a graphical user interface (GUI) for user interactions. It incorporates several design patterns such as Singleton, Observer, Factory, and State to enhance modularity, scalability, and maintainability

**Features**

1. **User Management**
   * Users can register and log in.
2. **Shopping Cart**
   * Users can browse products and add them to their cart.
   * Cart operations are handled using Singleton Design Pattern.
3. **Order Management**
   * Users can place orders.
   * Users receive notifications about order updates via Observer Design Pattern.
4. **Admin Management**
   * Admins can add products using the Factory Design Pattern.
   * Admins can update order states (Pending, Shipped, Delivered) using the State Design Pattern.
   * Admins can send notifications to users.

**Architecture and Design Patterns**

**1. Singleton Design Pattern**

* **Purpose**: Ensures a single instance of classes like CartManager.
* **Usage**:
  + CartManager: Manages the user's shopping cart.

**2. Observer Design Pattern**

* **Purpose**: Enables notifications to users about their order status.
* **Usage**:
  + NotificationService: Observes changes in order states and sends notifications to users.

**3. Factory Design Pattern**

* **Purpose**: Simplifies the creation of different types of products.
* **Usage**:
  + ProductFactory: Creates product instances based on type (e.g., Electronics, Clothing, etc.).

**4. State Design Pattern**

* **Purpose**: Manages the state transitions of orders.
* **Usage**:
  + OrderState: Defines different states (Pending, Shipped, Delivered) for an order.
  + OrderContext: Manages the current state of an order.

**Modules**

**1. User Module**

* **Registration**:
  + Input user details in database.
* **Login**:
  + Validate credentials against the database.

**2. Product Management Module**

* **For Users**:
  + View products fetched from the database.
  + Add selected products to the cart.
* **For Admins**:
  + Add products using the ProductFactory class.
  + Update product details.

**3. Cart Module**

* Add products to the cart.
* View items in the cart.
* Calculate total price.

**4. Order Module**

* Place an order from the cart.
* Update order states (Admin).
* Notify users of state changes.

**5. Notification Module**

* Notify users of order placement and updates using the NotificationService class.
* Admin can broadcast messages to all users.

**Class Design**

**1. Singleton Classes**

* UserSession: Manages logged-in user information.
* CartManager: Manages shopping cart operations.

**2. Observer Classes**

* NotificationService: Observes order state changes and sends notifications.

**3. Factory Classes**

* ProductFactory: Creates different types of product instances.

**4. State Classes**

* OrderState: Interface for order states.
  + PendingState
  + ShippedState
  + DeliveredState
* OrderContext: Manages the current state of an order.

**Tools and Technologies**

* **Language**: Java
* **Database**: SQL Server (using JDBC for interaction)
* **GUI Framework**: Swing/JavaFX
* **Version Control**: Git

**Conclusion**

This e-commerce project demonstrates the practical application of design patterns and best practices in Java. It provides a scalable and extensible architecture, ensuring a robust and user-friendly system for both customers and administrators.