**A Project Report on Online Shopping Portal**

**Name : Balisetti Vinay Kumar**

**Enrolment Number : 2250471271**

**Programme : BCA-ODL**

**Course Code : BCSP-064**

**Study Centre Code : 0103**

**Regional Centre Code : 3300**

**E-mail Id : vk.balisetti@gmail.com**

**Mobile Number : 7702766819**

**CONTENTS**

[**ABSTRACT** 4](#_Toc198971684)

[**OBJECTIVES** 5](#_Toc198971685)

[**TOOLS / ENVIRONMENTS USED** 7](#_Toc198971686)

[**SOFTWARE REQUIREMENTS SPECIFICATION (SRS)** 8](#_Toc198971687)

[**1. Introduction** 8](#_Toc198971688)

[**2. Overall Description** 14](#_Toc198971689)

[**3. Specific Requirements** 21](#_Toc198971690)

[**3.1. Functional Requirements** 21](#_Toc198971691)

[**3.2. External Interface Requirements** 27](#_Toc198971692)

[**3.3. Non-Functional Requirements** 29](#_Toc198971693)

[**3.4. Database Requirements** 32](#_Toc198971694)

[**4. Analysis Models** 33](#_Toc198971695)

[**ER Diagram** 36](#_Toc198971696)

[**Use Case Diagram** 38](#_Toc198971697)

[**Data Flow Diagram** 39](#_Toc198971698)

[**5. Appendices** 40](#_Toc198971699)

[**DESIGN DOCUMENT** 42](#_Toc198971700)

[**1. Introduction** 42](#_Toc198971701)

[**2. Architectural Design** 42](#_Toc198971702)

[**3. Database Design** 43](#_Toc198971703)

[**4. User Interface Design** 46](#_Toc198971704)

[**5. Security Design** 47](#_Toc198971705)

[**6. Performance Design** 47](#_Toc198971706)

[**7. Conclusion** 48](#_Toc198971707)

[**PROJECT CODE** 49](#_Toc198971708)

[**TESTING** 122](#_Toc198971709)

[**Testing Strategy** 122](#_Toc198971710)

[**Types of Testing** 122](#_Toc198971711)

[**Test Environment** 124](#_Toc198971712)

[**Test Cases** 124](#_Toc198971713)

[**Test Report** 125](#_Toc198971714)

[**INPUT & OUTPUT SCREENS** 126](#_Toc198971715)

[**LIMITATIONS OF THE PROJECT** 134](#_Toc198971716)

[**FUTURE ENHANCEMENTS OF THE PROJECT** 137](#_Toc198971717)

[**REFERENCES** 141](#_Toc198971718)

# **ABSTRACT**

The "Online Shopping Portal" is a comprehensive e-commerce platform designed to empower small and medium-scale businesses by providing them with a dynamic and user-friendly interface to showcase and sell their products online. This platform facilitates end users to register, browse products from local sellers, manage their profiles, add items to their cart, place orders, track deliveries, and handle payments seamlessly. For administrators, the system offers robust tools for product listings, and order fulfilment, ensuring efficient and effective operations. Built using PHP and MySQL, along with modern web technologies, the platform guarantees a responsive and interactive user experience. In an era where the internet is a crucial asset, the "Online Shopping Portal" bridges the gap for local retailers who often struggle with the technical and financial challenges of establishing an online presence. By leveraging this platform, businesses can tap into the vast network of online users, thereby expanding their reach and enhancing their sales potential.

# **OBJECTIVES**

The primary objectives of the "Online Shopping Portal" project are to develop a robust, user-friendly, and secure e-commerce platform that meets the needs of both customers and administrators. The specific objectives are as follows:

**User Experience**

* Provide an intuitive and seamless user experience for customers to browse, search, and purchase products.
* Offer a responsive design that adapts to various screen sizes and devices, including desktops, tablets, and smartphones.

**Functionality**

* Implement comprehensive user registration and authentication mechanisms to ensure secure access to the platform.
* Develop a robust product catalog system that allows users to browse, search and filter products efficiently.
* Create a reliable shopping cart and checkout process that supports multiple payment methods and ensures secure transactions.
* Provide order management functionalities, including order placement, tracking, and history viewing.
* Implement administrative functionalities for managing products, categories, orders, and users.

**Performance**

* Ensure the platform can handle a high volume of concurrent users and transactions without significant performance degradation.
* Optimize database queries and indexing to ensure fast data retrieval and processing.
* Implement caching mechanisms to improve data access performance and reduce database load.

**Maintainability**

* Design the system with modular and reusable components to facilitate easy maintenance and updates.
* Include comprehensive documentation to support maintenance and troubleshooting.
* Implement testing to ensure the reliability and stability of updates and changes.

**Portability**

* Ensure the platform is compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.
* Ensure the platform is responsive and adapts to different screen sizes and devices, including desktops, tablets, and smartphones.
* Ensure the platform is deployable on various hosting environments, including cloud-based servers.

By achieving these objectives, the "Online Shopping Portal" project aims to deliver a high-quality, reliable, and secure e-commerce platform that meets the needs and expectations of its users and stakeholders.

# **TOOLS / ENVIRONMENTS USED**

Programming Languages : PHP, HTML, CSS

Development Environment/ IDE : Visual Studio Code

Database : MYSQL

Backend Technology : PHP

Frontend Technology : HTML, CSS

Operating System : Windows 11

Hosting Service : XAMPP for local

AWS for cloud

Documentation Tools : Microsoft Word

Diagramming Tools : Draw.io

# **SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

## **1. Introduction**

**1.1. Purpose**

The purpose of this System Requirements Specification (SRS) document is to provide a detailed description of the requirements for the "Online Shopping Portal." This document will serve as the foundation for the design, development, testing, and eventual deployment of the web application. It aims to provide a clear and unambiguous understanding of the system's functionalities, capabilities, constraints, and interfaces for all stakeholders involved, including developer, tester, supervisor or guide, evaluator. This document is intended to be a living document and may be updated as the project progresses, with appropriate version control.

The "Online Shopping Portal" is designed to be a robust and dynamic platform enabling a central administration to showcase and sell products online. It will facilitate end-users (Customers) to register, browse products, manage their profiles, add items to a shopping cart, place orders, track deliveries, and simulate payment processes. Simultaneously, it will provide Administrators with comprehensive control over user management, product listings, category management, and order fulfilment.

**1.2. Scope**

The "Online Shopping Portal" is envisioned as a full-stack eCommerce web application designed to replicate the core functionalities of real-world online stores, specifically tailored for an academic project. It will be managed by a central administrator.

The system will include:

* Administrator Platform: A backend for administrators to list, manage and showcase products online.
* Customer Interface: A user-friendly interface for customers to discover, browse, search and filter products.
* User Authentication & Management: Secure registration, login, and profile management (details, shipping addresses) for customers.
* Core eCommerce Features:
  + Shopping cart management (add, remove, update quantity).
  + A streamlined order placement process.
  + Order history viewing and status tracking for customers.
* Simulated Payment Processing: A mechanism to demonstrate the checkout flow without actual financial transactions.
* Administrative Backend Tools for:
  + Product management.
  + Category and subcategory management.
  + User management (viewing customer details).
  + Order management (viewing orders, updating status).
* Responsive Design: Ensuring accessibility and usability across various devices (desktops, tablets, and smartphones).

**1.3. Intended Audience**

This SRS document is intended for:

* Project Developer: As a primary guide for designing, developing, implementing, and testing the system.
* Project Supervisor/Faculty: For evaluating the project scope, understanding the system's capabilities, assessing the design and implementation.
* Testers (primarily the developer in an academic project context): To create test plans and test cases to ensure the system meets the defined requirements.

**1.4. Intended Use**

The "Online Shopping Portal" is designed to provide a comprehensive and user-friendly platform for small and medium-scale businesses to showcase and sell their products online. This platform aims to bridge the gap between local retailers and the vast network of online consumers, enabling businesses to expand their reach and increase their sales potential.

For End Users:

* Browsing and Shopping: End users can easily browse through a wide range of products offered by local sellers. They can search for specific items, filter products by categories, and view detailed product descriptions and images.
* User Registration and Profile Management: Users can register on the platform, create and manage their profiles, and keep track of their order history.
* Shopping Cart and Order Placement: Users can add items to their shopping cart, review their selections, and place orders seamlessly. The platform supports multiple simulated payment methods to ensure a smooth checkout process.
* Order Tracking: Users can track the status of their orders in real-time. They can also manage their delivery preferences and addresses.

For Administrators:

* User Management: Administrators can manage user accounts.
* Product Listing and Management: Administrators can easily list their products on the platform, including detailed descriptions, images, and pricing information. They can manage their product catalogues and update inventory levels.
* Product and Category Management: Administrators can manage product categories, approve or reject product listings, and ensure that all products meet the platform's quality standards.
* Order Management: Administrators can view and manage orders placed by customers, update order statuses, and handle returns.
* Order and Payment Management: Administrators can resolve any issues related to orders and payments.
* Platform Maintenance and Updates: Administrators are responsible for maintaining the platform, ensuring its smooth operation, and implementing updates and improvements to enhance user experience and platform performance.

**1.5. Definitions, Acronyms, and Abbreviations**

* SRS: Software Requirements Specification
* SMB: Small and Medium-scale Business
* UI: User Interface
* UX: User Experience
* PHP: Hypertext Preprocessor (Server-side scripting language)
* MySQL: My Structured Query Language (Relational Database Management System)
* HTML: HyperText Markup Language
* CSS: Cascading Style Sheets
* CRUD: Create, Read, Update, Delete
* ERD: Entity-Relationship Diagram
* DFD: Data Flow Diagram
* PK: Primary Key
* FK: Foreign Key
* Admin: Administrator of the Online Shopping Portal.
* Customer/User: End-user who browses and purchases products.
* MRP: Maximum Retail Price.
* SQL: Structured Query Language.
* AWS: Amazon Web Services
* EC2: Elastic Compute Cloud (AWS service)

**1.6. References**

* IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications (Referenced as a guiding standard for the structure and content of this SRS).
* Sommerville, I. (2015). *Software Engineering*. Pearson: This textbook provides a comprehensive introduction to software engineering, including requirements analysis, design, and testing.
* BCSP-064 Project Guidelines-IGNOU: Official guidelines for structuring the SRS in Project Report.
* Wiegers, K. E., & Beatty, J. (2013). *Software Requirements*. Microsoft Press: A comprehensive guide to writing effective software requirements, including best practices and practical examples.

## **2. Overall Description**

**2.1. Product Perspective**

The "Online Shopping Portal" is a new, self-contained web application. It is designed to be an independent system that provides a complete end-to-end solution for a central administrator to conduct online sales and for Customers to shop online. The system is developed as part of the BCSP-064 project course.

The system will operate on a client-server architecture:

* **Client-Side (Frontend):** This is the presentation layer accessed by users via standard web browsers (e.g., Chrome, Firefox, Safari, Edge) on various devices, including desktops, laptops, tablets, and smartphones. It will be developed using HTML and CSS. The frontend will be responsible for rendering the user interface and handling user interactions.
* **Server-Side (Backend):** This is the application logic and data persistence layer. It will be developed using PHP as the server-side scripting language. The backend will handle business logic, process requests from the client, interact with the database, manage sessions, and enforce security measures.
* **Database:** MySQL will be used as the relational database management system (RDBMS) to store all persistent data, including user information, product details, categories, orders, etc.

While the current scope focuses on simulated payment processing, the system's architecture should be designed with modularity in mind to allow for potential future integration with actual payment gateways and other third-party services. The system does not interface with any legacy systems.

**2.2. Product Functions**

The Online Shopping Portal will provide the following key functionalities, which are detailed further in Section 3:

1. **User/Customer Functionalities:**
   * Secure registration and login.
   * Password management (change, forgot/reset).
   * Personal profile management (view/edit details, shipping addresses).
   * Product browsing, searching, and filtering.
   * Detailed product viewing.
   * Shopping cart management (add, remove, update quantities).
   * Order placement with simulated payment.
   * Viewing order history and status.
2. **Administrator Functionalities:**
   * Secure login to a dedicated admin panel.
   * Product management (add, edit, delete products, manage main image, stock levels, pricing).
   * Category and subcategory management (add, edit, delete).
   * Order management (view orders, update order status, view customer details for orders).
   * User management (view registered customer details, manage accounts).

The system aims to provide a seamless and intuitive online shopping experience for customers and efficient backend operations for administrators.

**2.3. User Classes and Characteristics**

The system will have two primary classes of users:

1. **Customer (End-User):**
   * **Characteristics:**
     + Individuals interested in browsing and purchasing products online.
     + Varying levels of technical proficiency, but generally expected to have basic web literacy (familiar with using web browsers, online forms, and standard e-commerce interactions).
     + May access the portal from various devices (desktops, tablets, smartphones).
   * **Goals/Motivations:**
     + To easily find and discover products.
     + To get detailed information about products.
     + To purchase products conveniently and securely (simulated security for payments).
     + To manage their account and track their orders.
   * **Permissions:**
     + Access to all public-facing product pages and information.
     + Ability to register, log in, and manage their own profile.
     + Ability to use shopping cart and place orders.
     + Ability to view their own order history.
2. **Administrator (Admin):**
   * **Characteristics:**
     + Personnel responsible for managing the entire online store.
     + Assumed to have moderate web literacy and understanding of e-commerce operations.
     + Will have dedicated credentials to access the admin panel.
   * **Goals/Motivations:**
     + To manage product listings effectively (add new products, update existing ones, manage inventory).
     + To organize products using categories and subcategories.
     + To process and manage customer orders efficiently.
     + To manage customer accounts.
     + To monitor basic site activity.
   * **Permissions:**
     + Full CRUD access to products and categories.
     + Ability to view and manage all customer orders.
     + Ability to view and manage registered customer accounts.
     + Access to the admin dashboard and site settings.

**2.4. Operating Environment**

* **Client-Side:**
  + **Web Browsers:** The system shall be compatible with the latest stable versions of the following web browsers: Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge and other chromium-based browsers.
  + **Operating Systems:** Platform independent (Windows, macOS, Linux, Android, iOS) as long as a compatible browser is used.
  + **Screen Resolutions:** The system shall be responsive and adapt to various screen resolutions, from mobile (e.g., 360x640) to desktop (e.g., 1920x1080 and higher).
  + **Network:** Requires an active internet connection.
* **Server-Side:**
  + **Operating System:** Linux (recommended, e.g., Ubuntu, CentOS) or Windows Server.
  + **Web Server:** Apache (version 2.4 or later) or Nginx (latest stable version).
  + **PHP:** Version 7.4 or later.
  + **MySQL:** Version 5.7 or later.
  + **Other Software:** Standard server utilities, SSL certificate for HTTPS
* **Development Environment:**
  + Asus VivoBook S14 I5 12th gen with Windows 11.
  + Development stack: XAMPP.
  + Code editor/IDE: VS Code.
  + Version control system: Git.

**2.5. Design and Implementation Constraints**

1. **Technology Stack:** PHP, MySQL, HTML, CSS.
2. **Academic Project Scope:** Core functionalities, technical competence demonstration.
3. **Simulated Payments:** Real-time payment gateway integration is out of scope and unsecure for academic purposes.
4. **No Third-Party Costs:** Primarily open-source technologies and free trial of some premium programs.
5. **Security:** cryptographic hash for passwords, XSS breach security.
6. **Timeline:** Within BCSP-064 course timeline.
7. **Deployment:** Deployable on standard web hosting or local XAMPP for demo.

**2.6. Assumptions and Dependencies**

* **Assumptions:**
  + Stable internet for users.
  + Basic user web literacy.
  + Developer access to necessary tools.
  + Product content provided/created by admin.
  + UTF-8-character set.
  + Project guide support.
* **Dependencies:**
  + Functioning web server, PHP interpreter, MySQL server.
  + Modern client web browsers.
  + Successful development environment setup.

## **3. Specific Requirements**

This section details all the specific functional and non-functional requirements that the Online Shopping Portal must satisfy. This section is the guide for the developer to meet his program needs and functional needs of making this system.

### **3.1. Functional Requirements**

Functional requirements define what the system is supposed to do. Functional requirements describe the specific behaviours and functions that the system must support. These requirements define what the system should do in response to user inputs and other stimuli.

**3.1.1. User Module**

This module handles all functionalities related to the customer/end-user.

**1. User Registration**

* The system shall allow a new user to register by providing a first name, last name, unique email address (username), password, confirm password, and contact number.
* The system shall validate all registration fields for correctness.
* The system shall ensure the email address provided during registration is unique across all existing users.
* The system shall securely hash and store user passwords.
* Upon successful registration, the system shall display a success message and may automatically log the user in or direct them to the login page.
* The system should provide clear error messages if registration fails (e.g., email already exists, password mismatch).

**2. User Login**

* The system shall provide a login interface for registered users to enter their email address and password.
* The system shall authenticate users based on the provided credentials against the stored records.
* Upon successful login, the system shall create a user session and redirect the user to their dashboard or the homepage.
* The system shall display an appropriate error message for invalid login attempts.
* The system should implement a basic mechanism to prevent brute-force login attacks.
* The system shall provide a "Logout" option that terminates the user session.

**3. Password Management**

* The system shall allow a logged-in user to change their password.
* The system shall provide a "Forgot Password" functionality.
* The system shall ask old password before entering new password.
* The system shall store the new password in place of old password and remove the old password from the database because of privacy issues.
* The system shall not show the password to the admins.

**.4. User Profile Management**

* Logged-in users shall be able to view their profile information.
* Logged-in users shall be able to edit their profile information excluding email.
* Logged-in users shall be able to manage multiple shipping addresses.

**5. Shopping Cart Management**

* Users shall be able to add products to their shopping cart.
* The system shall allow users to view their shopping cart contents.
* Users shall be able to update item quantities in their cart.
* Users shall be able to remove items from their cart.
* The system shall update cart totals dynamically.
* Cart contents should persist for logged-in users (session-based for guests).
* The system shall indicate stock availability issues in the cart.

**6. Order Placement and Checkout**

* The system shall allow users to proceed to checkout from their cart.
* Checkout shall require selection/confirmation of a shipping address.
* An order summary shall be displayed before final confirmation.
* The system shall provide a simulated payment option.
* A unique order ID shall be generated upon successful order placement.
* An order confirmation page shall be displayed.
* Order details shall be stored in the database.
* Order confirmation shall be indicated.
* Product stock levels shall be updated after order placement.

**7. Order History and Tracking**

* Logged-in users shall be able to view their order history.
* Order history shall list key order details.
* Users shall be able to view detailed information for each past order.
* The system shall display the current order status as updated by the Admin.

**3.1.2. Product Module**

**1. Product Display and Browsing**

* The system shall display products in a gallery or list format.
* Product listings shall display primary image, name, and price.
* Users shall be able to browse products by categories and subcategories.
* Pagination shall be supported for product listings.
* The homepage should feature a selection of products.

**2. Product Search**

* The system shall provide a keyword-based product search.
* Search results shall be displayed in a product listing format.
* A "No results found" message shall be displayed if applicable.

**3. Product Filtering**

* Users should be able to filter products.
* Users should be able to filter products by subcategories.

**4. Detailed Product View**

* Clicking a product shall navigate to its detail page.
* The product detail page shall display: Product Name, Main Product Image, Detailed Description, MRP and Selling Price, Stock Status, Category/Subcategory, quantity selection, and "Add to Cart" button.

**3.1.3. Admin Module (Backend Management)**

**1. Secure Admin Login**

* Secure login interface for administrators.
* Admin authentication distinct from customer authentication.
* Redirect to admin dashboard upon successful login.

**2. Admin Dashboard**

* Dashboard with summary site statistics.
* Quick navigation links to other admin modules.

**3. Product Management (CRUD)**

* Admins shall add products with name, description, MRP, selling price, stock, category, and a main image.
* Admins shall view all products with search/filter options.
* Admins shall edit existing product details.
* Admins shall soft delete (deactivate) products.
* Admins shall update stock levels.
* System should support uploading one main image per product.

**4. Category and Subcategory Management (CRUD)**

* Admins shall create, view, edit, and delete product categories.
* Admins shall manage subcategories and associate them with parent categories.

**5. Order Management**

* Admins shall view all customer orders with filtering options.
* Admins shall view detailed order information.
* Admins shall update order status.
* Admins should be able to add tracking information (simulated).

**3.1.4. Common Functionalities**

**1. Navigation**

* Clear, consistent, intuitive navigation.
* Breadcrumbs where appropriate.

**2. Session Management**

* Secure session management.
* Session expiration after inactivity.
* Logout invalidates session.

**3. Error Handling**

* Graceful error handling with user-friendly messages.
* Clear input validation errors.
* User-friendly HTTP error pages.

**4. Responsive Design**

* Responsive customer-facing portal.
* Admin panel reasonably usable on tablets.

### **3.2. External Interface Requirements**

This section describes the interfaces between the Online Shopping Portal and external entities, including users, hardware, and other software systems.

**3.2.1. User Interfaces (UI)**

* Clean, modern, visually appealing UI.
* Intuitive and easy navigation.
* Consistent branding.
* User-friendly forms.
* Prominent call-to-action buttons.
* Basic web accessibility adherence.

**3.2.2. Hardware Interfaces**

* The system shall support standard input devices, such as keyboards and mice, for user interactions.
* The system shall support touchscreen interfaces for users accessing the portal via tablets or smartphones.
* The system shall support standard output devices, such as monitors and displays, for visual output.
* The system shall be compatible with various network devices, such as routers, switches, and modems, for establishing and maintaining network connections.
* The system shall be compatible with major operating systems, including Windows, macOS, Linux, iOS, and Android, to ensure broad accessibility.

**3.2.3. Software Interfaces**

* The system shall work on every latest up to date browser which offer new html versions.
* The system connected databases shall be accessible from MySQL or PHPADMIN for controlling and data reporting.
* The system shall integrate with a relational database management system (RDBMS) for storing and retrieving data.
* The system shall ensure data integrity and consistency through the use of transactions and constraints.
* The system shall provide dashboards and reports for administrators to monitor and analyse the performance of the portal.

**3.2.4. Communication Interfaces**

* The system shall use HTTPS protocol for secure communication between the client and server.
* The system shall support HTTP/2 for improved performance and efficiency in data transmission.
* The system shall ensure that all data exchanged is well-formed and validated according to the specified data formats.
* The system shall provide detailed error messages and logs for troubleshooting and debugging communication issues.

### **3.3. Non-Functional Requirements**

This section outlines the non-functional requirements that the Online Shopping Portal must meet to ensure a high-quality user experience and system performance. These requirements are crucial for ensuring that the system meets the expectations of users and stakeholders in terms of performance, security, usability, reliability, and other quality attributes. In the context of the "Online Shopping Portal," non-functional requirements might include performance, security, usability, reliability, maintainability, and portability.

**3.3.1. Performance Requirements**

* The system shall load the page on average within 3 seconds.
* The system shall respond to user’s typical actions in less than 2 seconds.
* For academic purposes only, the system shall handle 20-50 concurrent users.
* The system shall optimize database queries and indexing to ensure fast data retrieval and processing.
* The system shall support a minimum of 1000 concurrent users without significant degradation in performance.

**3.3.2. Security Requirements**

* The system shall implement hashing for passwords and any other private info.
* The system shall be protected against common web vulnerabilities like SQL injection, XSS.
* The system shall implement secure session management.
* The system shall implement protected admin panel.
* The system shall implement client side and server-side input validation.
* The system shall validate the file uploads for virus free images.

**3.3.3. Usability Requirements**

* The system shall provide an intuitive and user-friendly interface that is easy to navigate.
* The system shall provide clear and concise error messages to help users understand and resolve issues.
* The system shall provide responsive design to ensure a consistent user experience across different devices and screen sizes.
* The system shall implement clear and consistent terminology.

**3.3.4. Reliability Requirements**

* The system shall have an uptime of at least 99.9%
* The system shall implement backup and recovery mechanisms to ensure data integrity and availability in case of failures.
* The system shall implement redundancy and failover mechanisms to ensure continuous operation in case of hardware or software failures.
* The system shall implement data integrity in database.

**3.3.5. Maintainability Requirements**

* The system shall be designed with modular and reusable components to facilitate easy maintenance and updates.
* The system shall include comprehensive documentation to support maintenance and troubleshooting.
* The system shall provide logging and monitoring tools to track system performance and identify issues.

**3.3.6. Portability Requirements**

* The system shall be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.
* The system shall be responsive and adapt to different screen sizes and devices, including desktops, tablets, and smartphones.
* The system shall support multiple operating systems, including Windows, macOS.

### **3.4. Database Requirements**

* The system shall use MySQL (or a compatible MariaDB version).
* The database schema shall be designed using normalization principles (at least 3NF).
* Key entities to be stored include:
  + **Users/Customers:** (UserID, FirstName, LastName, Email, PasswordHash, ContactNumber, RegistrationDate)
  + **Addresses:** (AddressID, UserID (FK), Street, City, State, ZipCode, Country, IsDefaultShipping)
  + **Categories:** (CategoryID, CategoryName, ParentCategoryID)
  + **Products:** (ProductID, ProductName, Description, MRP, SellingPrice, StockQuantity, CategoryID (FK), MainImageURL, DateAdded, IsActive)
  + **Orders:** (OrderID, UserID (FK), OrderDate, TotalAmount, ShippingAddressID (FK), OrderStatus, PaymentMethod (simulated), PaymentStatus (simulated), TrackingNumber (simulated))
  + **OrderItems:** (OrderItemID, OrderID (FK), ProductID (FK), Quantity, PriceAtPurchase, Subtotal)
  + **Administrators:** (AdminID, Username, PasswordHash, Email, Role, LastLogin)
* The system shall have appropriate data types, constraints, and indexes used.

## **4. Analysis Models**

This section presents the analysis models that capture the functional and behavioral requirements of the Online Shopping Portal.

**4.1. Entity-Relationship (ER) Diagram**

The Entity-Relationship Diagram is a crucial component of the database design for the Online Shopping Portal. It serves as a conceptual blueprint, visually representing the logical structure of the database. The ERD identifies the key "entities", the "attributes", and the "relationships" that exist between these entities. For this project, the ER Diagram provides a clear and unambiguous understanding of how different data elements are interconnected, forming the backbone of the portal's data management system.

**Entities and Attributes**

1. **Users**
   * UserId (PK)
   * FirstName
   * LastName
   * Email
   * PasswordHash
   * ContactNumber
   * RegistrationDate
2. **Addresses**
   * AddressId (PK)
   * Users\_UserId (FK)
   * Street
   * City
   * State
   * ZipCode
   * Country
   * IsDefaultShipping
3. **Categories**
   * CategoryID (PK)
   * CategoryName
   * ParentCategoryID
4. **Products**
   * ProductID (PK)
   * ProductName
   * Description
   * MRP
   * SellingPrice
   * StockQuantity
   * Categories\_CategoryID (FK)
   * MainImageURL
   * DateAdded
   * IsActive
5. **Order\_items**
   * OrderItemID (PK)
   * OrderID
   * ProductID
   * Quantity
   * PriceAtPurchase
   * Subtotal
6. **Orders**
   * OrderID (PK)
   * Users\_UserID (FK)
   * OrderDate
   * TotalAmount
   * ShippingAddressID
   * OrderStatus
   * PaymentMethod
   * PaymentStatus
   * TrackingNumber

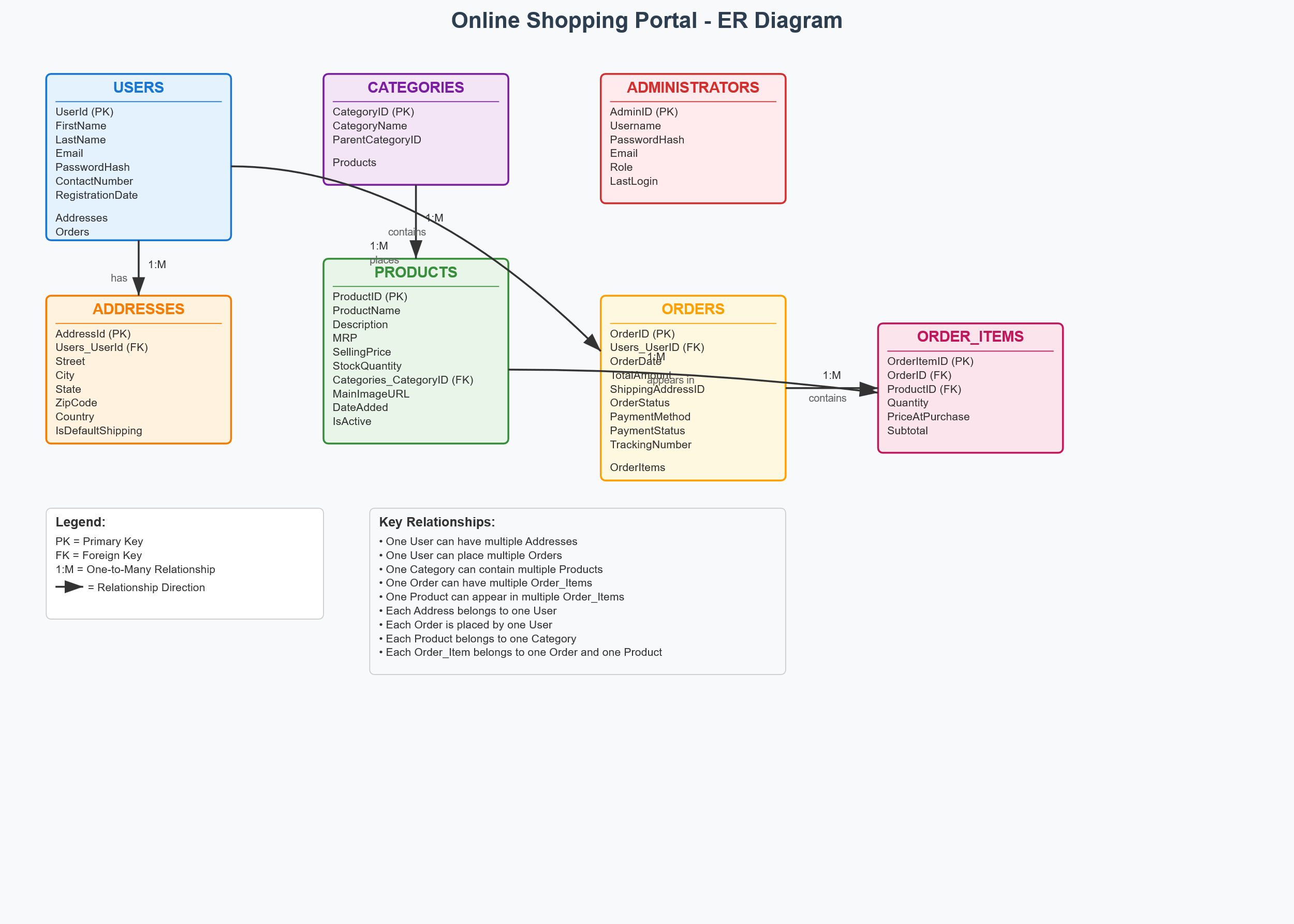
**7.Administrators**

* + AdminID (PK)
  + Username
  + PasswordHash
  + Email
  + Role
  + LastLogin

**Relationships**

* A users:
  + can have multiple Addresses, Orders
* An Address:
  + belongs to one Customer
* A Product:
  + belongs to one Category
  + can appear in many OrderITEMs, Orders
* An Order:
  + is placed by one user
  + is associated with one or more OrderITEMs
* OrderITEM:
  + connects Orders and Products
  + includes MRP and quantity per product per order

### **ER Diagram**



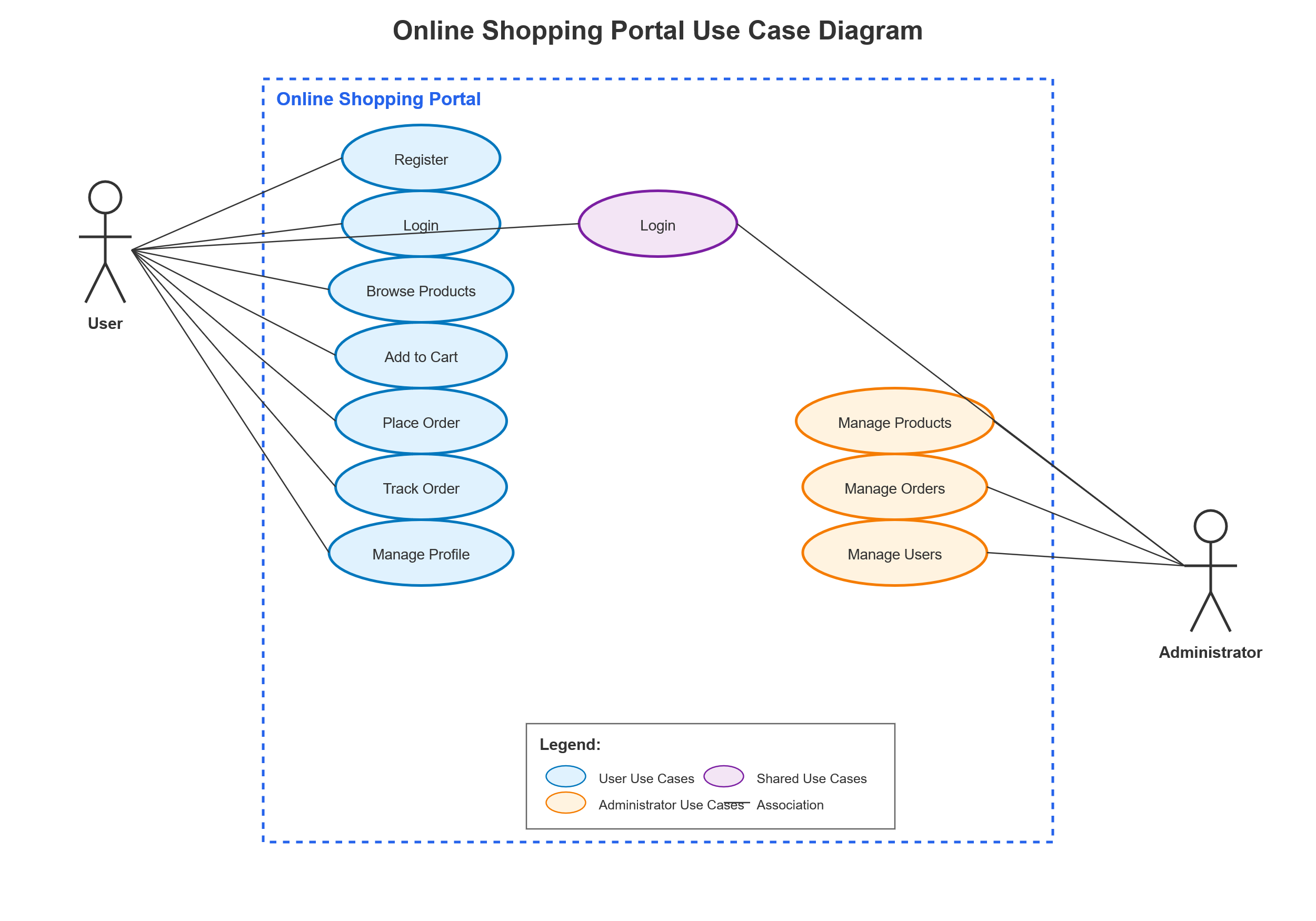
**4.2 Use Case Diagram**

The use case diagram illustrates the interactions between users (actors) and the system (use cases). The Use Case Diagram is a vital behavioural diagram that provides a high-level view of the functional requirements of the Online Shopping Portal. It illustrates the interactions between external users, known as "actors," and the system itself, represented by "use cases." Each use case describes a specific piece of functionality that the portal offers to an actor to achieve a particular goal. For this project, the Use Case Diagram serves as a primary tool to define the scope of the system and to ensure that all intended functionalities are captured from a user's perspective.

