Abstract

This project report presents the development of ESHOP, a comprehensive e-commerce software application designed and implemented as part of a database management system course. ESHOP serves as a platform for showcasing typical e-commerce functionalities, focusing on user-centric design and efficient database interactions. It provides a functional prototype that highlights the core capabilities of an e-commerce system, serving as a foundation for further expansion and deployment.

The application is built using JavaFX for a dynamic and interactive graphical user interface, MySQL for robust database management, and CSS for modern styling and layout. Key features of ESHOP include User Authentication and Registration, an Admin Panel for managing products and orders, a Dashboard for users for browsing items, and functionalities like Cart, Checkout, and Order Management.

The project demonstrates practical implementation of database management principles within a real-world e-commerce context, showcasing effective integration of front-end interface design, backend database operations, and user experience considerations. ESHOP serves as a practical illustration of contemporary software development techniques in creating scalable and interactive digital commerce platforms.

Table of Contents

Introduction

The rapid digitization of commerce has revolutionized consumer behavior, with e-commerce platforms emerging as indispensable tools for global trade. At the heart of these systems lies the efficient management of data, ranging from user credentials and product inventories to transaction records, underscoring the critical role of Database Management Systems (DBMS) in modern software development. This project, ESHOP, is an e-commerce platform designed to demonstrate the practical application of DBMS principles while addressing real-world requirements of online shopping systems.

This report documents the database designs, implementation of features and designs, tools and technologies utilized, lessons learned facing the challenges and potential future scopes. Beyond fulfilling academic objectives, this project serves as a practical model, offering valuable hands-on experience in developing a real-world application.

The objectives of this project are as follows:

- To design and implement a user-friendly interface for customers and administrators.
- To create a secure and efficient backend system using MySQL, ensuring robust data storage and retrieval.
- To provide essential features typical of e-commerce platforms, such as user authentication, product browsing, cart management, order processing, and administrative controls.
- To ensure scalability and modularity in the system, facilitating future enhancements.

With the growing reliance on online shopping, understanding the underlying architecture and functionality of such systems is crucial. ESHOP is designed as a functional prototype to explore and illustrate the integration of frontend technologies, backend processing, and database management in a cohesive application.

Tools Used

JavaFX

JavaFX served as the primary framework for developing the graphical user interface (GUI) of the application. It has extensive library and customizable rich components while providing Cross-platform compatibility.

Key roles of JavaFX include:

- Creating visually appealing layouts for the dashboard, product cards, and admin panel and other dynamic and interactive UI components.
- Managing dynamic user interactions, such as scene switching, viewing and managing products.
- Integration with MySQL via JDBC (Java Database Connectivity) to fetch and display real-time data (e.g., product availability, prices, user balances).

CSS

CSS (Cascading Style Sheets) were used to style and enhance the visual elements of the JavaFX-based GUI.

Key roles include:

- Customizing the appearance of buttons, labels, and other components for a pleasing design.
- Implementing responsive layouts and hover effects to improve usability.
- Applying themes to maintain a professional and modern look throughout the software.

MySQL

MySQL was used as the database management system (DBMS) for storing and retrieving application data. It played a critical role in managing the backend operations of the system.

Key roles of MySQL include:

- Storing data of users, products, cart, orders, and other essential entities.
- Ensuring data integrity and security through robust database integrity constraints.
- SQL queries for efficient data retrieval, manipulation, and reporting.

JDBC

JDBC (Java Database Connectivity) was used to bridge the **JavaFX frontend** and **MySQL backend** for real-time data exchange.

Key roles of JDBC include:

- Provides a standard interface for connecting to databases.
- Facilitates SQL queries likes SELECT, INSERT, UPDATE.
- Manages database connections efficiently and prevent resource leaks.

FXML

FXML, an XML-based markup language, was used for designing the user interface components in JavaFX.

Key roles of FXML include:

- Allows visual drag-and-drop UI design via tools like Scene Builder, accelerating frontend development.
- Separation of the UI design from the application logic.
- Allowing declarative specification of the layout and styles for UI components.
- Facilitates synchronization between JavaFX controllers and UI elements.

Git & Github

Git and Github was used for version control to manage changes in the codebase effectively.

It allowed for:

- Hosting the repository remotely on GitHub, ensuring 24/7 access to the project.
- Facilitating collaboration and backup of the source code.
- Tracking the evolution of the project across different commits.

Database Design

4.1 ER Diagram

ER (Entity Relationship) Diagram is a type of flowchart that illustrates how entities are related to each other withing a database system.

The ER Diagram for ESHOP Project is:

