Table of Contants

1. Introduction 2

2. Objective of the System 2

3. System Design and Implementation 2

3.1) Proposed Solution 2

3.2) List of Tables with Purpose 3

3.3) ER Diagram:..... 4

3.4) Physical Schema (Table Structure with Attributes) 5

4. Result and Discussion 7

4.1) Current State: The UI is currently static (HTML/CSS only) without database connectivity. 7

4.2) Integration Requirements: To connect the UI with the database, the following components are needed: 7

5. Application and Future Scope 8

5.1) Current Applications 8

5.2) UI-Database Integration Plan 8

5.3) Future Enhancements 8

6. Conclusion 8

7. Reference 9

7. Screen short of UI 9

7.1) Home page & Registation: 9

7.2) Log In & About: 10

7.3) Developer's & View Product 11

7.8) User Dashboard & Admin Dashboard 12

7.5) Shop address: 13

E-Commerce Database Management System with UI Integration Proposal

Abstract

This project presents the design and implementation of a comprehensive relational database management system for an e-commerce platform, along with a web-based user interface. The database system efficiently manages user accounts, product catalog, inventory, orders, payments, and customer reviews. The front-end interface, built with HTML and CSS, provides visual components for user interaction. While currently disconnected, this report outlines the integration pathway between the UI and Oracle database backend to create a fully functional e-commerce application.

Acknowledgement

This project report, entitled E-Commerce Database Management System with UI Integration Proposal, is the culmination of my work and learning during my second year. I wish to extend my genuine appreciation to those who contributed to its completion.

I am profoundly grateful to my project supervisor, ***Prof. Tasfia Tabassum Faija***, Department of ***Computer Science and Engineering*****,** for her exemplary guidance, motivation, and immense knowledge. Her continuous support and patience helped me immensely at every stage of this project.

I am also obliged to the faculty and staff of ***Northern University Bangladesh*** for their direct and indirect support, which created an excellent learning atmosphere.

My heartfelt thanks to my project group members, ***Farzana Afroz***, ***Mohtasim Billah Tanvir,*** and ***Asif Ahmed*** for their incredible teamwork, dedication, and hard work. It was a pleasure collaborating with them.

A special note of thanks to my parents and my brother/sister for their faith in me and for their endless love and support, which kept me going.Lastly, I thank Almighty **Allah** for giving me the strength and perseverance to complete this project.***Md Mahabub Hasan (41230301765)***

1. Introduction

The rapid growth of e-commerce has created a critical need for robust database systems that can handle complex transactions, user data, inventory management, and customer interactions. Traditional file-based systems or poorly designed databases often lead to data redundancy, inconsistency, and inefficient query processing. Without a properly structured database, e-commerce platforms face challenges in maintaining accurate inventory records, processing orders efficiently, managing user accounts securely, and providing personalized shopping experiences.

The motivation for this project stems from the necessity to create a scalable, efficient, and maintainable database backend for e-commerce operations that ensures data integrity, supports business intelligence through structured data storage, and enables seamless user experiences across various shopping activities.

2. Objective of the System

- To design a normalized database schema for e-commerce operations
- To implement tables, relationships, and constraints for data integrity
- To automate critical business processes through triggers and sequences
- To support essential e-commerce functionalities including:
    - User registration and role management
    - Product catalog organization with categories
    - Inventory management with stock tracking
    - Shopping cart functionality
    - Order processing and status tracking
    - Discount and promotion management
    - Product reviews and ratings system
    - Payment method management