The use case diagram illustrates the interactions between users (actors) and the system (use cases).

* **Actors:**
  + Users
  + Administrator
* **Use Cases:**
  + Register
  + Login
  + Browse Products
  + Add to Cart
  + Place Order
  + Track Order
  + Manage Profile
  + Manage Products
  + Manage Orders
  + Manage Users

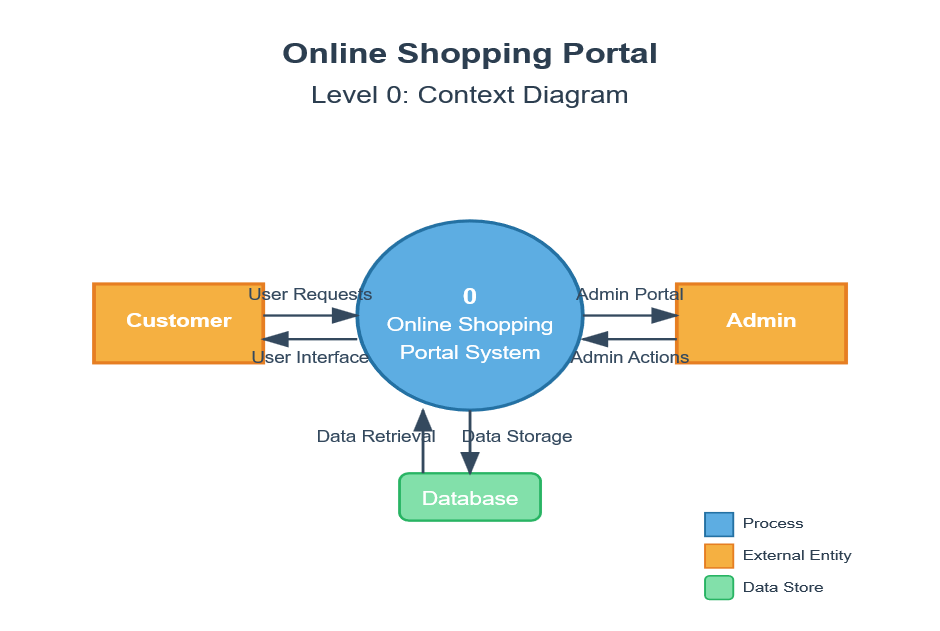
### **Use Case Diagram**

****

**4.3. Data Flow Diagrams (DFD)**

The Data Flow Diagram (DFD) is a graphical representation that illustrates the flow of data through the Online Shopping Portal system. Unlike Use Case Diagrams which focus on user interactions and system functionality, or ERDs which model data structure, DFDs emphasize how data is processed and transformed as it moves from its source to its destination. DFDs depict the system as a network of processes, data stores, external entities, and the data flows connecting them. For this project, DFDs will help in understanding how information is input, processed, stored, and output by the portal.

### **Data Flow Diagram**

****



## **5. Appendices**

**A. Code Files**

The code files for the "Online Shopping Portal" project are available in a GitHub repository. You can access and download the code files using the following link:

<https://github.com/Vinay-kumar-code/Online_Shopping_Portal>

If you encounter any issues or need further assistance, please contact the repository owner.

**B. Setup Instructions**

To set up and run the "Online Shopping Portal" project on your local machine, follow these steps:

1. Install XAMPP Control Panel:
   * Download and install the XAMPP control panel for Windows 11, which includes PHP, MySQL, and Apache Server.
2. Download Project Files:
   * Download the project files from the link provided in the SRS Document Appendices section.
3. Copy Files to htdocs:
   * Copy the downloaded files and maintain the folder structure as in the GitHub repository.
   * Paste them into a new folder within the htdocs folder of XAMPP located in the C drive (for Windows 11).
4. Rename the Folder:
   * Rename the new folder to online\_shopping\_portal.
5. Start Servers:
   * Open the XAMPP control panel.
   * Start the Apache server and MySQL server.
6. Set Up the Database:
   * Open phpMyAdmin.
   * Import the database and tables structure from the SQL file included in the downloaded GitHub files.
7. Access the Portal:
   * Open any updated web browser on your system.
   * Type the URL http://localhost/online\_shopping\_portal/index.php.
   * You will access the fully working online shopping portal if you followed the above steps correctly.

**C. Glossary**

* XAMPP: A free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.
* phpMyAdmin: A free and open-source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.
* GitHub: A web-based platform used for version control and collaboration. It allows developers to work together on projects from anywhere.

**D.References**

* XAMPP Download: <https://www.apachefriends.org/index.html>
* phpMyAdmin Documentation: <https://www.phpmyadmin.net/docs/>
* GitHub Documentation: <https://docs.github.com/en>

# **DESIGN DOCUMENT**

## **1. Introduction**

This document provides a detailed design of the "Online Shopping Portal" project. It outlines the architectural design, system components, database design, user interface design, and other key aspects of the system.

## **2. Architectural Design**

**2.1 Overview**

The Online Shopping Portal will be designed using a layered architecture to ensure modularity, scalability, and maintainability. The system will consist of the following layers:

* Presentation Layer: Responsible for handling user interactions and displaying info.
* Business Logic Layer: Responsible for processing business logic and managing data.
* Data Access Layer: Responsible for interacting with the database and managing data.

**2.2 System Components**

The system will be composed of the following main components:

* User Management: Handles user registration, authentication, and profile management.
* Product Management: Manages product catalog, including product details, categories, and inventory.
* Order Management: Handles order placement, processing, and tracking.
* Admin Management: Provides administrative functionalities for managing products, categories, orders, and users.

## **3. Database Design**

**3.1 Users**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| UserID | int(11) | Primary key, unique identifier for each user. |
| FirstName | varchar(100) | First name of the user. |
| LastName | varchar(100) | Last name of the user. |
| Email | varchar(255) | Email address of the user, used for login and communication. |
| PasswordHash | varchar(255) | Hashed password for secure authentication. |
| ContactNumber | varchar(20) | Contact number of the user. |
| RegistrationDate | timestamp | Date and time when the user registered. |

**3.2 Addresses**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| AddressID | int(11) | Primary key, unique identifier for each address. |
| UserID | int(11) | Foreign key, references UserId in the Users table. |
| Street | varchar(255) | Street address. |
| City | varchar(100) | City of the address. |
| State | varchar(100) | State of the address. |
| ZipCode | varchar(20) | Zip code of the address. |
| Country | varchar(100) | Country of the address. |
| IsDefaultShipping | tinyint(1) | Indicates if this is the default shipping address for the user. |

**3.3 Products**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ProductID | int(11) | Primary key, unique identifier for each product. |
| ProductName | varchar(255) | Name of the product. |
| Description | text | Description of the product. |
| MRP | decimal(10,2) | Maximum Retail Price of the product. |
| SellingPrice | decimal(10,2) | Selling price of the product. |
| StockQuantity | int(11) | Quantity of the product in stock. |
| CategoryID | int(11) | Foreign key, references CategoryId in the Categories table. |
| MainImageURL | varchar(255) | URL of the main product image. |
| DateAdded | timestamp | Date and time when the product was added. |
| IsActive | tinyint(1) | Indicates if the product is active. |

**3.4 Orders**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| OrderID | int(11) | Primary key, unique identifier for each order. |
| UserID | int(11) | Foreign key, references UserId in the Users table. |
| OrderDate | timestamp | Date and time when the order was placed. |
| TotalAmount | decimal(10,2) | Total amount of the order. |
| ShippingAddressID | int(11) | Foreign key, references AddressId in the Addresses table. |
| OrderStatus | varchar(50) | Status of the order (e.g., Pending, Processing, Shipped, Delivered, Cancelled). |
| PaymentMethod | varchar(50) | Method of payment. |
| PaymentStatus | varchar(50) | Status of the payment (e.g., Pending, Completed, Failed). |
| TrackingNumber | varchar(100) | Tracking number for the order. |

**3.4 Categories**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| CategoryID | int(11) | Primary key, unique identifier for each category. |
| CategoryName | varchar(100) | Name of the category. |
| ParentCategoryID | int(11) | Foreign key, references CategoryId in the Categories table. |

**3.6 OrderItems**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| OrderItemID | int(11) | Primary key, unique identifier for each order item. |
| OrderID | int(11) | Foreign key, references OrderId in the Orders table. |
| ProductID | int(11) | Foreign key, references ProductId in the Products table. |
| Quantity | int(11) | Quantity of the product in the order. |
| PriceAtPurchase | decimal(10,2) | Price of the product at the time of the order. |
| Subtotal | decimal(10,2) | Subtotal amount for the order item. |

**3.7 Administrators**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| AdminID | int(11) | Primary key, unique identifier for each admin. |
| Username | varchar(100) | Username of the admin. |
| PasswordHash | varchar(255) | Hashed password for secure authentication. |
| Email | varchar(255) | Email address of the admin. |
| Role | varchar(50) | Role of the admin (e.g., Admin, Super Admin). |
| LastLogin | timestamp | Date and time of the last login. |

## **4. User Interface Design**

**4.1 Overview**

The user interface (UI) of the Online Shopping Portal will be designed to provide an intuitive and seamless user experience. The UI will be responsive and adapt to various screen sizes and devices, including desktops, tablets, and smartphones.

**4.2 Key UI Components**

* **Homepage:** The main landing page that displays information about online shopping portal.
* **Product Listing Page:** Displays a list of products with filtering by category options.
* **Product Detail Page:** Provides detailed information about a specific product, including images, descriptions and prices.
* **Shopping Cart Page:** Shows the items in the user's shopping cart, with options to update quantities, remove items, and proceed to checkout.
* **Checkout Page:** Allows users to enter shipping and payment information to complete their purchase.
* **Order History Page:** Displays the user's order history, with options to view order details and track order status.
* **User Profile Page:** Allows users to view and update their profile information, including name, email, contact number, and addresses.
* **Admin Dashboard:** Provides administrative functionalities for managing products, categories, orders, and users.

## **5. Security Design**

**5.1 Overview**

Security is a critical aspect of the Online Shopping Portal. The system will implement various security measures to protect sensitive data and ensure secure transactions.

**5.2 Key Security Measures**

* **Data Encryption:** All sensitive data, including user passwords and payment information, will be encrypted using industry-standard encryption algorithms.
* **Role-Based Access Control:** The system will implement role-based access control to restrict access to sensitive data and functionalities.
* **Prevention of Common Security Vulnerabilities:** The system will implement measures to prevent common security vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

## **6. Performance Design**

**6.1 Overview**

Performance is a key consideration for the Online Shopping Portal. The system will be designed to handle a high volume of concurrent users and transactions without significant performance degradation.

**6.2 Key Performance Measures**

* **Optimized Database Queries and Indexing:** The system will optimize database queries and indexing to ensure fast data retrieval and processing.
* **Caching Mechanisms:** The system will implement caching mechanisms to improve data access performance and reduce database load.
* **Load Balancing:** The system will use load balancing techniques to distribute traffic across multiple servers and ensure high availability.

## **7. Conclusion**

This design document outlines the architectural design, system components, data models, user interface design, security design, and performance design for the "Online Shopping Portal" project. By following this design, the development team can ensure that the system meets the specified requirements and delivers a high-quality and reliable user experience.

# **PROJECT CODE**

**Config.php**

<?php

// Database Configuration and Utility Functions

// --- Error Reporting (Enable for Development) ---

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

// --- Session Management ---

if (session\_status() == PHP\_SESSION\_NONE) {

    session\_start();

}

//Database credentials

define('DB\_SERVER', 'localhost');

define('DB\_USERNAME', 'root');

define('DB\_PASSWORD', '');

define('DB\_NAME', 'online\_shopping\_portal\_bca');

//MySQL connection

$conn = mysqli\_connect(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD, DB\_NAME);

if ($conn === false) {

    die("ERROR: Could not connect to database. " . mysqli\_connect\_error());

}

mysqli\_set\_charset($conn, "utf8mb4");

//It correctly identify the root URL of project.

$protocol = (!empty($\_SERVER['HTTPS']) && $\_SERVER['HTTPS'] !== 'off' || $\_SERVER['SERVER\_PORT'] == 443) ? "https://" : "http://";

$host = $\_SERVER['HTTP\_HOST'];

$project\_root\_path\_segment = rtrim(str\_replace($\_SERVER['DOCUMENT\_ROOT'], '', str\_replace('\\', '/', \_\_DIR\_\_)), '/');

define('BASE\_URL', $protocol . $host . $project\_root\_path\_segment . '/');

define('UPLOAD\_DIR', \_\_DIR\_\_ . '/uploads/');

define('UPLOAD\_URL', BASE\_URL . 'uploads/');

// Ensure uploads directory exists and is writable

if (!is\_dir(UPLOAD\_DIR)) {

    // Attempt to create it

    if (!mkdir(UPLOAD\_DIR, 0775, true) && !is\_dir(UPLOAD\_DIR)) {

        die('Failed to create uploads directory. Please create it manually and ensure it is writable by the web server.');

    }

}

function sanitize\_input($data) {

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data, ENT\_QUOTES, 'UTF-8');

    return $data;

}

function redirect($url\_from\_root) {

    header("Location: " . BASE\_URL . $url\_from\_root);

    exit();

}

function display\_message($message, $type = 'info') {

    $class = 'message-info';

    if ($type === 'success') {

        $class = 'message-success';

    } elseif ($type === 'error') {

        $class = 'message-error';

    }

    $\_SESSION['message'] = "<div class='{$class}'>{$message}</div>";

}

function get\_session\_message() {

    if (isset($\_SESSION['message'])) {

        $message = $\_SESSION['message'];

        unset($\_SESSION['message']);

        return $message;

    }

    return '';

}

function display\_admin\_message($message, $type = 'info') {

    $class = 'message-info'; // Default

    if ($type === 'success') {

        $class = 'message-success';

    } elseif ($type === 'error') {

        $class = 'message-error';

    }

    $\_SESSION['admin\_message'] = "<div class='{$class}'>{$message}</div>";

}

function get\_admin\_session\_message() {

    if (isset($\_SESSION['admin\_message'])) {

        $message = $\_SESSION['admin\_message'];

        unset($\_SESSION['admin\_message']);

        return $message;

    }

    return '';

}

function is\_user\_logged\_in() {

    return isset($\_SESSION['user\_id']);

}

function is\_admin\_logged\_in() {

    return isset($\_SESSION['admin\_id']) && isset($\_SESSION['admin\_username']);

}

function require\_user\_login($redirect\_target\_after\_login = null) {

    if (!is\_user\_logged\_in()) {

        display\_message("You need to login to access this page.", "error");

        $login\_url = 'login.php';

        if ($redirect\_target\_after\_login) {

            $login\_url .= '?redirect=' . urlencode($redirect\_target\_after\_login);

        }

        redirect($login\_url);

    }

}

function require\_admin\_login($admin\_login\_page = 'index.php') {

    if (!is\_admin\_logged\_in()) {

        display\_admin\_message("You need to login to access the admin panel.", "error");

        redirect('admin/' . $admin\_login\_page);

    }

}

?>

**Header.php**

<?php

require\_once 'config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Online Shopping Portal</title>

    <link rel="stylesheet" href="<?php echo BASE\_URL; ?>style.css">

</head>

<body>

    <header>

        <div class="container">

            <div id="branding">

                <h1><a href="<?php echo BASE\_URL; ?>index.php">Online Shopping Portal</a></h1>

            </div>

            <nav>

                <ul>

                    <li><a href="<?php echo BASE\_URL; ?>products.php">Products</a></li>

                    <li><a href="<?php echo BASE\_URL; ?>cart.php">Cart

                        <?php

                        $cart\_item\_count = 0;

                        if (isset($\_SESSION['cart']) && !empty($\_SESSION['cart'])) {

                            foreach ($\_SESSION['cart'] as $item) {

                                $cart\_item\_count += $item['quantity'];

                            }

                        }

                        if ($cart\_item\_count > 0) {

                            echo " <span class='cart-count'>($cart\_item\_count)</span>";

                        }

                        ?>

                    </a></li>

                    <?php if (is\_user\_logged\_in()): ?>

                        <li><a href="<?php echo BASE\_URL; ?>profile.php">My Profile</a></li>

                        <li><a href="<?php echo BASE\_URL; ?>order\_history.php">Order History</a></li>

                        <li><a href="<?php echo BASE\_URL; ?>logout.php">Logout</a></li>

                    <?php else: ?>

                        <li><a href="<?php echo BASE\_URL; ?>login.php">Login</a></li>

                        <li><a href="<?php echo BASE\_URL; ?>register.php">Register</a></li>

                    <?php endif; ?>

                     <li><a href="<?php echo BASE\_URL; ?>admin/index.php">Admin</a></li>

                </ul>

            </nav>

        </div>

    </header>

    <div class="container main-content">

        <?php

        echo get\_session\_message();

        ?>

**Footer.php**

<?php

?>

    </div> <footer>

        <p>Online Shopping Portal &copy; <?php echo date("Y"); ?> - BCA Project by Vinay Kumar</p>

        <a href="<?php echo BASE\_URL; ?>terms.php" style="color:#fff; margin: 0 10px;">Terms & Conditions</a> |

        <a href="<?php echo BASE\_URL; ?>privacy.php" style="color:#fff; margin: 0 10px;">Privacy Policy</a>

    </footer>

</body>

</html>

**Index.php**

<?php

require\_once 'header.php';

?>

<h2>Welcome to the Online Shopping Portal!</h2>

<p>

This is the homepage of the Online Shopping Portal, developed as part of an academic project for the Bachelor of Computer Applications (BCA) program.</p>

<p>

⚠️ Please note: The Terms & Conditions and Privacy Policy pages are sample placeholders included only to demonstrate standard website structure. You may ignore their content.

</p><p>

✅ Key Features

<li>Browse products and categories</li>

<li>Add items to cart</li>

<li>Place orders</li>

<li>Mobile-friendly and responsive design</li>

<li>Sample pages included for layout and demonstration purposes</li>

</p>

<p>

🚀 Scalable and Customizable</p>

<p>

This portal is designed with scalability and customization in mind. It can be easily extended to support additional features such as:

<li>Secure payment gateway integration</li>

<li>Admin panel for product management</li>

<li>User accounts and order history</li>

<li>Email notifications and more</li>

</p>

<p>

💡 Try It Out</p>

<p>

Explore and interact with all the functionalities on the site to get a complete experience. Your feedback is appreciated as this is a learning-focused academic prototype.

</p>

<?php

require\_once 'footer.php';

?>

**Login.php**

<?php

require\_once 'config.php';

if (is\_user\_logged\_in()) {

    redirect('index.php');

}

$email = $password = "";

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $email = sanitize\_input($\_POST['email']);

    $password\_input = $\_POST['password'];

    if (empty($email)) {

        $errors[] = "Email is required.";

    } elseif (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

        $errors[] = "Invalid email format.";

    }

    if (empty($password\_input)) {

        $errors[] = "Password is required.";

    }

    if (empty($errors)) {

        $sql = "SELECT UserID, FirstName, PasswordHash FROM users WHERE Email = ?";

        if ($stmt = mysqli\_prepare($conn, $sql)) {

            mysqli\_stmt\_bind\_param($stmt, "s", $param\_email);

            $param\_email = $email;

            if (mysqli\_stmt\_execute($stmt)) {

                mysqli\_stmt\_store\_result($stmt);

                if (mysqli\_stmt\_num\_rows($stmt) == 1) {

                    mysqli\_stmt\_bind\_result($stmt, $user\_id, $first\_name, $hashed\_password\_from\_db);

                    if (mysqli\_stmt\_fetch($stmt)) {

                        if (password\_verify($password\_input, $hashed\_password\_from\_db)) {

                            session\_regenerate\_id(true);

                            $\_SESSION['user\_id'] = $user\_id;

                            $\_SESSION['user\_email'] = $email;

                            $\_SESSION['user\_first\_name'] = $first\_name;

                            display\_message("Login successful! Welcome back, " . htmlspecialchars($first\_name) . ".", "success");

                            redirect('index.php');

                        } else {

                            $errors[] = "Invalid email or password.";

                        }

                    }

                } else {

                    $errors[] = "Invalid email or password.";

                }

            } else {

                $errors[] = "Oops! Something went wrong. Please try again later. (Execute failed)";

            }

            mysqli\_stmt\_close($stmt);

        } else {

             $errors[] = "Database error. Please try again later. (Prepare failed)";

        }

    }

    if (!empty($errors)) {

        $error\_message\_html = "<ul>";

        foreach ($errors as $error) {

            $error\_message\_html .= "<li>" . htmlspecialchars($error) . "</li>";

        }

        $error\_message\_html .= "</ul>";

        $\_SESSION['form\_errors'] = $error\_message\_html;

        $\_SESSION['form\_data'] = ['email' => $email];

        redirect('login.php');

    }

}

$form\_errors\_html = '';

if (isset($\_SESSION['form\_errors'])) {

    $form\_errors\_html = $\_SESSION['form\_errors'];

    unset($\_SESSION['form\_errors']);

}

$form\_data\_email = '';

if (isset($\_SESSION['form\_data']['email'])) {

    $form\_data\_email = htmlspecialchars($\_SESSION['form\_data']['email'], ENT\_QUOTES, 'UTF-8');

    unset($\_SESSION['form\_data']);

}

require\_once 'header.php';

?>

<h2>User Login</h2>

<?php

if (!empty($form\_errors\_html)) {

    echo "<div class='message-error'>" . $form\_errors\_html . "</div>";

}

?>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="post">

    <div class="form-group">

        <label for="email">Email:</label>

        <input type="email" name="email" id="email" value="<?php echo $form\_data\_email; ?>" required>

    </div>

    <div class="form-group">

        <label for="password">Password:</label>

        <input type="password" name="password" id="password" required>

    </div>

    <div class="form-group">

        <button type="submit" class="btn">Login</button>

    </div>

</form>

<p>Don't have an account? <a href="register.php">Register here</a>.</p>

<p><a href="register.php">Forgot Password?</a></p>

<?php

require\_once 'footer.php';

mysqli\_close($conn);

?>

**Logout.php**

<?php

require\_once 'config.php';

$\_SESSION = array();

if (ini\_get("session.use\_cookies")) {

    $params = session\_get\_cookie\_params();

    setcookie(session\_name(), '', time() - 42000,

        $params["path"], $params["domain"],

        $params["secure"], $params["httponly"]

    );

}

session\_destroy();

display\_message("You have been logged out successfully.", "success");

redirect('login.php');

?>

**Order\_detail\_user.php**

<?php

require\_once 'header.php';

require\_user\_login();

if (!isset($\_GET['order\_id']) || empty($\_GET['order\_id'])) {

    display\_message("No order specified.", "error");

    redirect('order\_history.php');

}

$order\_id\_get = (int)$\_GET['order\_id'];

$user\_id = $\_SESSION['user\_id'];

$order\_details = null;

$order\_items\_details = [];

$shipping\_address\_details = null;

$sql\_order = "SELECT o.OrderID, o.OrderDate, o.TotalAmount, o.OrderStatus, o.PaymentMethod, o.PaymentStatus, o.TrackingNumber,

                     a.Street, a.City, a.State, a.ZipCode, a.Country

              FROM orders o

              JOIN addresses a ON o.ShippingAddressID = a.AddressID

              WHERE o.OrderID = ? AND o.UserID = ?";

$stmt\_order = mysqli\_prepare($conn, $sql\_order);

if ($stmt\_order) {

    mysqli\_stmt\_bind\_param($stmt\_order, "ii", $order\_id\_get, $user\_id);

    mysqli\_stmt\_execute($stmt\_order);

    $result\_order = mysqli\_stmt\_get\_result($stmt\_order);

    $order\_details = mysqli\_fetch\_assoc($result\_order);

    mysqli\_stmt\_close($stmt\_order);

    if ($order\_details) {

        $sql\_items = "SELECT p.ProductName, p.MainImageURL, oi.Quantity, oi.PriceAtPurchase, oi.Subtotal, p.ProductID

                      FROM order\_items oi

                      JOIN products p ON oi.ProductID = p.ProductID

                      WHERE oi.OrderID = ?";

        $stmt\_items = mysqli\_prepare($conn, $sql\_items);

        if ($stmt\_items) {

            mysqli\_stmt\_bind\_param($stmt\_items, "i", $order\_id\_get);

            mysqli\_stmt\_execute($stmt\_items);

            $result\_items = mysqli\_stmt\_get\_result($stmt\_items);

            while ($row = mysqli\_fetch\_assoc($result\_items)) {

                $order\_items\_details[] = $row;

            }

            mysqli\_stmt\_close($stmt\_items);

        }

    } else {

        display\_message("Order not found or you do not have permission to view it.", "error");

        redirect('order\_history.php');

    }

} else {

    display\_message("Error fetching order details: " . mysqli\_error($conn), "error");

    redirect('order\_history.php');

}

?>

<h2>Order Details #<?php echo htmlspecialchars($order\_details['OrderID']); ?></h2>

<div style="display: flex; flex-wrap: wrap; gap: 30px;">

    <div style="flex: 1; min-width: 280px; background-color: #f9f9f9; padding:15px; border-radius:5px;">

        <h4>Order Information</h4>

        <p><strong>Order Date:</strong> <?php echo date("F j, Y, g:i a", strtotime($order\_details['OrderDate'])); ?></p>

        <p><strong>Total Amount:</strong> ₹<?php echo number\_format($order\_details['TotalAmount'], 2); ?></p>

        <p><strong>Payment Method:</strong> <?php echo htmlspecialchars($order\_details['PaymentMethod']); ?></p>

        <p><strong>Payment Status:</strong> <?php echo htmlspecialchars($order\_details['PaymentStatus']); ?></p>

        <p><strong>Order Status:</strong> <?php echo htmlspecialchars($order\_details['OrderStatus']); ?></p>

        <?php if (!empty($order\_details['TrackingNumber'])): ?>

            <p><strong>Tracking Number:</strong> <?php echo htmlspecialchars($order\_details['TrackingNumber']); ?></p>

        <?php endif; ?>

    </div>

    <div style="flex: 1; min-width: 280px; background-color: #f9f9f9; padding:15px; border-radius:5px;">

        <h4>Shipping Address</h4>

        <p>

            <?php echo htmlspecialchars($order\_details['Street']); ?><br>

            <?php echo htmlspecialchars($order\_details['City']); ?>, <?php echo htmlspecialchars($order\_details['State']); ?> - <?php echo htmlspecialchars($order\_details['ZipCode']); ?><br>

            <?php echo htmlspecialchars($order\_details['Country']); ?>

        </p>

    </div>

</div>

<h3 style="margin-top:30px;">Items in this Order</h3>

<?php if (!empty($order\_items\_details)): ?>

    <table class="cart-table" style="margin-top:10px;">

        <thead>

            <tr>

                <th>Product</th>

                <th>Name</th>

                <th>Quantity</th>

                <th>Price Paid</th>

                <th>Subtotal</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($order\_items\_details as $item): ?>

            <tr>

                <td>

                    <a href="product\_detail.php?id=<?php echo $item['ProductID']; ?>">

                        <img src="<?php echo UPLOAD\_URL . htmlspecialchars($item['MainImageURL']); ?>" alt="<?php echo htmlspecialchars($item['ProductName']); ?>" style="width:60px; height:60px; object-fit:cover;">

                    </a>

                </td>

                <td>

                     <a href="product\_detail.php?id=<?php echo $item['ProductID']; ?>" style="text-decoration:none; color:#333;">

                        <?php echo htmlspecialchars($item['ProductName']); ?>

                    </a>

                </td>

                <td><?php echo $item['Quantity']; ?></td>

                <td>₹<?php echo number\_format($item['PriceAtPurchase'], 2); ?></td>

                <td>₹<?php echo number\_format($item['Subtotal'], 2); ?></td>

            </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

<?php else: ?>

    <p class="message-info">No items found for this order. This should not happen.</p>

<?php endif; ?>

<p style="margin-top: 30px;">

    <a href="order\_history.php" class="btn btn-secondary">&laquo; Back to Order History</a>

</p>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**Order\_history.php**

<?php

require\_once 'header.php';

require\_user\_login();

$user\_id = $\_SESSION['user\_id'];

$orders = [];

$sql = "SELECT OrderID, OrderDate, TotalAmount, OrderStatus, PaymentMethod

        FROM orders

        WHERE UserID = ?

        ORDER BY OrderDate DESC";

$stmt = mysqli\_prepare($conn, $sql);

if ($stmt) {

    mysqli\_stmt\_bind\_param($stmt, "i", $user\_id);

    mysqli\_stmt\_execute($stmt);

    $result = mysqli\_stmt\_get\_result($stmt);

    while ($row = mysqli\_fetch\_assoc($result)) {

        $orders[] = $row;

    }

    mysqli\_stmt\_close($stmt);

} else {

    display\_message("Error fetching order history: " . mysqli\_error($conn), "error");

}

?>

<h2>My Order History</h2>

<?php if (empty($orders)): ?>

    <p class="message-info">You have not placed any orders yet. <a href="products.php">Start shopping!</a></p>

<?php else: ?>

    <table class="cart-table" style="width:100%;"> <thead>

            <tr>

                <th>Order ID</th>

                <th>Order Date</th>

                <th>Total Amount</th>

                <th>Payment Method</th>

                <th>Status</th>

                <th>Action</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($orders as $order): ?>

                <tr>

                    <td>#<?php echo htmlspecialchars($order['OrderID']); ?></td>

                    <td><?php echo date("M j, Y, g:i a", strtotime($order['OrderDate'])); ?></td>

                    <td>₹<?php echo number\_format($order['TotalAmount'], 2); ?></td>

                    <td><?php echo htmlspecialchars($order['PaymentMethod']); ?></td>

                    <td><?php echo htmlspecialchars($order['OrderStatus']); ?></td>

                    <td>

                        <a href="order\_detail\_user.php?order\_id=<?php echo $order['OrderID']; ?>" class="btn btn-secondary" style="font-size:0.9em; padding: 5px 10px;">View Details</a>

                    </td>

                </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

<?php endif; ?>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**Order\_success.php**

<?php

require\_once 'header.php';

require\_user\_login();

if (!isset($\_SESSION['last\_order\_id'])) {

    display\_message("No recent order found. View your order history for details.", "info");

    redirect('order\_history.php');

}

$last\_order\_id = $\_SESSION['last\_order\_id'];

unset($\_SESSION['last\_order\_id']);

$order\_details = null;

$order\_items\_details = [];

$sql\_order = "SELECT OrderID, OrderDate, TotalAmount, PaymentMethod, OrderStatus

              FROM orders

              WHERE OrderID = ? AND UserID = ?";

$stmt\_order = mysqli\_prepare($conn, $sql\_order);

if ($stmt\_order) {

    mysqli\_stmt\_bind\_param($stmt\_order, "ii", $last\_order\_id, $\_SESSION['user\_id']);

    mysqli\_stmt\_execute($stmt\_order);

    $result\_order = mysqli\_stmt\_get\_result($stmt\_order);

    $order\_details = mysqli\_fetch\_assoc($result\_order);

    mysqli\_stmt\_close($stmt\_order);

    if ($order\_details) {

        $sql\_items = "SELECT p.ProductName, oi.Quantity, oi.PriceAtPurchase, oi.Subtotal

                      FROM order\_items oi

                      JOIN products p ON oi.ProductID = p.ProductID

                      WHERE oi.OrderID = ?";

        $stmt\_items = mysqli\_prepare($conn, $sql\_items);

        if ($stmt\_items) {

            mysqli\_stmt\_bind\_param($stmt\_items, "i", $last\_order\_id);

            mysqli\_stmt\_execute($stmt\_items);

            $result\_items = mysqli\_stmt\_get\_result($stmt\_items);

            while ($row = mysqli\_fetch\_assoc($result\_items)) {

                $order\_items\_details[] = $row;

            }

            mysqli\_stmt\_close($stmt\_items);

        }

    }

}

?>

<h2>Order Placed Successfully!</h2>

<?php if ($order\_details): ?>

    <div class="message-success">

        <p>Thank you for your order, <?php echo htmlspecialchars($\_SESSION['user\_first\_name']); ?>!</p>

        <p>Your Order ID is: <strong>#<?php echo htmlspecialchars($order\_details['OrderID']); ?></strong></p>

        <p>Order Date: <?php echo date("F j, Y, g:i a", strtotime($order\_details['OrderDate'])); ?></p>

        <p>Total Amount: ₹<?php echo number\_format($order\_details['TotalAmount'], 2); ?></p>

        <p>Payment Method: <?php echo htmlspecialchars($order\_details['PaymentMethod']); ?></p>

        <p>Current Status: <?php echo htmlspecialchars($order\_details['OrderStatus']); ?></p>

        <p>A confirmation (simulated) has been sent to your email.</p>

    </div>

    <?php if (!empty($order\_items\_details)): ?>

        <h3>Order Summary:</h3>

        <table class="cart-table" style="margin-bottom: 20px;">

            <thead>

                <tr>

                    <th>Product Name</th>

                    <th>Quantity</th>

                    <th>Price Paid</th>

                    <th>Subtotal</th>

                </tr>

            </thead>

            <tbody>

                <?php foreach ($order\_items\_details as $item): ?>

                <tr>

                    <td><?php echo htmlspecialchars($item['ProductName']); ?></td>

                    <td><?php echo $item['Quantity']; ?></td>

                    <td>₹<?php echo number\_format($item['PriceAtPurchase'], 2); ?></td>

                    <td>₹<?php echo number\_format($item['Subtotal'], 2); ?></td>

                </tr>

                <?php endforeach; ?>

            </tbody>

        </table>

    <?php endif; ?>

<?php else: ?>

    <p class="message-info">Thank you for your order! You can view your order details in your <a href="order\_history.php">order history</a>.</p>

<?php endif; ?>

<p style="margin-top: 20px;">

    <a href="products.php" class="btn btn-secondary">Continue Shopping</a>

    <a href="order\_history.php" class="btn">View Order History</a>

</p>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**Privacy.php**

<?php

require\_once 'header.php';

$page\_title = "Privacy Policy";

?>

<h1>Privacy Policy</h1>

    <p>This Privacy Policy outlines how we handle your personal information on our Online Shopping Portal. This website is developed solely for academic purposes as part of a BCA project.</p>

    <h4>1. Information Collection</h4>

    <p>We may collect basic user information such as name, email address, and contact details for demonstration purposes only.</p>

    <h4>2. Use of Information</h4>

    <p>All collected information is used only to simulate a functional eCommerce platform. No real transactions or data sharing occur.</p>

    <h4>3. Data Security</h4>

    <p>This is a demo website. While basic security practices may be implemented, it is not intended for real-world commercial use.</p>

    <h4>4. Third-Party Sharing</h4>

    <p>We do not share any information with third parties. All data remains local and is used for academic testing only.</p>

    <h4>5. Contact</h4>

    <p>If you have any questions about this privacy policy, you may contact the project developer.</p>

    <p><em>Last updated: May 3, 2025</em></p>

<?php

require\_once 'footer.php';

?>

**Product\_detail.php**

<?php

require\_once 'header.php';

if (!isset($\_GET['id']) || empty($\_GET['id'])) {

    display\_message("No product specified.", "error");

    redirect('products.php');

}

$product\_id = (int)$\_GET['id'];

$sql = "SELECT p.ProductID, p.ProductName, p.Description, p.MRP, p.SellingPrice,

               p.StockQuantity, p.MainImageURL, c.CategoryName, c.CategoryID

        FROM products p

        JOIN categories c ON p.CategoryID = c.CategoryID

        WHERE p.ProductID = ? AND p.IsActive = TRUE";

$stmt = mysqli\_prepare($conn, $sql);

if ($stmt) {

    mysqli\_stmt\_bind\_param($stmt, "i", $product\_id);

    mysqli\_stmt\_execute($stmt);

    $result = mysqli\_stmt\_get\_result($stmt);

    $product = mysqli\_fetch\_assoc($result);

    mysqli\_stmt\_close($stmt);

    if (!$product) {

        display\_message("Product not found or is inactive.", "error");

        redirect('products.php');

    }

} else {

    display\_message("Error fetching product details: " . mysqli\_error($conn), "error");

    redirect('products.php');

}

?>

<div class="product-detail-layout">

    <div class="product-detail-image">

        <img src="<?php echo UPLOAD\_URL . htmlspecialchars($product['MainImageURL']); ?>" alt="<?php echo htmlspecialchars($product['ProductName']); ?>">

    </div>

    <div class="product-detail-info">

        <h1><?php echo htmlspecialchars($product['ProductName']); ?></h1>

        <p class="category">Category: <a href="<?php echo BASE\_URL; ?>products.php?category\_id=<?php echo $product['CategoryID']; ?>"><?php echo htmlspecialchars($product['CategoryName']); ?></a></p>

        <p class="price">

            ₹<?php echo number\_format($product['SellingPrice'], 2); ?>

            <?php if ($product['MRP'] > $product['SellingPrice']): ?>

                <span class="mrp">₹<?php echo number\_format($product['MRP'], 2); ?></span>

                <?php

                $discount = (($product['MRP'] - $product['SellingPrice']) / $product['MRP']) \* 100;

                echo " <span style='color:green; font-size:0.9em;'>(" . round($discount) . "% off)</span>";

                ?>

            <?php endif; ?>

        </p>

        <p class="stock <?php echo ($product['StockQuantity'] > 0) ? 'in-stock' : 'out-of-stock'; ?>">

            <?php echo ($product['StockQuantity'] > 0) ? 'In Stock (' . $product['StockQuantity'] . ' available)' : 'Out of Stock'; ?>

        </p>

        <div class="description" style="margin-top: 20px; margin-bottom: 20px;">

            <h4>Product Description:</h4>

            <p><?php echo nl2br(htmlspecialchars($product['Description'])); ?></p>

        </div>

        <?php if ($product['StockQuantity'] > 0): ?>

            <form action="<?php echo BASE\_URL; ?>cart\_actions.php" method="POST">

                <input type="hidden" name="product\_id" value="<?php echo $product['ProductID']; ?>">

                <div class="form-group" style="max-width: 150px;">

                    <label for="quantity">Quantity:</label>

                    <input type="number" name="quantity" id="quantity" value="1" min="1" max="<?php echo $product['StockQuantity']; ?>" class="form-control">

                </div>

                <button type="submit" name="add\_to\_cart" class="btn">Add to Cart</button>

            </form>

        <?php else: ?>

            <p class="message-info">This product is currently out of stock.</p>

        <?php endif; ?>

    </div>

</div>

<?php

if(isset($conn)) mysqli\_close($conn);

require\_once 'footer.php';

?>

**Products.php**

<?php

require\_once 'header.php';

$category\_id = isset($\_GET['category\_id']) ? (int)$\_GET['category\_id'] : 0;

$search\_term = isset($\_GET['search']) ? sanitize\_input($\_GET['search']) : '';

$page\_title = "Our Products";

$sql = "SELECT p.ProductID, p.ProductName, p.SellingPrice, p.MRP, p.MainImageURL, c.CategoryName

        FROM products p

        JOIN categories c ON p.CategoryID = c.CategoryID

        WHERE p.IsActive = TRUE";

if ($category\_id > 0) {

    $cat\_sql = "SELECT CategoryName FROM categories WHERE CategoryID = ?";

    if ($cat\_stmt = mysqli\_prepare($conn, $cat\_sql)) {

        mysqli\_stmt\_bind\_param($cat\_stmt, "i", $category\_id);

        mysqli\_stmt\_execute($cat\_stmt);

        mysqli\_stmt\_bind\_result($cat\_stmt, $category\_name\_title);

        if (mysqli\_stmt\_fetch($cat\_stmt)) {

            $page\_title = "Products in " . htmlspecialchars($category\_name\_title);

        }

        mysqli\_stmt\_close($cat\_stmt);

    }

    $sql .= " AND (p.CategoryID = ? OR p.CategoryID IN (SELECT CategoryID FROM categories WHERE ParentCategoryID = ?))";

}

if (!empty($search\_term)) {

    $page\_title = "Search Results for \"" . htmlspecialchars($search\_term) . "\"";

    $sql .= " AND (p.ProductName LIKE ? OR p.Description LIKE ? OR c.CategoryName LIKE ?)";

}

$sql .= " ORDER BY p.DateAdded DESC";

$stmt = mysqli\_prepare($conn, $sql);

if ($stmt) {

    if ($category\_id > 0 && !empty($search\_term)) {

        $search\_like = "%" . $search\_term . "%";

        mysqli\_stmt\_bind\_param($stmt, "iisss", $category\_id, $category\_id, $search\_like, $search\_like, $search\_like);

    } elseif ($category\_id > 0) {

        mysqli\_stmt\_bind\_param($stmt, "ii", $category\_id, $category\_id);

    } elseif (!empty($search\_term)) {

        $search\_like = "%" . $search\_term . "%";

        mysqli\_stmt\_bind\_param($stmt, "sss", $search\_like, $search\_like, $search\_like);

    }

    mysqli\_stmt\_execute($stmt);

    $result = mysqli\_stmt\_get\_result($stmt);

} else {

    echo "<p class='message-error'>Error preparing product query: " . mysqli\_error($conn) . "</p>";

    $result = false;

}

$categories\_sql = "SELECT CategoryID, CategoryName FROM categories WHERE ParentCategoryID IS NULL ORDER BY CategoryName ASC";

$categories\_result = mysqli\_query($conn, $categories\_sql);

?>

<h2><?php echo $page\_title; ?></h2>

<form action="<?php echo BASE\_URL; ?>products.php" method="GET" class="search-form" style="margin-bottom: 20px; display: flex; gap: 10px;">

    <input type="text" name="search" placeholder="Search products..." value="<?php echo htmlspecialchars($search\_term); ?>" style="flex-grow: 1; padding: 8px; border: 1px solid #ccc; border-radius: 4px;">

    <button type="submit" class="btn">Search</button>

</form>

<div style="display: flex; gap: 20px;">

    <?php if ($categories\_result && mysqli\_num\_rows($categories\_result) > 0): ?>

    <aside style="width: 200px; border-right: 1px solid #eee; padding-right: 20px;">

        <h4>Categories</h4>

        <ul style="list-style: none; padding: 0;">

            <li><a href="<?php echo BASE\_URL; ?>products.php" style="text-decoration: none; color: #333; display:block; padding: 5px 0;">All Products</a></li>

            <?php while($category = mysqli\_fetch\_assoc($categories\_result)): ?>

                <li>

                    <a href="<?php echo BASE\_URL; ?>products.php?category\_id=<?php echo $category['CategoryID']; ?>" style="text-decoration: none; color: #0779e4; display:block; padding: 5px 0;">

                        <?php echo htmlspecialchars($category['CategoryName']); ?>

                    </a>

                </li>

            <?php endwhile; ?>

        </ul>

    </aside>

    <?php endif; ?>

    <section class="product-grid" style="flex-grow: 1;">

        <?php

        if ($result && mysqli\_num\_rows($result) > 0) {

            while ($product = mysqli\_fetch\_assoc($result)) {

                ?>

                <div class="product-card">

                    <a href="<?php echo BASE\_URL; ?>product\_detail.php?id=<?php echo $product['ProductID']; ?>" style="text-decoration:none; color:inherit;">

                        <img src="<?php echo UPLOAD\_URL . htmlspecialchars($product['MainImageURL']); ?>" alt="<?php echo htmlspecialchars($product['ProductName']); ?>">

                        <h3><?php echo htmlspecialchars($product['ProductName']); ?></h3>

                        <p class="category" style="font-size:0.9em; color:#777;"><?php echo htmlspecialchars($product['CategoryName']); ?></p>

                        <p class="price">

                            ₹<?php echo number\_format($product['SellingPrice'], 2); ?>

                            <?php if ($product['MRP'] > $product['SellingPrice']): ?>

                                <span class="mrp">₹<?php echo number\_format($product['MRP'], 2); ?></span>

                            <?php endif; ?>

                        </p>

                    </a>

                    <form action="<?php echo BASE\_URL; ?>cart\_actions.php" method="POST">

                        <input type="hidden" name="product\_id" value="<?php echo $product['ProductID']; ?>">

                        <input type="hidden" name="quantity" value="1">

                        <button type="submit" name="add\_to\_cart" class="btn">Add to Cart</button>

                    </form>

                </div>

                <?php

            }

        } else {

            if (empty($search\_term) && $category\_id == 0) {

                 echo "<p>No products found. Please check back later or contact support if you believe this is an error.</p>";

            } elseif (!empty($search\_term)) {

                 echo "<p>No products found matching your search term: \"" . htmlspecialchars($search\_term) . "\". Try a different search or browse categories.</p>";

            } else {

                 echo "<p>No products found in this category. Try browsing other categories or all products.</p>";

            }

        }

        if ($stmt) {

            mysqli\_stmt\_close($stmt);

        }

        ?>

    </section>

</div>

<?php

if(isset($conn)) mysqli\_close($conn);

require\_once 'footer.php';

?>

**Profile.php**

<?php

require\_once 'header.php';

require\_user\_login();

$user\_id = $\_SESSION['user\_id'];

$current\_user = null;

$addresses = [];

$update\_errors = [];

$update\_success\_msg = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['update\_profile'])) {

    $first\_name = sanitize\_input($\_POST['first\_name']);

    $last\_name = sanitize\_input($\_POST['last\_name']);

    $contact\_number = sanitize\_input($\_POST['contact\_number']);

    if (empty($first\_name)) $update\_errors[] = "First name is required.";

    if (empty($last\_name)) $update\_errors[] = "Last name is required.";

    if (empty($update\_errors)) {

        $sql\_update = "UPDATE users SET FirstName = ?, LastName = ?, ContactNumber = ? WHERE UserID = ?";

        $stmt\_update = mysqli\_prepare($conn, $sql\_update);

        if ($stmt\_update) {

            mysqli\_stmt\_bind\_param($stmt\_update, "sssi", $first\_name, $last\_name, $contact\_number, $user\_id);

            if (mysqli\_stmt\_execute($stmt\_update)) {

                $update\_success\_msg = "Profile updated successfully!";

                $\_SESSION['user\_first\_name'] = $first\_name;

            } else {

                $update\_errors[] = "Failed to update profile. Database error.";

            }

            mysqli\_stmt\_close($stmt\_update);

        } else {

            $update\_errors[] = "Failed to prepare profile update. Database error.";

        }

    }

}

$pw\_change\_errors = [];

$pw\_change\_success\_msg = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['change\_password'])) {

    $current\_password = $\_POST['current\_password'];

    $new\_password = $\_POST['new\_password'];

    $confirm\_new\_password = $\_POST['confirm\_new\_password'];

    if (empty($current\_password)) $pw\_change\_errors[] = "Current password is required.";

    if (empty($new\_password)) $pw\_change\_errors[] = "New password is required.";

    elseif (strlen($new\_password) < 6) $pw\_change\_errors[] = "New password must be at least 6 characters long.";

    if ($new\_password !== $confirm\_new\_password) $pw\_change\_errors[] = "New passwords do not match.";

    if (empty($pw\_change\_errors)) {

        $sql\_pw = "SELECT PasswordHash FROM users WHERE UserID = ?";

        $stmt\_pw = mysqli\_prepare($conn, $sql\_pw);

        mysqli\_stmt\_bind\_param($stmt\_pw, "i", $user\_id);

        mysqli\_stmt\_execute($stmt\_pw);

        mysqli\_stmt\_bind\_result($stmt\_pw, $current\_password\_hash);

        mysqli\_stmt\_fetch($stmt\_pw);

        mysqli\_stmt\_close($stmt\_pw);

        if (password\_verify($current\_password, $current\_password\_hash)) {

            $new\_password\_hash = password\_hash($new\_password, PASSWORD\_DEFAULT);

            $sql\_update\_pw = "UPDATE users SET PasswordHash = ? WHERE UserID = ?";

            $stmt\_update\_pw = mysqli\_prepare($conn, $sql\_update\_pw);

            mysqli\_stmt\_bind\_param($stmt\_update\_pw, "si", $new\_password\_hash, $user\_id);

            if (mysqli\_stmt\_execute($stmt\_update\_pw)) {

                $pw\_change\_success\_msg = "Password changed successfully!";

            } else {

                $pw\_change\_errors[] = "Failed to change password. Database error.";

            }

            mysqli\_stmt\_close($stmt\_update\_pw);

        } else {

            $pw\_change\_errors[] = "Incorrect current password.";

        }

    }

}

$address\_action\_msg = '';

$address\_action\_type = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['add\_address'])) {

    $street = sanitize\_input($\_POST['street']);

    $city = sanitize\_input($\_POST['city']);

    $state = sanitize\_input($\_POST['state']);

    $zipcode = sanitize\_input($\_POST['zipcode']);

    $country = sanitize\_input($\_POST['country'] ?? 'India');

    if (empty($street) || empty($city) || empty($state) || empty($zipcode)) {

        $address\_action\_msg = "All address fields (Street, City, State, Zipcode) are required.";

        $address\_action\_type = 'error';

    } else {

        $sql\_add\_addr = "INSERT INTO addresses (UserID, Street, City, State, ZipCode, Country) VALUES (?, ?, ?, ?, ?, ?)";

        $stmt\_add\_addr = mysqli\_prepare($conn, $sql\_add\_addr);

        mysqli\_stmt\_bind\_param($stmt\_add\_addr, "isssss", $user\_id, $street, $city, $state, $zipcode, $country);

        if (mysqli\_stmt\_execute($stmt\_add\_addr)) {

            $address\_action\_msg = "Address added successfully!";

            $address\_action\_type = 'success';

        } else {

            $address\_action\_msg = "Failed to add address. Database error.";

            $address\_action\_type = 'error';

        }

        mysqli\_stmt\_close($stmt\_add\_addr);

    }

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['delete\_address'])) {

    $address\_id\_to\_delete = (int)$\_POST['address\_id'];

    $sql\_delete\_addr = "DELETE FROM addresses WHERE AddressID = ? AND UserID = ?";

    $stmt\_delete\_addr = mysqli\_prepare($conn, $sql\_delete\_addr);

    mysqli\_stmt\_bind\_param($stmt\_delete\_addr, "ii", $address\_id\_to\_delete, $user\_id);

    if (mysqli\_stmt\_execute($stmt\_delete\_addr)) {

        if (mysqli\_stmt\_affected\_rows($stmt\_delete\_addr) > 0) {

            $address\_action\_msg = "Address deleted successfully!";

            $address\_action\_type = 'success';

        } else {

            $address\_action\_msg = "Address not found or you do not have permission to delete it.";

            $address\_action\_type = 'error';

        }

    } else {

        $address\_action\_msg = "Failed to delete address. Database error.";

        $address\_action\_type = 'error';

    }

    mysqli\_stmt\_close($stmt\_delete\_addr);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['set\_default\_address'])) {

    $address\_id\_to\_default = (int)$\_POST['address\_id'];

    mysqli\_begin\_transaction($conn);

    try {

        $sql\_unset\_default = "UPDATE addresses SET IsDefaultShipping = FALSE WHERE UserID = ? AND IsDefaultShipping = TRUE";

        $stmt\_unset = mysqli\_prepare($conn, $sql\_unset\_default);

        mysqli\_stmt\_bind\_param($stmt\_unset, "i", $user\_id);

        mysqli\_stmt\_execute($stmt\_unset);

        mysqli\_stmt\_close($stmt\_unset);

        $sql\_set\_default = "UPDATE addresses SET IsDefaultShipping = TRUE WHERE AddressID = ? AND UserID = ?";

        $stmt\_set = mysqli\_prepare($conn, $sql\_set\_default);

        mysqli\_stmt\_bind\_param($stmt\_set, "ii", $address\_id\_to\_default, $user\_id);

        mysqli\_stmt\_execute($stmt\_set);

        if (mysqli\_stmt\_affected\_rows($stmt\_set) > 0) {

            mysqli\_commit($conn);

            $address\_action\_msg = "Default shipping address updated!";

            $address\_action\_type = 'success';

        } else {

            throw new Exception("Address not found or permission denied.");

        }

        mysqli\_stmt\_close($stmt\_set);

    } catch (Exception $e) {

        mysqli\_rollback($conn);

        $address\_action\_msg = "Failed to set default address: " . $e->getMessage();

        $address\_action\_type = 'error';

    }

}

$sql\_user = "SELECT UserID, FirstName, LastName, Email, ContactNumber, RegistrationDate FROM users WHERE UserID = ?";

$stmt\_user = mysqli\_prepare($conn, $sql\_user);

if ($stmt\_user) {

    mysqli\_stmt\_bind\_param($stmt\_user, "i", $user\_id);

    mysqli\_stmt\_execute($stmt\_user);

    $result\_user = mysqli\_stmt\_get\_result($stmt\_user);

    $current\_user = mysqli\_fetch\_assoc($result\_user);

    mysqli\_stmt\_close($stmt\_user);

}

$addresses = [];

$sql\_addresses = "SELECT AddressID, Street, City, State, ZipCode, Country, IsDefaultShipping FROM addresses WHERE UserID = ? ORDER BY IsDefaultShipping DESC, AddressID DESC";

$stmt\_addresses = mysqli\_prepare($conn, $sql\_addresses);

if ($stmt\_addresses) {

    mysqli\_stmt\_bind\_param($stmt\_addresses, "i", $user\_id);

    mysqli\_stmt\_execute($stmt\_addresses);

    $result\_addresses = mysqli\_stmt\_get\_result($stmt\_addresses);

    while ($row = mysqli\_fetch\_assoc($result\_addresses)) {

        $addresses[] = $row;

    }

    mysqli\_stmt\_close($stmt\_addresses);

}

if (!$current\_user) {

    display\_message("Could not retrieve user profile.", "error");

    redirect("index.php");

}

?>

<h2>My Profile</h2>

<?php

if (!empty($update\_success\_msg)) echo "<div class='message-success'>{$update\_success\_msg}</div>";

if (!empty($update\_errors)) {

    echo "<div class='message-error'><ul>";

    foreach ($update\_errors as $err) echo "<li>{$err}</li>";

    echo "</ul></div>";

}

if (!empty($pw\_change\_success\_msg)) echo "<div class='message-success'>{$pw\_change\_success\_msg}</div>";

if (!empty($pw\_change\_errors)) {

    echo "<div class='message-error'><ul>";

    foreach ($pw\_change\_errors as $err) echo "<li>{$err}</li>";

    echo "</ul></div>";

}

if (!empty($address\_action\_msg)) {

    echo "<div class='message-{$address\_action\_type}'>{$address\_action\_msg}</div>";

}

?>

<div style="display: flex; flex-wrap: wrap; gap: 30px;">

    <div style="flex: 1; min-width: 300px;">

        <h3>Account Details</h3>

        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

            <div class="form-group">

                <label for="email">Email (Username):</label>

                <input type="email" id="email" value="<?php echo htmlspecialchars($current\_user['Email']); ?>" readonly disabled style="background-color:#e9ecef;">

            </div>

            <div class="form-group">

                <label for="first\_name">First Name:</label>

                <input type="text" name="first\_name" id="first\_name" value="<?php echo htmlspecialchars($current\_user['FirstName']); ?>" required>

            </div>

            <div class="form-group">

                <label for="last\_name">Last Name:</label>

                <input type="text" name="last\_name" id="last\_name" value="<?php echo htmlspecialchars($current\_user['LastName']); ?>" required>

            </div>

            <div class="form-group">

                <label for="contact\_number">Contact Number:</label>

                <input type="text" name="contact\_number" id="contact\_number" value="<?php echo htmlspecialchars($current\_user['ContactNumber'] ?? ''); ?>">

            </div>

            <div class="form-group">

                <button type="submit" name="update\_profile" class="btn">Update Profile</button>

            </div>

        </form>

        <hr style="margin: 30px 0;">

        <h3>Change Password</h3>

        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

            <div class="form-group">

                <label for="current\_password">Current Password:</label>

                <input type="password" name="current\_password" id="current\_password" required>

            </div>

            <div class="form-group">

                <label for="new\_password">New Password:</label>

                <input type="password" name="new\_password" id="new\_password" required>

                <small>Must be at least 6 characters.</small>

            </div>

            <div class="form-group">

                <label for="confirm\_new\_password">Confirm New Password:</label>

                <input type="password" name="confirm\_new\_password" id="confirm\_new\_password" required>

            </div>

            <div class="form-group">

                <button type="submit" name="change\_password" class="btn">Change Password</button>

            </div>

        </form>

    </div>

    <div style="flex: 1; min-width: 300px;" id="addresses">

        <h3>My Shipping Addresses</h3>

        <?php if (!empty($addresses)): ?>

            <ul style="list-style: none; padding: 0;">

                <?php foreach ($addresses as $address): ?>

                    <li style="border: 1px solid #ddd; padding: 15px; margin-bottom: 10px; border-radius: 5px; background-color: #f9f9f9;">

                        <p>

                            <?php echo htmlspecialchars($address['Street']); ?><br>

                            <?php echo htmlspecialchars($address['City']); ?>, <?php echo htmlspecialchars($address['State']); ?> - <?php echo htmlspecialchars($address['ZipCode']); ?><br>

                            <?php echo htmlspecialchars($address['Country']); ?>

                            <?php if ($address['IsDefaultShipping']) echo " <strong>(Default)</strong>"; ?>

                        </p>

                        <div style="margin-top: 10px;">

                            <?php if (!$address['IsDefaultShipping']): ?>

                            <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>#addresses" method="POST" style="display: inline-block; margin-right: 5px;">

                                <input type="hidden" name="address\_id" value="<?php echo $address['AddressID']; ?>">

                                <button type="submit" name="set\_default\_address" class="btn btn-secondary" style="font-size:0.9em; padding: 5px 10px;">Set as Default</button>

                            </form>

                            <?php endif; ?>

                            <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>#addresses" method="POST" style="display: inline-block;">

                                <input type="hidden" name="address\_id" value="<?php echo $address['AddressID']; ?>">

                                <button type="submit" name="delete\_address" class="btn btn-danger" style="font-size:0.9em; padding: 5px 10px;" onclick="return confirm('Are you sure you want to delete this address?');">Delete</button>

                            </form>

                        </div>

                    </li>

                <?php endforeach; ?>

            </ul>

        <?php else: ?>

            <p>You have no saved shipping addresses.</p>

        <?php endif; ?>

        <hr style="margin: 30px 0;">

        <h4 id="add-address">Add New Address</h4>

        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>#add-address" method="POST">

            <div class="form-group">

                <label for="street">Street Address:</label>

                <input type="text" name="street" id="street" required>

            </div>

            <div class="form-group">

                <label for="city">City:</label>

                <input type="text" name="city" id="city" required>

            </div>

            <div class="form-group">

                <label for="state">State:</label>

                <input type="text" name="state" id="state" required>

            </div>

            <div class="form-group">

                <label for="zipcode">Zip Code:</label>

                <input type="text" name="zipcode" id="zipcode" required>

            </div>

            <div class="form-group">

                <label for="country">Country:</label>

                <input type="text" name="country" id="country" value="India"> </div>

            <div class="form-group">

                <button type="submit" name="add\_address" class="btn">Add Address</button>

            </div>

        </form>

    </div>

</div>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**Register.php**

<?php

require\_once 'config.php';

if (is\_user\_logged\_in()) {

    redirect('index.php');

}

$first\_name = $last\_name = $email = $contact\_number = $password = $confirm\_password = "";

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $first\_name = sanitize\_input($\_POST['first\_name']);

    $last\_name = sanitize\_input($\_POST['last\_name']);

    $email = sanitize\_input($\_POST['email']);

    $contact\_number = sanitize\_input($\_POST['contact\_number']);

    $password = $\_POST['password'];

    $confirm\_password = $\_POST['confirm\_password'];

    if (empty($first\_name)) {

        $errors[] = "First name is required.";

