E-COMMERCE PROJECT

PRESENTED TO DR.MAHMOUD KHALIL

PRESENTED TO ENG.MAHMOUD SOHEIL

YOUSSEF AHMED FAROUK 23P0054

YOUSSEF GEORGE 23P0227

MARWAN AHMEDABDELAZIZ 23P0242

ZIYAD MOHAMED ELNEMR 23P0013

AHMED MOHAMED ELSAYED 23P0035

Contents

Project Description

An **E-commerce application** built with **JavaFX** provides a desktop-based GUI for users to browse products, add items to the cart, and complete transactions. The backend, developed in Java, handles business logic, data storage, and processing.

Core Features

- GitHub Repository link: https://github.com/Knighler/OOPproject.git
- YouTube Video: https://youtu.be/vJtGlRv9Wq8

Core Features

Frontend (JavaFX)

- **Homepage**
  - A welcoming page displaying featured products and categories.
  - Implemented using JavaFX layouts like VBox, GridPane, and ScrollPane.
- **Product Details Page**
  - A detailed view of a product with an image, description, price, and "Buy Now" option.
  - Includes components like ImageView, Label, and Button.
- **Shopping Cart**
  - A separate page to view added items, total price, and checkout button.
  - Uses TableView for item details.
- **User Account Section**
  - Allows users to view their profile, order history, and update details.

Backend (Java)

- **Data Management**
    - A Product class to store product information (ID, name, price, description, stock).
    - A User class to manage user details.
    - A Cart class to manage items added by the user.

- **Business Logic**
    - Implemented using Java classes to handle operations like adding to the cart, checking stock, and calculating the total price.
- **Payment Processing**
    - Simulate payment processing logic.

Technology Stack

Frontend

- **JavaFX**: For building a responsive and modern user interface.
- **FXML**: This is used to define UI layout in XML format.
- **CSS**: For styling the JavaFX components.

Backend

- **Java**: For application logic and data processing.

Workflow

User Workflow

- The user registers/logs in.
- Browse the product catalog and view product details.
- Adds items to the cart and proceeds to checkout.

- Confirms the order and receives a summary.

Admin Workflow

- Admin adds, updates, or deletes products.
- Views Database

Problem Description

Challenges Faced in E-commerce Applications:

1. **User Navigation**
   - Users need an easy way to browse products and navigate between sections like the product catalog and cart.
2. **Data Management**
   - Efficiently storing, retrieving, and updating product and user data.
3. **Dynamic UI Updates**
   - The interface must reflect real-time changes, such as adding items to the cart.
4. **Order Processing**
   - Ensuring accurate calculations of prices and availability.

Detailed Solution

User Features

1. **Homepage**
   - Displays featured products with options to navigate to categories or a search bar for quick access.
2. **Product Details**
   - Displays detailed information about a selected product (image, description, price).
   - Adds an "Add to Cart" button with a click event to update the cart.