3. System Design and Implementation

3.1) Proposed Solution

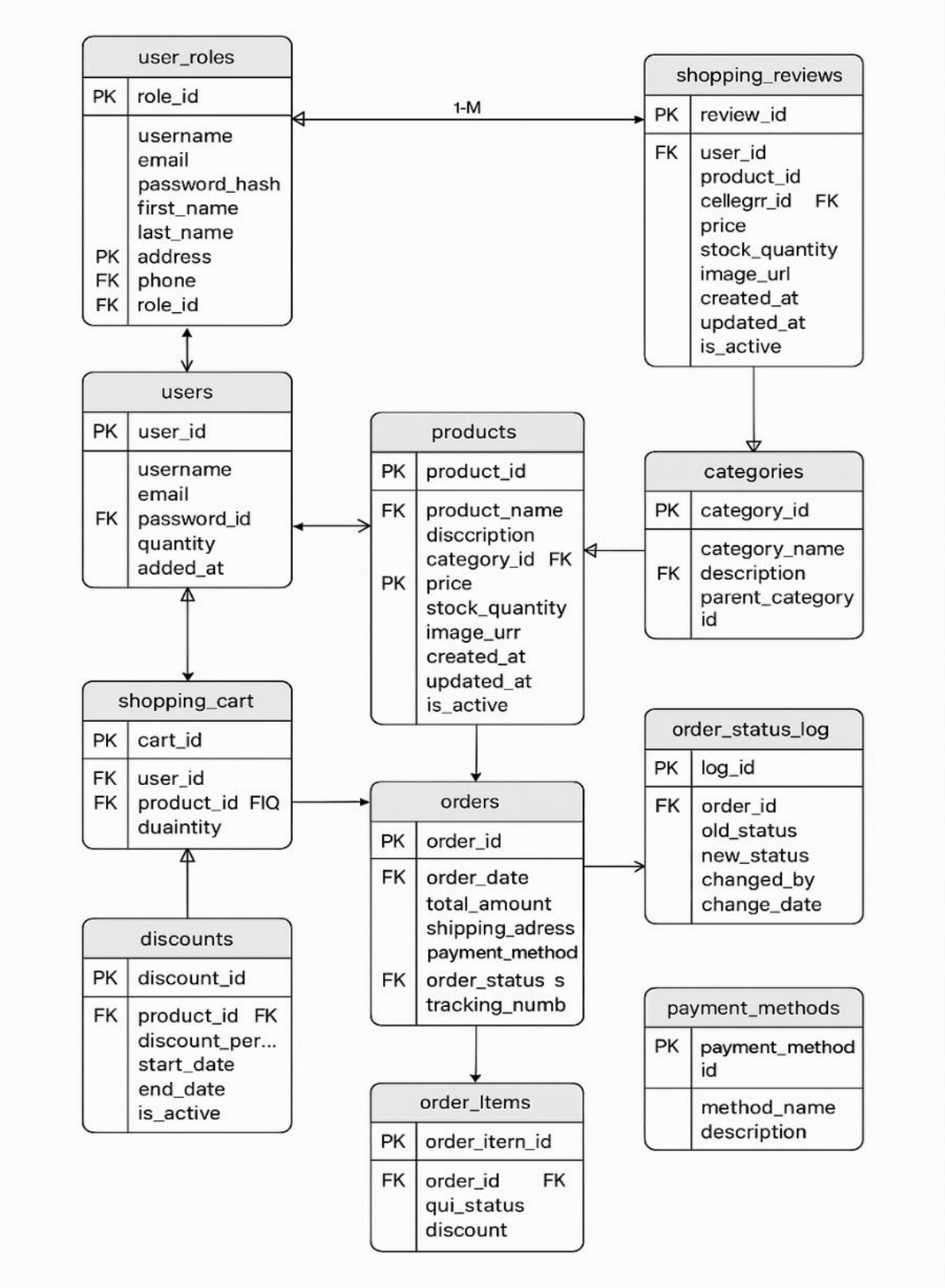
The solution implements a relational database model with carefully designed tables, relationships, and constraints. The system uses:

- Primary and foreign keys for relationship enforcement
- Check constraints for data validation
- Sequences for automated ID generation
- Triggers for automated stock management and rating calculations
- Normalized structure to minimize redundancy

3.2) List of Tables with Purpose

Table Name	Purpose
user_roles	Stores different user roles (Admin, User)
users	Contains user account information
categories	Organizes products into hierarchical categories
products	Stores product details and inventory information
discounts	Manages product discounts and promotions
payment_methods	Defines available payment options
shopping_cart	Temporarily stores items users intend to purchase
orders	Records customer orders and status
order_items	Contains individual products within each order
product_reviews	Stores customer ratings and reviews
order_status_log	Tracks order status changes for auditing

3.3) ER Diagram:



3.4) Physical Schema (Table Structure with Attributes)

1. user_roles

- role_id (PK, NUMBER)
- role_name (VARCHAR2(50))
- description (VARCHAR2(200))

2. users

- user_id (PK, NUMBER)
- username (VARCHAR2(50))
- email (VARCHAR2(100))
- password_hash (VARCHAR2(200))
- first_name (VARCHAR2(50))
- last_name (VARCHAR2(50))
- address (VARCHAR2(200))
- phone (VARCHAR2(20))
- role_id (FK, NUMBER)
- created_at (DATE)
- last_login (DATE)
- is_active (NUMBER(1))

3. categories

- category_id (PK, NUMBER)
- category_name (VARCHAR2(100))
- description (VARCHAR2(200))
- parent_category_id (FK, NUMBER)