    }

    if (empty($last\_name)) {

        $errors[] = "Last name is required.";

    }

    if (empty($email)) {

        $errors[] = "Email is required.";

    } elseif (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

        $errors[] = "Invalid email format.";

    }

    if (empty($password)) {

        $errors[] = "Password is required.";

    } elseif (strlen($password) < 6) {

        $errors[] = "Password must be at least 6 characters long.";

    }

    if ($password !== $confirm\_password) {

        $errors[] = "Passwords do not match.";

    }

    if (empty($errors)) {

        $sql\_check\_email = "SELECT UserID FROM users WHERE Email = ?";

        if ($stmt\_check\_email = mysqli\_prepare($conn, $sql\_check\_email)) {

            mysqli\_stmt\_bind\_param($stmt\_check\_email, "s", $param\_email);

            $param\_email = $email;

            if (mysqli\_stmt\_execute($stmt\_check\_email)) {

                mysqli\_stmt\_store\_result($stmt\_check\_email);

                if (mysqli\_stmt\_num\_rows($stmt\_check\_email) == 1) {

                    $errors[] = "This email address is already registered.";

                }

            } else {

                $errors[] = "Oops! Something went wrong. Please try again later. (Email check failed)";

            }

            mysqli\_stmt\_close($stmt\_check\_email);

        } else {

            $errors[] = "Database error. Please try again later. (Prepare failed)";

        }

    }

    if (empty($errors)) {

        $sql\_insert\_user = "INSERT INTO users (FirstName, LastName, Email, PasswordHash, ContactNumber) VALUES (?, ?, ?, ?, ?)";

        if ($stmt\_insert\_user = mysqli\_prepare($conn, $sql\_insert\_user)) {

            mysqli\_stmt\_bind\_param($stmt\_insert\_user, "sssss", $param\_first\_name, $param\_last\_name, $param\_email, $param\_password\_hash, $param\_contact\_number);

            $param\_first\_name = $first\_name;

            $param\_last\_name = $last\_name;

            $param\_email = $email;

            $param\_password\_hash = password\_hash($password, PASSWORD\_DEFAULT);

            $param\_contact\_number = $contact\_number;

            if (mysqli\_stmt\_execute($stmt\_insert\_user)) {

                display\_message("Registration successful! You can now login.", "success");

                redirect('login.php');

            } else {

                display\_message("Something went wrong. Please try again later. (Registration failed)", "error");

            }

            mysqli\_stmt\_close($stmt\_insert\_user);

        } else {

             display\_message("Database error. Please try again later. (Prepare insert failed)", "error");

        }

    }

    if (!empty($errors)) {

        $error\_message\_html = "<ul>";

        foreach ($errors as $error) {

            $error\_message\_html .= "<li>" . htmlspecialchars($error) . "</li>";

        }

        $error\_message\_html .= "</ul>";

        $\_SESSION['form\_errors'] = $error\_message\_html;

        $\_SESSION['form\_data'] = $\_POST;

        redirect('register.php');

    }

}

$form\_errors\_html = '';

if (isset($\_SESSION['form\_errors'])) {

    $form\_errors\_html = $\_SESSION['form\_errors'];

    unset($\_SESSION['form\_errors']);

}

$form\_data = [];

if (isset($\_SESSION['form\_data'])) {

    $form\_data = $\_SESSION['form\_data'];

    foreach ($form\_data as $key => $value) {

        if ($key !== 'password' && $key !== 'confirm\_password') {

            $form\_data[$key] = htmlspecialchars($value, ENT\_QUOTES, 'UTF-8');

        } else {

            $form\_data[$key] = '';

        }

    }

    unset($\_SESSION['form\_data']);

}

require\_once 'header.php';

?>

<h2>User Registration</h2>

<?php

if (!empty($form\_errors\_html)) {

    echo "<div class='message-error'>" . $form\_errors\_html . "</div>";

}

?>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="post">

    <div class="form-group">

        <label for="first\_name">First Name:</label>

        <input type="text" name="first\_name" id="first\_name" value="<?php echo $form\_data['first\_name'] ?? ''; ?>" required>

    </div>

    <div class="form-group">

        <label for="last\_name">Last Name:</label>

        <input type="text" name="last\_name" id="last\_name" value="<?php echo $form\_data['last\_name'] ?? ''; ?>" required>

    </div>

    <div class="form-group">

        <label for="email">Email:</label>

        <input type="email" name="email" id="email" value="<?php echo $form\_data['email'] ?? ''; ?>" required>

    </div>

    <div class="form-group">

        <label for="contact\_number">Contact Number (Optional):</label>

        <input type="text" name="contact\_number" id="contact\_number" value="<?php echo $form\_data['contact\_number'] ?? ''; ?>">

    </div>

    <div class="form-group">

        <label for="password">Password:</label>

        <input type="password" name="password" id="password" required>

        <small>Password must be at least 6 characters long.</small>

    </div>

    <div class="form-group">

        <label for="confirm\_password">Confirm Password:</label>

        <input type="password" name="confirm\_password" id="confirm\_password" required>

    </div>

    <div class="form-group">

        <button type="submit" class="btn">Register</button>

    </div>

</form>

<p>Already have an account? <a href="login.php">Login here</a>.</p>

<?php

require\_once 'footer.php';

mysqli\_close($conn);

?>

**Style.css**

:root {

    --primary-color: #006994;

    --primary-color-dark: #004D6B;

    --secondary-color: #87CEEB;

    --secondary-color-dark: #6A9FB5;

    --accent-color: #FFD700;

    --accent-color-dark: #E6C200;

    --light-gray: #F0F8FF;

    --medium-gray: #D1E0E0;

    --dark-gray: #2F4F4F;

    --success-color: #3CB371;

    --success-color-dark: #2E8B57;

    --danger-color: #CD5C5C;

    --danger-color-dark: #B22222;

    --warning-color: #FFA500;

    --info-color: #4682B4;

    --white: #FFFFFF;

    --black: #1C1C1C;

    --border-color: var(--medium-gray);

    --text-color: #333333;

    --link-color: var(--primary-color);

    --font-family-sans-serif: system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", "Liberation Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";

    --box-shadow-sm: 0 .125rem .25rem rgba(0, 0, 0, .075);

    --box-shadow: 0 .5rem 1rem rgba(0, 0, 0, .15);

    --border-radius: .25rem;

    --border-radius-lg: .3rem;

    --border-radius-pill: 50rem;

}

\*,

\*::before,

\*::after {

    box-sizing: border-box;

}

body {

    font-family: var(--font-family-sans-serif);

    margin: 0;

    padding: 0;

    background-color: var(--light-gray);

    color: var(--text-color);

    line-height: 1.6;

    display: flex;

    flex-direction: column;

    min-height: 100vh;

}

main {

    flex-grow: 1;

    padding-top: 20px;

    padding-bottom: 40px;

}

.main-content {

    padding: 25px 20px;

    background: var(--white);

    margin-top: 25px;

    margin-bottom: 25px;

    border-radius: var(--border-radius-lg);

    box-shadow: var(--box-shadow-sm);

    flex: 1 0 auto;

}

.container {

    width: 100%;

    padding-right: 15px;

    padding-left: 15px;

    margin-right: auto;

    margin-left: auto;

}

@media (min-width: 576px) { .container { max-width: 540px; } }

@media (min-width: 768px) { .container { max-width: 720px; } }

@media (min-width: 992px) { .container { max-width: 960px; } }

@media (min-width: 1200px) { .container { max-width: 1140px; } }

.visually-hidden {

  position: absolute !important;

  height: 1px; width: 1px;

  overflow: hidden;

  clip: rect(1px, 1px, 1px, 1px);

  white-space: nowrap;

}

h1, h2, h3, h4, h5, h6 {

    margin-top: 0;

    margin-bottom: 0.5rem;

    font-weight: 600;

    line-height: 1.2;

    color: var(--dark-gray);

}

h1 { font-size: 2.2rem; }

h2 { font-size: 1.8rem; }

h3 { font-size: 1.5rem; }

h4 { font-size: 1.2rem; }

p {

    margin-top: 0;

    margin-bottom: 1rem;

}

a {

    color: var(--link-color);

    text-decoration: none;

}

a:hover {

    color: var(--primary-color-dark);

    text-decoration: underline;

}

/\* --- Buttons --- \*/

.btn, .button-primary,

button {

    display: inline-block;

    font-weight: 400;

    color: var(--white);

    text-align: center;

    vertical-align: middle;

    cursor: pointer;

    user-select: none;

    background-color: var(--primary-color);

    border: 1px solid var(--primary-color);

    padding: .375rem .75rem;

    font-size: 1rem;

    line-height: 1.5;

    border-radius: var(--border-radius);

    text-decoration: none;

}

.btn:hover, .button-primary:hover, button:hover {

    background-color: var(--primary-color-dark);

    border-color: var(--primary-color-dark);

    color: var(--white);

    text-decoration: none;

}

.btn:focus, .button-primary:focus, button:focus {

    outline: 0;

    box-shadow: 0 0 0 .2rem rgba(0, 105, 148, .5);

}

.btn:disabled, .button-primary:disabled, button:disabled {

    background-color: var(--secondary-color);

    border-color: var(--secondary-color);

    opacity: 0.65;

    cursor: not-allowed;

}

.btn-secondary, .button-secondary {

    background-color: var(--secondary-color);

    border-color: var(--secondary-color);

    color: var(--dark-gray);

}

.btn-secondary:hover, .button-secondary:hover {

    background-color: var(--secondary-color-dark);

    border-color: var(--secondary-color-dark);

    color: var(--black);

}

.btn-danger, .button-danger {

    background-color: var(--danger-color);

    border-color: var(--danger-color);

    color: var(--white);

}

.btn-danger:hover, .button-danger:hover {

    background-color: var(--danger-color-dark);

    border-color: var(--danger-color-dark);

    color: var(--white);

}

.btn-success, .button-success {

    background-color: var(--success-color);

    border-color: var(--success-color);

    color: var(--white);

}

.btn-success:hover, .button-success:hover {

    background-color: var(--success-color-dark);

    border-color: var(--success-color-dark);

    color: var(--white);

}

.form-group {

    margin-bottom: 1rem;

}

.form-group label {

    display: inline-block;

    margin-bottom: .5rem;

    font-weight: 500;

    color: var(--dark-gray);

}

.form-group input[type="text"],

.form-group input[type="email"],

.form-group input[type="password"],

.form-group input[type="number"],

.form-group input[type="search"],

.form-group input[type="url"],

.form-group input[type="date"],

.form-group textarea,

.form-group select {

    display: block;

    width: 100%;

    padding: .375rem .75rem;

    font-size: 1rem;

    font-weight: 400;

    line-height: 1.5;

    color: var(--text-color);

    background-color: var(--white);

    border: 1px solid var(--border-color);

    border-radius: var(--border-radius);

}

.form-group input:focus,

.form-group textarea:focus,

.form-group select:focus {

    color: var(--text-color);

    background-color: var(--white);

    border-color: var(--primary-color);

    outline: 0;

    box-shadow: 0 0 0 .2rem rgba(0, 105, 148, .25);

}

.form-group textarea {

    resize: vertical;

    min-height: 100px;

}

.form-group small {

    font-size: 0.8em;

    color: var(--secondary-color-dark);

    display: block;

    margin-top: .25rem;

}

.message-success, .message-error, .message-info, .message-warning {

    padding: .75rem 1.25rem;

    margin-bottom: 1rem;

    border: 1px solid transparent;

    border-radius: var(--border-radius);

}

.message-success { color: var(--success-color-dark); background-color: #e6f6e9; border-color: var(--success-color);}

.message-error { color: var(--danger-color-dark); background-color: #f9eaea; border-color: var(--danger-color);}

.message-info { color: #2E597C; background-color: #e8f0f6; border-color: var(--info-color);}

.message-warning { color: #B37400; background-color: #fff6e0; border-color: var(--warning-color);}

header {

    background-color: var(--dark-gray);

    color: var(--light-gray);

    padding: 10px 0;

    border-bottom: 3px solid var(--primary-color);

    box-shadow: var(--box-shadow-sm);

    position: sticky;

    top: 0;

    z-index: 1020;

}

header .container {

    display: flex;

    justify-content: space-between;

    align-items: center;

    flex-wrap: wrap;

}

header #branding h1 {

    margin: 0;

    color: var(--white);

    font-weight: 700;

    font-size: 1.8rem;

}

header #branding h1 a {

    color: inherit;

    text-decoration: none;

}

header nav ul {

    list-style: none;

    padding: 0;

    margin: 0;

    display: flex;

    align-items: center;

}

header nav ul li {

    margin-left: 20px;

}

header nav ul li a {

    text-decoration: none;

    color: var(--medium-gray);

    font-weight: 500;

    padding: 8px 10px;

    border-radius: var(--border-radius);

    text-transform: uppercase;

    font-size: 16px;

}

header nav ul li a:hover,

header nav ul li a.active {

    color: var(--white);

    background-color: var(--primary-color);

    text-decoration: none;

}

.cart-count {

    background-color: var(--danger-color);

    color: var(--white);

    border-radius: 50%;

    padding: 1px 5px;

    font-size: 0.75em;

    position: relative;

    top: -8px;

    left: -3px;

}

.product-grid {

    display: grid;

    grid-template-columns: repeat(auto-fill, minmax(230px, 1fr));

    gap: 20px;

    margin-top: 20px;

}

.product-card {

    border: 1px solid var(--border-color);

    padding: 15px;

    text-align: center;

    border-radius: var(--border-radius-lg);

    background-color: var(--white);

    box-shadow: var(--box-shadow-sm);

    display: flex;

    flex-direction: column;

    justify-content: space-between;

}

.product-card:hover {

    box-shadow: var(--box-shadow);

}

.product-card img {

    max-width: 100%;

    height: 180px;

    object-fit: contain;

    margin-bottom: 10px;

    border-radius: var(--border-radius);

}

.product-card h3 {

    font-size: 1.1em;

    margin: 0 0 8px;

    font-weight: 600;

    min-height: 2.6em;

    color: var(--dark-gray);

}

.product-card .category, .product-card .brand {

    font-size: 0.85em;

    color: var(--secondary-color-dark);

    margin-bottom: 5px;

}

.product-card .price {

    font-weight: bold;

    color: var(--success-color);

    font-size: 1.15em;

    margin-bottom: 15px;

}

.product-card .mrp {

    font-size: 0.85em;

    color: var(--secondary-color-dark);

    text-decoration: line-through;

    margin-left: 8px;

}

.product-card .btn {

    margin-top: auto;

    width: 100%;

}

.product-detail-layout {

    display: flex;

    gap: 30px;

    margin-top: 20px;

    flex-wrap: wrap;

}

.product-detail-image {

    flex: 1 1 350px;

    max-width: 450px;

    text-align: center;

}

.product-detail-image img {

    max-width: 100%;

    max-height: 450px;

    object-fit: contain;

    border: 1px solid var(--border-color);

    border-radius: var(--border-radius-lg);

    padding: 10px;

    background-color: var(--white);

}

.product-detail-info {

    flex: 2 1 400px;

    padding: 10px;

}

.product-detail-info h1 {

    margin-top: 0;

    margin-bottom: 10px;

    color: var(--dark-gray);

    font-size: 2em;

    font-weight: 600;

}

.product-detail-info .category {

    font-size: 0.9em;

    color: var(--secondary-color-dark);

    margin-bottom: 15px;

    text-transform: uppercase;

    letter-spacing: 0.5px;

}

.product-detail-info .price {

    font-size: 2em;

    font-weight: bold;

    color: var(--success-color);

    margin-bottom: 20px;

}

.product-detail-info .mrp {

    font-size: 1em;

    color: var(--secondary-color-dark);

    text-decoration: line-through;

    margin-left: 10px;

}

.product-detail-info .stock {

    font-size: 1em;

    margin-bottom: 15px;

    font-weight: 500;

}

.product-detail-info .stock.in-stock { color: var(--success-color); }

.product-detail-info .stock.out-of-stock { color: var(--danger-color); }

.product-detail-info .description {

    margin-bottom: 25px;

    line-height: 1.7;

    color: var(--text-color);

}

.product-detail-info .description h4 {

    font-size: 1.1em;

    margin-bottom: 0.5em;

    color: var(--dark-gray);

}

.cart-table, .admin-table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 20px;

    background-color: var(--white);

    box-shadow: var(--box-shadow-sm);

    font-size: 0.95em;

}

.cart-table th, .cart-table td,

.admin-table th, .admin-table td {

    border: 1px solid var(--border-color);

    padding: .75rem;

    text-align: left;

    vertical-align: middle;

}

.cart-table th, .admin-table th {

    background-color: var(--medium-gray);

    font-weight: 600;

    color: var(--dark-gray);

}

.cart-table img, .admin-table img {

    width: 60px;

    height: 60px;

    object-fit: contain;

    margin-right: 10px;

    vertical-align: middle;

    border-radius: var(--border-radius);

}

.cart-table input[type="number"] {

    width: 70px;

    padding: 5px;

    text-align: center;

}

.cart-summary {

    margin-top: 25px;

    padding: 20px;

    background-color: var(--light-gray);

    border: 1px solid var(--border-color);

    border-radius: var(--border-radius-lg);

    float: right;

    width: 100%;

    max-width: 350px;

    box-sizing: border-box;

}

.cart-summary h3 {

    margin-top: 0;

    font-size: 1.3em;

    color: var(--dark-gray);

}

.cart-actions {

    margin-top: 25px;

    clear: both;

    text-align: right;

}

.cart-actions .btn, .cart-actions button {

    margin-left: 10px;

}

footer {

    background-color: var(--dark-gray);

    color: var(--light-gray);

    padding: 20px 0;

    margin-top: auto;

    font-size: 0.9em;

    flex-shrink: 0;

}

footer .container {

    padding-bottom: 10px;

    text-align: center;

}

footer a {

    color: var(--medium-gray);

    margin: 0 10px;

}

footer a:hover {

    color: var(--white);

}

footer p {

    text-align: center;

    margin-bottom: 0;

}

.admin-container-wrapper {

    flex: 1 0 auto;

    display: flex;

}

.admin-login-page-wrapper {

    flex: 1 0 auto;

    display: flex;

    flex-direction: column;

    justify-content: center;

}

.admin-sidebar {

    width: 240px;

    background: var(--dark-gray);

    color: var(--light-gray);

    padding: 20px 15px;

    flex-shrink: 0;

    min-height: 100%;

}

.admin-sidebar h2 {

    text-align: center;

    margin-bottom: 25px;

    color: var(--white);

    font-size: 1.5em;

}

.admin-sidebar ul li a {

    display: block;

    padding: 10px 15px;

    color: var(--medium-gray);

    border-radius: var(--border-radius);

    margin-bottom: 6px;

    font-size: 0.95em;

}

.admin-main-content {

    flex-grow: 1;

    padding: 25px;

    background-color: var(--light-gray);

}

.admin-main-content h1, .admin-main-content h2 {

    color: var(--dark-gray);

    margin-bottom: 25px;

}

.admin-table th {

    background-color: var(--medium-gray);

    color: var(--dark-gray);

}

.admin-table tr:nth-child(even) {

    background-color: var(--white);

}

.admin-table tr:hover {

    background-color: var(--medium-gray);

}

.actions a, .actions button {

    margin-right: 8px;

    text-decoration: none;

    font-size: 0.9em;

    padding: 3px 8px;

}

.actions .edit-link { color: var(--primary-color); }

.actions .delete-link {

    color: var(--danger-color);

    border: none;

    background: none;

    cursor: pointer;

    padding: 0;

}

.actions .delete-link:hover { color: var(--danger-color-dark); }

@media (max-width: 992px) {

    .admin-container-wrapper {

        flex-direction: column;

    }

    .admin-sidebar {

        width: 100%;

        min-height: auto;

        height: auto;

        margin-bottom: 20px;

    }

    .admin-sidebar h2 { margin-bottom: 15px; }

    .admin-sidebar ul { display: flex; flex-wrap: wrap; justify-content: center; }

    .admin-sidebar ul li { margin: 5px; }

}

@media (max-width: 768px) {

    .container {

        width: 95%;

    }

    header .container {

        flex-direction: column;

        align-items: center;

    }

    header #branding { margin-bottom: 10px; text-align:center; }

    header nav { width: 100%; }

    header nav ul {

        flex-direction: column;

        width: 100%;

        text-align: center;

        margin-top: 10px;

    }

    header nav ul li {

        margin-left: 0;

        padding: 8px 0;

        width: 100%;

        border-top: 1px solid var(--primary-color-dark);

    }

    header nav ul li:first-child { border-top: none; }

    .product-grid {

        grid-template-columns: repeat(auto-fill, minmax(180px, 1fr));

        gap: 15px;

    }

    .product-card h3 { font-size: 1em; min-height: 40px; }

    .product-card .price { font-size: 1em; }

    .product-card img { height: 150px; }

    .product-detail-layout { flex-direction: column; }

    .product-detail-image, .product-detail-info { flex-basis: auto; max-width:100%; }

    .cart-summary { width: 100%; float: none; margin-bottom: 20px; max-width: none; }

    .cart-actions { text-align: center; }

    .cart-actions .btn, .cart-actions button { display: block; width: 100%; margin: 10px 0 0 0; }

    .admin-main-content { padding: 15px; }

    .admin-table, .cart-table { font-size: 0.9em; }

    .admin-table th, .admin-table td,

    .cart-table th, .cart-table td { padding: 0.5rem; }

}

**Terms.php**

<?php

require\_once 'header.php';

$page\_title = "Terms and Conditions";

?>

    <h1>Terms and Conditions</h1>

    <p>Welcome to the Online Shopping Portal. This website is developed solely for academic purposes as part of a BCA project. By using this site, you agree to the following terms:</p>

    <h4>1. Academic Use Only</h4>

    <p>This website is a demo project and is not intended for real commercial transactions or actual purchases.</p>

    <h4>2. No Real Payments</h4>

    <p>Any payment gateways, cart functions, or orders shown on the site are for simulation only. No actual financial processing occurs.</p>

    <h4>3. Content Disclaimer</h4>

    <p>All products, user data, and transaction details are fictional and used to demonstrate the working of an eCommerce portal.</p>

    <h4>4. Limited Liability</h4>

    <p>The developer of this academic project is not liable for any misuse, misunderstanding, or issues arising from the demo website.</p>

    <h4>5. Changes to Terms</h4>

    <p>These terms may be updated without notice as the project evolves for academic submission or testing purposes.</p>

    <p><em>Last updated: May 3, 2025</em></p>

<?php

require\_once 'footer.php';

?>

**Cart\_actions.php**

<?php

require\_once 'config.php';

if (!isset($\_SESSION['cart'])) {

    $\_SESSION['cart'] = array();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    if (isset($\_POST['add\_to\_cart']) && isset($\_POST['product\_id'])) {

        $product\_id = (int)$\_POST['product\_id'];

        $quantity = isset($\_POST['quantity']) ? (int)$\_POST['quantity'] : 1;

        if ($quantity <= 0) {

            display\_message("Invalid quantity specified.", "error");

            redirect($\_SERVER['HTTP\_REFERER'] ?? 'products.php');

        }

        $sql = "SELECT ProductName, SellingPrice, StockQuantity FROM products WHERE ProductID = ? AND IsActive = TRUE";

        $stmt = mysqli\_prepare($conn, $sql);

        if ($stmt) {

            mysqli\_stmt\_bind\_param($stmt, "i", $product\_id);

            mysqli\_stmt\_execute($stmt);

            $result = mysqli\_stmt\_get\_result($stmt);

            $product = mysqli\_fetch\_assoc($result);

            mysqli\_stmt\_close($stmt);

            if ($product) {

                if ($product['StockQuantity'] < $quantity && !isset($\_SESSION['cart'][$product\_id])) {

                     display\_message("Not enough stock available for " . htmlspecialchars($product['ProductName']) . ". Only " . $product['StockQuantity'] . " left.", "error");

                     redirect('products.php');

                }

                if (isset($\_SESSION['cart'][$product\_id])) {

                    $new\_quantity = $\_SESSION['cart'][$product\_id]['quantity'] + $quantity;

                    if ($product['StockQuantity'] < $new\_quantity) {

                        display\_message("Cannot add more " . htmlspecialchars($product['ProductName']) . " to cart. Not enough stock. Max available: " . $product['StockQuantity'] . ", In cart: " . $\_SESSION['cart'][$product\_id]['quantity'], "error");

                        redirect('cart.php');

                    }

                    $\_SESSION['cart'][$product\_id]['quantity'] = $new\_quantity;

                    display\_message(htmlspecialchars($product['ProductName']) . " quantity updated in cart.", "success");

                } else {

                     if ($product['StockQuantity'] < $quantity) {

                        display\_message("Not enough stock available for " . htmlspecialchars($product['ProductName']) . ". Only " . $product['StockQuantity'] . " left.", "error");

                        redirect('products.php');

                    }

                    $\_SESSION['cart'][$product\_id] = array(

                        'product\_id' => $product\_id,

                        'name' => $product['ProductName'],

                        'price' => $product['SellingPrice'],

                        'quantity' => $quantity,

                        'image' => ''

                    );

                    display\_message(htmlspecialchars($product['ProductName']) . " added to cart.", "success");

                }

            } else {

                display\_message("Product not found.", "error");

            }

        } else {

            display\_message("Database error preparing product fetch.", "error");

        }

        redirect('products.php');

    }

    elseif (isset($\_POST['update\_cart\_item']) && isset($\_POST['product\_id'])) {

        $product\_id = (int)$\_POST['product\_id'];

        $quantity = isset($\_POST['quantity']) ? (int)$\_POST['quantity'] : 0;

        if (isset($\_SESSION['cart'][$product\_id])) {

            if ($quantity > 0) {

                $sql\_stock = "SELECT StockQuantity, ProductName FROM products WHERE ProductID = ?";

                $stmt\_stock = mysqli\_prepare($conn, $sql\_stock);

                mysqli\_stmt\_bind\_param($stmt\_stock, "i", $product\_id);

                mysqli\_stmt\_execute($stmt\_stock);

                $result\_stock = mysqli\_stmt\_get\_result($stmt\_stock);

                $product\_stock\_info = mysqli\_fetch\_assoc($result\_stock);

                mysqli\_stmt\_close($stmt\_stock);

                if ($product\_stock\_info && $product\_stock\_info['StockQuantity'] < $quantity) {

                    display\_message("Not enough stock for " . htmlspecialchars($product\_stock\_info['ProductName']) . ". Only " . $product\_stock\_info['StockQuantity'] . " available.", "error");

                } else {

                    $\_SESSION['cart'][$product\_id]['quantity'] = $quantity;

                    display\_message("Cart updated.", "success");

                }

            } else {

                unset($\_SESSION['cart'][$product\_id]);

                display\_message("Item removed from cart.", "success");

            }

        }

        redirect('cart.php');

    }

    elseif (isset($\_POST['remove\_cart\_item']) && isset($\_POST['product\_id'])) {

        $product\_id = (int)$\_POST['product\_id'];

        if (isset($\_SESSION['cart'][$product\_id])) {

            unset($\_SESSION['cart'][$product\_id]);

            display\_message("Item removed from cart.", "success");

        }

        redirect('cart.php');

    }

    elseif (isset($\_POST['clear\_cart'])) {

        $\_SESSION['cart'] = array();

        display\_message("Cart has been cleared.", "success");

        redirect('cart.php');

    }

    else {

        display\_message("Invalid cart action.", "error");

        redirect('index.php');

    }

} else {

    redirect('index.php');

}

mysqli\_close($conn);

?>

**Cart.php**

<?php

require\_once 'header.php';

$cart\_items = isset($\_SESSION['cart']) ? $\_SESSION['cart'] : array();

$grand\_total = 0;

?>

<h2>Your Shopping Cart</h2>

<?php if (empty($cart\_items)): ?>

    <p class="message-info">Your shopping cart is empty. <a href="products.php">Continue shopping!</a></p>

<?php else: ?>

    <form action="cart\_actions.php" method="POST">

        <table class="cart-table">

            <thead>

                <tr>

                    <th>Product</th>

                    <th>Name</th>

                    <th>Price</th>

                    <th>Quantity</th>

                    <th>Subtotal</th>

                    <th>Action</th>

                </tr>

            </thead>

            <tbody>

                <?php foreach ($cart\_items as $item):

                    $product\_info\_sql = "SELECT MainImageURL, StockQuantity FROM products WHERE ProductID = ?";

                    $stmt\_info = mysqli\_prepare($conn, $product\_info\_sql);

                    $product\_image\_url = 'default\_product.png';

                    $current\_stock = 0;

                    if ($stmt\_info) {

                        mysqli\_stmt\_bind\_param($stmt\_info, "i", $item['product\_id']);

                        mysqli\_stmt\_execute($stmt\_info);

                        $info\_result = mysqli\_stmt\_get\_result($stmt\_info);

                        if ($prod\_info = mysqli\_fetch\_assoc($info\_result)) {

                            $product\_image\_url = $prod\_info['MainImageURL'];

                            $current\_stock = $prod\_info['StockQuantity'];

                        }

                        mysqli\_stmt\_close($stmt\_info);

                    }

                    $subtotal = $item['price'] \* $item['quantity'];

                    $grand\_total += $subtotal;

                    ?>

                    <tr>

                        <td>

                            <a href="product\_detail.php?id=<?php echo $item['product\_id']; ?>">

                                <img src="<?php echo UPLOAD\_URL . htmlspecialchars($product\_image\_url); ?>" alt="<?php echo htmlspecialchars($item['name']); ?>">

                            </a>

                        </td>

                        <td>

                            <a href="product\_detail.php?id=<?php echo $item['product\_id']; ?>" style="text-decoration:none; color:#333;">

                                <?php echo htmlspecialchars($item['name']); ?>

                            </a>

                        </td>

                        <td>₹<?php echo number\_format($item['price'], 2); ?></td>

                        <td>

                            <form action="cart\_actions.php" method="POST" style="display:inline;">

                                <input type="hidden" name="product\_id" value="<?php echo $item['product\_id']; ?>">

                                <input type="number" name="quantity" value="<?php echo $item['quantity']; ?>" min="1" max="<?php echo $current\_stock; ?>" style="width: 60px; padding: 5px;">

                                <button type="submit" name="update\_cart\_item" class="btn btn-secondary" style="padding: 5px 8px; font-size: 0.9em;">Update</button>

                            </form>

                        </td>

                        <td>₹<?php echo number\_format($subtotal, 2); ?></td>

                        <td>

                            <form action="cart\_actions.php" method="POST" style="display:inline;">

                                <input type="hidden" name="product\_id" value="<?php echo $item['product\_id']; ?>">

                                <button type="submit" name="remove\_cart\_item" class="btn btn-danger" style="padding: 5px 8px; font-size: 0.9em;">Remove</button>

                            </form>

                        </td>

                    </tr>

                <?php endforeach; ?>

            </tbody>

        </table>

        <div class="cart-summary">

            <h3>Cart Summary</h3>

            <p><strong>Grand Total: ₹<?php echo number\_format($grand\_total, 2); ?></strong></p>

        </div>

        <div class="cart-actions">

            <button type="submit" name="clear\_cart" class="btn btn-danger">Clear Cart</button>

            <a href="products.php" class="btn btn-secondary">Continue Shopping</a>

            <?php if (is\_user\_logged\_in()): ?>

                <a href="checkout.php" class="btn">Proceed to Checkout</a> <?php else: ?>

                <p style="margin-top:10px;"><a href="login.php?redirect=cart.php" class="btn">Login to Checkout</a></p>

            <?php endif; ?>

        </div>

    </form>

<?php endif; ?>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**Checkout.php**

<?php

require\_once 'header.php';

require\_user\_login();

$user\_id = $\_SESSION['user\_id'];

$cart\_items = isset($\_SESSION['cart']) ? $\_SESSION['cart'] : array();