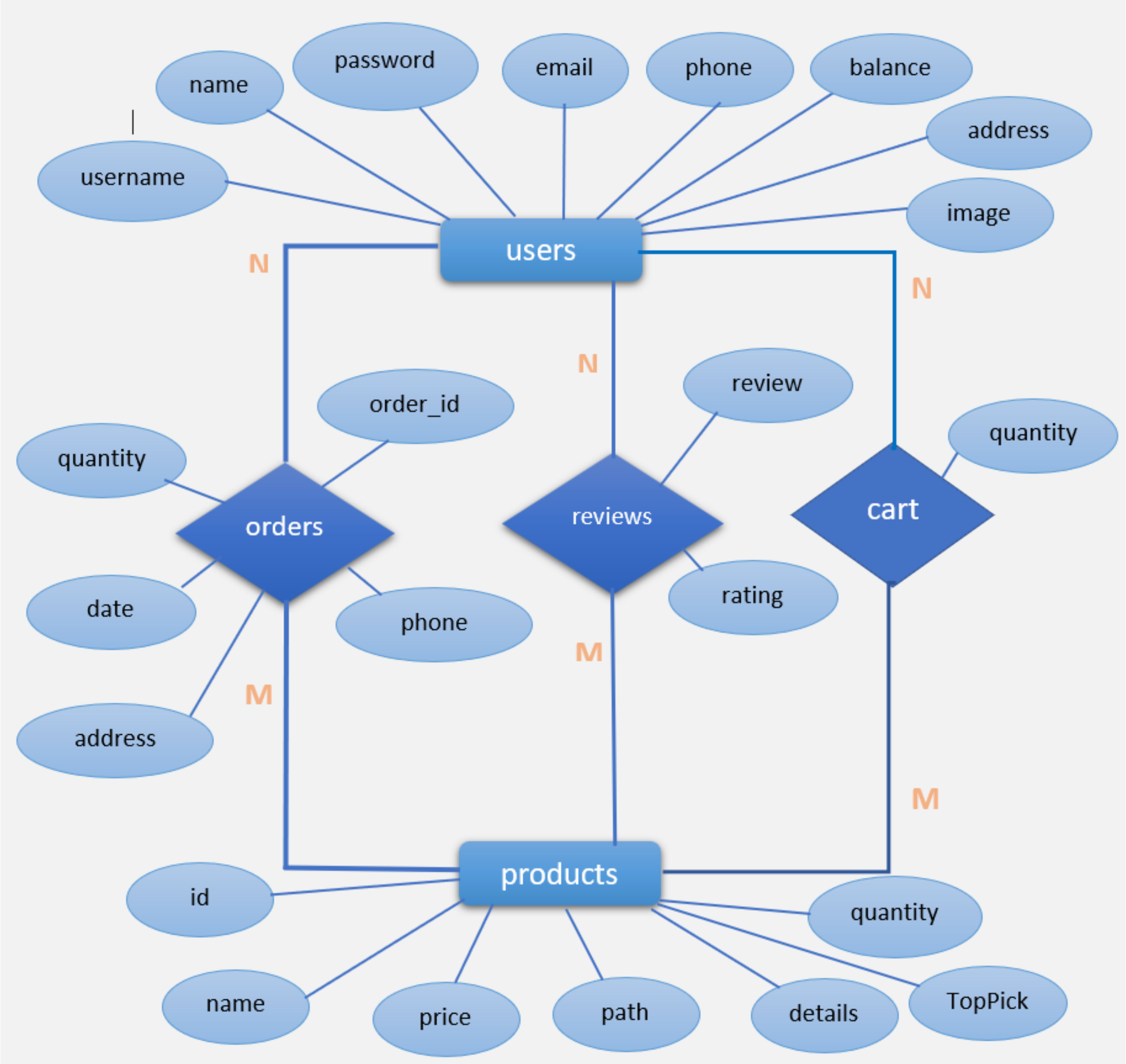
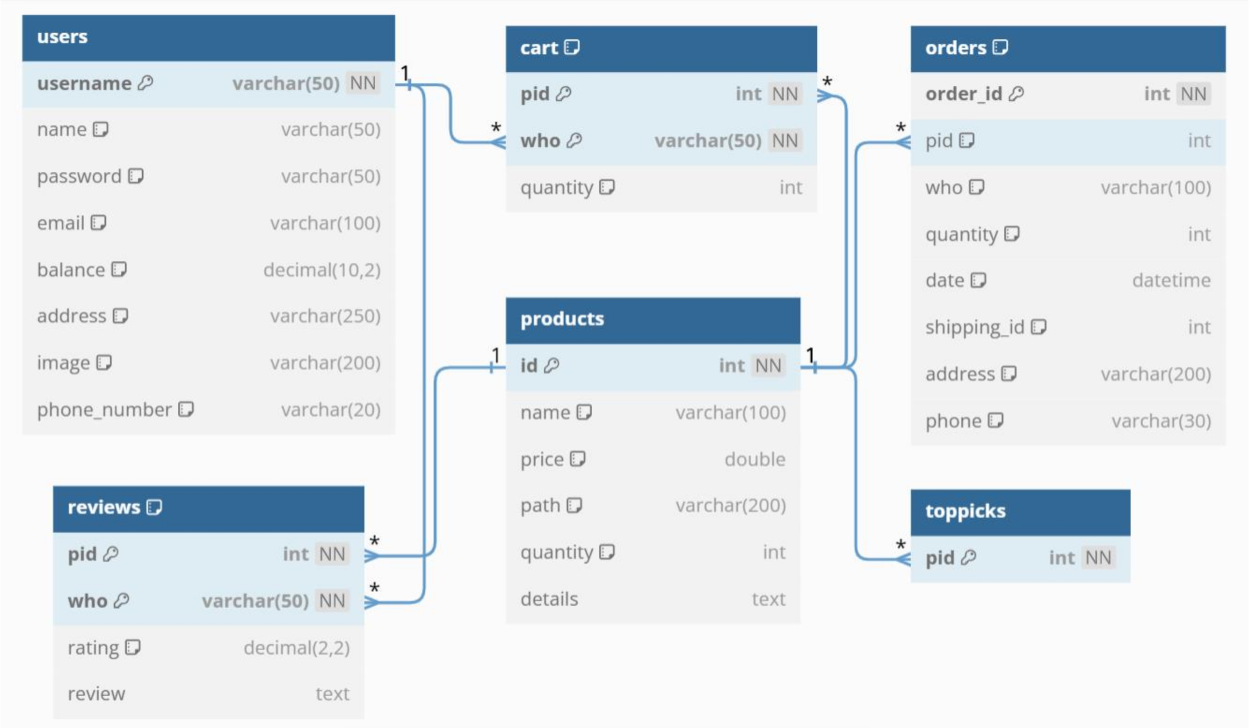


Figure 4.1: ER Diagram

4.2 Schema Diagram

The schema diagram represents the translated form of the Entity-Relationship (ER) diagram into **physical database tables**. It specifies data types, primary and foreign keys, and the relationships between the tables.