4. products

- product_id (PK, NUMBER)
- product_name (VARCHAR2(100))
- description (VARCHAR2(500))
- category_id (FK, NUMBER)
- price (NUMBER(10,2))
- stock_quantity (NUMBER)
- image_url (VARCHAR2(200)
- created_at (DATE)
- updated_at (DATE)

- is_active (NUMBER(1))
- rating (NUMBER(3,2))

5, discounts

- discount_id (PK, NUMBER)
- product_id (FK, NUMBER)
- discount_percent (NUMBER(5,2))
- start_date (DATE)
- end_date (DATE)
- is_active (NUMBER(1))

6. payment_methods

- payment_method_id (PK, NUMBER)
- method_name (VARCHAR2(50))
- description (VARCHAR2(200))

7. shopping_cart

cart_id (PK, NUMBER)
user_id (FK, NUMBER)
product_id (FK, NUMBER)
quantity (NUMBER)
added_at (DATE)

8. orders

order_id (PK, NUMBER)
user_id (FK, NUMBER)
order_date (DATE)
total_amount (NUMBER(10,2))
shipping_address (VARCHAR2(200))
payment_method_id (FK, NUMBER)
order_status (VARCHAR2(20))
tracking_number (VARCHAR2(50))

9. order_items

order_item_id (PK, NUMBER)
order_id (FK, NUMBER)
product_id (FK, NUMBER)
quantity (NUMBER)

unit_price (NUMBER(10,2))
discount_amount (NUMBER(10,2))

10. product_reviews

review_id (PK, NUMBER)
product_id (FK, NUMBER)
user_id (FK, NUMBER)
order_id (FK, NUMBER)
rating (NUMBER(1))
review_text (VARCHAR2(1000))
review_date (DATE)

11, order_status_log

log_id (PK, NUMBER)
order_id (FK, NUMBER)
old_status (VARCHAR2(20))
new_status (VARCHAR2(20))
changed_by (VARCHAR2(50))
change_date (DATE)

4. Result and Discussion

The implemented database system successfully addresses all core requirements of an e-commerce platform. The normalization process eliminated data redundancy while maintaining referential integrity through well-defined relationships.The UI components have been designed with modern e-commerce principles:

- Responsive design for various device sizes
- Intuitive navigation and product browsing
- Clean shopping cart interface
- Streamlined checkout process
- User account management pages

4.1) Current State:

The UI is currently static (HTML/CSS only) without database connectivity.

4.2) Integration Requirements:

To connect the UI with the database, the following components are needed:

- Server-side technology (Node.js, PHP, Java, or Python)
- REST API or server-side scripts to handle database operations

- Connection middleware (like Oracle Instant Client)
- Authentication and session management

5. Application and Future Scope

5.1) Current Applications

- Online retail stores
- Inventory management systems
- Order processing systems
- Customer relationship management
- Product review and rating systems

5.2) UI-Database Integration Plan

- **Phase 1**: Connect product catalog and shopping features
- **Phase 2**: Implement user authentication and profiles
- **Phase 3**: Develop order processing and payment integration
- **Phase 4**: Add admin dashboard for inventory and order management

5.3) Future Enhancements

- **Real-time Inventory Updates**: WebSocket integration for stock level updates
- **Payment Gateway Integration**: Secure connection to payment processors
- **Image Upload**: Product image management system
- **Search Functionality**: Advanced product search with filters
- **Order Tracking**: Real-time shipment tracking integration
- **Email Notifications**: Order confirmation and status updates
- **Recommendation Engine**: Personalized product suggestions
- **Mobile App**: React Native or Flutter application

6. Conclusion

The e-commerce database management system designed and implemented in this project provides a robust foundation for online retail operations. The front-end interface offers a user-friendly experience that, when connected to the database, will create a complete e-commerce solution. The integration pathway outlined demonstrates how the static UI can be transformed

