Mini E_Commerce Platform - Java Swing Project Designed By Team:

Name	ID No.
1.Ayana Samuel	UGR/30201/15
2.Ayub Nasir	UGR/30219/15
3.Gemechu Alemu	UGR/30589/15
4. Megersa Tekalign	UGR/31494/15
5.Meklit H/Michael	UGR/30878/15
6.Million Mengstu	UGR/30932/15

Submitted To: Mr. Addisu

Table of Contents

1. Introduction

This **CommerceCore** project is an e-commerce platform designed using Java Swing to provide an interactive graphical user interface (GUI). The system allows companies to create accounts, post commodities for sale, and manage their inventory, while users can view these products, log in to their accounts, and make purchases. The application is integrated with an SQL database for managing user and company data, product listings, and transactions.

2. System Overview

The platform features two main types of users:

1. Companies: Can register, log in, post products, and manage their listings.

2. Customers: Can register, log in, browse products, and purchase items.

This Java-based desktop application leverages the Swing framework for GUI and JDBC for database management.

3. Functionalities

Company Features

- Account Creation: Companies can create new accounts, which are stored in the SQL database.
- Login: Companies can log in to manage their product listings.
- Product Posting: Companies can post products, including details such as name, description, price, and quantity.
- Inventory Management: Companies can update or delete product listings.

User Features

- Account Creation: Users can sign up and have their credentials securely stored in the database.
- Login: Users can log in to view and purchase products.
- Product Viewing: Users can browse products posted by various companies.
- Purchase Products: Users can select products, add them to the cart, and complete transactions.

4. Technology Stack

1. Frontend: Java Swing (JFrame, JPanel, JButton, JTextField)

2. Backend: Java with JDBC for SQL database integration

3. Database: MySQL or SQLite

4. Libraries/Packages:

- `**javax.swing**.` for GUI design

- `**java.sql.**` for database operations

- `**java.awt**.` for layout management

- Custom packages for authentication and panels

5. Class Structure and Key Components

- Main Class: Initializes the JFrame and sets the default layout and panels.
- CompanyPanel: Handles all operations related to the company's functionalities such as login, posting products, and managing inventory.

- ❖ UserPanel: Manages user operations like browsing products and completing purchases.
- ❖ SQLConnection: Manages database connectivity for user authentication and product management.
- ❖ CardLayout: Enables switching between user views (company, user, and home panels).

6. Database Integration

The platform connects to a SQL database that stores user accounts, company accounts, and product details. The `SQLConnection` class manages all interactions with the database, including user authentication, adding new products, and handling transactions. Tables include:

- ✓ Users: Stores user account information.
- ✓ Companies: Stores company account details.
- ✓ Products: Contains product listings, with references to the company that posted each product.
- ✓ Transactions: Logs purchase history and transaction details.

7. User Interface Design

The platform employs a simple and intuitive design. The home screen displays navigation buttons for users and companies. Companies are directed to a panel where they can log in and manage their products. Users can browse available products and add items to the cart for purchasing. The layout is managed using `CardLayout`, ensuring smooth transitions between different sections of the application.

8. Planning and Development Process

Step 1: Requirements Gathering

- Defined the main users (companies and customers) and their actions.
- Planned for user authentication and product management.

Step 2: GUI Design

- Used Java Swing to design a user-friendly interface with panels for each functionality.
- Implemented the `CardLayout` to handle panel transitions.

Step 3: Database Design

- Designed the SQL database schema with tables for users, companies, products, and transactions.

Step 4: Integration

- Connected the GUI to the SQL database using JDBC.
- Ensured seamless data flow between user actions (e.g., logging in, posting products) and database records.

Step 5: Testing and Debugging

- Tested user authentication, product posting, and transactions.
- Fixed bugs related to panel transitions and database queries.

9. Conclusion

This Java-based e-commerce platform offers a robust and scalable solution for companies and customers to interact. With a modular design using Java Swing and SQL database integration, it provides the essential functionalities for online retail operations, including account management, product posting, and purchasing.