The Schema Diagram for ESHOP Project is:



users		
username	varchar(50)	NN
name	varchar(50)	
password	varchar(50)	
email	varchar(100)	
balance	decimal(10,2)	
address	varchar(250)	
image	varchar(200)	
phone_number	varchar(20)	

cart		
pid	int	NN
who	varchar(50)	NN
quantity	int	

orders		
order_id	int	NN
pid	int	
who	varchar(100)	
quantity	int	
date	datetime	
shipping_id	int	
address	varchar(200)	
phone	varchar(30)	

products		
id	int	NN
name	varchar(100)	
price	double	
path	varchar(200)	
quantity	int	
details	text	

reviews		
pid	int	NN
who	varchar(50)	NN
rating	decimal(2,2)	
review	text	

toppicks		
pid	int	NN

Figure 4.2: Schema Diagram

Implementation and Designs

5.1 Login Interface

- **Front-End:** The login form has input fields for username and password. It provides feedback for incorrect credentials.
- **Back-End:** User credentials are verified against the **users table** in the database.

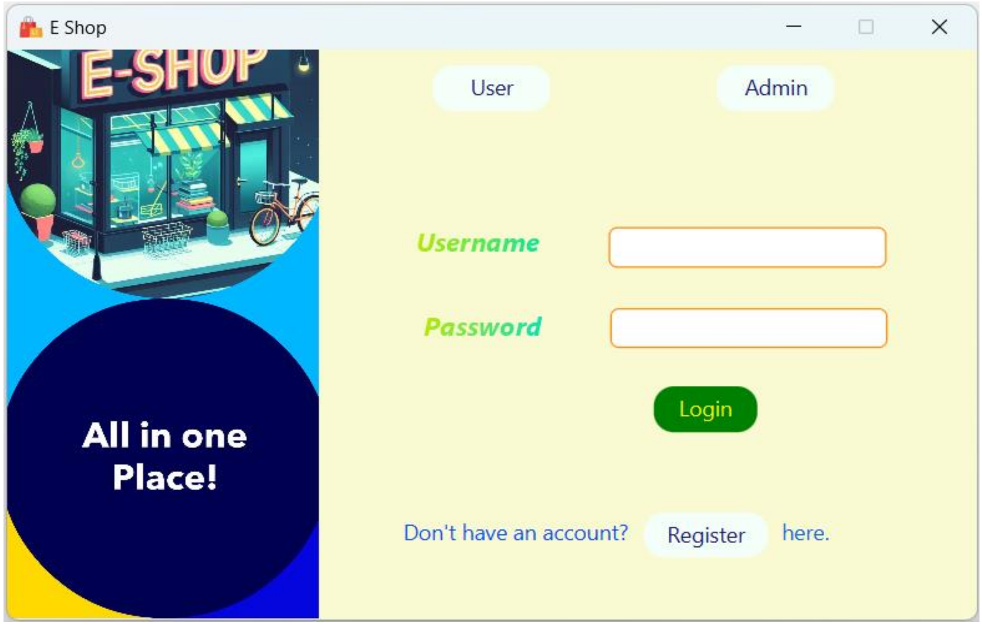


Figure 5.1: Login Interface

5.2 Registration Interface

- **Front-End:** The registration page collects user details such as username, email, and password. Success/failure alerts using JavaFX dialogs.
- **Back-End:** The registration details are inserted into the **users table**. The system checks for duplicate username to maintain primary key of the **users table**.

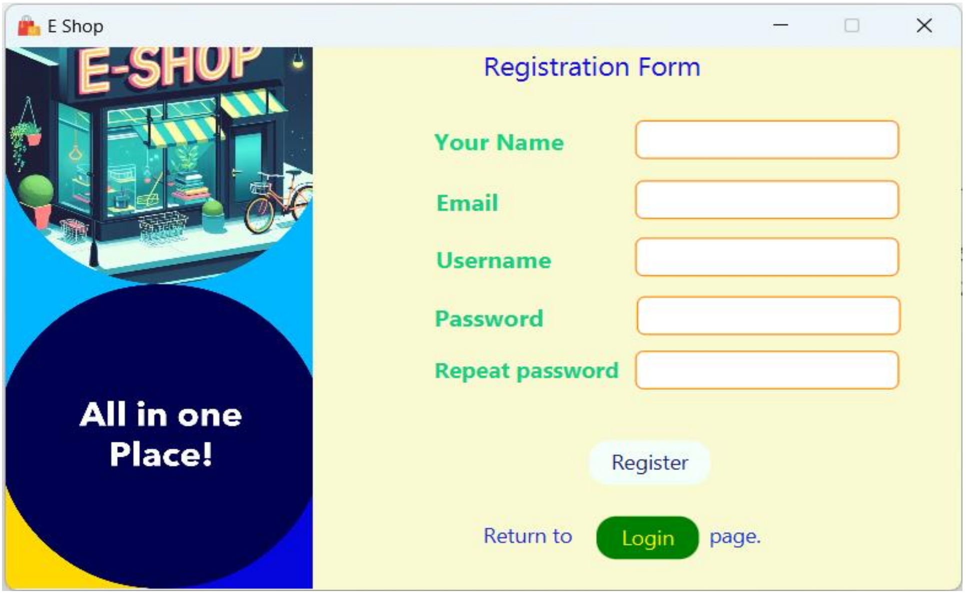


Figure 5.2: Registration Interface

5.3 Admin Login Interface

- **Front-End:** The login form has input field for admin password. It provides feedback for incorrect credentials.
- **Back-End:** The credentials are verified against the admin's password.