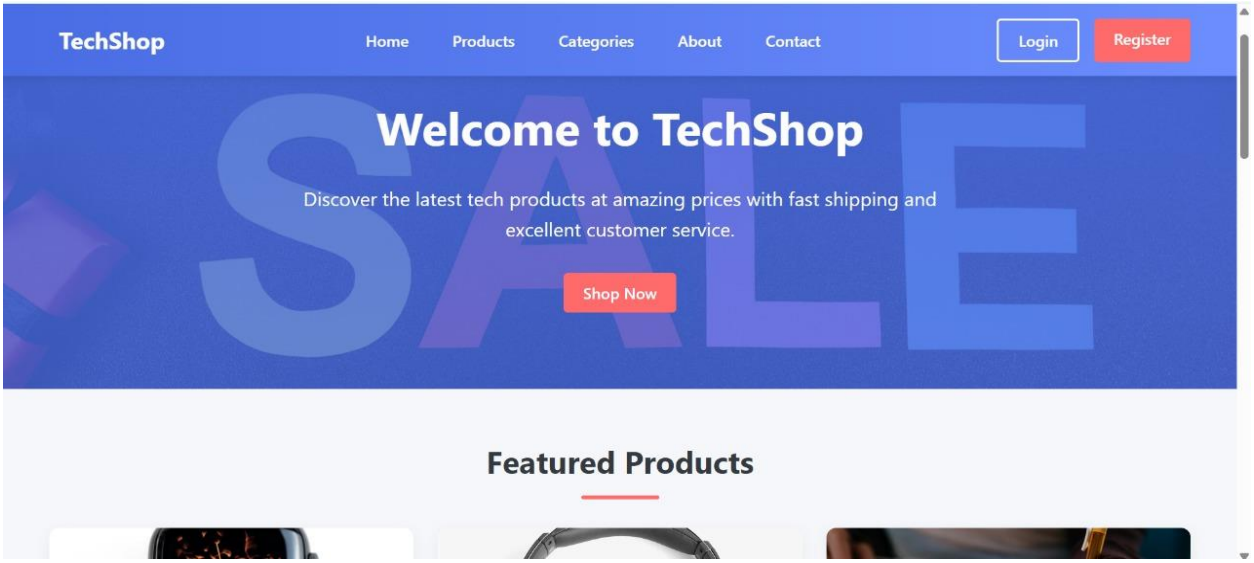
into a dynamic web application through server-side scripting and API development. This comprehensive system provides all necessary components for a modern e-commerce platform that can scale with business growth.

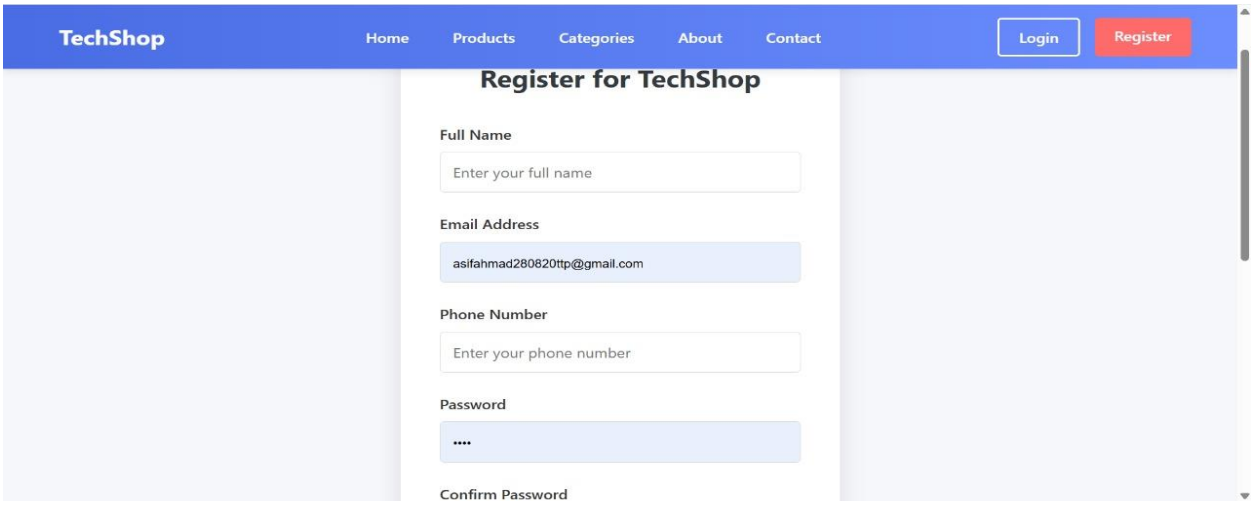
7. Reference

- Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson Education.
- Oracle Database Documentation. (2023). Oracle Help Center.
- MDN Web Docs. (2023). JavaScript and Web APIs.