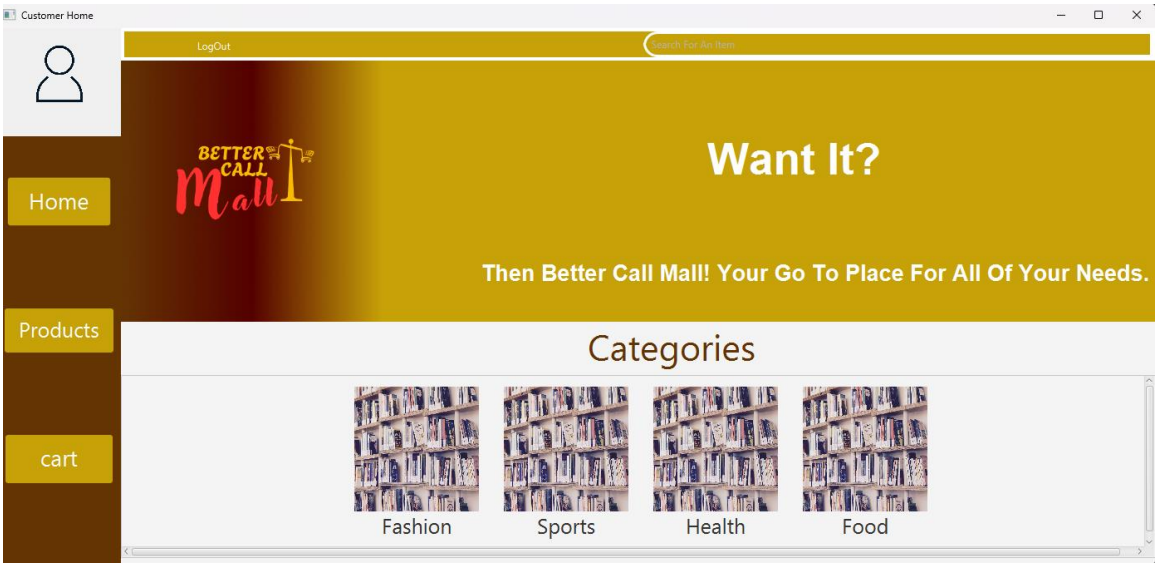
3. **Shopping Cart**
    - Shows a list of added items, their quantities, and the total price.
    - Implements a TableView to display cart details.
4. **Order Confirmation**
    - Simulates a payment process and generates an order summary.

Limitations

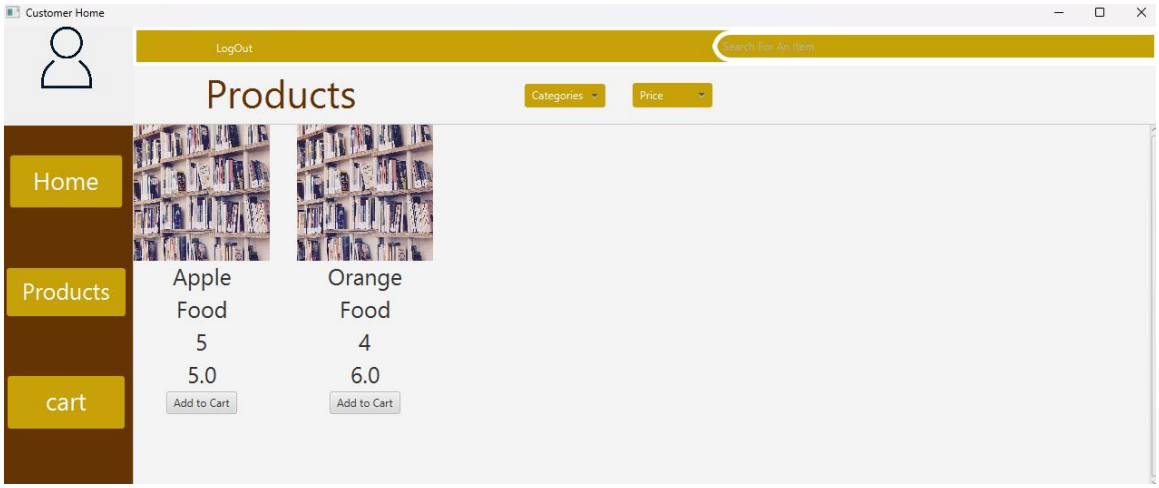
1. **Scalability**
    - The desktop application may struggle with very large datasets compared to web-based solutions.
2. **Payment Simulation**
    - Lacks real integration with payment gateways (e.g., PayPal).
3. **Multi-User Support**
    - Not optimized for concurrent access, as it is a desktop application.
4. **Platform Dependency**
    - Requires JavaFX runtime, which may not be pre-installed on all machines.

Sample Output

Customer Home Page Design



Product Page Design



Admin Database

Additional Information

Future Enhancements

- Adding a web-based version for broader access.
- Implementing machine learning algorithms for product recommendations.
- Real-time updates using WebSockets for live inventory tracking.

Future Testing Tools

- Use JUnit for unit testing backend logic.
- Perform GUI testing with TestFX.