$grand\_total = 0;

if (empty($cart\_items)) {

    display\_message("Your cart is empty. Please add items before checkout.", "info");

    redirect('cart.php');

}

foreach ($cart\_items as $item) {

    $grand\_total += $item['price'] \* $item['quantity'];

}

$addresses = [];

$sql\_addresses = "SELECT AddressID, Street, City, State, ZipCode, Country, IsDefaultShipping FROM addresses WHERE UserID = ? ORDER BY IsDefaultShipping DESC, AddressID DESC";

$stmt\_addresses = mysqli\_prepare($conn, $sql\_addresses);

if ($stmt\_addresses) {

    mysqli\_stmt\_bind\_param($stmt\_addresses, "i", $user\_id);

    mysqli\_stmt\_execute($stmt\_addresses);

    $result\_addresses = mysqli\_stmt\_get\_result($stmt\_addresses);

    while ($row = mysqli\_fetch\_assoc($result\_addresses)) {

        $addresses[] = $row;

    }

    mysqli\_stmt\_close($stmt\_addresses);

}

$selected\_address\_id = null;

if (!empty($addresses)) {

    foreach ($addresses as $addr) {

        if ($addr['IsDefaultShipping']) {

            $selected\_address\_id = $addr['AddressID'];

            break;

        }

    }

    if ($selected\_address\_id === null) {

        $selected\_address\_id = $addresses[0]['AddressID'];

    }

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['place\_order'])) {

    $selected\_address\_id\_posted = isset($\_POST['shipping\_address']) ? (int)$\_POST['shipping\_address'] : null;

    $payment\_method = isset($\_POST['payment\_method']) ? sanitize\_input($\_POST['payment\_method']) : 'Simulated COD';

    $errors = [];

    if (empty($selected\_address\_id\_posted) && empty($addresses)) {

        $errors[] = "Please select or add a shipping address.";

    } elseif (!empty($selected\_address\_id\_posted)) {

        $valid\_address = false;

        foreach($addresses as $addr) {

            if ($addr['AddressID'] == $selected\_address\_id\_posted) {

                $valid\_address = true;

                $selected\_address\_id = $selected\_address\_id\_posted;

                break;

            }

        }

        if (!$valid\_address) {

            $errors[] = "Invalid shipping address selected.";

        }

    } elseif (empty($selected\_address\_id\_posted) && !empty($addresses)) {

    }

    if (empty($payment\_method)) {

        $errors[] = "Please select a payment method.";

    }

    $current\_grand\_total = 0;

    foreach ($cart\_items as $item) {

        $current\_grand\_total += $item['price'] \* $item['quantity'];

    }

    if ($current\_grand\_total <= 0) {

        $errors[] = "Cannot place an order with an empty cart or zero total.";

    }

    if (empty($errors)) {

        mysqli\_begin\_transaction($conn);

        try {

            foreach ($cart\_items as $product\_id\_cart => $item\_cart) {

                $sql\_stock\_check = "SELECT StockQuantity, ProductName FROM products WHERE ProductID = ? FOR UPDATE";

                $stmt\_stock\_check = mysqli\_prepare($conn, $sql\_stock\_check);

                mysqli\_stmt\_bind\_param($stmt\_stock\_check, "i", $product\_id\_cart);

                mysqli\_stmt\_execute($stmt\_stock\_check);

                $result\_stock\_check = mysqli\_stmt\_get\_result($stmt\_stock\_check);

                $product\_db = mysqli\_fetch\_assoc($result\_stock\_check);

                mysqli\_stmt\_close($stmt\_stock\_check);

                if (!$product\_db || $product\_db['StockQuantity'] < $item\_cart['quantity']) {

                    throw new Exception("Not enough stock for " . htmlspecialchars($item\_cart['name']) . ". Available: " . ($product\_db ? $product\_db['StockQuantity'] : 0) . ".");

                }

            }

            $order\_status = 'Pending';

            $payment\_status = ($payment\_method === 'Simulated COD') ? 'Pending' : 'Paid (Simulated)';

            $sql\_insert\_order = "INSERT INTO orders (UserID, TotalAmount, ShippingAddressID, OrderStatus, PaymentMethod, PaymentStatus) VALUES (?, ?, ?, ?, ?, ?)";

            $stmt\_insert\_order = mysqli\_prepare($conn, $sql\_insert\_order);

            mysqli\_stmt\_bind\_param($stmt\_insert\_order, "idssss", $user\_id, $current\_grand\_total, $selected\_address\_id, $order\_status, $payment\_method, $payment\_status);

            if (!mysqli\_stmt\_execute($stmt\_insert\_order)) {

                throw new Exception("Failed to place order. DB Error (Order Insert): " . mysqli\_stmt\_error($stmt\_insert\_order));

            }

            $order\_id = mysqli\_insert\_id($conn);

            mysqli\_stmt\_close($stmt\_insert\_order);

            $sql\_insert\_order\_item = "INSERT INTO order\_items (OrderID, ProductID, Quantity, PriceAtPurchase, Subtotal) VALUES (?, ?, ?, ?, ?)";

            $stmt\_insert\_order\_item = mysqli\_prepare($conn, $sql\_insert\_order\_item);

            foreach ($cart\_items as $product\_id\_cart => $item\_cart) {

                $subtotal\_item = $item\_cart['price'] \* $item\_cart['quantity'];

                mysqli\_stmt\_bind\_param($stmt\_insert\_order\_item, "iiidd", $order\_id, $item\_cart['product\_id'], $item\_cart['quantity'], $item\_cart['price'], $subtotal\_item);

                if (!mysqli\_stmt\_execute($stmt\_insert\_order\_item)) {

                     throw new Exception("Failed to save order items. DB Error: " . mysqli\_stmt\_error($stmt\_insert\_order\_item));

                }

                $new\_stock = $product\_db['StockQuantity'] - $item\_cart['quantity'];

                $sql\_current\_stock = "SELECT StockQuantity FROM products WHERE ProductID = ?";

                $stmt\_current\_stock = mysqli\_prepare($conn, $sql\_current\_stock);

                mysqli\_stmt\_bind\_param($stmt\_current\_stock, "i", $item\_cart['product\_id']);

                mysqli\_stmt\_execute($stmt\_current\_stock);

                $res\_current\_stock = mysqli\_stmt\_get\_result($stmt\_current\_stock);

                $prod\_current\_stock\_data = mysqli\_fetch\_assoc($res\_current\_stock);

                mysqli\_stmt\_close($stmt\_current\_stock);

                $current\_item\_stock = $prod\_current\_stock\_data['StockQuantity'];

                $new\_stock\_for\_item = $current\_item\_stock - $item\_cart['quantity'];

                $sql\_update\_stock = "UPDATE products SET StockQuantity = ? WHERE ProductID = ?";

                $stmt\_update\_stock = mysqli\_prepare($conn, $sql\_update\_stock);

                mysqli\_stmt\_bind\_param($stmt\_update\_stock, "ii", $new\_stock\_for\_item, $item\_cart['product\_id']);

                 if (!mysqli\_stmt\_execute($stmt\_update\_stock)) {

                    throw new Exception("Failed to update stock for product ID " . $item\_cart['product\_id'] . ". DB Error: " . mysqli\_stmt\_error($stmt\_update\_stock));

                }

                mysqli\_stmt\_close($stmt\_update\_stock);

            }

            mysqli\_stmt\_close($stmt\_insert\_order\_item);

            mysqli\_commit($conn);

            $\_SESSION['cart'] = array();

            $\_SESSION['last\_order\_id'] = $order\_id;

            display\_message("Order placed successfully! Your Order ID is #" . $order\_id, "success");

            redirect('order\_success.php');

        } catch (Exception $e) {

            mysqli\_rollback($conn);

            display\_message("Order placement failed: " . $e->getMessage(), "error");

        }

    } else {

        $error\_message\_html = "<ul>";

        foreach ($errors as $error) {

            $error\_message\_html .= "<li>" . htmlspecialchars($error) . "</li>";

        }

        $error\_message\_html .= "</ul>";

        $\_SESSION['checkout\_form\_errors'] = $error\_message\_html;

        redirect('checkout.php');

    }

}

$checkout\_form\_errors\_html = '';

if (isset($\_SESSION['checkout\_form\_errors'])) {

    $checkout\_form\_errors\_html = $\_SESSION['checkout\_form\_errors'];

    unset($\_SESSION['checkout\_form\_errors']);

}

?>

<h2>Checkout</h2>

<?php

if (!empty($checkout\_form\_errors\_html)) {

    echo "<div class='message-error'>" . $checkout\_form\_errors\_html . "</div>";

}

?>

<div style="display: flex; gap: 30px; flex-wrap: wrap;">

    <div style="flex: 2; min-width: 300px;">

        <h3>Order Summary</h3>

        <table class="cart-table" style="margin-bottom: 20px;">

            <thead>

                <tr>

                    <th>Product</th>

                    <th>Price</th>

                    <th>Quantity</th>

                    <th>Subtotal</th>

                </tr>

            </thead>

            <tbody>

                <?php foreach ($cart\_items as $item): ?>

                    <tr>

                        <td><?php echo htmlspecialchars($item['name']); ?></td>

                        <td>₹<?php echo number\_format($item['price'], 2); ?></td>

                        <td><?php echo $item['quantity']; ?></td>

                        <td>₹<?php echo number\_format($item['price'] \* $item['quantity'], 2); ?></td>

                    </tr>

                <?php endforeach; ?>

            </tbody>

            <tfoot>

                <tr>

                    <td colspan="3" style="text-align:right; font-weight:bold;">Grand Total:</td>

                    <td style="font-weight:bold;">₹<?php echo number\_format($grand\_total, 2); ?></td>

                </tr>

            </tfoot>

        </table>

        <p><a href="cart.php" class="btn btn-secondary">Edit Cart</a></p>

    </div>

    <div style="flex: 1; min-width: 280px;">

        <h3>Shipping & Payment</h3>

        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

            <div class="form-group">

                <label for="shipping\_address">Shipping Address:</label>

                <?php if (!empty($addresses)): ?>

                    <select name="shipping\_address" id="shipping\_address" class="form-control" required>

                        <?php foreach ($addresses as $address): ?>

                            <option value="<?php echo $address['AddressID']; ?>" <?php echo ($address['AddressID'] == $selected\_address\_id) ? 'selected' : ''; ?>>

                                <?php echo htmlspecialchars($address['Street'] . ", " . $address['City'] . ", " . $address['State'] . " - " . $address['ZipCode']); ?>

                                <?php echo ($address['IsDefaultShipping']) ? ' (Default)' : ''; ?>

                            </option>

                        <?php endforeach; ?>

                    </select>

                    <p style="margin-top: 5px;"><small><a href="profile.php#addresses">Manage Addresses</a></small></p>

                <?php else: ?>

                    <p class="message-info">You have no saved shipping addresses. Please <a href="profile.php#add-address">add an address</a> in your profile before proceeding.</p>

                    <p><small>For this project, address management is on the profile page. A real site might allow adding an address here.</small></p>

                <?php endif; ?>

            </div>

            <div class="form-group">

                <label for="payment\_method">Payment Method:</label>

                <select name="payment\_method" id="payment\_method" class="form-control" required>

                    <option value="Simulated COD" selected>Simulated Cash on Delivery (COD)</option>

                    <option value="Simulated Card">Simulated Card Payment</option>

                    </select>

                <p><small>This is a simulated payment process for project demonstration.</small></p>

            </div>

            <?php if (!empty($addresses)): ?>

            <div class="form-group">

                <button type="submit" name="place\_order" class="btn" style="width:100%;">Place Order (Total: ₹<?php echo number\_format($grand\_total, 2); ?>)</button>

            </div>

            <?php else: ?>

                 <button type="submit" name="place\_order" class="btn" style="width:100%;" disabled>Place Order</button>

                 <p class="message-error" style="margin-top:10px;">Please add a shipping address in your profile to place an order.</p>

            <?php endif; ?>

        </form>

    </div>

</div>

<?php

mysqli\_close($conn);

require\_once 'footer.php';

?>

**/admin/add\_product.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

$categories = [];

$sql\_cat = "SELECT CategoryID, CategoryName FROM categories ORDER BY CategoryName ASC";

$res\_cat = mysqli\_query($conn, $sql\_cat);

if ($res\_cat) {

    while ($row\_cat = mysqli\_fetch\_assoc($res\_cat)) {

        $categories[] = $row\_cat;

    }

}

$product\_name = $description = $sku = $mrp = $selling\_price = $stock\_quantity = $category\_id = '';

$main\_image\_url = 'default\_product.png';

$is\_active = 1;

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $product\_name = sanitize\_input($\_POST['product\_name']);

    $description = sanitize\_input($\_POST['description']);

    $mrp = filter\_var($\_POST['mrp'], FILTER\_VALIDATE\_FLOAT);

    $selling\_price = filter\_var($\_POST['selling\_price'], FILTER\_VALIDATE\_FLOAT);

    $stock\_quantity = filter\_var($\_POST['stock\_quantity'], FILTER\_VALIDATE\_INT);

    $category\_id = (int)$\_POST['category\_id'];

    $is\_active = isset($\_POST['is\_active']) ? 1 : 0;

    if (empty($product\_name)) $errors[] = "Product name is required.";

    if ($mrp === false || $mrp < 0) $errors[] = "Valid MRP is required.";

    if ($selling\_price === false || $selling\_price < 0) $errors[] = "Valid selling price is required.";

    if ($selling\_price > $mrp) $errors[] = "Selling price cannot be greater than MRP.";

    if ($stock\_quantity === false || $stock\_quantity < 0) $errors[] = "Valid stock quantity is required.";

    if (empty($category\_id)) $errors[] = "Category is required.";

    $uploaded\_image\_name = $main\_image\_url;

    if (isset($\_FILES['main\_image']) && $\_FILES['main\_image']['error'] == UPLOAD\_ERR\_OK) {

        $file\_info = $\_FILES['main\_image'];

        $file\_name = $file\_info['name'];

        $file\_tmp\_name = $file\_info['tmp\_name'];

        $file\_size = $file\_info['size'];

        $file\_error = $file\_info['error'];

        $file\_ext\_parts = explode('.', $file\_name);

        $file\_ext = strtolower(end($file\_ext\_parts));

        $allowed\_exts = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

        if (in\_array($file\_ext, $allowed\_exts)) {

            if ($file\_error === 0) {

                if ($file\_size <= 2000000) {

                    $new\_file\_name = uniqid('prod\_', true) . '.' . $file\_ext;

                    $file\_destination = UPLOAD\_DIR . $new\_file\_name;

                    if (move\_uploaded\_file($file\_tmp\_name, $file\_destination)) {

                        $uploaded\_image\_name = $new\_file\_name;

                    } else {

                        $errors[] = "Failed to move uploaded image to destination.";

                    }

                } else {

                    $errors[] = "Image file is too large (Max 2MB).";

                }

            } else {

                $errors[] = "Error uploading image: error code " . $file\_error;

            }

        } else {

            $errors[] = "Invalid image file type. Allowed: " . implode(', ', $allowed\_exts);

        }

    } elseif (isset($\_FILES['main\_image']) && $\_FILES['main\_image']['error'] != UPLOAD\_ERR\_NO\_FILE) {

        $errors[] = "There was an error with the image upload (Code: " . $\_FILES['main\_image']['error'] . ").";

    }

    if (empty($errors)) {

        $sql = "INSERT INTO products (ProductName, Description, MRP, SellingPrice, StockQuantity, CategoryID, MainImageURL, IsActive)

                VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

        $stmt = mysqli\_prepare($conn, $sql);

        if ($stmt) {

            mysqli\_stmt\_bind\_param($stmt, "ssddiisi",

                $product\_name, $description, $mrp, $selling\_price,

                $stock\_quantity, $category\_id, $uploaded\_image\_name, $is\_active

            );

            if (mysqli\_stmt\_execute($stmt)) {

                $\_SESSION['admin\_message'] = "<div class='message-success'>Product '{$product\_name}' added successfully!</div>";

                redirect('admin/manage\_products.php');

            } else {

                $errors[] = "Failed to add product: " . mysqli\_stmt\_error($stmt);

            }

            mysqli\_stmt\_close($stmt);

        } else {

            $errors[] = "Database error preparing statement: " . mysqli\_error($conn);

        }

    }

}

?>

<h1>Add New Product</h1>

<?php

if (!empty($errors)) {

    echo "<div class='message-error'><ul>";

    foreach ($errors as $error) {

        echo "<li>" . htmlspecialchars($error) . "</li>";

    }

    echo "</ul></div>";

}

?>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST" enctype="multipart/form-data">

    <div class="form-group">

        <label for="product\_name">Product Name:</label>

        <input type="text" name="product\_name" id="product\_name" value="<?php echo htmlspecialchars($product\_name); ?>" required>

    </div>

    <div class="form-group">

        <label for="description">Description:</label>

        <textarea name="description" id="description" rows="5"><?php echo htmlspecialchars($description); ?></textarea>

    </div>

    <div class="form-group">

        <label for="mrp">MRP (₹):</label>

        <input type="number" name="mrp" id="mrp" step="0.01" min="0" value="<?php echo htmlspecialchars($mrp); ?>" required>

    </div>

    <div class="form-group">

        <label for="selling\_price">Selling Price (₹):</label>

        <input type="number" name="selling\_price" id="selling\_price" step="0.01" min="0" value="<?php echo htmlspecialchars($selling\_price); ?>" required>

    </div>

    <div class="form-group">

        <label for="stock\_quantity">Stock Quantity:</label>

        <input type="number" name="stock\_quantity" id="stock\_quantity" min="0" value="<?php echo htmlspecialchars($stock\_quantity); ?>" required>

    </div>

    <div class="form-group">

        <label for="category\_id">Category:</label>

        <select name="category\_id" id="category\_id" required>

            <option value="">-- Select Category --</option>

            <?php foreach ($categories as $cat): ?>

                <option value="<?php echo $cat['CategoryID']; ?>" <?php echo ($category\_id == $cat['CategoryID'] ? 'selected' : ''); ?>>

                    <?php echo htmlspecialchars($cat['CategoryName']); ?>

                </option>

            <?php endforeach; ?>

        </select>

    </div>

    <div class="form-group">

        <label for="main\_image">Main Product Image:</label>

        <input type="file" name="main\_image" id="main\_image" accept="image/jpeg, image/png, image/gif, image/webp">

        <small>Max 2MB. Allowed types: JPG, PNG, GIF, WEBP.</small>

    </div>

    <div class="form-group">

        <label for="is\_active">

            <input type="checkbox" name="is\_active" id="is\_active" value="1" <?php echo ($is\_active ? 'checked' : ''); ?>>

            Product is Active (visible to customers)

        </label>

    </div>

    <div class="form-group">

        <button type="submit" class="btn">Add Product</button>

        <a href="manage\_products.php" class="btn btn-secondary">Cancel</a>

    </div>

</form>

<?php

require\_once 'admin\_footer.php';

mysqli\_close($conn);

?>

**/admin/admin\_header.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin Panel - Online Shopping Portal</title>

    <link rel="stylesheet" href="<?php echo BASE\_URL; ?>style.css">

    <style>

        body, html { height: 100%; margin: 0; }

        body { display: flex; flex-direction: column; }

        .admin-container-wrapper { flex: 1 0 auto; display: flex; }

        .admin-login-page-wrapper { flex: 1 0 auto; display: flex; flex-direction: column; justify-content: center; }

        .admin-login-form {

            width: 100%;

            max-width: 400px;

            margin: 20px auto;

            padding: 30px;

            background-color: #fff;

            border: 1px solid #ddd;

            border-radius: 8px;

            box-shadow: 0 4px 10px rgba(0,0,0,0.1);

        }

        .admin-login-form h2 {

            text-align: center;

            margin-bottom: 25px;

            color: #333;

        }

        .admin-dashboard-stats {

            display: flex;

            gap: 20px;

            flex-wrap: wrap;

            margin-bottom: 30px;

        }

        .stat-card {

            background-color: #fff;

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0 2px 5px rgba(0,0,0,0.08);

            flex: 1;

            min-width: 200px;

            text-align: center;

        }

        .stat-card h4 {

            margin-top: 0;

            color: #2c3e50;

        }

        .stat-card p {

            font-size: 2em;

            font-weight: bold;

            color: #0779e4;

            margin-bottom: 0;

        }

        .admin-sidebar {

            width: 250px;

            background: #2c3e50;

            color: #ecf0f1;

            padding: 20px;

            min-height: 100%;

        }

        .admin-sidebar h2 { text-align: center; margin-bottom: 30px; color: #fff; }

        .admin-sidebar ul { list-style: none; padding: 0; }

        .admin-sidebar ul li a { display: block; padding: 12px 15px; color: #ecf0f1; text-decoration: none; border-radius: 4px; margin-bottom: 5px; transition: background-color 0.3s ease; }

        .admin-sidebar ul li a:hover,

        .admin-sidebar ul li a.active { background-color: #34495e; color: #fff; }

        .admin-main-content { flex-grow: 1; padding: 20px; background-color: #ecf0f1; }

        footer.admin-footer {

            text-align:center;

            padding:15px;

            background-color:#2c3e50;

            color:#ecf0f1;

            flex-shrink: 0;

        }

        footer.admin-footer-login {

             background-color:#f4f4f4; color:#333; margin-top: auto;

        }

    </style>

</head>

<body>

    <?php if (is\_admin\_logged\_in()): ?>

    <div class="admin-container-wrapper">

        <aside class="admin-sidebar">

            <h2>Admin Menu</h2>

            <ul>

                <li><a href="<?php echo BASE\_URL; ?>admin/dashboard.php" class="<?php echo (basename($\_SERVER['PHP\_SELF']) == 'dashboard.php' ? 'active' : ''); ?>">Dashboard</a></li>

                <li><a href="<?php echo BASE\_URL; ?>admin/manage\_products.php" class="<?php echo (basename($\_SERVER['PHP\_SELF']) == 'manage\_products.php' ? 'active' : ''); ?>">Products</a></li>

                <li><a href="<?php echo BASE\_URL; ?>admin/manage\_categories.php" class="<?php echo (basename($\_SERVER['PHP\_SELF']) == 'manage\_categories.php' ? 'active' : ''); ?>">Categories</a></li>

                <li><a href="<?php echo BASE\_URL; ?>admin/manage\_orders.php" class="<?php echo (basename($\_SERVER['PHP\_SELF']) == 'manage\_orders.php' ? 'active' : ''); ?>">Orders</a></li>

                <li><a href="<?php echo BASE\_URL; ?>admin/logout.php">Logout (<?php echo htmlspecialchars($\_SESSION['admin\_username'] ?? 'Admin'); ?>)</a></li>

            </ul>

            <p style="text-align:center; margin-top:20px;"><a href="<?php echo BASE\_URL; ?>" target="\_blank" style="color:#fff; font-size:0.9em;">View Live Site &rarr;</a></p>

        </aside>

        <main class="admin-main-content">

            <?php

            echo get\_admin\_session\_message();

            ?>

    <?php else: ?>

    <?php endif; ?>

**/admin/admin\_footer.php**

<?php

?>

    <?php if (is\_admin\_logged\_in()): ?>

    <footer class="admin-footer">

        <p>&copy; <?php echo date("Y"); ?> Online Shopping Portal - Admin Panel</p>

    </footer>

    <?php else: ?>

    <footer class="admin-footer admin-footer-login">

        <p>&copy; <?php echo date("Y"); ?> Online Shopping Portal - Admin Login</p>

    </footer>

    <?php endif; ?>

</body>

</html>

**/admin/dashboard.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

$sql\_users = "SELECT COUNT(UserID) as total\_users FROM users";

$res\_users = mysqli\_query($conn, $sql\_users);

$total\_users = ($res\_users && mysqli\_num\_rows($res\_users) > 0) ? mysqli\_fetch\_assoc($res\_users)['total\_users'] : 0;

$sql\_products = "SELECT COUNT(ProductID) as total\_products FROM products WHERE IsActive = TRUE";

$res\_products = mysqli\_query($conn, $sql\_products);

$total\_products = ($res\_products && mysqli\_num\_rows($res\_products) > 0) ? mysqli\_fetch\_assoc($res\_products)['total\_products'] : 0;

$sql\_orders = "SELECT COUNT(OrderID) as total\_orders FROM orders";

$res\_orders = mysqli\_query($conn, $sql\_orders);

$total\_orders = ($res\_orders && mysqli\_num\_rows($res\_orders) > 0) ? mysqli\_fetch\_assoc($res\_orders)['total\_orders'] : 0;

$sql\_pending\_orders = "SELECT COUNT(OrderID) as pending\_orders FROM orders WHERE OrderStatus = 'Pending'";

$res\_pending\_orders = mysqli\_query($conn, $sql\_pending\_orders);

$pending\_orders = ($res\_pending\_orders && mysqli\_num\_rows($res\_pending\_orders) > 0) ? mysqli\_fetch\_assoc($res\_pending\_orders)['pending\_orders'] : 0;

$recent\_orders = [];

$sql\_recent = "SELECT o.OrderID, u.FirstName, u.LastName, o.OrderDate, o.TotalAmount, o.OrderStatus

               FROM orders o

               JOIN users u ON o.UserID = u.UserID

               ORDER BY o.OrderDate DESC LIMIT 5";

$res\_recent = mysqli\_query($conn, $sql\_recent);

if ($res\_recent) {

    while ($row = mysqli\_fetch\_assoc($res\_recent)) {

        $recent\_orders[] = $row;

    }

}

?>

<h1>Admin Dashboard</h1>

<p>Welcome back, <?php echo htmlspecialchars($\_SESSION['admin\_username']); ?>!</p>

<div class="admin-dashboard-stats">

    <div class="stat-card"><h4>Total Customers</h4><p><?php echo $total\_users; ?></p></div>

    <div class="stat-card"><h4>Active Products</h4><p><?php echo $total\_products; ?></p></div>

    <div class="stat-card"><h4>Total Orders</h4><p><?php echo $total\_orders; ?></p></div>

    <div class="stat-card"><h4>Pending Orders</h4><p><?php echo $pending\_orders; ?></p></div>

</div>

<h3>Recent Orders</h3>

<?php if (!empty($recent\_orders)): ?>

<table class="admin-table">

    <thead>

        <tr><th>Order ID</th><th>Customer</th><th>Date</th><th>Total</th><th>Status</th><th>Action</th></tr>

    </thead>

    <tbody>

        <?php foreach ($recent\_orders as $order): ?>

        <tr>

            <td>#<?php echo $order['OrderID']; ?></td>

            <td><?php echo htmlspecialchars($order['FirstName'] . ' ' . $order['LastName']); ?></td>

            <td><?php echo date("M j, Y, g:i a", strtotime($order['OrderDate'])); ?></td>

            <td>₹<?php echo number\_format($order['TotalAmount'], 2); ?></td>

            <td><?php echo htmlspecialchars($order['OrderStatus']); ?></td>

            <td><a href="<?php echo BASE\_URL; ?>admin/manage\_orders.php?view\_order\_id=<?php echo $order['OrderID']; ?>" class="btn btn-secondary" style="font-size:0.8em; padding:3px 8px;">View</a></td>

        </tr>

        <?php endforeach; ?>

    </tbody>

</table>

<?php else: ?>

<p class="message-info">No recent orders found.</p>

<?php endif; ?>

<?php

require\_once 'admin\_footer.php';

if ($conn) mysqli\_close($conn);

?>

**/admin/edit\_product.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

if (!isset($\_GET['id']) || empty((int)$\_GET['id'])) {

    $\_SESSION['admin\_message'] = "<div class='message-error'>No product ID specified for editing.</div>";

    redirect(BASE\_URL . 'admin/manage\_products.php');

}

$product\_id\_to\_edit = (int)$\_GET['id'];

$product = null;

$sql\_fetch = "SELECT \* FROM products WHERE ProductID = ?";

$stmt\_fetch = mysqli\_prepare($conn, $sql\_fetch);

if ($stmt\_fetch) {

    mysqli\_stmt\_bind\_param($stmt\_fetch, "i", $product\_id\_to\_edit);

    mysqli\_stmt\_execute($stmt\_fetch);

    $result\_fetch = mysqli\_stmt\_get\_result($stmt\_fetch);

    $product = mysqli\_fetch\_assoc($result\_fetch);

    mysqli\_stmt\_close($stmt\_fetch);

}

if (!$product) {

    $\_SESSION['admin\_message'] = "<div class='message-error'>Product not found with ID {$product\_id\_to\_edit}.</div>";

    redirect(BASE\_URL . 'admin/manage\_products.php');

}

$categories = [];

$sql\_cat = "SELECT CategoryID, CategoryName FROM categories ORDER BY CategoryName ASC";

$res\_cat = mysqli\_query($conn, $sql\_cat);

if ($res\_cat) {

    while ($row\_cat = mysqli\_fetch\_assoc($res\_cat)) {

        $categories[] = $row\_cat;

    }

}

$product\_name = $product['ProductName'];

$description = $product['Description'];

$mrp = $product['MRP'];

$selling\_price = $product['SellingPrice'];

$stock\_quantity = $product['StockQuantity'];

$category\_id = $product['CategoryID'];

$current\_main\_image\_url = $product['MainImageURL'];

$is\_active = $product['IsActive'];

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $product\_name = sanitize\_input($\_POST['product\_name']);

    $description = sanitize\_input($\_POST['description']);

    $mrp = filter\_var($\_POST['mrp'], FILTER\_VALIDATE\_FLOAT);

    $selling\_price = filter\_var($\_POST['selling\_price'], FILTER\_VALIDATE\_FLOAT);

    $stock\_quantity = filter\_var($\_POST['stock\_quantity'], FILTER\_VALIDATE\_INT);

    $category\_id = (int)$\_POST['category\_id'];

    $is\_active = isset($\_POST['is\_active']) ? 1 : 0;

    if (empty($product\_name)) $errors[] = "Product name is required.";

    if ($mrp === false || $mrp < 0) $errors[] = "Valid MRP is required.";

    if ($selling\_price === false || $selling\_price < 0) $errors[] = "Valid selling price is required.";

    if ($selling\_price > $mrp) $errors[] = "Selling price cannot be greater than MRP.";

    if ($stock\_quantity === false || $stock\_quantity < 0) $errors[] = "Valid stock quantity is required.";

    if (empty($category\_id)) $errors[] = "Category is required.";

    $new\_uploaded\_image\_name = $current\_main\_image\_url;

    if (isset($\_FILES['main\_image']) && $\_FILES['main\_image']['error'] == UPLOAD\_ERR\_OK) {

        $file\_info = $\_FILES['main\_image'];

        $file\_name = $file\_info['name'];

        $file\_tmp\_name = $file\_info['tmp\_name'];

        $file\_size = $file\_info['size'];

        $file\_error = $file\_info['error'];

        $file\_ext\_parts = explode('.', $file\_name);

        $file\_ext = strtolower(end($file\_ext\_parts));

        $allowed\_exts = ['jpg', 'jpeg', 'png', 'gif', 'webp'];

        if (in\_array($file\_ext, $allowed\_exts)) {

            if ($file\_error === 0) {

                if ($file\_size <= 2000000) {

                    $new\_file\_name\_unique = uniqid('prod\_', true) . '.' . $file\_ext;

                    $file\_destination = UPLOAD\_DIR . $new\_file\_name\_unique;

                    if (move\_uploaded\_file($file\_tmp\_name, $file\_destination)) {

                        if ($current\_main\_image\_url && $current\_main\_image\_url != 'default\_product.png' && file\_exists(UPLOAD\_DIR . $current\_main\_image\_url)) {

                            unlink(UPLOAD\_DIR . $current\_main\_image\_url);

                        }

                        $new\_uploaded\_image\_name = $new\_file\_name\_unique;

                    } else {

                        $errors[] = "Failed to move uploaded image.";

                    }

                } else {

                    $errors[] = "Image file too large (Max 2MB).";

                }

            } else {

                $errors[] = "Error uploading image: code " . $file\_error;

            }

        } else {

            $errors[] = "Invalid image file type.";

        }

    } elseif (isset($\_FILES['main\_image']) && $\_FILES['main\_image']['error'] != UPLOAD\_ERR\_NO\_FILE) {

        $errors[] = "There was an error with the image upload (Code: " . $\_FILES['main\_image']['error'] . ").";

    }

    if (empty($errors)) {

        $sql\_update = "UPDATE products SET

                        ProductName = ?, Description = ?, MRP = ?, SellingPrice = ?,

                        StockQuantity = ?, CategoryID = ?, MainImageURL = ?, IsActive = ?

                       WHERE ProductID = ?";

        $stmt\_update = mysqli\_prepare($conn, $sql\_update);

        if ($stmt\_update) {

            mysqli\_stmt\_bind\_param($stmt\_update, "ssddiisii",

                $product\_name, $description, $mrp, $selling\_price,

                $stock\_quantity, $category\_id, $new\_uploaded\_image\_name, $is\_active,

                $product\_id\_to\_edit

            );

            if (mysqli\_stmt\_execute($stmt\_update)) {

                $\_SESSION['admin\_message'] = "<div class='message-success'>Product '{$product\_name}' updated successfully!</div>";

                redirect('admin/manage\_products.php');

            } else {

                $errors[] = "Failed to update product: " . mysqli\_stmt\_error($stmt\_update);

            }

            mysqli\_stmt\_close($stmt\_update);

        } else {

            $errors[] = "Database error preparing update statement: " . mysqli\_error($conn);

        }

    }

}

?>

<h1>Edit Product: <?php echo htmlspecialchars($product['ProductName']); ?></h1>

<?php

if (!empty($errors)) {

    echo "<div class='message-error'><ul>";

    foreach ($errors as $error) {

        echo "<li>" . htmlspecialchars($error) . "</li>";

    }

    echo "</ul></div>";

}

?>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>?id=<?php echo $product\_id\_to\_edit; ?>" method="POST" enctype="multipart/form-data">

    <div class="form-group">

        <label for="product\_name">Product Name:</label>

        <input type="text" name="product\_name" id="product\_name" value="<?php echo htmlspecialchars($product\_name); ?>" required>

    </div>

    <div class="form-group">

        <label for="description">Description:</label>

        <textarea name="description" id="description" rows="5"><?php echo htmlspecialchars($description); ?></textarea>

    </div>

    <div class="form-group">

        <label for="mrp">MRP (₹):</label>

        <input type="number" name="mrp" id="mrp" step="0.01" min="0" value="<?php echo htmlspecialchars($mrp); ?>" required>

    </div>

    <div class="form-group">

        <label for="selling\_price">Selling Price (₹):</label>

        <input type="number" name="selling\_price" id="selling\_price" step="0.01" min="0" value="<?php echo htmlspecialchars($selling\_price); ?>" required>

    </div>

    <div class="form-group">

        <label for="stock\_quantity">Stock Quantity:</label>

        <input type="number" name="stock\_quantity" id="stock\_quantity" min="0" value="<?php echo htmlspecialchars($stock\_quantity); ?>" required>

    </div>

    <div class="form-group">

        <label for="category\_id">Category:</label>

        <select name="category\_id" id="category\_id" required>

            <option value="">-- Select Category --</option>

            <?php foreach ($categories as $cat): ?>

                <option value="<?php echo $cat['CategoryID']; ?>" <?php echo ($category\_id == $cat['CategoryID'] ? 'selected' : ''); ?>>

                    <?php echo htmlspecialchars($cat['CategoryName']); ?>

                </option>

            <?php endforeach; ?>

        </select>

    </div>

    <div class="form-group">

        <label for="main\_image">Main Product Image (leave blank to keep current):</label>

        <input type="file" name="main\_image" id="main\_image" accept="image/jpeg, image/png, image/gif, image/webp">

        <small>Current Image: <?php echo htmlspecialchars($current\_main\_image\_url); ?></small><br>

        <?php if ($current\_main\_image\_url && $current\_main\_image\_url != 'default\_product.png'): ?>

            <img src="<?php echo UPLOAD\_URL . htmlspecialchars($current\_main\_image\_url); ?>" alt="Current Image" style="max-width: 100px; max-height: 100px; margin-top: 5px;">

        <?php endif; ?>

    </div>

    <div class="form-group">

        <label for="is\_active">

            <input type="checkbox" name="is\_active" id="is\_active" value="1" <?php echo ($is\_active ? 'checked' : ''); ?>>

            Product is Active

        </label>

    </div>

    <div class="form-group">

        <button type="submit" class="btn">Update Product</button>

        <a href="manage\_products.php" class="btn btn-secondary">Cancel</a>

    </div>

</form>

<?php

require\_once 'admin\_footer.php';

mysqli\_close($conn);

?>

**/admin/index.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

if (is\_admin\_logged\_in()) {

    redirect('admin/dashboard.php');

}

$username\_input = "";

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username\_input = sanitize\_input($\_POST['username']);

    $password\_input = $\_POST['password'];

    if (empty($username\_input)) $errors[] = "Username is required.";

    if (empty($password\_input)) $errors[] = "Password is required.";

    if (empty($errors)) {

        $sql = "SELECT AdminID, Username, PasswordHash, Role FROM administrators WHERE Username = ?";

        if ($stmt = mysqli\_prepare($conn, $sql)) {

            mysqli\_stmt\_bind\_param($stmt, "s", $username\_input\_param);

            $username\_input\_param = $username\_input;

            if (mysqli\_stmt\_execute($stmt)) {

                mysqli\_stmt\_store\_result($stmt);

                if (mysqli\_stmt\_num\_rows($stmt) == 1) {

                    mysqli\_stmt\_bind\_result($stmt, $admin\_id, $admin\_username\_db, $hashed\_password\_from\_db, $admin\_role);

                    if (mysqli\_stmt\_fetch($stmt)) {

                        if (password\_verify($password\_input, $hashed\_password\_from\_db)) {

                            session\_regenerate\_id(true);

                            $\_SESSION['admin\_id'] = $admin\_id;

                            $\_SESSION['admin\_username'] = $admin\_username\_db;

                            $\_SESSION['admin\_role'] = $admin\_role;

                            $update\_login\_sql = "UPDATE administrators SET LastLogin = CURRENT\_TIMESTAMP WHERE AdminID = ?";

                            if ($update\_stmt = mysqli\_prepare($conn, $update\_login\_sql)) {

                                mysqli\_stmt\_bind\_param($update\_stmt, "i", $admin\_id);

                                mysqli\_stmt\_execute($update\_stmt);

                                mysqli\_stmt\_close($update\_stmt);

                            }

                            redirect('admin/dashboard.php');

                        } else {

                            $errors[] = "Invalid username or password.";

                        }

                    }

                } else {

                    $errors[] = "Invalid username or password.";

                }

            } else {

                $errors[] = "Login execution failed. DB Error: " . mysqli\_stmt\_error($stmt);

            }

            mysqli\_stmt\_close($stmt);

        } else {

            $errors[] = "Database statement preparation failed. DB Error: " . mysqli\_error($conn);

        }

    }

    if (!empty($errors)) {

        $\_SESSION['admin\_login\_errors'] = $errors;

        $\_SESSION['admin\_login\_username'] = $username\_input;

        redirect('admin/index.php');

    }

}

$login\_errors\_display = [];

if (isset($\_SESSION['admin\_login\_errors'])) {

    $login\_errors\_display = $\_SESSION['admin\_login\_errors'];

    unset($\_SESSION['admin\_login\_errors']);

}

$form\_username\_value = '';

if (isset($\_SESSION['admin\_login\_username'])) {

    $form\_username\_value = htmlspecialchars($\_SESSION['admin\_login\_username'], ENT\_QUOTES, 'UTF-8');

    unset($\_SESSION['admin\_login\_username']);

}

require\_once 'admin\_header.php';

?>

<div class="admin-login-form">

    <h2>Admin Portal Login</h2>

    <?php

    if (!empty($login\_errors\_display)) {

        echo "<div class='message-error'><ul>";

        foreach ($login\_errors\_display as $error) {

            echo "<li>" . htmlspecialchars($error) . "</li>";

        }

        echo "</ul></div>";

    }

    echo get\_admin\_session\_message();

    ?>

    <form action="<?php echo htmlspecialchars(BASE\_URL . 'admin/index.php'); ?>" method="post">

        <div class="form-group">

            <label for="username">Username:</label>

            <input type="text" name="username" id="username" value="<?php echo $form\_username\_value; ?>" required>

        </div>

        <div class="form-group">

            <label for="password">Password:</label>

            <input type="password" name="password" id="password" required>

        </div>

        <div class="form-group">

            <button type="submit" class="btn" style="width:100%;">Login</button>

        </div>

    </form>

</div>

<?php

require\_once 'admin\_footer.php';

if ($conn) mysqli\_close($conn);

?>

**/admin/logout.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

unset($\_SESSION['admin\_id']);

unset($\_SESSION['admin\_username']);

unset($\_SESSION['admin\_role']);

$\_SESSION['admin\_message'] = "<div class='message-success'>You have been logged out successfully.</div>";

redirect('admin/index.php');

?>

**/admin/manage\_categories.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

$errors = [];

$success\_msg = '';

$action = $\_GET['action'] ?? 'view';

$edit\_id = isset($\_GET['id']) ? (int)$\_GET['id'] : 0;

$category\_to\_edit = null;

function fetch\_all\_categories($conn, $parent\_id = null, $prefix = '') {

    $categories\_list = [];

    $sql = "SELECT CategoryID, CategoryName, ParentCategoryID FROM categories WHERE ";

    if ($parent\_id === null) {

        $sql .= "ParentCategoryID IS NULL ";

    } else {

        $sql .= "ParentCategoryID = ? ";

    }

    $sql .= "ORDER BY CategoryName ASC";

    $stmt = mysqli\_prepare($conn, $sql);

    if ($parent\_id !== null) {

        mysqli\_stmt\_bind\_param($stmt, "i", $parent\_id);

    }

    if (mysqli\_stmt\_execute($stmt)) {

        $result = mysqli\_stmt\_get\_result($stmt);

        while ($row = mysqli\_fetch\_assoc($result)) {

            $row['PrefixedName'] = $prefix . $row['CategoryName'];

            $categories\_list[] = $row;

            $sub\_categories = fetch\_all\_categories($conn, $row['CategoryID'], $prefix . "&nbsp;&nbsp;&nbsp;&nbsp;");

            $categories\_list = array\_merge($categories\_list, $sub\_categories);

        }

    }

    mysqli\_stmt\_close($stmt);

    return $categories\_list;

}

$all\_categories\_for\_dropdown = fetch\_all\_categories($conn);

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    if (isset($\_POST['save\_category'])) {

        $category\_id\_form = isset($\_POST['category\_id']) ? (int)$\_POST['category\_id'] : 0;

        $category\_name = sanitize\_input($\_POST['category\_name']);

        $parent\_category\_id = !empty($\_POST['parent\_category\_id']) ? (int)$\_POST['parent\_category\_id'] : null;

        if (empty($category\_name)) {

            $errors[] = "Category name is required.";

        }

        if ($parent\_category\_id !== null && $parent\_category\_id == $category\_id\_form && $category\_id\_form != 0) {

            $errors[] = "A category cannot be its own parent.";

        }

        if (empty($errors)) {

            if ($category\_id\_form > 0) {

                $sql = "UPDATE categories SET CategoryName = ?, ParentCategoryID = ? WHERE CategoryID = ?";

                $stmt = mysqli\_prepare($conn, $sql);

                mysqli\_stmt\_bind\_param($stmt, "sii", $category\_name, $parent\_category\_id, $category\_id\_form);

                $action\_word = 'updated';

            } else {

                $sql = "INSERT INTO categories (CategoryName, ParentCategoryID) VALUES (?, ?)";

                $stmt = mysqli\_prepare($conn, $sql);

                mysqli\_stmt\_bind\_param($stmt, "si", $category\_name, $parent\_category\_id);

                $action\_word = 'added';

            }

            if ($stmt && mysqli\_stmt\_execute($stmt)) {

                $\_SESSION['admin\_message'] = "<div class='message-success'>Category '{$category\_name}' {$action\_word} successfully!</div>";

                redirect('admin/manage\_categories.php');

            } else {

                $errors[] = "Failed to {$action\_word} category: " . ($stmt ? mysqli\_stmt\_error($stmt) : mysqli\_error($conn));

            }

            if ($stmt) mysqli\_stmt\_close($stmt);

        }

    }

    elseif (isset($\_POST['delete\_category'])) {

        $category\_id\_to\_delete = (int)$\_POST['category\_id\_delete'];

        $sql\_check\_sub = "SELECT COUNT(\*) as sub\_count FROM categories WHERE ParentCategoryID = ?";

        $stmt\_check\_sub = mysqli\_prepare($conn, $sql\_check\_sub);

        mysqli\_stmt\_bind\_param($stmt\_check\_sub, "i", $category\_id\_to\_delete);

        mysqli\_stmt\_execute($stmt\_check\_sub);

        $res\_sub = mysqli\_stmt\_get\_result($stmt\_check\_sub);

        $sub\_count = mysqli\_fetch\_assoc($res\_sub)['sub\_count'];

        mysqli\_stmt\_close($stmt\_check\_sub);

        if ($sub\_count > 0) {

            $errors[] = "Cannot delete category: It has subcategories. Please delete or reassign subcategories first.";

        }

        $sql\_check\_prod = "SELECT COUNT(\*) as prod\_count FROM products WHERE CategoryID = ?";

        $stmt\_check\_prod = mysqli\_prepare($conn, $sql\_check\_prod);

        mysqli\_stmt\_bind\_param($stmt\_check\_prod, "i", $category\_id\_to\_delete);

        mysqli\_stmt\_execute($stmt\_check\_prod);

        $res\_prod = mysqli\_stmt\_get\_result($stmt\_check\_prod);

        $prod\_count = mysqli\_fetch\_assoc($res\_prod)['prod\_count'];

        mysqli\_stmt\_close($stmt\_check\_prod);

        if ($prod\_count > 0) {

            $errors[] = "Cannot delete category: It has products assigned to it. Please reassign products first.";

        }

        if (empty($errors)) {

            $sql\_delete = "DELETE FROM categories WHERE CategoryID = ?";

            $stmt\_delete = mysqli\_prepare($conn, $sql\_delete);

            mysqli\_stmt\_bind\_param($stmt\_delete, "i", $category\_id\_to\_delete);

            if (mysqli\_stmt\_execute($stmt\_delete)) {

                if (mysqli\_stmt\_affected\_rows($stmt\_delete) > 0) {

                    $\_SESSION['admin\_message'] = "<div class='message-success'>Category deleted successfully.</div>";

                } else {

                    $errors[] = "Category not found or already deleted.";

                }

            } else {

                $errors[] = "Error deleting category: " . mysqli\_stmt\_error($stmt\_delete);

            }

            mysqli\_stmt\_close($stmt\_delete);

        }

        if (!empty($errors)) {

            $error\_msg\_html = "<ul>";

            foreach($errors as $err) $error\_msg\_html .= "<li>{$err}</li>";

            $error\_msg\_html .= "</ul>";

            $\_SESSION['admin\_message'] = "<div class='message-error'>{$error\_msg\_html}</div>";

        }

        redirect('admin/manage\_categories.php');

    }

}

if ($action === 'edit' && $edit\_id > 0) {

    $sql\_edit = "SELECT CategoryID, CategoryName, ParentCategoryID FROM categories WHERE CategoryID = ?";

    $stmt\_edit = mysqli\_prepare($conn, $sql\_edit);

    if ($stmt\_edit) {

        mysqli\_stmt\_bind\_param($stmt\_edit, "i", $edit\_id);

        mysqli\_stmt\_execute($stmt\_edit);

        $result\_edit = mysqli\_stmt\_get\_result($stmt\_edit);

        $category\_to\_edit = mysqli\_fetch\_assoc($result\_edit);

        mysqli\_stmt\_close($stmt\_edit);

        if (!$category\_to\_edit) {

            $\_SESSION['admin\_message'] = "<div class='message-error'>Category not found for editing.</div>";

            redirect('admin/manage\_categories.php');

        }

    }

}

$display\_categories = fetch\_all\_categories($conn);

?>

<h1>Manage Categories</h1>

<?php

echo get\_admin\_session\_message();

if (!empty($errors) && $\_SERVER["REQUEST\_METHOD"] != "POST") {

    echo "<div class='message-error'><ul>";

    foreach ($errors as $error) {

        echo "<li>" . htmlspecialchars($error) . "</li>";

    }

    echo "</ul></div>";

}

?>

<div style="background-color: #fff; padding: 20px; border-radius: 8px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 30px;">

    <h3><?php echo ($action === 'edit' && $category\_to\_edit) ? 'Edit Category: ' . htmlspecialchars($category\_to\_edit['CategoryName']) : 'Add New Category'; ?></h3>

    <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?><?php echo ($action === 'edit' && $edit\_id) ? '?action=edit&id='.$edit\_id : ''; ?>" method="POST">

        <?php if ($action === 'edit' && $category\_to\_edit): ?>

            <input type="hidden" name="category\_id" value="<?php echo $category\_to\_edit['CategoryID']; ?>">

        <?php endif; ?>

        <div class="form-group">

            <label for="category\_name">Category Name:</label>

            <input type="text" name="category\_name" id="category\_name" value="<?php echo htmlspecialchars($category\_to\_edit['CategoryName'] ?? ''); ?>" required>

        </div>

        <div class="form-group">

            <label for="parent\_category\_id">Parent Category (Optional - for subcategory):</label>

            <select name="parent\_category\_id" id="parent\_category\_id">

                <option value="">-- None (Top Level Category) --</option>

                <?php foreach ($all\_categories\_for\_dropdown as $cat\_opt): ?>

                    <?php

                    if ($action === 'edit' && $category\_to\_edit && $cat\_opt['CategoryID'] == $category\_to\_edit['CategoryID']) continue;

                    ?>

                    <option value="<?php echo $cat\_opt['CategoryID']; ?>"

                        <?php echo (($category\_to\_edit['ParentCategoryID'] ?? null) == $cat\_opt['CategoryID'] ? 'selected' : ''); ?>>

                        <?php echo htmlspecialchars($cat\_opt['PrefixedName']); ?>

                    </option>

                <?php endforeach; ?>

            </select>

        </div>

        <div class="form-group">

            <button type="submit" name="save\_category" class="btn">

                <?php echo ($action === 'edit' && $category\_to\_edit) ? 'Update Category' : 'Add Category'; ?>

            </button>

            <?php if ($action === 'edit' && $category\_to\_edit): ?>

                <a href="manage\_categories.php" class="btn btn-secondary">Cancel Edit</a>

            <?php endif; ?>

        </div>

    </form>

</div>

<h3>Existing Categories</h3>

<?php if (empty($display\_categories)): ?>

    <p class="message-info">No categories found. Add your first category using the form above!</p>

<?php else: ?>

    <table class="admin-table">

        <thead>

            <tr>

                <th>ID</th>

                <th>Category Name</th>

                <th>Parent ID</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($display\_categories as $category): ?>

                <tr>

                    <td><?php echo $category['CategoryID']; ?></td>

                    <td><?php echo $category['PrefixedName'];  ?></td>

                    <td><?php echo $category['ParentCategoryID'] ?? 'N/A (Top)'; ?></td>

                    <td class="actions">

                        <a href="manage\_categories.php?action=edit&id=<?php echo $category['CategoryID']; ?>" class="edit-link" title="Edit">&#9998;</a>

                        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST" style="display:inline;" onsubmit="return confirm('Are you sure you want to delete this category? This action cannot be undone and may fail if products or subcategories are linked.');">

                            <input type="hidden" name="category\_id\_delete" value="<?php echo $category['CategoryID']; ?>">

                            <button type="submit" name="delete\_category" class="delete-link" title="Delete" style="border:none; background:none; cursor:pointer; padding:0; color:red;">&#10006;</button>

                        </form>

                    </td>

                </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

<?php endif; ?>

<?php

require\_once 'admin\_footer.php';

mysqli\_close($conn);

?>

**/admin/manage\_orders.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

$errors = [];

$success\_msg = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['update\_order\_status'])) {

    $order\_id\_to\_update = (int)$\_POST['order\_id'];

    $new\_status = sanitize\_input($\_POST['order\_status']);

    $tracking\_number = isset($\_POST['tracking\_number']) ? sanitize\_input($\_POST['tracking\_number']) : null;

    if (empty($new\_status)) {

        $errors[] = "Order status cannot be empty.";

    }

    $allowed\_statuses = ['Pending', 'Processing', 'Shipped', 'Delivered', 'Cancelled', 'Refunded (Simulated)'];

    if (!in\_array($new\_status, $allowed\_statuses)) {

        $errors[] = "Invalid order status selected.";

    }

    if (empty($errors)) {

        $sql\_update = "UPDATE orders SET OrderStatus = ?";

        $params\_types = "s";

        $params\_values = [$new\_status];

        if ($new\_status === 'Shipped' && !empty($tracking\_number)) {

            $sql\_update .= ", TrackingNumber = ?";

            $params\_types .= "s";

            $params\_values[] = $tracking\_number;

        } elseif ($new\_status !== 'Shipped') {

            $sql\_update .= ", TrackingNumber = NULL";

        }

        $sql\_update .= " WHERE OrderID = ?";

        $params\_types .= "i";

        $params\_values[] = $order\_id\_to\_update;

        $stmt\_update = mysqli\_prepare($conn, $sql\_update);

        if ($stmt\_update) {

            mysqli\_stmt\_bind\_param($stmt\_update, $params\_types, ...$params\_values);

            if (mysqli\_stmt\_execute($stmt\_update)) {

                $\_SESSION['admin\_message'] = "<div class='message-success'>Order #{$order\_id\_to\_update} status updated to '{$new\_status}'.</div>";

            } else {

                $errors[] = "Failed to update order status: " . mysqli\_stmt\_error($stmt\_update);

            }

            mysqli\_stmt\_close($stmt\_update);

        } else {

            $errors[] = "Database error preparing update statement: " . mysqli\_error($conn);

        }

    }

    if (!empty($errors)) {

        $error\_msg\_html = "<ul>";

        foreach($errors as $err) $error\_msg\_html .= "<li>{$err}</li>";

        $error\_msg\_html .= "</ul>";

        $\_SESSION['admin\_message'] = "<div class='message-error'>{$error\_msg\_html}</div>";

    }

    $redirect\_url ='admin/manage\_orders.php';

    if(isset($\_POST['filter\_status\_redirect']) && !empty($\_POST['filter\_status\_redirect'])) {

        $redirect\_url .= '?filter\_status=' . urlencode($\_POST['filter\_status\_redirect']);

    }

    redirect($redirect\_url);

}

$orders = [];

$filter\_status = isset($\_GET['filter\_status']) ? sanitize\_input($\_GET['filter\_status']) : '';

$view\_order\_id = isset($\_GET['view\_order\_id']) ? (int)$\_GET['view\_order\_id'] : 0;

$sql = "SELECT o.OrderID, o.OrderDate, o.TotalAmount, o.OrderStatus, o.PaymentMethod, o.TrackingNumber,

               u.UserID, u.FirstName, u.LastName, u.Email as UserEmail,

               a.Street, a.City, a.State, a.ZipCode, a.Country

        FROM orders o

        JOIN users u ON o.UserID = u.UserID

        JOIN addresses a ON o.ShippingAddressID = a.AddressID ";

if (!empty($filter\_status)) {

    $sql .= " WHERE o.OrderStatus = ? ";

} elseif ($view\_order\_id > 0) {

    $sql .= " WHERE o.OrderID = ? ";

}

$sql .= " ORDER BY o.OrderDate DESC";

$stmt = mysqli\_prepare($conn, $sql);

if ($stmt) {

    if (!empty($filter\_status)) {

        mysqli\_stmt\_bind\_param($stmt, "s", $filter\_status);

    } elseif ($view\_order\_id > 0) {

        mysqli\_stmt\_bind\_param($stmt, "i", $view\_order\_id);

    }

    mysqli\_stmt\_execute($stmt);

    $result = mysqli\_stmt\_get\_result($stmt);

    while ($row = mysqli\_fetch\_assoc($result)) {

        $orders[] = $row;

    }

    mysqli\_stmt\_close($stmt);

} else {

    $\_SESSION['admin\_message'] = "<div class='message-error'>Error fetching orders: " . mysqli\_error($conn) . "</div>";

}

$order\_statuses = ['Pending', 'Processing', 'Shipped', 'Delivered', 'Cancelled', 'Refunded (Simulated)'];

?>

<h1>Manage Orders</h1>

<?php

echo get\_admin\_session\_message();

?>

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="GET" style="margin-bottom: 20px; background-color: #fff; padding: 15px; border-radius: 5px; box-shadow: 0 1px 3px rgba(0,0,0,0.1);">

    <div class="form-group" style="display: inline-block; margin-right: 10px;">

        <label for="filter\_status">Filter by Status:</label>

        <select name="filter\_status" id="filter\_status" onchange="this.form.submit()">

            <option value="">All Statuses</option>

            <?php foreach ($order\_statuses as $status): ?>

                <option value="<?php echo $status; ?>" <?php echo ($filter\_status == $status ? 'selected' : ''); ?>>

                    <?php echo htmlspecialchars($status); ?>

                </option>

            <?php endforeach; ?>

        </select>

    </div>

    <?php if (!empty($filter\_status) || $view\_order\_id > 0): ?>

        <a href="manage\_orders.php" class="btn btn-secondary">Clear Filter/View</a>

    <?php endif; ?>

</form>

<?php if ($view\_order\_id > 0 && count($orders) === 1): ?>

    <?php $order = $orders[0]; ?>

    <div style="background-color: #fff; padding: 20px; border-radius: 8px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 30px;">

        <h3>Order Details #<?php echo htmlspecialchars($order['OrderID']); ?></h3>

        <div style="display: flex; flex-wrap: wrap; gap: 20px;">

            <div style="flex:1; min-width:250px;">

                <p><strong>Customer:</strong> <?php echo htmlspecialchars($order['FirstName'] . ' ' . $order['LastName']); ?> (<?php echo htmlspecialchars($order['UserEmail']); ?>)</p>

                <p><strong>Order Date:</strong> <?php echo date("M j, Y, g:i a", strtotime($order['OrderDate'])); ?></p>

                <p><strong>Total Amount:</strong> ₹<?php echo number\_format($order['TotalAmount'], 2); ?></p>

                <p><strong>Payment Method:</strong> <?php echo htmlspecialchars($order['PaymentMethod']); ?></p>

            </div>

            <div style="flex:1; min-width:250px;">

                <p><strong>Shipping Address:</strong><br>

                    <?php echo htmlspecialchars($order['Street']); ?><br>

                    <?php echo htmlspecialchars($order['City']); ?>, <?php echo htmlspecialchars($order['State']); ?> - <?php echo htmlspecialchars($order['ZipCode']); ?><br>

                    <?php echo htmlspecialchars($order['Country']); ?>

                </p>

            </div>

        </div>

        <h4 style="margin-top:20px;">Items Ordered:</h4>

        <?php

        $order\_items = [];

        $sql\_items = "SELECT oi.ProductID, p.ProductName, oi.Quantity, oi.PriceAtPurchase, oi.Subtotal

                      FROM order\_items oi JOIN products p ON oi.ProductID = p.ProductID

                      WHERE oi.OrderID = ?";

        $stmt\_items = mysqli\_prepare($conn, $sql\_items);

        mysqli\_stmt\_bind\_param($stmt\_items, "i", $order['OrderID']);

        mysqli\_stmt\_execute($stmt\_items);

        $res\_items = mysqli\_stmt\_get\_result($stmt\_items);

        while($item\_row = mysqli\_fetch\_assoc($res\_items)) $order\_items[] = $item\_row;

        mysqli\_stmt\_close($stmt\_items);

        ?>

        <table class="admin-table" style="font-size:0.9em;">

            <thead><tr><th>Product ID</th><th>Name</th><th>Qty</th><th>Price Paid</th><th>Subtotal</th></tr></thead>

            <tbody>

            <?php foreach($order\_items as $item): ?>

                <tr>

                    <td><?php echo $item['ProductID']; ?></td>

                    <td><?php echo htmlspecialchars($item['ProductName']); ?></td>

                    <td><?php echo $item['Quantity']; ?></td>

                    <td>₹<?php echo number\_format($item['PriceAtPurchase'], 2); ?></td>

                    <td>₹<?php echo number\_format($item['Subtotal'], 2); ?></td>

                </tr>

            <?php endforeach; ?>

            </tbody>

        </table>

        <h4 style="margin-top:20px;">Update Order Status:</h4>

        <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

            <input type="hidden" name="order\_id" value="<?php echo $order['OrderID']; ?>">

            <input type="hidden" name="filter\_status\_redirect" value="<?php echo htmlspecialchars($filter\_status); ?>"> <div class="form-group">

                <label for="order\_status\_update\_<?php echo $order['OrderID']; ?>">Status:</label>

                <select name="order\_status" id="order\_status\_update\_<?php echo $order['OrderID']; ?>">

                    <?php foreach ($order\_statuses as $status\_option): ?>

                        <option value="<?php echo $status\_option; ?>" <?php echo ($order['OrderStatus'] == $status\_option ? 'selected' : ''); ?>>

                            <?php echo htmlspecialchars($status\_option); ?>

                        </option>

                    <?php endforeach; ?>

                </select>

            </div>

            <div class="form-group" id="tracking\_number\_div\_<?php echo $order['OrderID']; ?>" style="<?php echo ($order['OrderStatus'] == 'Shipped' || (isset($\_POST['order\_status']) && $\_POST['order\_status'] == 'Shipped') ? '' : 'display:none;'); ?>">

                <label for="tracking\_number\_<?php echo $order['OrderID']; ?>">Tracking Number:</label>

                <input type="text" name="tracking\_number" id="tracking\_number\_<?php echo $order['OrderID']; ?>" value="<?php echo htmlspecialchars($order['TrackingNumber'] ?? ''); ?>">

            </div>

            <button type="submit" name="update\_order\_status" class="btn">Update Status</button>

        </form>

        <script>

            document.getElementById('order\_status\_update\_<?php echo $order['OrderID']; ?>').addEventListener('change', function() {

                var trackingDiv = document.getElementById('tracking\_number\_div\_<?php echo $order['OrderID']; ?>');

                if (this.value === 'Shipped') {

                    trackingDiv.style.display = 'block';

                } else {

                    trackingDiv.style.display = 'none';

                    document.getElementById('tracking\_number\_<?php echo $order['OrderID']; ?>').value = '';

                }

            });

        </script>

    </div>

<?php elseif (empty($orders)): ?>

    <p class="message-info">No orders found matching your criteria.</p>

<?php else:  ?>

    <table class="admin-table">

        <thead>

            <tr>

                <th>Order ID</th>

                <th>Customer</th>

                <th>Date</th>

                <th>Total</th>

                <th>Status</th>

                <th>Tracking #</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($orders as $order): ?>

                <tr>

                    <td>#<?php echo $order['OrderID']; ?></td>

                    <td><?php echo htmlspecialchars($order['FirstName'] . ' ' . $order['LastName']); ?><br><small><?php echo htmlspecialchars($order['UserEmail']); ?></small></td>

                    <td><?php echo date("M j, Y, g:i a", strtotime($order['OrderDate'])); ?></td>

                    <td>₹<?php echo number\_format($order['TotalAmount'], 2); ?></td>

                    <td><?php echo htmlspecialchars($order['OrderStatus']); ?></td>

                    <td><?php echo htmlspecialchars($order['TrackingNumber'] ?? 'N/A'); ?></td>

                    <td class="actions">

                        <a href="manage\_orders.php?view\_order\_id=<?php echo $order['OrderID']; ?><?php echo !empty($filter\_status) ? '&filter\_status='.urlencode($filter\_status) : ''; ?>" class="btn btn-secondary" style="font-size:0.8em; padding:3px 8px;">View/Update</a>

                    </td>

                </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

<?php endif; ?>

<?php

require\_once 'admin\_footer.php';

mysqli\_close($conn);

?>

**/admin/manage\_products.php**

<?php

require\_once \_\_DIR\_\_ . '/../config.php';

require\_admin\_login('index.php');

require\_once 'admin\_header.php';

$delete\_msg = '';

$delete\_type = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['delete\_product'])) {

    $product\_id\_to\_delete = (int)$\_POST['product\_id'];

    $sql\_delete = "UPDATE products SET IsActive = FALSE WHERE ProductID = ?";

    $stmt\_delete = mysqli\_prepare($conn, $sql\_delete);

    if ($stmt\_delete) {

        mysqli\_stmt\_bind\_param($stmt\_delete, "i", $product\_id\_to\_delete);

        if (mysqli\_stmt\_execute($stmt\_delete)) {

            if (mysqli\_stmt\_affected\_rows($stmt\_delete) > 0) {

                $\_SESSION['admin\_message'] = "<div class='message-success'>Product (ID: {$product\_id\_to\_delete}) deactivated successfully.</div>";

            } else {

                $\_SESSION['admin\_message'] = "<div class='message-error'>Product not found or already inactive.</div>";

            }

        } else {

            $\_SESSION['admin\_message'] = "<div class='message-error'>Error deactivating product: " . mysqli\_stmt\_error($stmt\_delete) . "</div>";

        }

        mysqli\_stmt\_close($stmt\_delete);

    } else {

        $\_SESSION['admin\_message'] = "<div class='message-error'>Error preparing deactivation statement.</div>";

    }

    redirect('admin/manage\_products.php');

}

$products = [];

$sql = "SELECT p.ProductID, p.ProductName, p.SellingPrice, p.StockQuantity, p.IsActive, c.CategoryName, p.MainImageURL

        FROM products p

        JOIN categories c ON p.CategoryID = c.CategoryID

        ORDER BY p.DateAdded DESC";

$result = mysqli\_query($conn, $sql);

if ($result) {

    while ($row = mysqli\_fetch\_assoc($result)) {

        $products[] = $row;

    }

} else {

    $\_SESSION['admin\_message'] = "<div class='message-error'>Error fetching products: " . mysqli\_error($conn) . "</div>";

}

?>

<h1>Manage Products</h1>

<p><a href="add\_product.php" class="btn">Add New Product</a></p>

<?php

echo get\_admin\_session\_message();

?>

<?php if (empty($products)): ?>

    <p class="message-info">No products found. <a href="add\_product.php">Add your first product!</a></p>

<?php else: ?>

    <table class="admin-table">

        <thead>

            <tr>

                <th>Image</th>

                <th>ID</th>

                <th>Name</th>

                <th>Category</th>

                <th>Price</th>

                <th>Stock</th>

                <th>Status</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>

            <?php foreach ($products as $product): ?>

                <tr>

                    <td>

                        <img src="<?php echo UPLOAD\_URL . htmlspecialchars($product['MainImageURL'] ? $product['MainImageURL'] : 'default\_product.png'); ?>" alt="<?php echo htmlspecialchars($product['ProductName']); ?>" style="width: 60px; height: 60px; object-fit: cover;">

                    </td>

                    <td><?php echo $product['ProductID']; ?></td>

                    <td><?php echo htmlspecialchars($product['ProductName']); ?></td>

                    <td><?php echo htmlspecialchars($product['CategoryName']); ?></td>

                    <td>₹<?php echo number\_format($product['SellingPrice'], 2); ?></td>

                    <td><?php echo $product['StockQuantity']; ?></td>

                    <td><?php echo $product['IsActive'] ? '<span style="color:green;">Active</span>' : '<span style="color:red;">Inactive</span>'; ?></td>

                    <td class="actions">

                        <a href="edit\_product.php?id=<?php echo $product['ProductID']; ?>" class="edit-link" title="Edit">&#9998;</a> <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST" style="display:inline;" onsubmit="return confirm('Are you sure you want to deactivate this product? Users will not be able to see or purchase it.');">

                            <input type="hidden" name="product\_id" value="<?php echo $product['ProductID']; ?>">

                            <button type="submit" name="delete\_product" class="delete-link" title="Deactivate" style="border:none; background:none; cursor:pointer; padding:0; color:red;">&#10006;</button> </form>

                    </td>

                </tr>

            <?php endforeach; ?>

        </tbody>

    </table>

<?php endif; ?>

<?php

require\_once 'admin\_footer.php';

mysqli\_close($conn);

?

# **TESTING**

## **Testing Strategy**

The testing strategy will involve manual tests. The primary goal is to ensure all functional requirements are met, the system is usable, secure against common vulnerabilities, and performs adequately under expected load for a demonstration. Testing will be an iterative process throughout the development lifecycle.

## **Types of Testing**

**1. Unit Testing**

* **Objective:** To test individual components or modules of the software in isolation.
* **Scope:** Test critical functions like input validation, database interaction functions, calculation logic (e.g., cart total), and authentication logic.
* **Method:** Primarily manual testing by the developer during coding.

**2. Integration Testing**

* **Objective:** To test the interaction between different modules of the system.
* **Scope:** Verify data flow and control flow between modules, e.g., user registration leading to database entry and then successful login; adding a product by admin and its display on the customer frontend; cart addition leading to order placement and stock update.
* **Method:** Manual testing by combining modules and testing their combined functionality.

**3. System Testing**

* **Objective:** To test the complete, integrated system against the specified requirements in this SRS.
* **Scope:** End-to-end testing of all use cases, including user registration, login, product browsing, cart management, checkout, order placement, admin functionalities (product, category, order, user management).
* **Method:** Manual execution of test scenarios based on the functional and non-functional requirements.

**4. User Acceptance Testing (UAT)**

* **Objective:** To validate that the system meets the needs of the user.
* **Scope:** Developer will perform UAT by using the system as an end-user and as an administrator to ensure it is fit for purpose as defined by the project goals. Supervisor feedback will also form part of UAT.
* **Method:** Scenario-based testing.

**5. Security Testing**

* **Objective:** To identify and address common web vulnerabilities.
* **Scope:** Test for SQL injection vulnerabilities (especially in forms and URL parameters), Cross-Site Scripting (XSS) by attempting to input malicious scripts, ensure password hashing is implemented, and check session management security.
* **Method:** Manual penetration testing techniques for common vulnerabilities.

## **Test Environment**

* **Development & Testing:** XAMPP environment on Windows 11 (Apache, MySQL)
* **Browsers:** Testing will be performed on the latest versions of Google Chrome.

## **Test Cases**

* User Registration (valid, invalid inputs, existing email)
* User Login (valid, invalid credentials, deactivated account)
* Password Change & Reset
* Profile & Address Management
* Product Browsing & Search (by keyword, category)
* Product Filtering
* Shopping Cart (add, update quantity, remove, clear)
* Checkout Process (address selection, simulated payment)
* Order Placement & Stock Update
* Order History Viewing
* Admin Login
* Admin Product Management (CRUD, image upload)
* Admin Category Management (CRUD, parent-child relationships)
* Admin Order Management (view details, update status)
* Responsiveness on different screen sizes (desktop, simulated mobile).
* Basic security checks (e.g., attempting SQL injection in search).

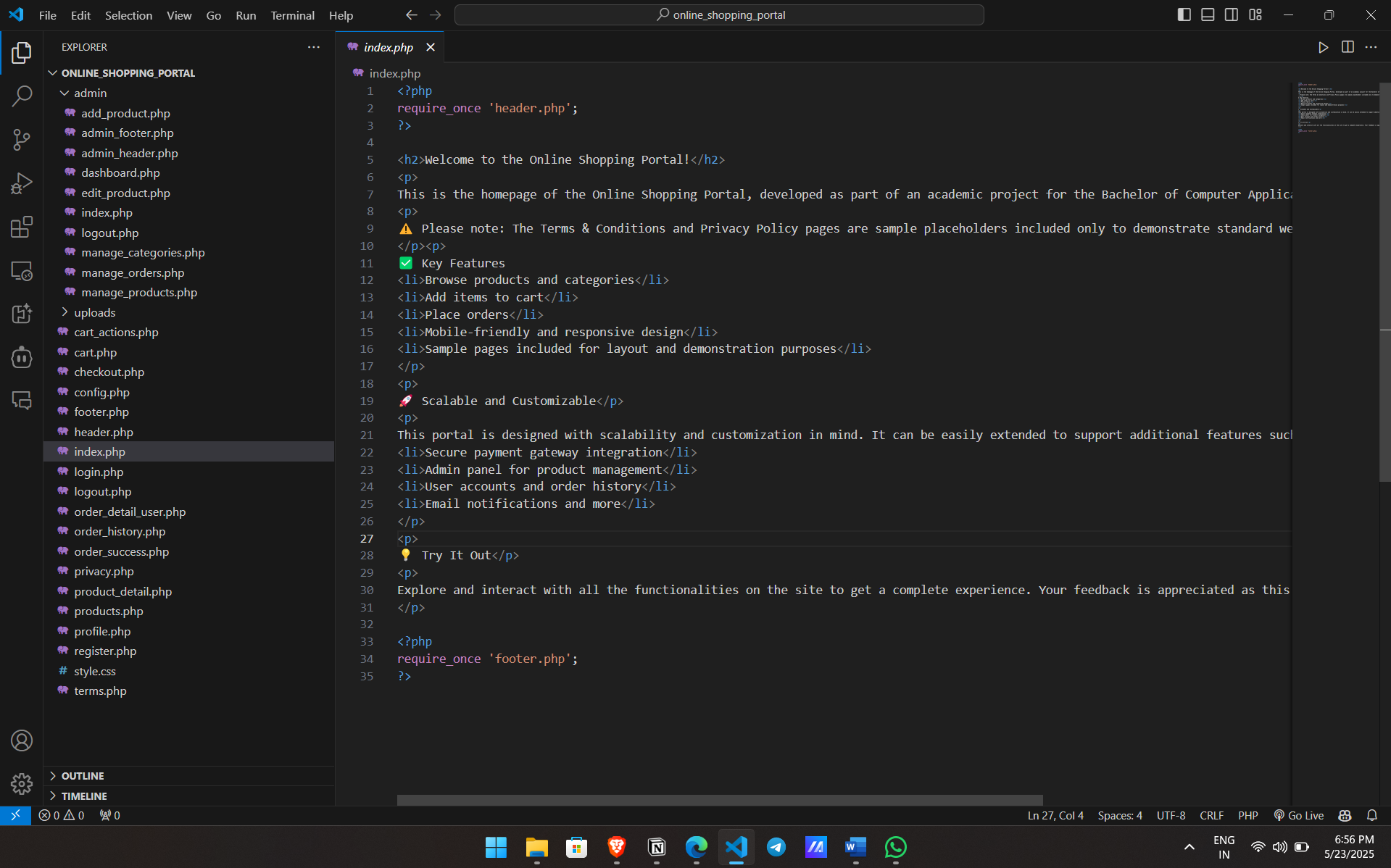
## **Test Report**

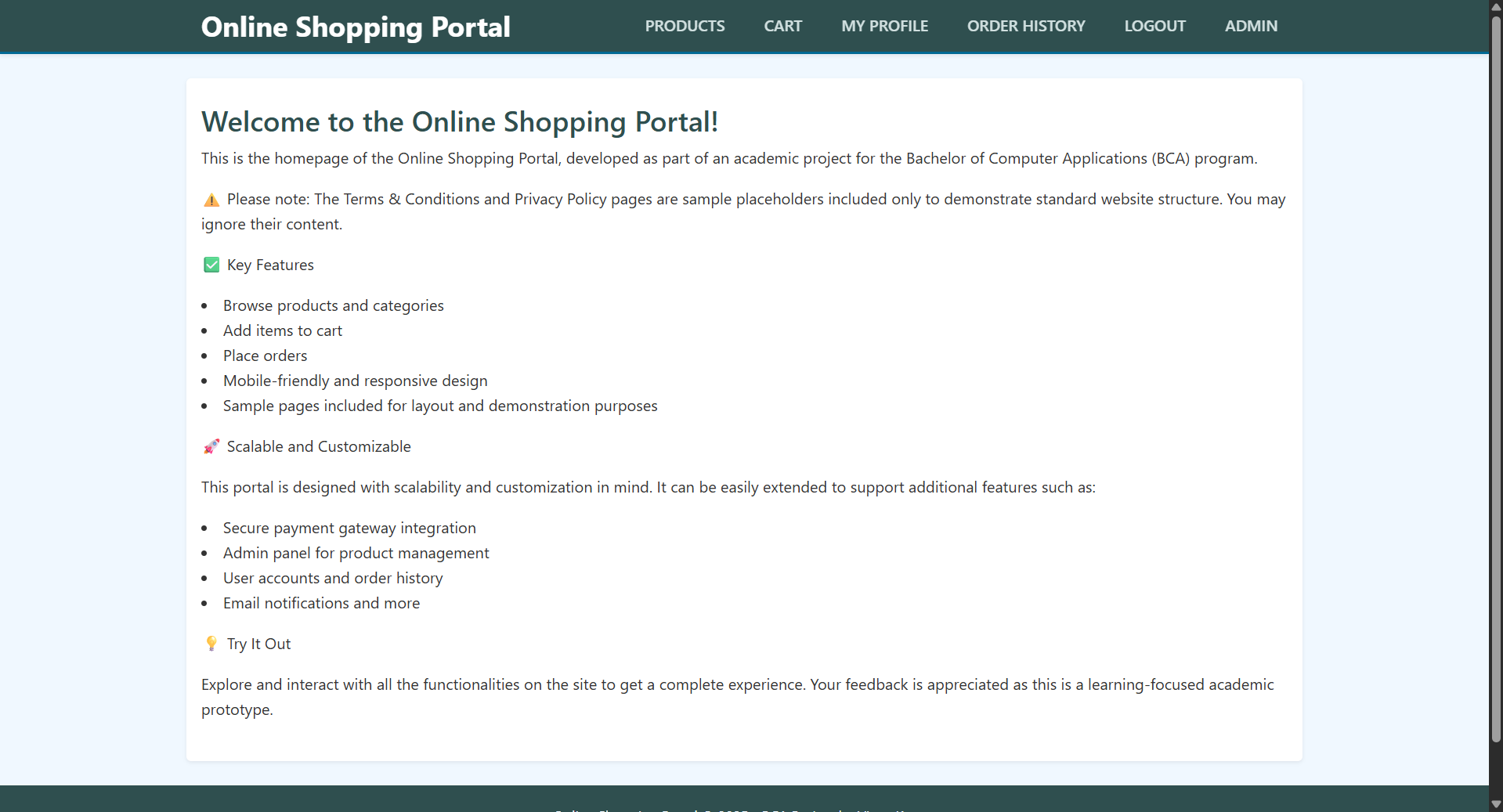
|  |  |  |  |
| --- | --- | --- | --- |
| **Test case id** | **Test case description** | **Test Date** | **Pass / Fail** |
| 001 | User registration | 18-04-2025 | Pass |
| 002 | User Login | 20-04-2025 | Pass |
| 003 | Password change & reset | 21-04-2025 | Pass |
| 004 | Profile & Address Management | 21-04-2025 | Pass |
| 005 | Product Browsing & Search | 23-04-2025 | Pass |
| 006 | Product Filtering | 23-04-2025 | Pass |
| 007 | Shopping Cart | 27-04-2025 | Pass |
| 008 | Checkout Process | 3-05-2025 | Pass |
| 009 | Order Placing & stock update | 4-05-2025 | Pass |
| 010 | Order History Viewing | 4-05-2025 | Pass |
| 011 | Admin Login | 10-05-2025 | Pass |
| 012 | Admin Product Management | 12-05-2025 | Pass |
| 013 | Admin Category Management | 13-05-2025 | Pass |
| 014 | Admin Order Management | 14-05-2025 | Pass |
| 015 | Responsiveness on different screen sizes | 18-05-2025 | Fail |
| 016 | Basic Security Checks | 18-05-2025 | Pass |

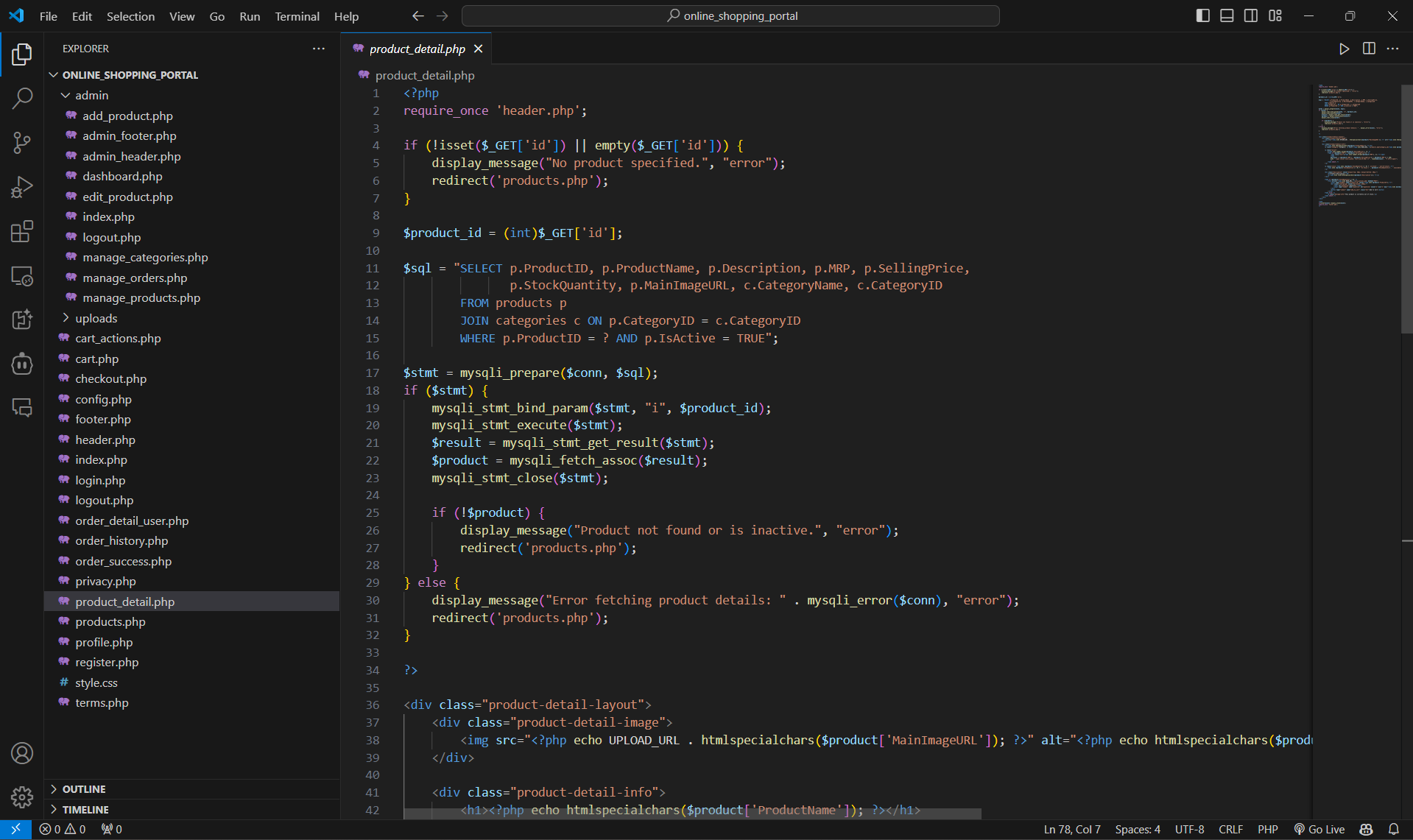
**For Failed Test case id 015**

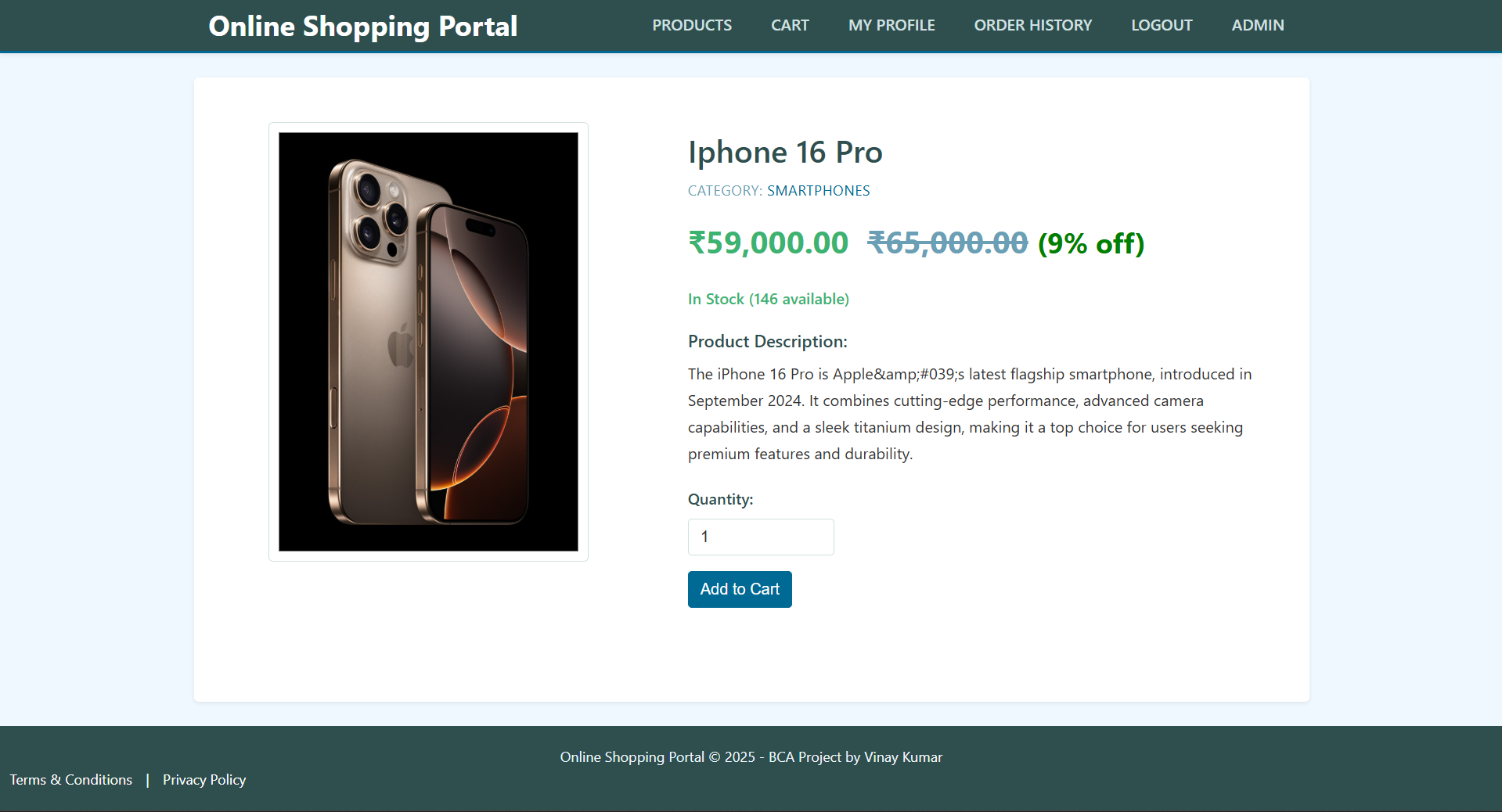
After Deployment to cloud I tested the responsive design of the webpage and its full functionalities in different devices like laptops, computers, tabs, android phones and iPhone. In all those only Android and iPhone are not able to show the webpage as well written and designed. But the functionality is still working. I will try to fix this issue in future.

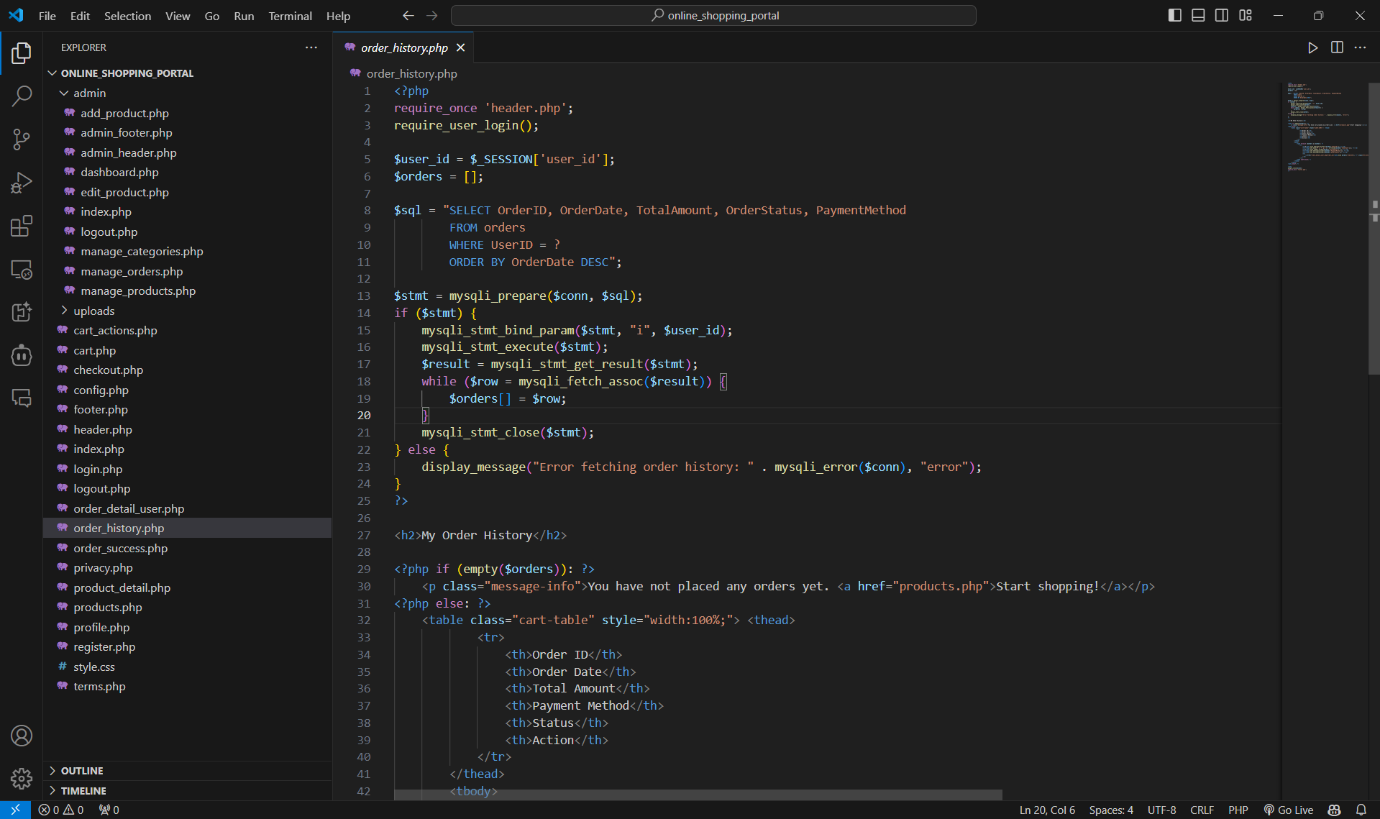
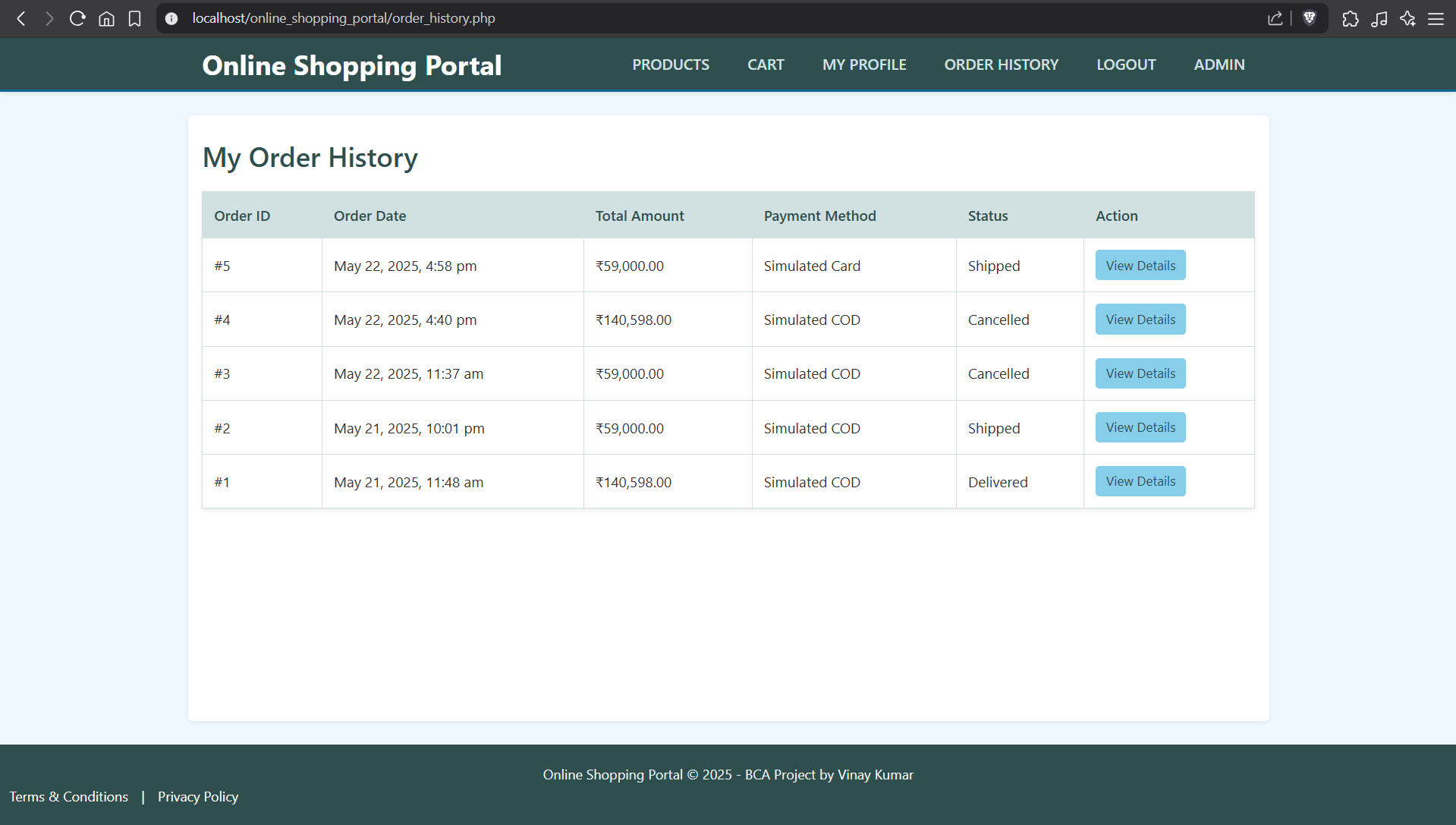
# **INPUT & OUTPUT SCREENS**

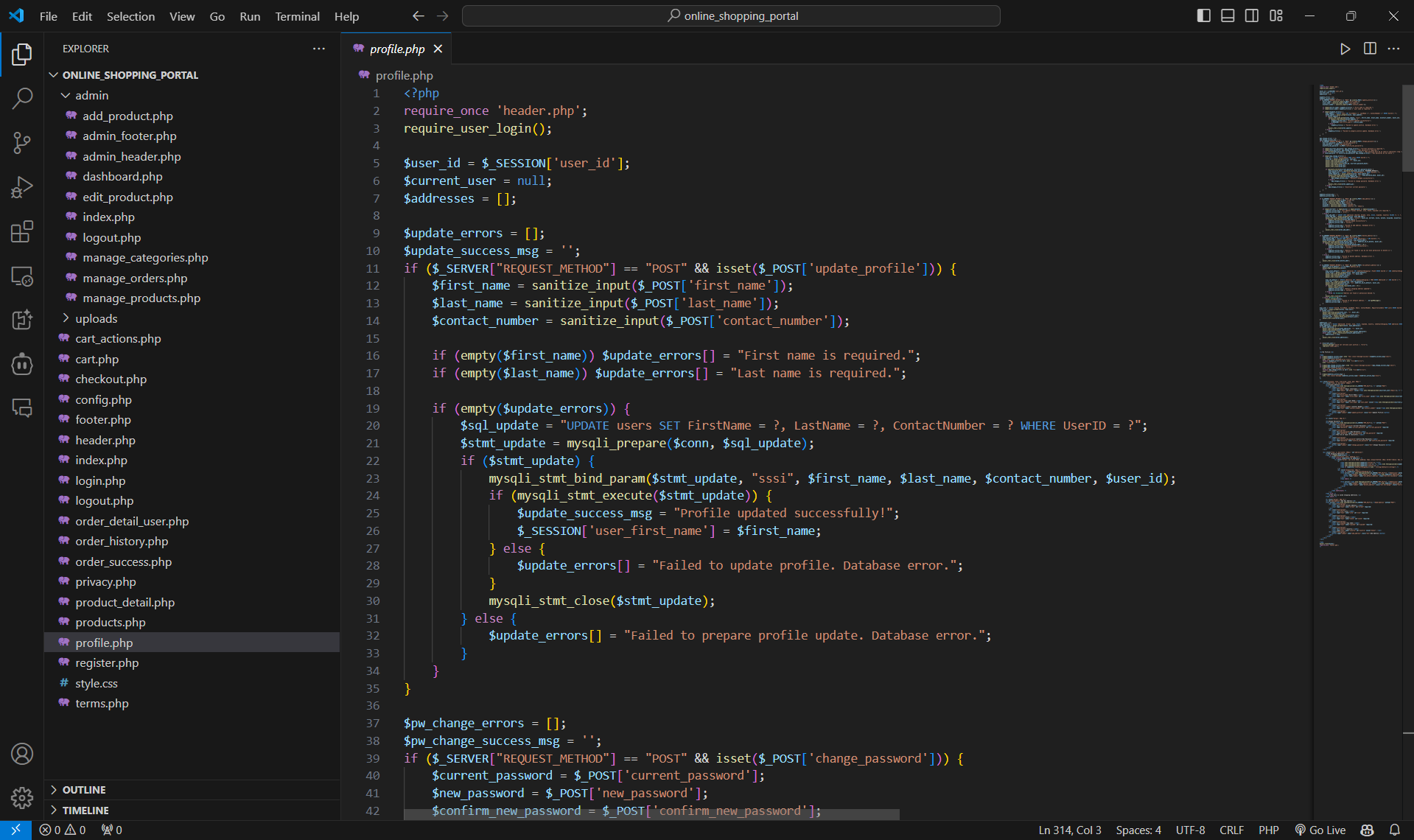


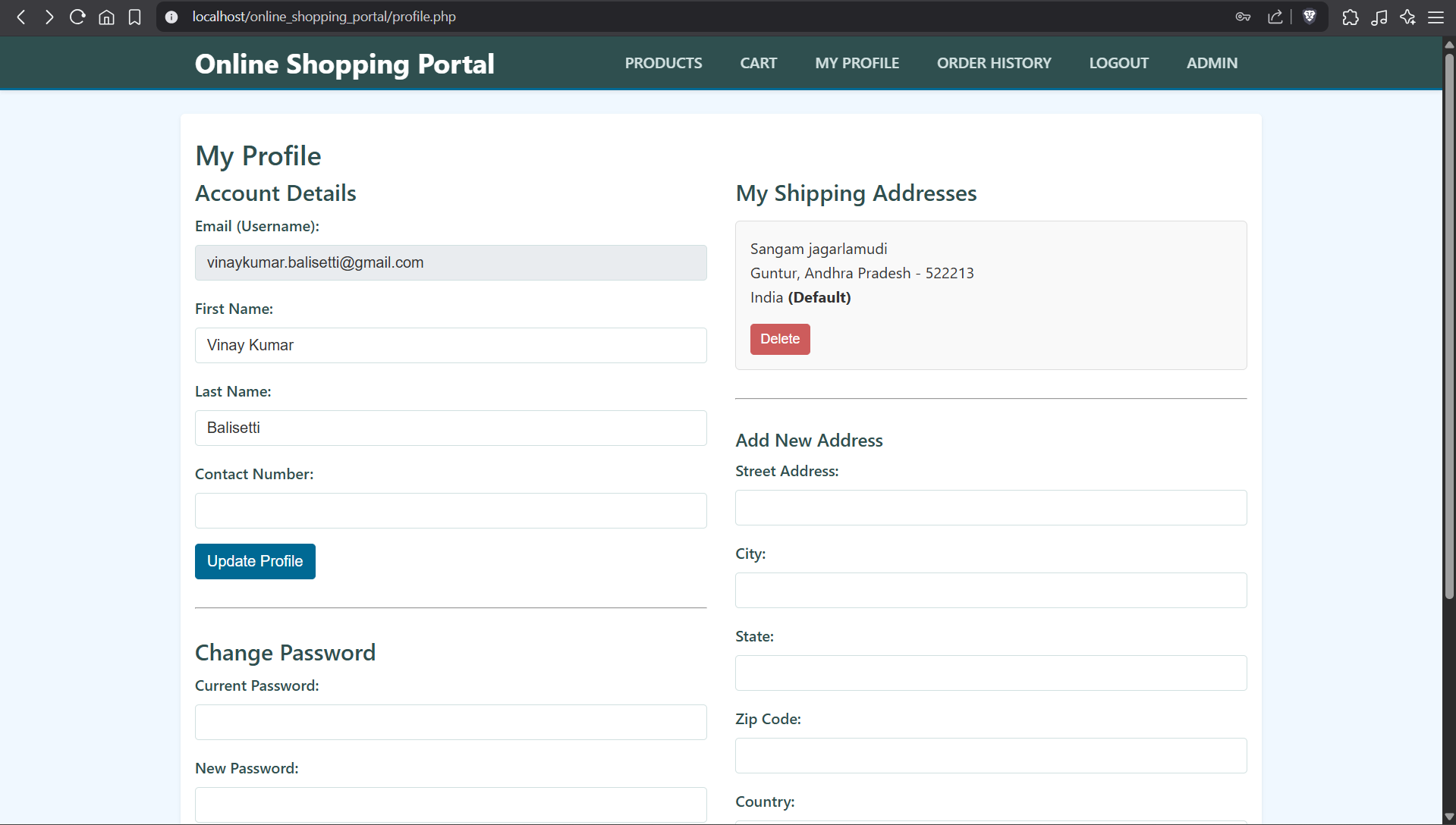


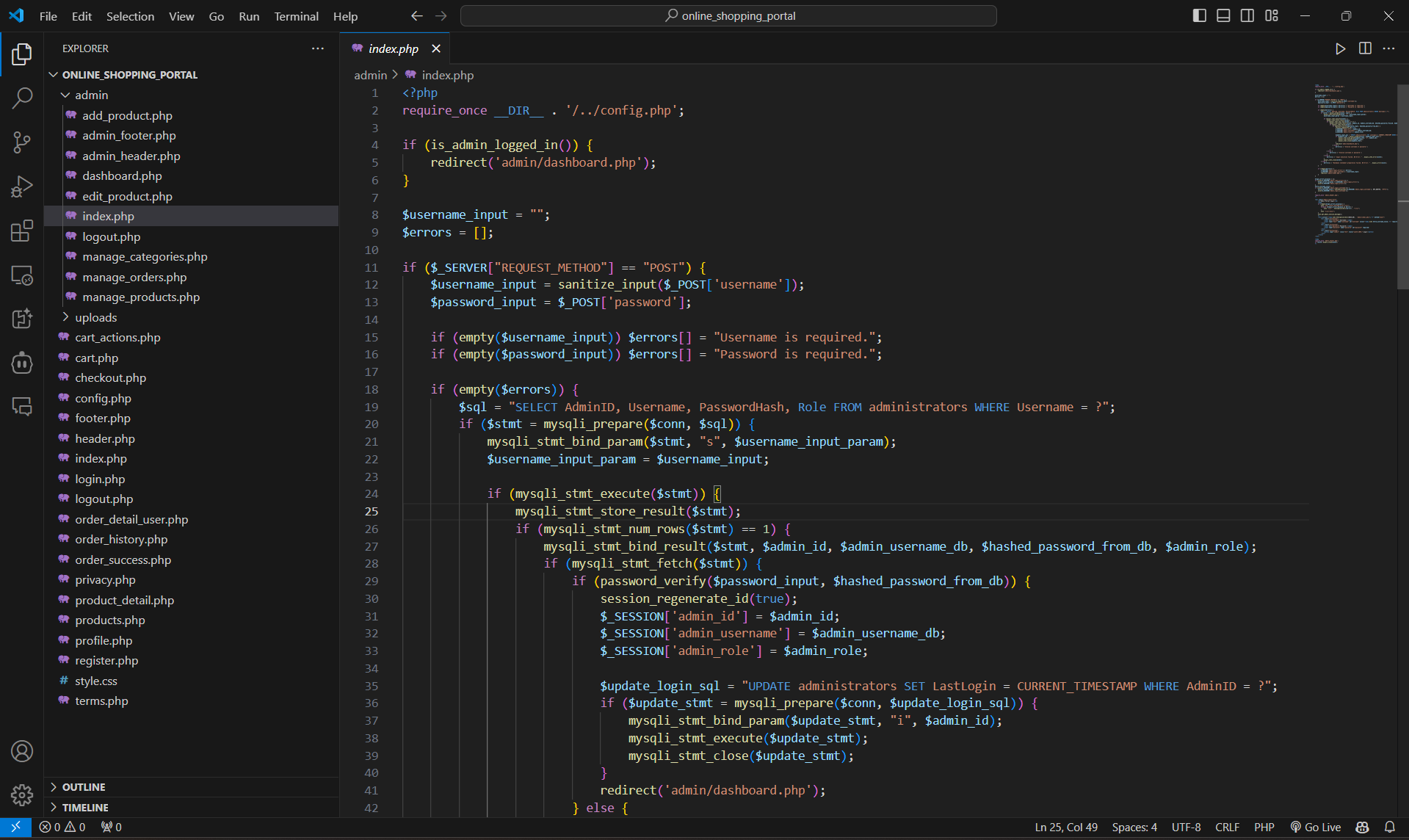


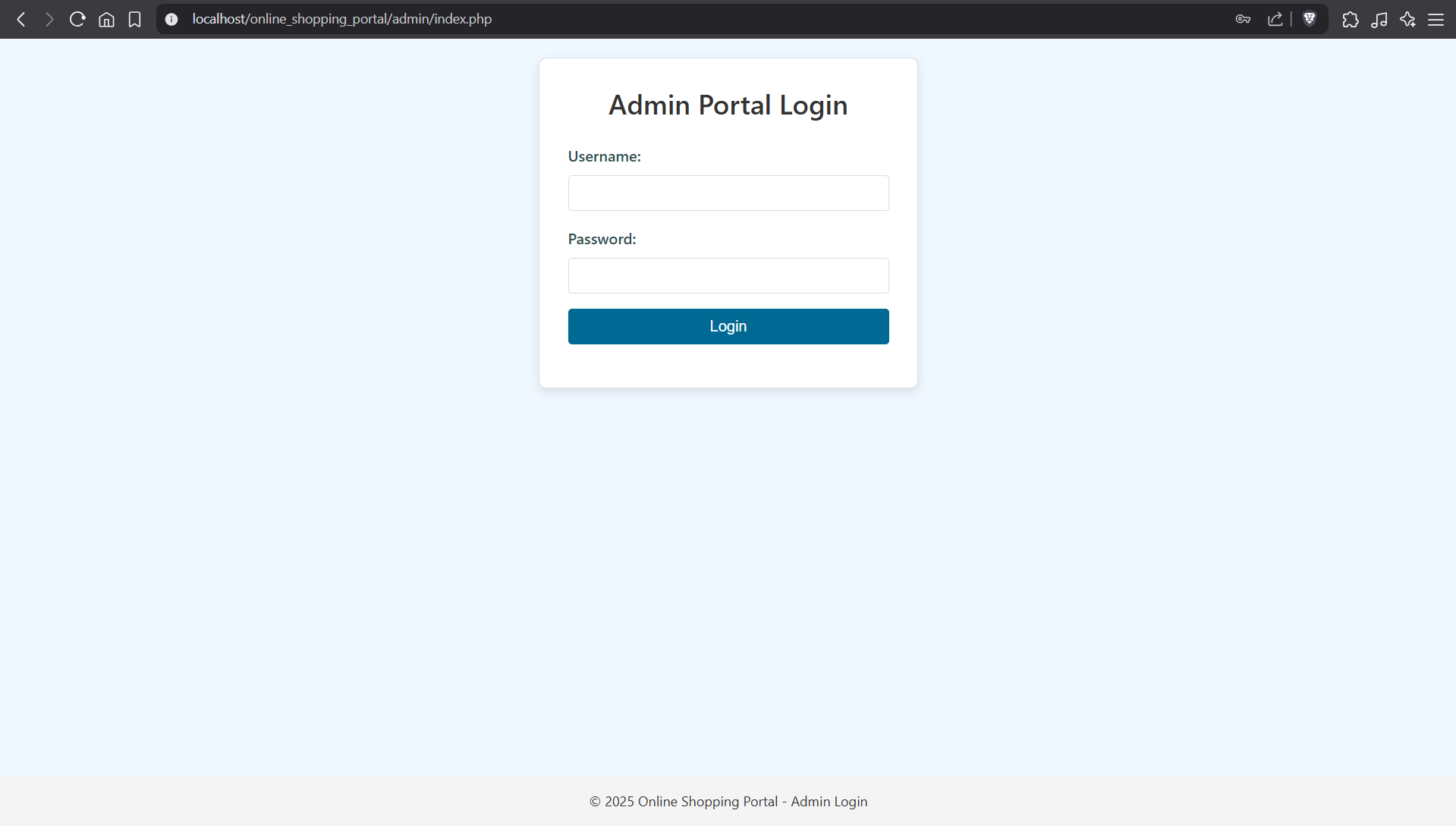


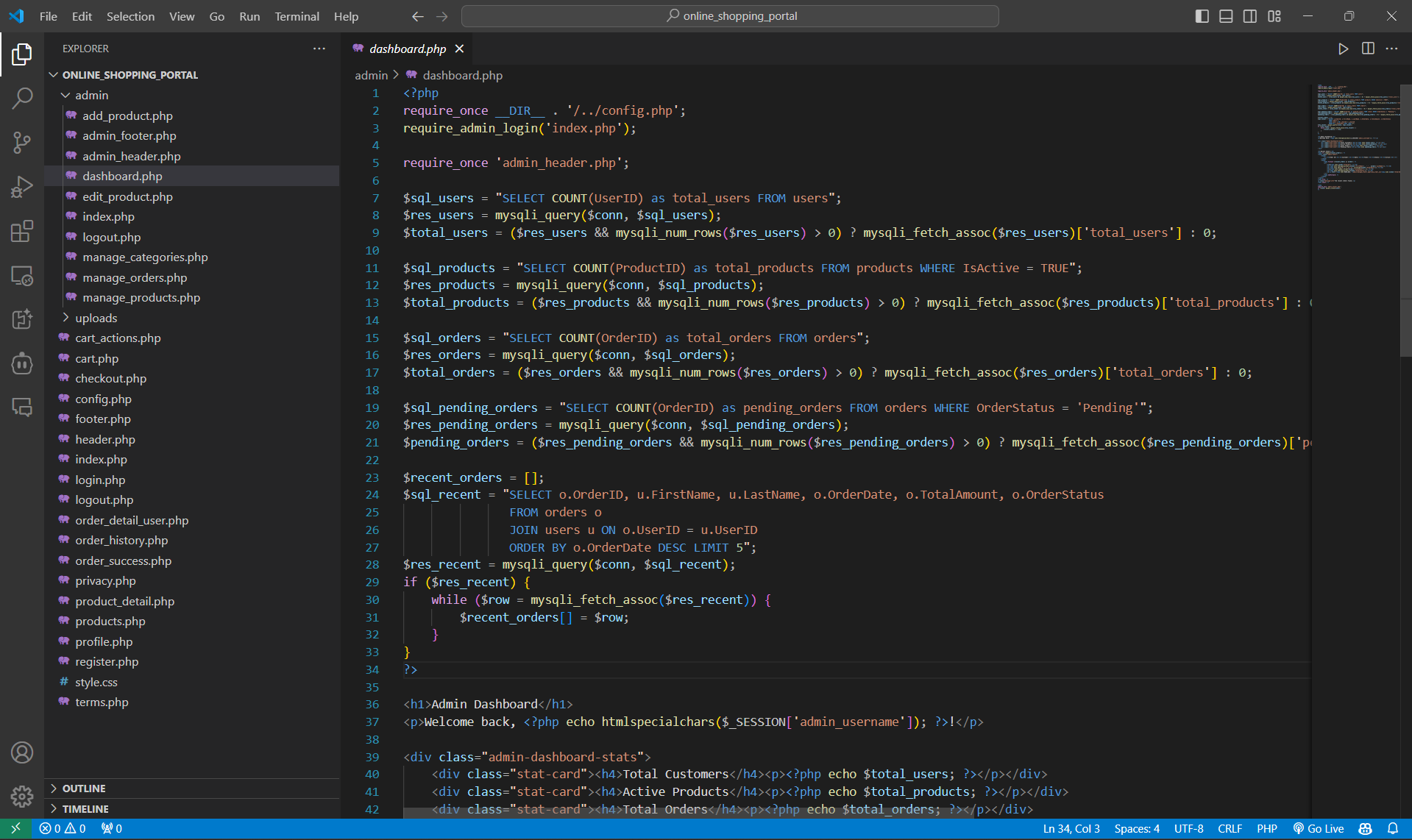
****

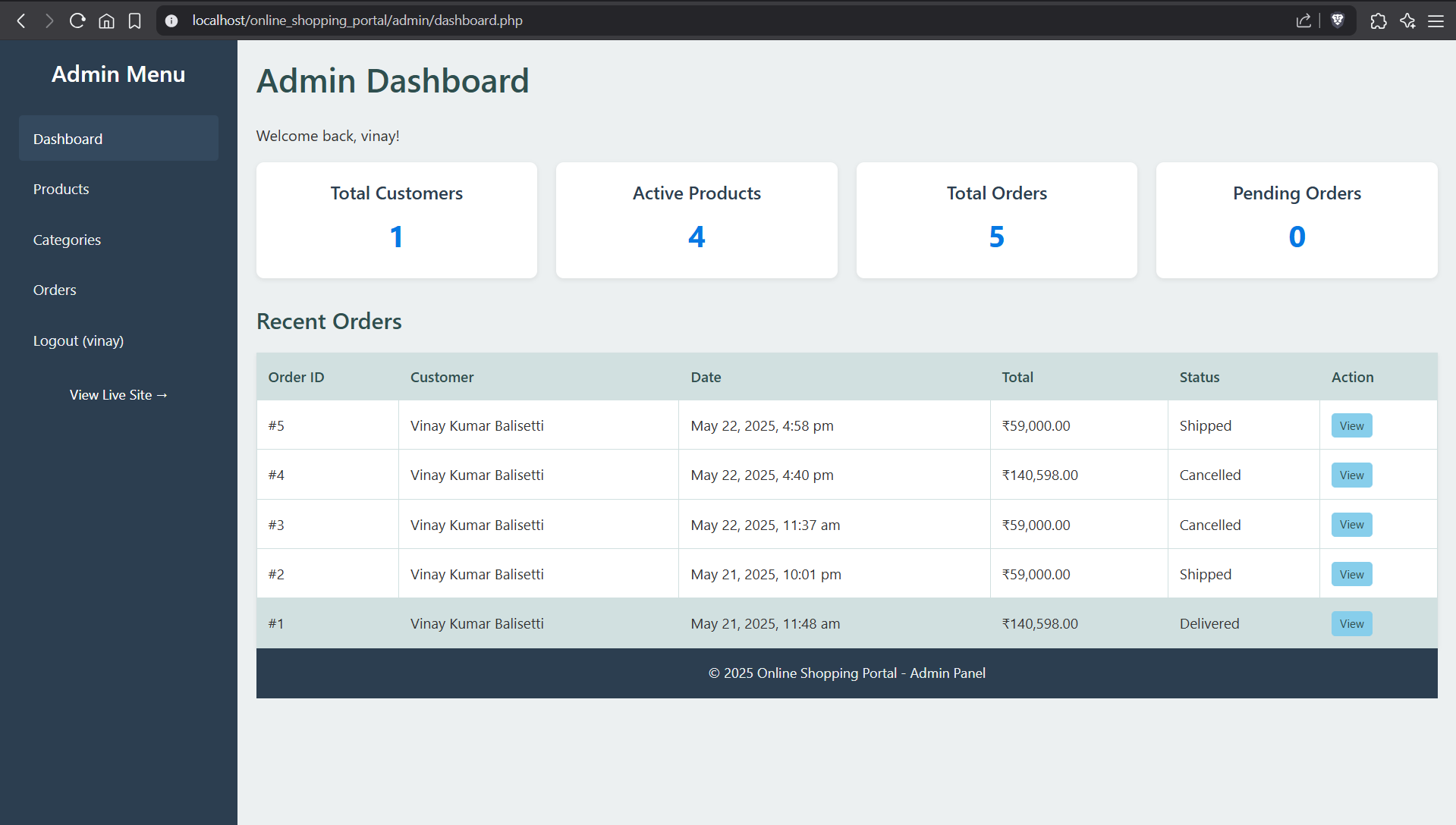
****

****

****

****

****

**Deployment to Amazon AWS Cloud**

In this section, we will deploy the online shopping portal program on Amazon Web Services (AWS). We will utilize the free trial instance, which is available for one year, adhering to the project guidelines that specify the use of only free or open-source methods for building the project. Specifically, we will use an EC2 instance from Amazon for hosting. Below are the detailed steps to follow for hosting the program in the cloud:

**Steps to Host the Program:**

1. **Start the EC2 Instance:**
   * Initiate the EC2 instance using SSH with the private key file.
2. **Configure Security Group:**
   * Set up inbound rules in the security group of the instance to allow access to HTTP, HTTPS, and SSH from anywhere.
3. **Update APT Packages:**
   * Update the list of APT packages to ensure all software is up-to-date.
4. **Install and Configure Server Software:**
   * Install the Apache server, PHP, and MySQL server.
   * Configure these components to work seamlessly with one another.
5. **Clone the Repository:**
   * Clone the repository from GitHub as provided in the appendices section of the SRS document.
   * Alternatively, copy the files from the development system and paste them into the EC2 instance using the SCP command.
6. **Import the SQL File:**
   * Import the SQL file located in the repository folder to set up the database.
7. **Copy PHP Files:**
   * Transfer the PHP files to the Apache server folder in the AWS instance.
8. **Note the Public IP Address:**
   * Record the public IP address of the instance for future reference.
9. **Access the Website:**
   * Access the website from any device using the URL: http://public-ip-address/folder-name/index.php.
10. **Test Database Connection:**
    * Test the database connection using the automatic database credentials.
    * If the connection is not working, modify the config.php file to include instance-specific MySQL credentials.

For Demo Purpose I have added some dummy data to showcase the functioning of the website.

To access the live demo of the deployed website on Amazon AWS, please visit the following URL:

<http://16.170.236.248/online_shopping_portal/index.php>

# **LIMITATIONS OF THE PROJECT**

While the "Online Shopping Portal" project aims to provide a comprehensive and user-friendly e-commerce platform, there are several limitations that should be considered:

**Technical Limitations**

* **Browser Compatibility:** The system is designed to be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge. However, there may be compatibility issues with older or less common browsers.
* **Device Compatibility:** The system is responsive and adapts to various screen sizes and devices, including desktops, tablets, and smartphones. However, there may be limitations in functionality or user experience on older or less common devices.
* **Performance:** The system is designed to handle a high volume of concurrent users and transactions. However, performance may degrade under extreme load conditions or with a large number of concurrent users.
* **Security:** The system implements various security measures to protect sensitive data and ensure secure transactions. However, no system is completely immune to security threats, and there may be vulnerabilities that could be exploited by malicious actors.

**Functional Limitations**

* **Product Catalogue:** The system provides a robust product catalogue system that allows users to browse, search and filter products. However, the catalogue may not include all possible products or categories, and there may be limitations in the depth or breadth of the product information.
* **Shopping Cart and Checkout:** The system provides a reliable shopping cart and checkout process that supports multiple payment methods. However, there may be limitations in the payment methods supported or the checkout process, and there may be issues with payment processing or order fulfilment.
* **Order Management:** The system provides order management functionalities, including order placement, tracking, and history viewing. However, there may be limitations in the order tracking or management capabilities, and there may be issues with order status updates or delivery tracking.
* **User Management:** The system provides user management functionalities, including user registration, authentication, and profile management. However, there may be limitations in the user management capabilities, and there may be issues with user authentication or profile updates.

**Operational Limitations**

* **Deployment:** The system is designed to be deployable on various hosting environments, including cloud-based and on-premises servers. However, there may be limitations in the deployment options or configurations, and there may be issues with server setup or configuration.
* **Maintenance:** The system is designed to be maintainable, with modular and reusable components and comprehensive documentation. However, there may be limitations in the maintenance capabilities or resources, and there may be issues with system updates or troubleshooting.
* **Support:** The system is designed to be supported by the development team and other stakeholders. However, there may be limitations in the support resources or capabilities, and there may be issues with system support or issue resolution.

**Legal and Ethical Limitations**

* **Ethical Considerations:** The system is designed to address ethical considerations, including privacy, fairness, and transparency. However, there may be limitations in the ethical measures or considerations, and there may be issues with ethical compliance or practices.

By outlining these limitations, this section provides a clear understanding of the constraints and potential issues that may arise with the "Online Shopping Portal" project, ensuring that stakeholders are aware of the system's limitations and can plan accordingly.

# **FUTURE ENHANCEMENTS OF THE PROJECT**

The "Online Shopping Portal" project is designed to be scalable and adaptable, with several potential enhancements planned for future development. These enhancements aim to improve the functionality, user experience, and overall performance of the system.

**1. Real Payment Gateway Integration**

* Description: Integrate real payment gateways to support secure and seamless transactions.
* Benefits: Enhance the checkout process by providing users with multiple payment options and ensuring secure transactions.

**2. Advanced Inventory Management**

* Description: Implement an advanced inventory management system to track stock levels, manage product availability, and automate reordering.
* Benefits: Improve inventory accuracy, reduce stockouts, and enhance the overall efficiency of the supply chain.

**3. Shipping API Integration**

* Description: Integrate with shipping APIs to provide real-time shipping rates, tracking, and delivery updates.
* Benefits: Enhance the shipping process by providing users with accurate shipping information and improving delivery tracking.

**4. Customer Support Module**

* Description: Develop a customer support module to provide users with assistance, including live chat, chatbots, and customer service portals.
* Benefits: Improve user satisfaction by providing timely and effective support, addressing user queries and issues.

**5. Advanced Reporting and Analytics**

* Description: Implement advanced reporting and analytics tools to provide insights into user behaviour, sales performance, and system performance.
* Benefits: Enhance decision-making by providing data-driven insights and improving the overall performance of the platform.

**6. Mobile Application**

* Description: Develop a mobile application to provide users with a convenient and accessible shopping experience on their mobile devices.
* Benefits: Increase user engagement and satisfaction by providing a seamless mobile shopping experience.

**7. Personalized Recommendations**

* Description: Implement machine learning algorithms to provide personalized product recommendations based on user behaviour and preferences.
* Benefits: Enhance the user experience by providing relevant and personalized product suggestions, increasing user engagement and sales.

**8. Loyalty Programs and Rewards**

* Description: Develop loyalty programs and rewards to incentivize repeat purchases and user engagement.
* Benefits: Increase user retention and satisfaction by providing rewards and incentives for continued use of platform.

**9. Social Media Integration**

* Description: Integrate with social media platforms to allow users to share products, reviews, and purchases on their social media profiles.
* Benefits: Enhance user engagement and marketing by leveraging social media platforms to promote products and the platform.

**10. Promotions and Discount Engine**

* Description: Implement a promotions and discount engine to provide users with special offers, discounts, and promotional pricing.
* Benefits: Increase sales and user engagement by providing attractive promotions and discounts.

**11. Bulk Product Upload/Update**

* Description: Develop a bulk product upload/update feature to allow administrators to upload and update multiple products simultaneously.
* Benefits: Improve the efficiency of product management by providing a convenient and time-saving feature for administrators.

**12. API for Third-Party Integrations**

* Description: Develop APIs to support third-party integrations, including payment gateways, shipping providers, and marketing tools.
* Benefits: Enhance the functionality and flexibility of the platform by supporting integrations with various third-party services.

**13. Enhanced SEO Features**

* Description: Implement enhanced SEO features to improve the visibility and ranking of the platform on search engines.
* Benefits: Increase the platform's visibility and attract more users by improving search engine optimization.

**14. Multi-Language and Multi-Currency Support**

* Description: Develop multi-language and multi-currency support to provide users with a localized and convenient shopping experience.
* Benefits: Enhance the user experience by providing support for multiple languages and currencies, catering to a global audience.

**15. Data Privacy and Protection**

* Description: Implement advanced data privacy and protection measures to ensure compliance with relevant regulations and standards.
* Benefits: Enhance user trust and satisfaction by ensuring the protection and privacy of user data.

# **REFERENCES**

1. Wiegers, K. E., & Beatty, J. (2013). *Software Requirements*. Microsoft Press.
   * A comprehensive guide to writing effective software requirements, including best practices and practical examples.
2. BCSP-064 Project Guidelines, IGNOU BCA.

* For Structure of Project Report.

1. Sommerville, I. (2015). *Software Engineering*. Pearson.
   * A widely used textbook that covers software engineering principles, including requirements engineering and SRS documentation.
2. Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements: A Use Case Approach*. Addison-Wesley Professional.
   * A book that focuses on managing software requirements using use cases, providing practical guidance for creating effective SRS documents.
3. **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**
   * The official IEEE standard for software requirements specifications, providing guidelines and best practices for creating SRS documents.
4. **Cockburn, A.** (2001). Writing Effective Use Cases. Addison-Wesley Professional.
   * A book that provides practical guidance on writing effective use cases, which are a key component of many SRS documents.
5. **W3Schools HTML Tutorial**
   * A beginner-friendly tutorial that covers the basics of HTML, including elements, attributes, forms, and more.
6. **HTML & CSS by Jon Duckett**
   * A highly regarded book that provides a visual and engaging introduction to HTML and CSS.
7. **W3Schools PHP Tutorial**
   * A beginner-friendly tutorial that covers the basics of PHP, including syntax, variables, loops, functions, and more.
8. **MySQL Official Documentation**
   * [MySQL 8.0 Reference Manual](https://dev.mysql.com/doc/refman/8.0/en/)
   * The official MySQL documentation covers all aspects of MySQL, including installation, configuration, and SQL syntax.
9. **PHP Official Documentation**
   * [PHP Manual](https://www.php.net/manual/en/)
   * The official PHP manual provides comprehensive documentation on PHP functions, language syntax, and best practices.
10. **HTML Living Standard**
    * [HTML Living Standard](https://html.spec.whatwg.org/multipage/)
    * The official living standard for HTML.