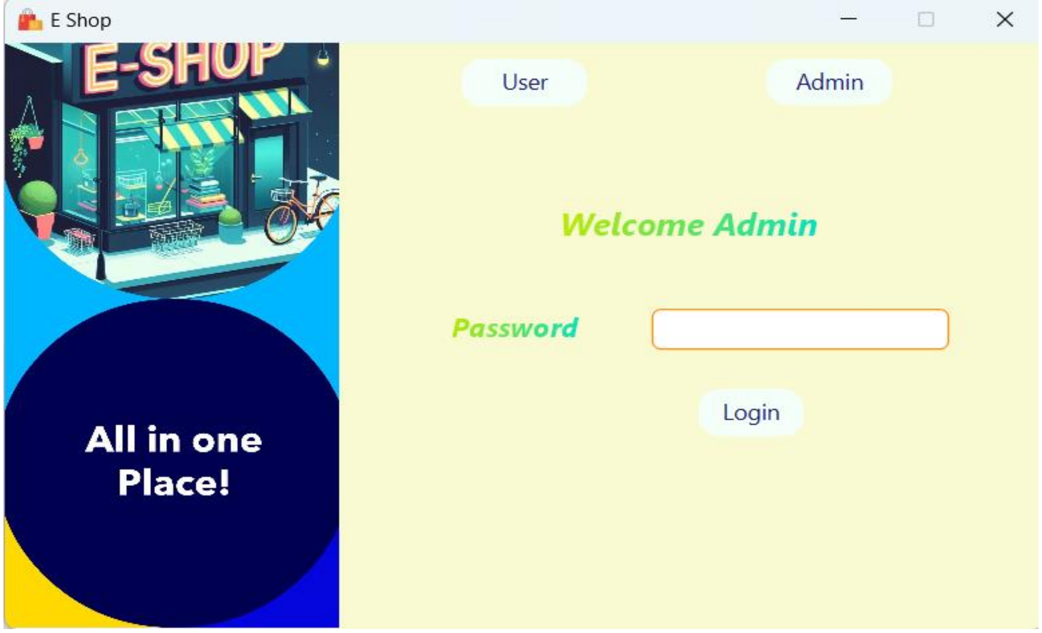


Figure 5.3: Admin Login Interface

5.4 Dashboard

- **Front-End**: The dashboard provides users with an interactive interface to browse available products built using JavaFX and FXML. Products are displayed as cards with details such as name, price, image and available quantity. CSS are used to create an appealing and responsive layout.

- **Back-End**: Data for the products is retrieved from the **products table** in the MySQL database. A query fetches product details dynamically, which are then displayed on the dashboard.

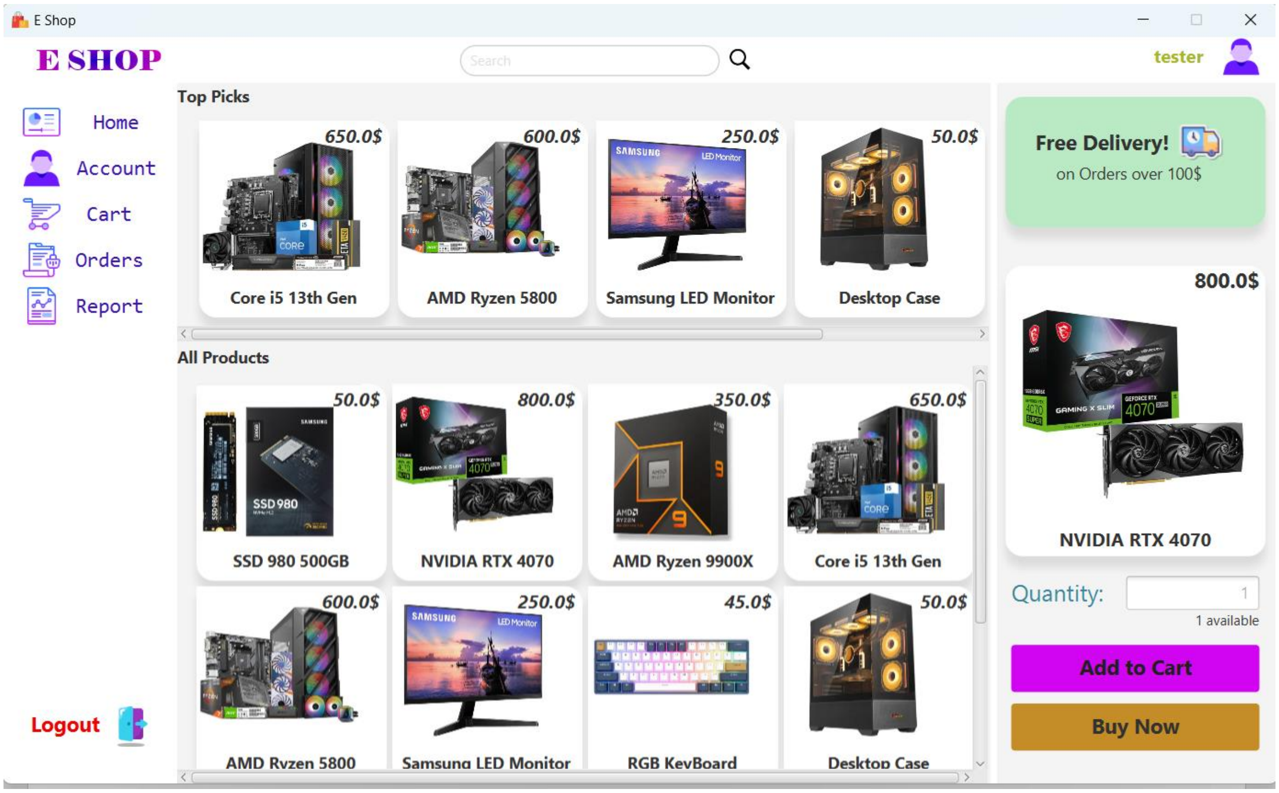
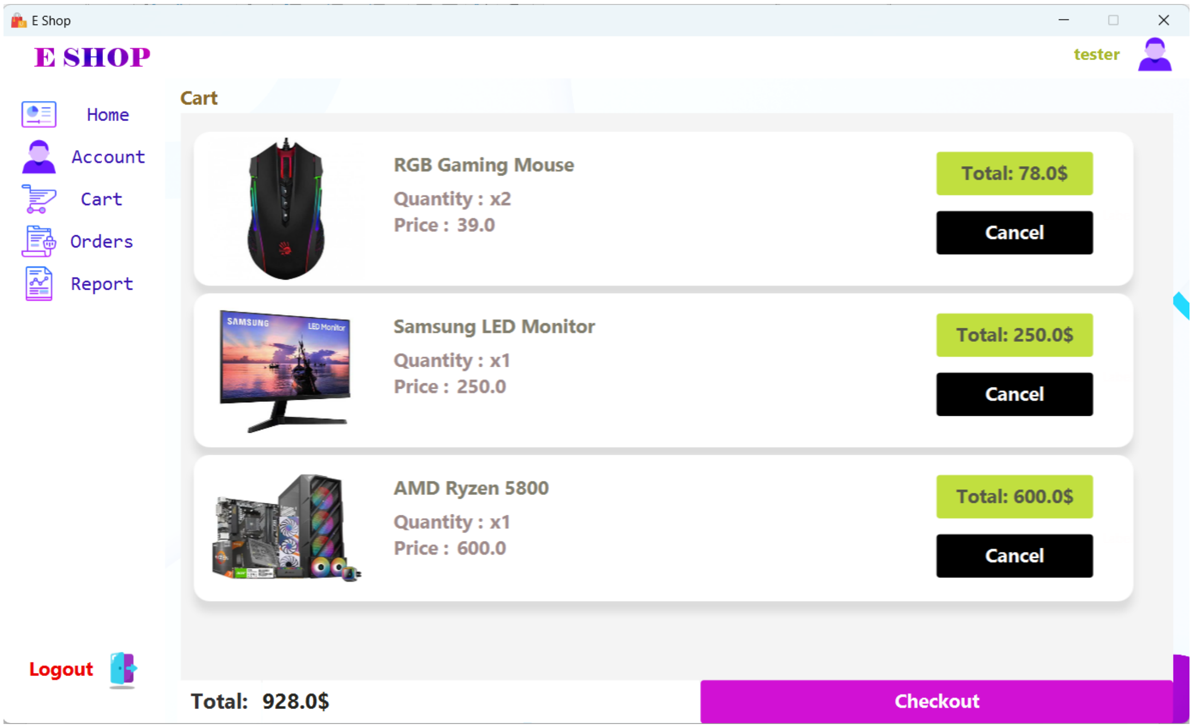


Figure 5.4: Dashboard for Customers

5.5 Cart Interface

- **Front-End**: The cart allows a user to see and remove items added to the cart by that customer and view the total price. It dynamically updates as items are added or removed. The cart interface is designed with JavaFX components for clarity and usability.

- **Back-End**: The items on cart are retrieved from **cart table**. SQL queries handle the addition, deletion, and retrieval of cart items. Foreign key relationships are used between cart and products for retrieving all the needed details.



Cart

RGB Gaming Mouse
Quantity : x2
Price : 39.0
Total: 78.0$

Samsung LED Monitor
Quantity : x1
Price : 250.0
Total: 250.0$

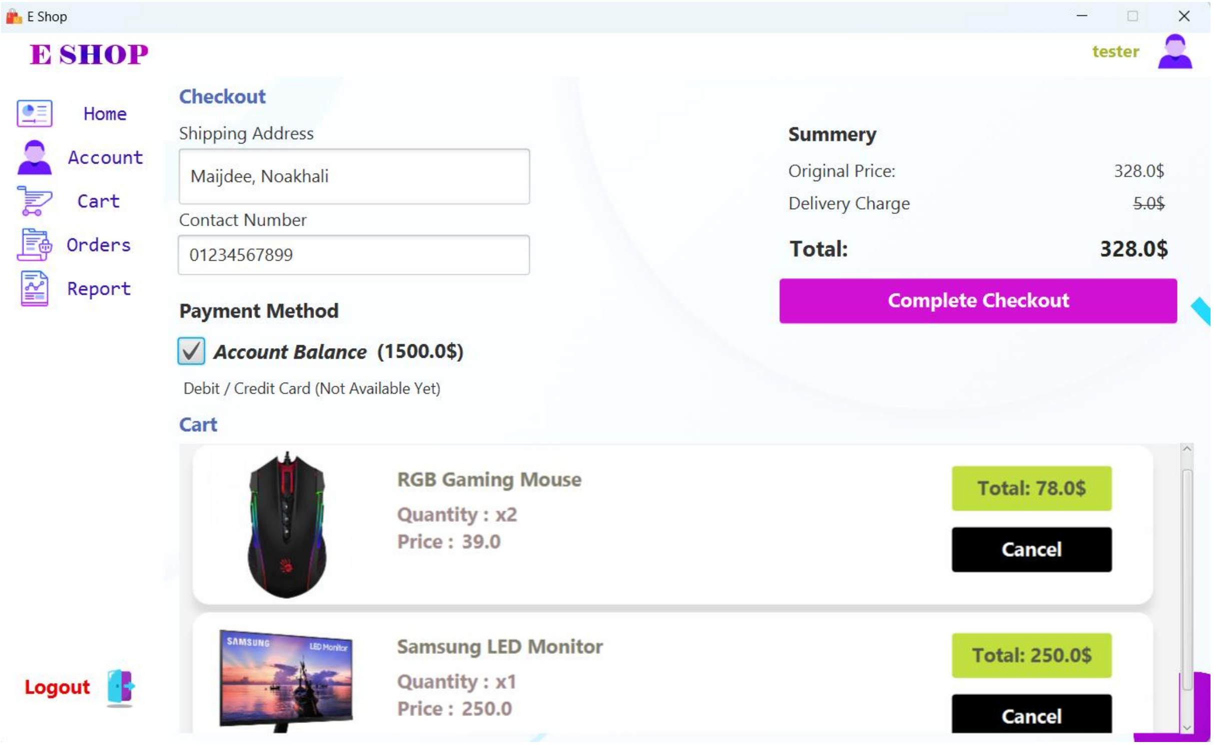
AMD Ryzen 5800
Quantity : x1
Price : 600.0
Total: 600.0$

Total: 928.0$

Figure 5.5: Cart Interface

5.6 Checkout Interface

- **Front-End**: The checkout interface captures order details such as shipping address, receiver’s number and payment method. It displays a summary of items and the total payable amount. In case of any failure, a corresponding error dialog box is displayed.
- **Back-End**: During checkout, customer’s balance is checked before proceeding further. Order details are stored in the orders table, and the inventory in the products table is updated to reflect the purchase.



Checkout

Shipping Address: Maijdee, Noakhali

Contact Number: 01234567899

Payment Method

☑ *Account Balance* (1500.0$)

Debit / Credit Card (Not Available Yet)

Summery

Original Price:	328.0$
Delivery Charge	~~5.0$~~
Total:	**328.0$**

Cart

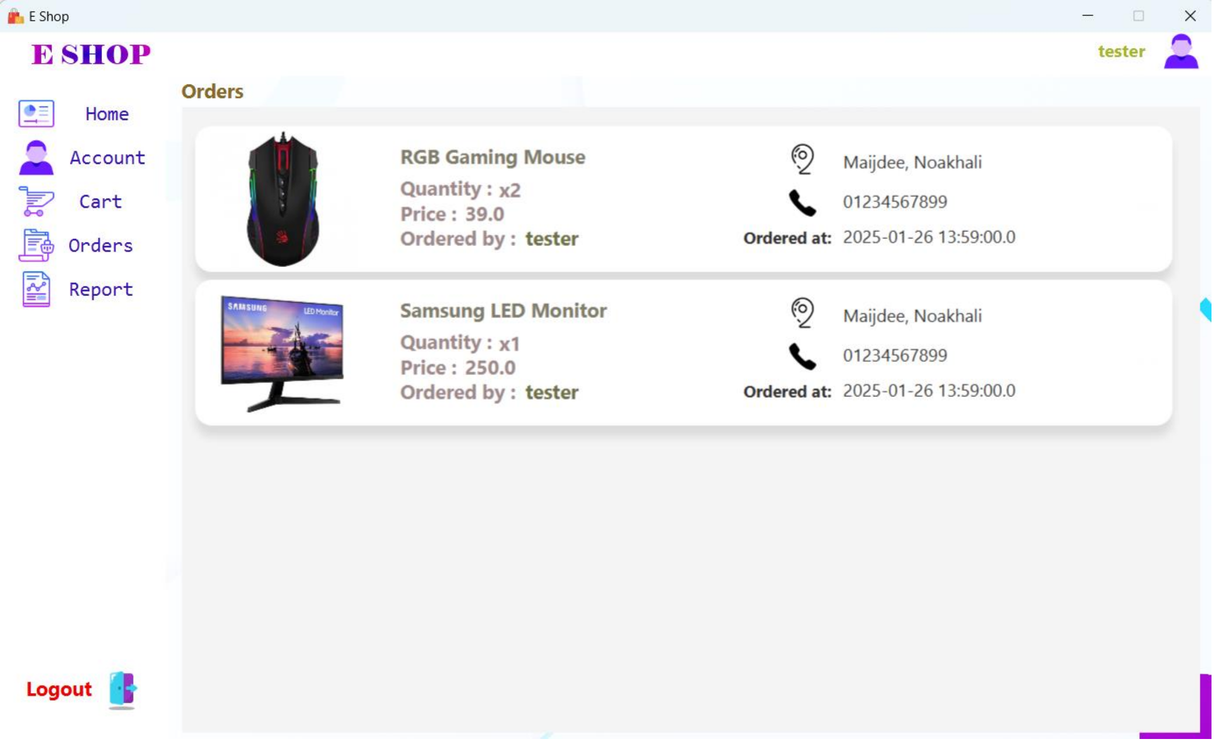
RGB Gaming Mouse
Quantity : x2
Price : 39.0
Total: 78.0$

Samsung LED Monitor
Quantity : x1
Price : 250.0
Total: 250.0$

Figure 5.6: Checkout Interface

5.7 Orders Interface

- **Front-End**: Users can view a summary of their previous orders, including order ID, product details, total amount, and order date.
- **Back-End**: Order data is stored in an orders table in the database. Queries retrieve the details of the orders of the customer. Foreign key relationships are used between orders and products for retrieving all the needed details.



RGB Gaming Mouse
Quantity : x2
Price : 39.0
Ordered by : tester
Maijdee, Noakhali
01234567899
Ordered at: 2025-01-26 13:59:00.0

Samsung LED Monitor
Quantity : x1
Price : 250.0
Ordered by : tester
Maijdee, Noakhali
01234567899
Ordered at: 2025-01-26 13:59:00.0

Figure 5.7: Orders Interface

5.8 Admin Panel

- **Front-End**: The admin panel provides functionality for managing products like adding, products, adding products to toppicks and viewing orders of the customers. It has fields for taking inputs of products such as product name, price, quantity and image.
- **Back-End**: Products are added to the products table after validating the input data. All orders are retrieved from the orders table for display to the admin. Secure authentication mechanisms ensure that only authorized admins can access this feature.

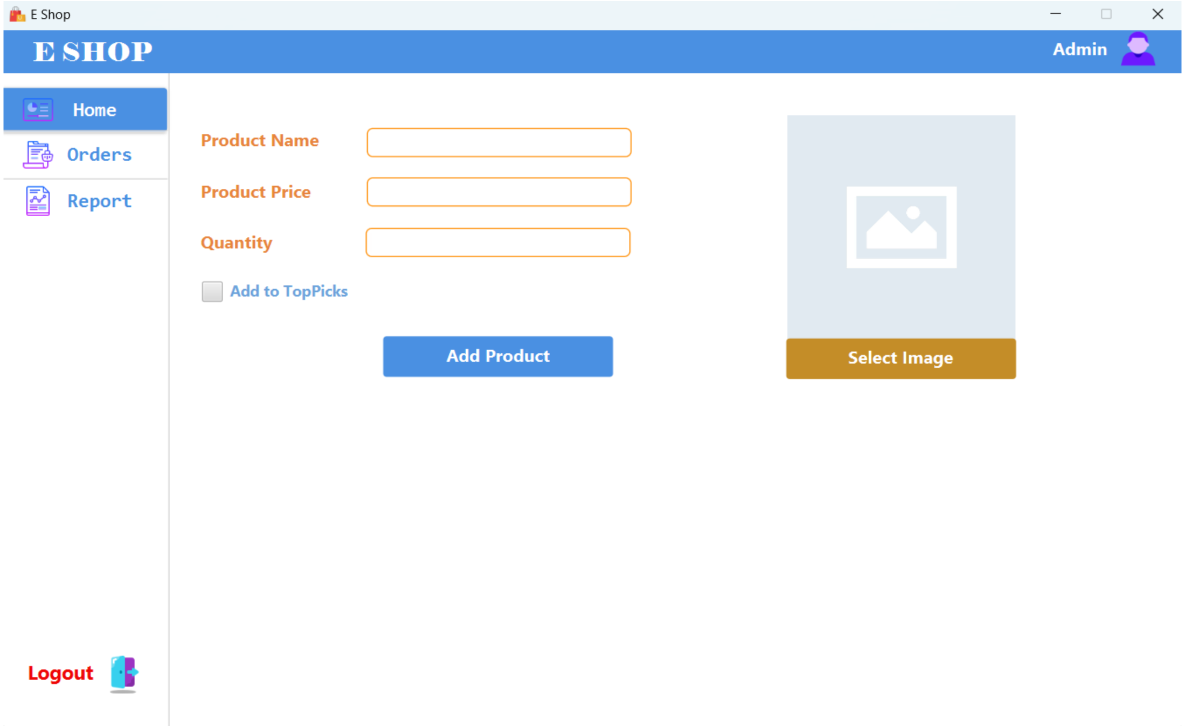
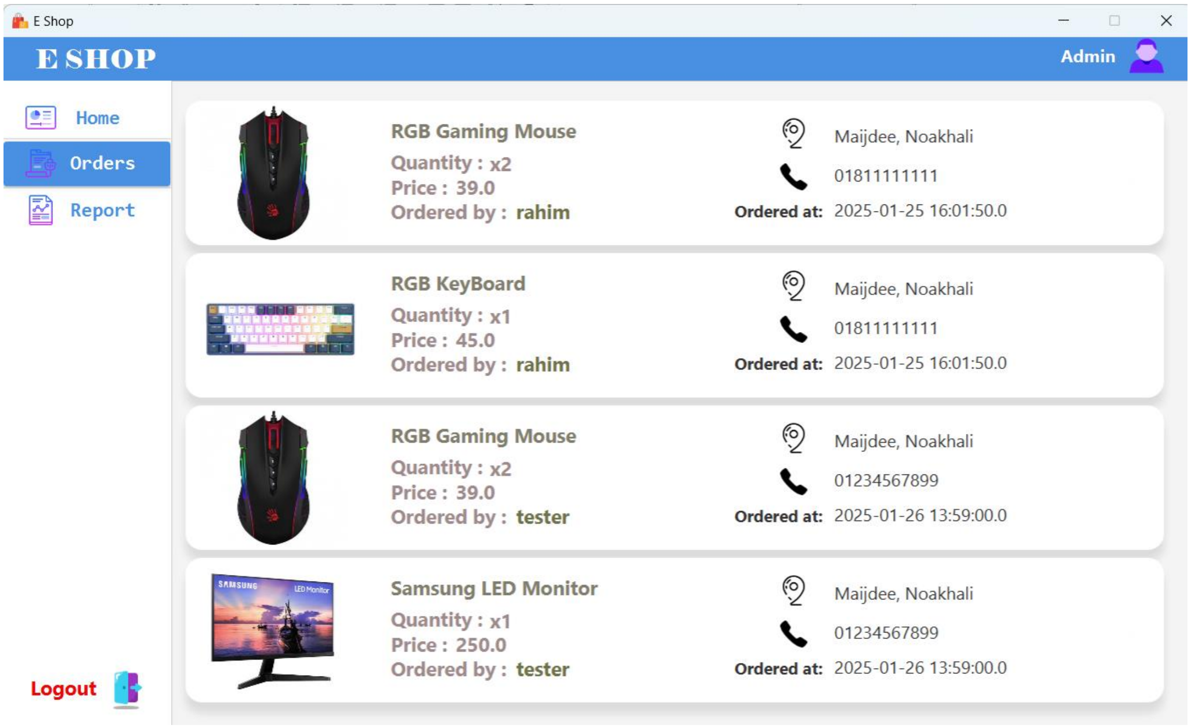


Figure 5.8.1: Admin Panel for Product Management



RGB Gaming Mouse
Quantity : x2
Price : 39.0
Ordered by : rahim
Maijdee, Noakhali
01811111111
Ordered at: 2025-01-25 16:01:50.0

RGB KeyBoard
Quantity : x1
Price : 45.0
Ordered by : rahim
Maijdee, Noakhali
01811111111
Ordered at: 2025-01-25 16:01:50.0

RGB Gaming Mouse
Quantity : x2
Price : 39.0
Ordered by : tester
Maijdee, Noakhali
01234567899
Ordered at: 2025-01-26 13:59:00.0

Samsung LED Monitor
Quantity : x1
Price : 250.0
Ordered by : tester
Maijdee, Noakhali
01234567899
Ordered at: 2025-01-26 13:59:00.0

Figure 5.8.2: All Orders Panel for Admin

Challenges Faced

1. Limited Access to Online Database Storage

- **Issue:** Inability to afford cloud-based database solutions (e.g., AWS, Azure) due to lack of credit cards and unreliable free-tier options.
- **Resolution:** Relied on local MySQL storage.

2. UI Performance and Scene Switching

- **Issue:** Slow UI loading and lag during some scene transitions .
- **Resolution:** Fixed issue for some transitions by multithreading the heavy background tasks.

3. Efficient Query Handling

- **Issue:** Writing efficient queries while ensuring data integrity was challenging, particularly for complex relationships in the database.
- **Resolution:** Used my database knowledge and problem-solving skills to handle queries effectively and maintain integrity constraints.

4. Static Layout and Window Sizing

- **Issue:** The use of fixed layouts (e.g., AnchorPane) instead of dynamic layouts (e.g., HBox, VBox) limited the window resizing capability.
- **Resolution:** Focused on fixed-size UI design due to time and knowledge constraints, with plans to improve in future projects.

5. Time Management and Learning Curve

- **Issue:** Balancing academic coursework alongside the project posed a significant challenge, especially as this was my first software development endeavor. Moreover, the project's scope was quite extensive given the limited time and resources available to us.
- **Resolution:** Focused on implementing the core-functionalities as much possible while adhering to the project timeline.

Lessons Learned

Throughout the development of the ESHOP project, we gained valuable insights and practical experience in software development. Some of the key lessons learned include:

1. **Software Development Process**
    - Acquired a comprehensive understanding of the developing a functional software application, from planning and designing to implementation and testing.

2. **Database Integration**
    - Learned how to design and integrate databases effectively in an application while ensuring data integrity and seamless interaction between the application and the database.

3. **User Interface Design**
    - Developed skills in designing user interfaces using JavaFX and FXML and in enhancing their appearance with CSS for better user experience.

4. **Multithreading**
    - Gained hands-on experience in implementing multithreading to optimize application performance, particularly for handling background tasks and ensuring smooth user interactions.

5. **Efficient and Manageable Code Practices**
    - Improved my ability to write clean, efficient, and maintainable code by following best practices and adhering to modular design principles.

6. **Database Design and Query Optimization**
    - Learned to design normalized database schemas, define relationships between tables, and write efficient queries for retrieving and manipulating data.

7. **Problem-Solving and Debugging**
    - Enhanced my problem-solving skills by debugging issues, resolving bugs, and optimizing the application's functionality under various constraints.

8. **Git and GitHub**
    - Acquired essential skills in using Git and GitHub for collaborative development and project backup, including writing clear commit messages, managing branches, and tracking project progress.

These lessons have not only helped us implement the project but have also laid a strong foundation for future endeavors in software development and database management.

Future Scopes

The ESHOP project has immense potential for further improvement and expansion. Below are some future enhancements that can elevate the functionality, usability, and performance of the software:

1. **Dynamic Layout Implementation**
   - Transitioning to dynamic layouts such as HBox, VBox to enable responsive design, allowing the application to adapt seamlessly to different screen sizes and resolutions.
2. **Recommendation System**
   - Integrating a recommendation system to suggest products to users based on their browsing history, purchase patterns, and popular trends.
3. **Statistical Analysis and Reporting**
   - Adding features to analyze user behavior and purchase data, such as identifying the most frequently bought products or analyzing sales trends .
4. **Enhanced Optimization**
   - Further refining performance through optimization techniques to ensure faster UI loading, seamless scene transitions, and efficient handling of background tasks.
5. **Improved User Interface (UI)**
   - Redesigning the UI to make it more visually appealing, modern, and user-friendly. This could include better use of color schemes, animations, and interactive elements.
6. **Integration with Payment Gateways**
   - Adding secure payment gateway options for a fully functional e-commerce experience, supporting a variety of payment methods like credit cards, mobile wallets, and online banking.
7. **Search and Filters**
   - Functionality with search and filters for price range, categories, etc. can provide a more tailored shopping experience for users.
8. **Enhanced Admin Features**
   - Enhanced dashboards for admins to manage inventory, monitor sales performance, and manage customers issues in real time.
9. **Addition of Product Details and Reviews**
   - Providing users with comprehensive product details, including specifications, descriptions, along with the option to leave reviews and ratings to assist other users in making informed purchasing decisions.

Conclusion

The ESHOP project means a significant milestone in the journey of software development and database management for us. Designed as an e-commerce application, it successfully incorporates essential features such as user authentication, product browsing, cart management, order processing, and an admin panel for efficient management. The project demonstrates the integration of modern tools and technologies like JavaFX, MySQL, and CSS, along with fundamental principles of database design, multithreading, and UI development.

Through the challenges faced, including resource limitations, optimization issues, and the steep learning curve of first-time development, this project provided valuable hands-on experience and fostered critical problem-solving skills. The lessons learned, from designing databases to writing efficient and maintainable code, have laid a strong foundation for future endeavors.

While the project meets its core objectives, it also offers room for significant enhancements, including dynamic layouts, advanced analytics, improved UI, and expanded functionality such as recommendation systems and product reviews. These future scopes outline the potential to transform ESHOP into a fully-featured, scalable e-commerce solution.

In conclusion, the ESHOP project not only fulfilled its academic objectives but also served as a practical introduction to real-world software development, equipping us with skills and insights that will be invaluable in future projects and professional pursuits.