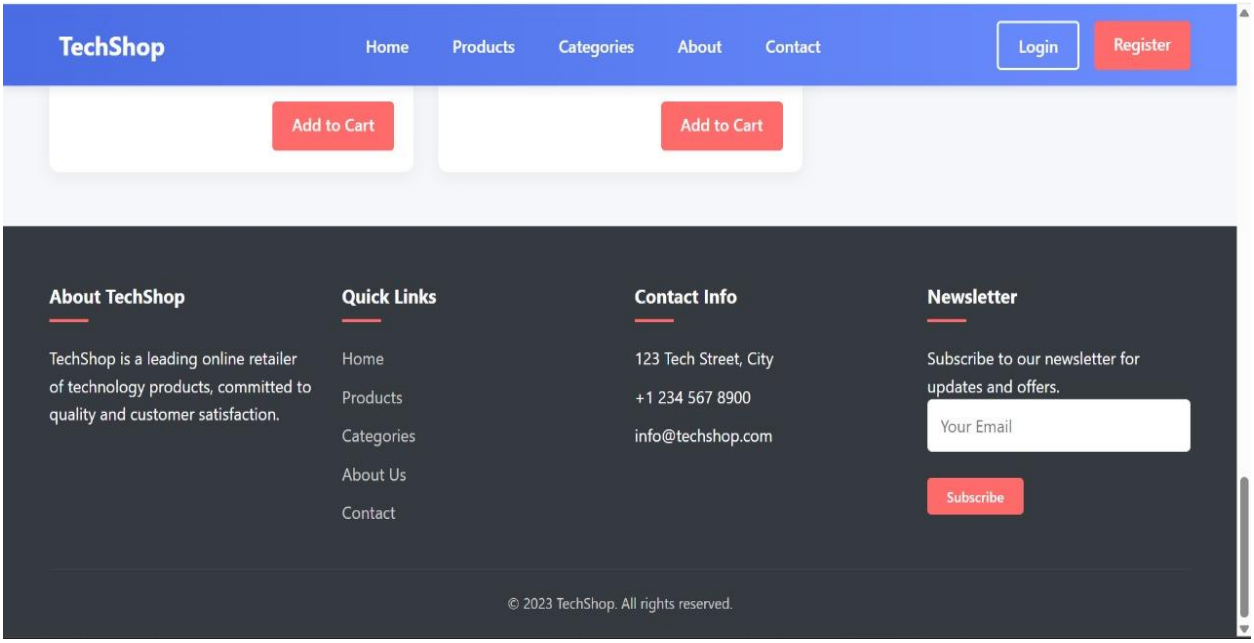
7. Screen short of UI

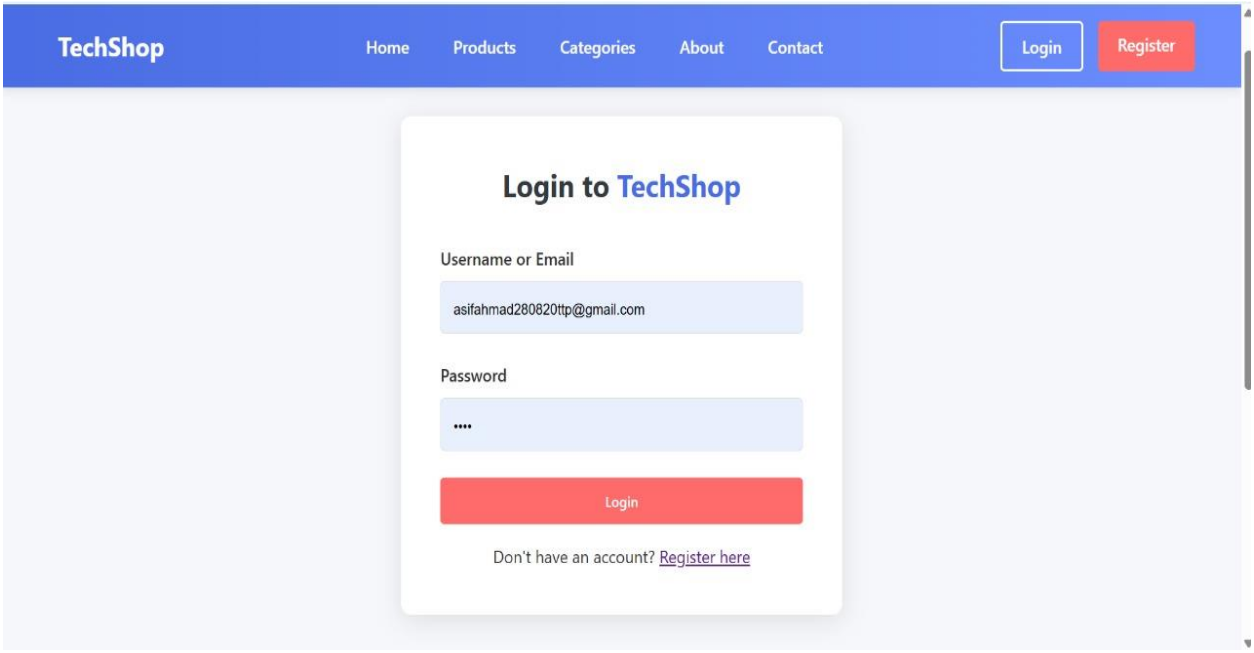
7.1) Home page & Registation:



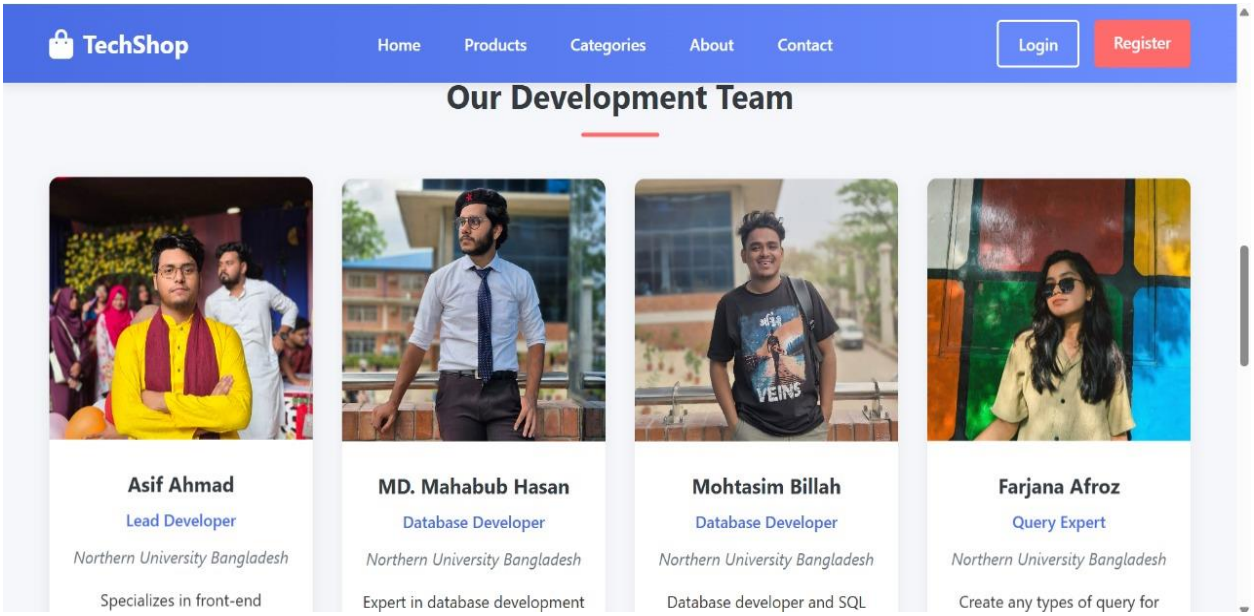


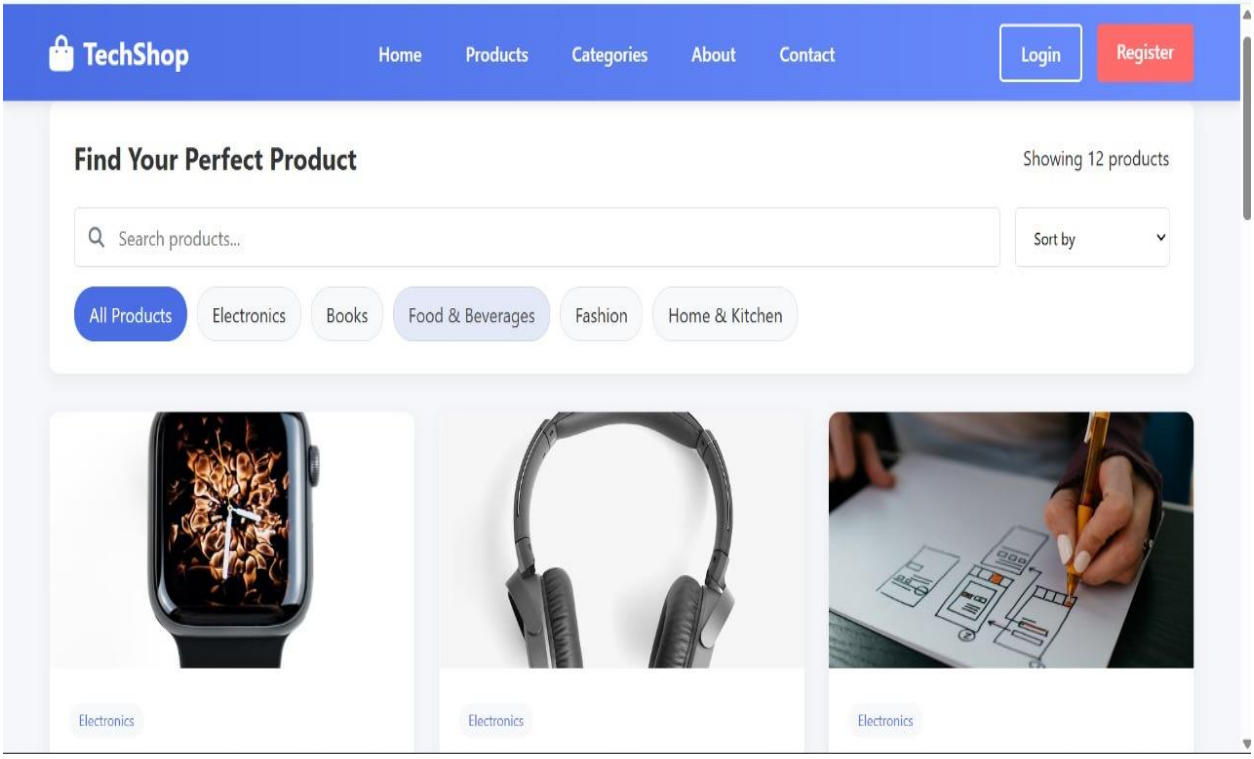
7.2) Log In & About:



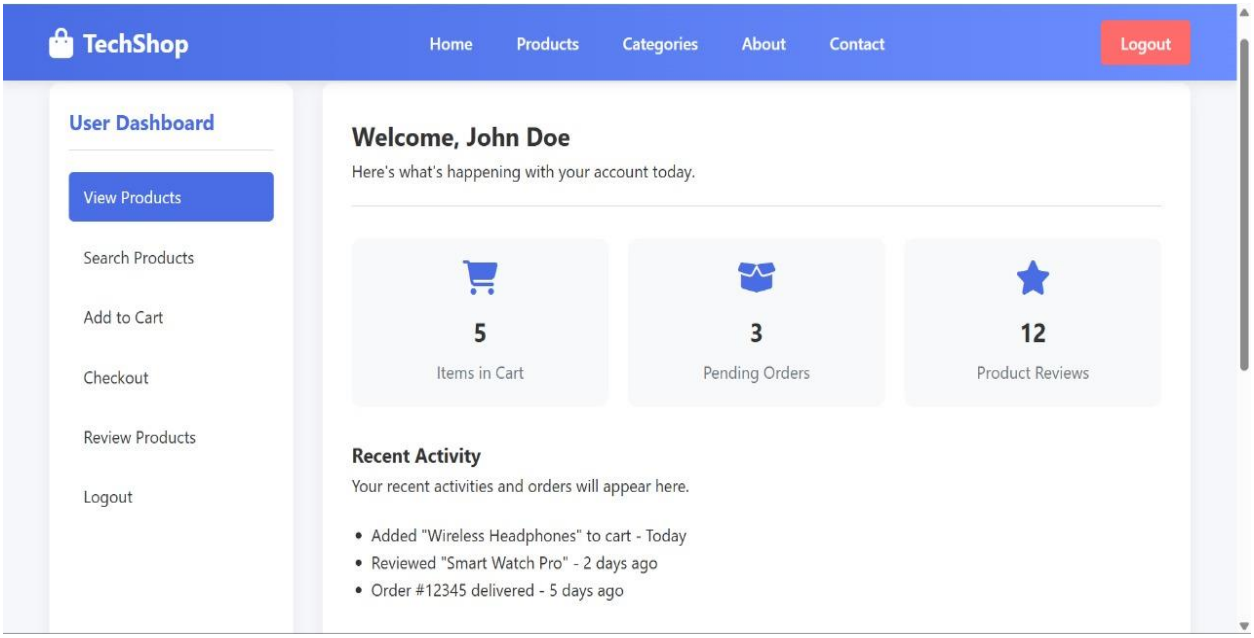


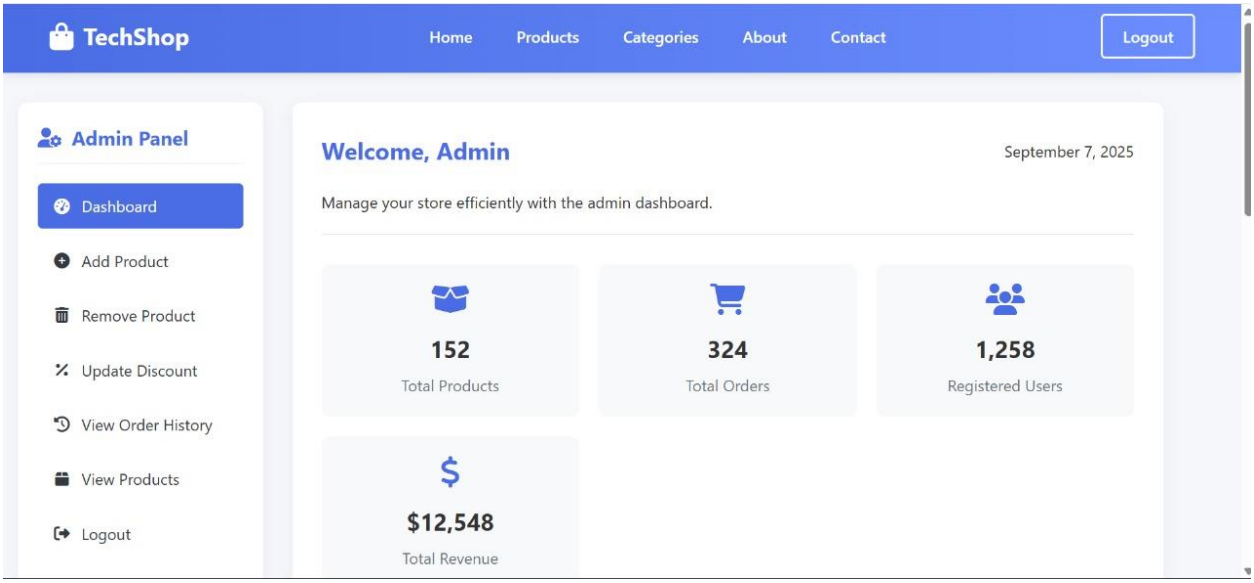
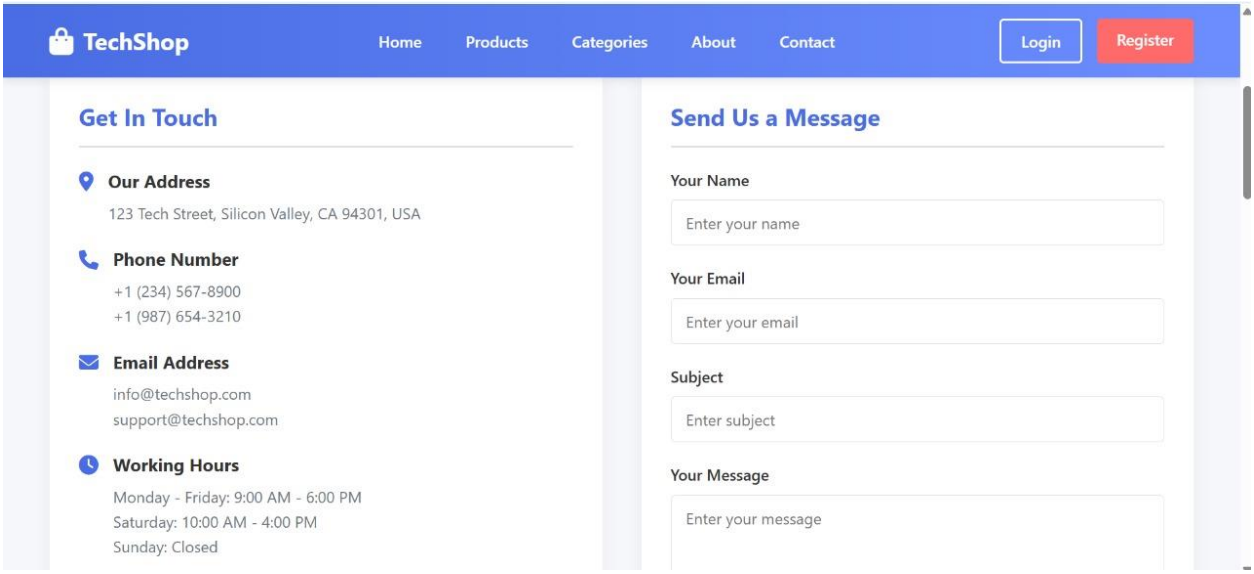
7.3) Developer's & View Product :





7.8) User Dashboard & Admin Dashboard:





7.5) Shop address: