|  |  |
| --- | --- |
| online shoe store | Abstract "This abstract outlines the development and implementation of a comprehensive online shoe store system. The project's primary objective is to overcome the limitations of traditional brick-and-mortar retail by providing customers with a convenient and accessible platform for purchasing footwear. The system is designed to be user-friendly and secure, offering a wide selection of shoes for various demographics and occasions. Key features include a robust user authentication system, advanced search and filtering capabilities, a secure shopping cart with multiple payment options, and an order tracking module. The administrative panel provides tools for efficient inventory management, sales tracking, and customer data handling. This online platform aims to streamline the shopping process for consumers while enabling the business to operate with greater efficiency, reliability, and security, ultimately leading to enhanced customer satisfaction and market growth."  Ashish  [Web development] |

**Mr. Jaiswar Ashish Kumar**

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
|  | **Index** |  |
| **Sr.no** | **Chapter 1** | **Page Number** |
| **1** | **Introduction** | **1-4** |
| 1.1 | Background |  |
| 1.2 | Objective |  |
| 1.3 | Purpose |  |
| 1.4 | Scope |  |
| 1.5 | Applicability |  |
|  | **Chapter 2** |  |
| **2** | **System Planning** | **5-8** |
| 2.1 | Survey of technologies |  |
| 2.2 | Fact finding technique |  |
| 2.3 | Feasibility Study |  |
| 2.4 | Stakeholders |  |
|  | **Chapter 3** |  |
| **3** | **Requirement and Analysis** | **9-24** |
| 3.1 | Problem Definition |  |
| 3.2 | Requirement Specification |  |
| 3.3 | Planning and Scheduling |  |
| 3.4 | Software and Hardware Requirement |  |

|  |  |  |
| --- | --- | --- |
| 3.5 | Conceptual Models |  |
| 3.5.1 | ER Diagram |  |
| 3.5.2 | Use case Diagram |  |

|  |  |  |
| --- | --- | --- |
| 3.5.3 | Class Diagram |  |
| 3.5.4 | Sequence Diagram |  |
| 3.5.5 | Package Diagram |  |
| 3.5.6 | Deployment Diagram |  |
| 3.5.7 | System Flowchart |  |
|  | **Chapter 4** |  |
| **4** | **System Design** | **25-34** |
| 4.1 | Basic Modules |  |
| 4.2 | Data Design and Data integrity and Constraints |  |
| 4.3 | User Interface and Design |  |
| 4.4 | Security Issues |  |
| 4.5 | Test Cases |  |
|  | **Chapter 5** |  |
| **5** | **System Coding, Implementation and Testing** | **35-62** |
| 5.1 | Coding Details |  |
| 5.2 | Coding Efficiency |  |
| 5.3 | Testing Approach |  |

|  |  |  |
| --- | --- | --- |
| 5.3.1 | Unit Testing |  |
| 5.3.2 | Integrated Testing |  |
|  | **Chapter 6** | **62-63** |
| **6** | **Conclusion and Future Work** |  |
|  | **Chapter 7** | **63-64** |
| **7** | **References** |  |

### CHAPTER 1 INTRODUCTION

#### 1.1 Background

The rapid growth of e-commerce has significantly transformed the retail landscape, giving rise to online platforms that cater to the evolving needs of consumers. Among these, online shoe stores have gained substantial popularity by offering convenience, variety, and competitive pricing. Traditional shoe shopping often involved limited options and time-consuming visits to physical stores, whereas online platforms provide customers with access to a wide range of brands, styles, and sizes from the comfort of their homes. The rise in smartphone usage, secure online payment systems, and efficient delivery services has further accelerated the shift towards digital shopping experiences. In this context, the development of an online shoe store addresses the modern consumer's demand for a seamless, user-friendly, and personalized shopping journey, combining the power of technology with the essentials of retail. The primary objective of this project is to design and develop a user-friendly, responsive, and efficient Business-to-Consumer (B2C) online shoe store that enables customers to browse, search, and purchase footwear seamlessly through a digital platform. The system aims to streamline the buying process by incorporating essential e-commerce features such as product categorization, search functionality, secure user authentication, real-time inventory management, online payment integration, and order tracking. Additionally, the platform seeks to enhance the user experience through intuitive navigation, aesthetic design, and personalized recommendations. This project not only aims to deliver a functional and scalable web application but also to demonstrate the effective use of modern web technologies in solving real-world

#### 1.2 Objectives

The primary objective of the **Online Shoes Store** is to design and develop a user-friendly, responsive, and efficient Business-toConsumer (B2C) online shoe store that enables customers to browse, search, and purchase footwear seamlessly through a digital platform. The system aims to streamline the buying process by incorporating essential e-commerce features such as product categorization, search functionality, secure user authentication, real-time inventory management, online payment integration, and order tracking. Additionally, the platform seeks to enhance the user experience through intuitive navigation, aesthetic design, and personalized recommendations. By addressing these objectives, the Online Shoes Store this project not only aims to deliver a functional and scalable web application but also to demonstrate the effective use of modern web technologies in solving real-world retail challenges.

Objectives In Points

**To develop a fully functional B2C online shoe store web application.**

**To allow users to browse and search for shoes based on categories, sizes, brands, and price.**

**To implement secure user registration, login, and profile management features.**

**To integrate a secure online payment gateway for smooth transactions.**

**To enable real-time inventory management for product availability tracking.**

**To provide order placement, tracking, and history features for users.**

**To design a responsive and intuitive user interface accessible across devices.**

## 1.3 PURPOSE

The purpose of the Online Shoes Store is to develop a B2C online shoe store that modernizes the traditional shoe retail process by offering a convenient and accessible digital platform for customers. This system aims to overcome the limitations of physical shopping by providing users with a wide range of footwear options, available anytime and from anywhere. By streamlining product display, order management, and payment processing, the platform not only enhances the overall customer experience but also supports business growth through increased reach and efficiency. Additionally, it seeks to establish a scalable and secure online presence that can adapt to the evolving demands of the digital market place. The application serves multiple purposes, including:

**Online Retailing:** Enables customers to browse and purchase shoes from the comfort of their homes.

**Inventory Management:** Allows the business to track stock levels in real time and manage product availability.

**Customer Relationship Management (CRM):** Stores user data to offer personalized shopping experiences and marketing campaigns.

**Sales Analytics:** Tracks user behavior, sales trends, and product performance to support business decisions.

**Order and Delivery Tracking:** Facilitates seamless order placement, status updates, and estimated delivery times for customers.

**Mobile Shopping:** Offers access to the store via mobile devices, increasing customer engagement and convenience.

**Digital Marketing Integration:** Supports promotional campaigns through features like discount codes, banners, and newsletters.

**User Feedback System:** Allows customers to rate and review products, helping future buyers and improving brand credibility.

## 1.4 SCOPE

The scope of this project includes the design and development of a B2C online shoe store that facilitates seamless interaction between customers and the business. On the user side, the platform will allow customers to register, log in, browse products by category, view detailed product information, add items to a cart, make secure online payments, and track their orders. Users will also be able to provide feedback and manage their profiles.

On the admin side, the system will include features for managing inventory, updating product details, processing orders, monitoring sales, and managing user queries. The platform will be designed to be responsive across devices, ensuring accessibility from desktops, tablets, and smartphones. Additionally, the system will include basic data analytics tools to track user behavior and sales trends, helping the business improve marketing strategies and inventory planning. Future scalability will also be considered, allowing for the integration of advanced features like AI-based recommendations, multilingual support, and mobile apps.

## 1.5 APPLICABILITY

The online shoe store system is highly applicable across various domains within the retail and e-commerce industry. It is ideal for small to large-scale footwear businesses looking to establish or expand their digital presence. The platform can be used by startup brands launching online-first stores, as well as by existing physical retailers transitioning into hybrid (offline + online) business models. It is also applicable for niche markets such as sports footwear, luxury shoes, or eco-friendly brands targeting specific customer segments. Furthermore, the system can be integrated into larger multi-vendor marketplaces or white-label for other fashion-related businesses. Its modular design and scalability make it suitable for deployment across different regions with customizable features such as currency, language, and payment options. Educational institutions can also use this project as a base for academic learning in fields like software development, UI/UX design, and business analytics.

.

## CHAPTER 2: SYSTEM PLANNING

**2.1 Survey of Technologies**

Technology selection is critical to ensuring system efficiency, maintainability, and scalability. Below is an overview of the technologies used:

* **Frontend Technologies**

The frontend provides the user interface, allowing Customers and administrators to interact with the system.

HTML (Hyper Text Markup Language): Defines the structure and layout of web pages. CSS (Cascading Style Sheets): Used to style and format the appearance of the application. JavaScript: Adds interactivity, allowing real-time updates and dynamic content loading.

Bootstrap: Ensures a responsive layout across different devices.

* **Backend Technologies**

The backend processes user requests and manages data.

PHP (Hypertext Preprocessor): A powerful scripting language used for server side logic. MySQL: A relational database management system for efficient data handling.

XAMPP: A local server environment that integrates Apache, PHP, and MySQL..

* **Database Technologies**

Efficient data storage and retrieval ensure smooth operations.

MySQL: Manages structured data including users, transactions, and sessions. phpMyAdmin: Provides a user-friendly interface for database administration. SQL Queries: Used to insert, retrieve, update, and delete data.

### 2.2 Fact Finding Technique

1. **Interviews:**
   * + **Purpose:** To gather detailed insights from stakeholders (e.g., business owners, target customers, or domain experts) about their expectations, preferences, and pain points.
     + **Application:** Conduct one-on-one or group interviews with business owners or customers to understand what features are needed in the online shoe store (e.g., user login, product search, payment methods, etc.).
     + **Outcome:** Clear understanding of user requirements and expectations.
2. **Surveys and Questionnaires:**
   * + **Purpose:** To collect quantitative data on user preferences, shoe types, budget, and online shopping behaviors.
     + **Application:** Design online surveys that target potential customers to gather feedback on preferred shoe categories, sizes, payment preferences, and delivery expectations.
     + **Outcome:** Statistical data to guide the design and functionality of the website.
3. **Observation:**
   * + **Purpose:** To observe user behaviour and how potential customers interact with existing shoe e-commerce platforms.
     + **Application:** Analyze how users navigate through current shoe websites, what features they frequently use, and where they tend to drop off in the shopping process.
     + **Outcome:** Insight into user preferences and pain points that can help improve the online shopping experience.
4. **Document Review:**
   * + **Purpose:** To A nalyze existing documents, reports, and records related to the shoe retail business and its customers.
     + **Application:** Review any existing sales reports, customer feedback, and website analytics (if applicable) to understand past trends, successful features, and areas of improvement.

**2.3 Feasibility Study**

A feasibility study was conducted to ensure the Online Shoes Store project is viable across technical, economic, and operational dimensions, demonstrating its potential for successful implementation and long-term sustainability.

1. **Technical Feasibility:** 
   * **Technology:** PHP, MySQL, HTML, CSS, and JavaScript are sufficient for building the platform.
   * **Server:** Local server (XAMPP/WAMP) for initial testing; scalable to cloud hosting later.
   * **Conclusion:** The technologies and infrastructure needed are available and suitable for the project.

1. **Economic Feasibility:** 
   * **Low Development Cost:** Uses open-source technologies (PHP, MySQL), reducing upfront costs.
   * **Operational Cost:** Affordable hosting options; costs increase with scale.
   * **Revenue Potential:** The platform opens a wider market reach, potentially increasing sales and reducing operational costs.
   * **Conclusion:** Economically viable with low initial investment and good potential returns.

1. **Operational Feasibility:** 
   * **Ease of Use:** User-friendly for customers; simple admin panel for order and inventory management.
   * **Maintenance:** Ongoing maintenance is manageable; future scalability is possible.
   * **Employee Training:** Minimal training required for staff to manage the System.
   * **Conclusion:** The system is operationally feasible and easy to maintain.

### 2.4 Stakeholders

Key stakeholders involved in the Online Shoes Store project include:

2.4.1 **Customers:**

* **Role:** The primary users who will browse, select, and purchase shoes from the online store.
* **Interest:** A seamless, easy-to-use shopping experience with fast checkout and reliable delivery.

2.4.2 **Business Owners/Management:**

* **Role:** The owners or managers of the shoe store, responsible for running and maintaining the online platform.
* **Interest:** The successful operation of the platform, increased sales, and profitability.

2.4.3 **Developers (Frontend and Backend):**

* **Role:** Responsible for designing, coding, and maintaining the website.
* **Interest:** Ensuring the platform functions smoothly, is user-friendly, and performs as required.

2.4.4 **Admin Staff:**

* **Role:** Manage product listings, customer orders, and customer support.
* **Interest:** An easy-to-use admin panel to manage products, orders, and user accounts effectively.

2.4.5 **Suppliers:**

* **Role:** Provide the shoes and inventory for the online store.
* **Interest:** Timely orders and inventory management, as well as the successful sale of their products.

2.4.6 **Payment Gateway Providers:**

* **Role:** Facilitate online payment processing for customer transactions.
* **Interest:** Secure and efficient transactions, ensuring payment is received promptly.

2.4.7 **Logistics/Delivery Partners:**

* **Role:** Responsible for delivering the shoes to customers after orders are placed.
* **Interest:** Timely and accurate order fulfillment, customer satisfaction.

2.4.8 **Marketing Team:**

* **Role:** Promote the online shoe store and drive traffic to the website.
* **Interest:** Increase in website visits, conversion rates, and overall brand visibility.

## CHAPTER 3: REQUIREMENT AND ANALYSIS

### 3.1 Problem Definition

The traditional shoe retail model is limited by physical store locations, high operational costs, and restricted customer reach. Customers often face challenges such as limited product variety, inconvenient store hours, and a lack of personalized shopping experiences. Additionally, businesses struggle with inventory management, order tracking, and offering seamless payment systems.

## Solution

The proposed solution to the challenges identified in the problem definition is the development of a **B2C Online Shoe Store** that offers a seamless, user-friendly, and efficient shopping experience. The platform provides **24/7 accessibility**, allowing customers to browse, search, and purchase shoes at their convenience, regardless of time or location, eliminating the need for physical store visits. It features an intuitive interface that enables easy navigation through categories, product filtering, and detailed product descriptions. Secure and **seamless transactions** are ensured thro- ugh integration with reliable payment gateways, offering multiple payment options such as credit/debit cards and PayPal. Additionally, the system includes **real-time inventory management**, allowing customers to see product availability in real time, preventing overselling and ensuring accurate stock information. The **admin panel** streamlines order management, enabling business owners to manage inventory, track orders, and handle customer queries efficiently. To enhance the customer experience, users can create **personalized profiles**, track their orders, and receive tailored product recommendations based on their browsing and purchase history. This comprehensive solution addresses both customer needs for convenience and the business requirements for efficient operation, making the platform scalable and userfriendly.

### 3.2 Requirement Specification

1. Functional Requirements:

* User Authentication:
  + Users must be able to register, log in, and manage their profiles.
  + Password recovery and secure session management must be implemented.
* Product Management:
  + Admins should be able to add, edit, and remove products from the catalog. o Products should have attributes like size, color, brand, and price, and users should be able to filter by these attributes.
* Search and Filters:
  + Customers must be able to search for shoes by name, size, brand, and category. o Filters should be available to refine search results based on various product features.
* Shopping Cart:
  + Users should be able to add products to a shopping cart and modify quantities.
  + Cart should store user selections until the purchase is completed.
* Checkout and Payment:
  + Users must be able to proceed with secure checkout by providing shipping details. o The system should integrate with a secure payment gateway (credit/debit cards, PayPal, etc.) for payment processing.
* Order Management: o Users should be able to view their order history and track current orders. o Admin should be able to manage orders, update statuses, and handle returns/refunds.

2. Non-Functional Requirements:

* Usability:
  + - The website must have an intuitive, responsive design accessible across devices (desktop, tablet, mobile). o The user interface should be clean, easy to navigate, and aesthetically appealing.  Security:
    - User data (especially sensitive information like payment details) must be encrypted using secure protocols (e.g., HTTPS). o The system should protect against common security threats like SQL injection and cross-site scripting (XSS).  Performance:
    - The website should load quickly and handle a reasonable number of concurrent users without performance degradation.
    - Optimized database queries to ensure fast product search and retrieval.  Scalability:
    - The system should be able to handle an increasing number of products and users as the business grows. o The architecture should allow for easy addition of new features, such as mobile app integration or support for additional payment methods.
* Availability:
  + The platform should be available 24/7 with minimal downtime.
  + Backup and disaster recovery strategies should be in place to prevent data loss.

3. Technical Requirements:

* Frontend:
  + Technologies: HTML, CSS, JavaScript (for dynamic functionality). o Responsive design using frameworks like Bootstrap or custom CSS to ensure compatibility across devices.
* Backend:
  + Technologies: PHP (for server-side scripting) and MySQL (for database management). o A secure and efficient server environment (e.g., XAMPP/WAMP for local testing, cloud hosting for live deployment).
* Payment Gateway Integration:
  + Integration with popular payment providers like PayPal, Stripe, or credit card gateways.
* Hosting:
  + Initially, local hosting (XAMPP/WAMP) for development and testing.
  + Cloud-based hosting (e.g., AWS, Bluehost) for live deployment.

4. System Requirements:  Hardware:

o Local server setup: Desktop or laptop with enough storage and RAM for development. o Cloud hosting: A reliable hosting service with sufficient bandwidth and storage capacity for scaling.

 Software:

* Web browser (Google Chrome, Firefox, etc.) for testing.
* Development tools: Code editor (VS Code, Sublime Text), MySQL Workbench, and XAMPP/WAMP for local server setup.

### 3.3 Planning and Scheduling

To ensure the successful completion of the online shoe store project, the development process is divided into clearly defined phases with specific goals and timelines. This structured planning helps in better resource management, on-time delivery, and systematic progress tracking.

**Project Phases and Timeline:**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Description** | **Duration** |
| **1. Requirement**  **Gathering** | Identify system needs, define objectives, and gather stakeholder inputs. | Week 1 –  Week 2 |
| **2. System Design** | Prepare wireframes, UI layout, and database schema design. | Week 3 –  Week 4 |
| **3. Frontend Development** | Design responsive web pages using HTML, CSS, and JavaScript. | Week 5 –  Week 6 |
| **4. Backend**  **Development** | Develop core functionalities using PHP and integrate with MySQL database. | Week 7 –  Week 9 |
| **5. Admin Panel Setup** | Create admin features for managing products, orders, and users. | Week 10 |
| **6. Testing and**  **Debugging** | Perform unit testing, bug fixing, and security checks. | Week 11 |
| **7. Final Deployment** | Host the project on localhost/XAMPP; optionally prepare for live hosting. | Week 12 |
| **8. Documentation** | Compile project report, user manual, and black book submission material. | Week 13 |

### 3.4 Software and Hardware Requirements

Software Requirements:

1. Development Environment:

* + Visual Studio Code or Sublime Text: For PHP and frontend development, providing a lightweight yet powerful coding and debugging environment.
  + XAMPP: A local development environment that includes Apache (web server), MySQL (database), and PHP for smooth backend development.

1. Backend Development:

* + PHP: The server-side scripting language used to handle business logic, database interactions, and API requests.
  + MySQL: A relational database management system (RDBMS) for storing user data, PC usage records, event details, and membership information.
  + Payment Gateway Integration: Used for processing secure online transactions and automating billing.

1. Frontend Development:

* + HTML, CSS, JavaScript: For designing the user interface and ensuring responsiveness.

* + React or Vue.js: For creating interactive and dynamic frontend components that enhance user experience.

1. Testing:

* + Unit Testing Frameworks: PHP Unit is used to test individual PHP components and functions.
  + Integration Testing Tools: Postman and Selenium to test system-wide functionalities and API endpoints.
  + Performance Testing Tools: JMeter for assessing system performance under high load conditions.

5.Deployment

Cloud Hosting Services (e.g., AWS, cPanel, Digital Ocean): For hosting the application and ensuring reliability and scalability.

* + CI/CD Tools (GitHub Actions, Jenkins): Automates testing and deployment processes for smooth system updates.

Hardware Requirements:

1. Development Workstations:

* + Desktops or Laptops: Equipped with at least 1GB RAM, SSD storage, and multi-core processors to efficiently handle PHP development and database operations.
  + High-Resolution Monitors: For effective coding and debugging.

1. Testing Devices:

* + Desktop Computers: For testing web applications across different browsers and operating systems to ensure compatibility.
  + Simulated Test Scenarios: Various user conditions and loads to validate performance.

1. Server Infrastructure:

* + Cloud-Based Servers: Hosted on cPanel, AWS, or Digital Ocean to manage PHP-based server logic and MySQL databases.
  + Stable Internet Connection: Required for seamless cloud-based development and testing.

1. Networking Equipment:

* + High-Speed Internet: Ensures smooth communication with cloud services and deployment pipelines.
  + Secure Network Configuration: Implements security protocols to protect user data and financial transactions.

By adhering to these software and hardware requirements, the Online Shoes Store will be equipped to handle seamless development, deployment, and operation, ensuring a robust and scalable Online Shoes Store system.

### 3.5 Conceptual Models

Conceptual models are essential in understanding and designing a system by representing its core concepts and relationships. They offer a structured overview of the system's components and how they interact, facilitating effective design and implementation. For the **Online Shoes Store** project, the conceptual models are outlined below to provide a clear representation of the system’s entities, their attributes, relationships, and actions.

1. User (Customer):
   * Registers, logs in, and maintains a profile.
   * Browses products, adds to cart, and places orders.
2. Product:
   * Shoes listed with details like name, size, brand, category, price, description, and stock quantity.
   * Managed by the admin.
3. Cart:
   * Temporary storage for selected products before checkout.
   * Linked to the user session.
4. Order:
   * Contains product details, quantity, price, shipping info, and payment status.
   * Created when a user completes checkout.
5. Admin: o Manages product listings, updates stock, views user activity, and processes orders.
6. Payment:
   * Handles transaction processing via available gateways.
   * Stores payment status and history.

Basic Relationships:

* A User can have multiple Orders.
* Each Order contains multiple Products.
* A Product can appear in multiple Orders.
* A User has one Cart.

.

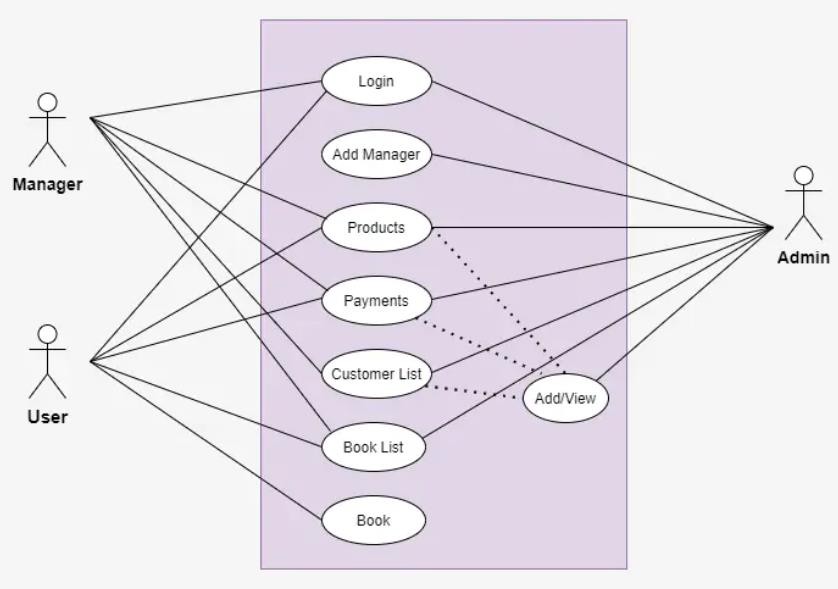
**3.5.1 ER DIAGRAM**

The Entity-Relationship (ER) diagram for the **Online Shoe Store** represents the core structure of the database and the relationships among different entities in the system. The main entities include **User**, **Product**, **Cart**, **Order**, **Order Detail**, **Payment**, and **Admin**. Each **User** can register, login, and maintain a personal profile, and they can place multiple **Orders** over time. Every user has one **Cart**, which temporarily holds selected products before checkout. The **Cart** consists of multiple **Cart Items**, each linked to a specific **Product** along with a specified quantity. Once an order is placed, the system creates an **Order** entry, and the details of each product in the order are stored in the **Order Detail** entity. For every order, there is one corresponding **Payment** entity that captures the method, date, and status of the transaction. The **Product** entity holds information such as product name, brand, category, size, price, and stock quantity, and it is managed by the **Admin**, who also monitors orders and handles inventory. The relationships between these entities are mostly one-to-many or one-to-one, ensuring proper linkage and data flow throughout the system. This ER model ensures a well-structured and normalized database for efficient data management and seamless e-commerce functionality.



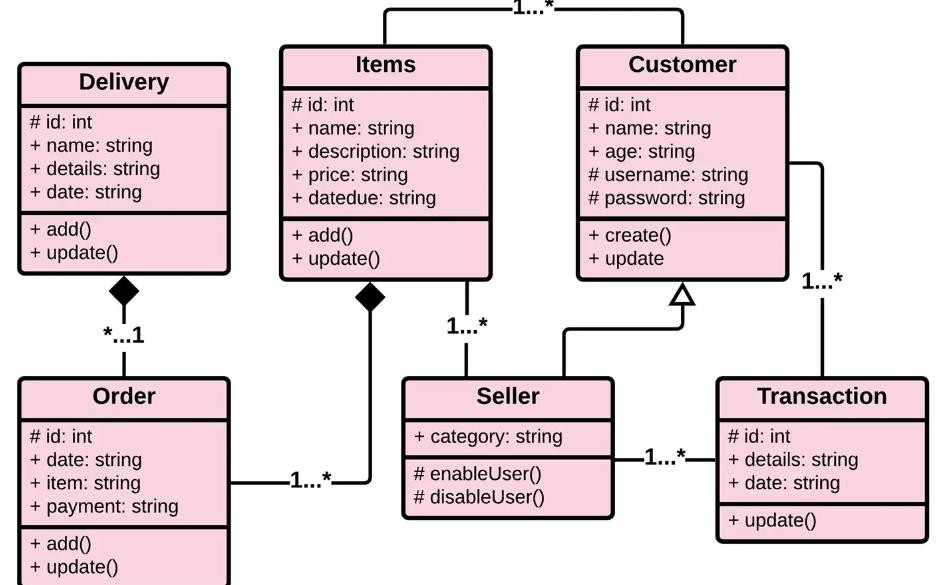
### 3.5.2 USE CASE DIAGRAM

The use case model for the **Online Shoe Store** outlines the key interactions between the system and its users, highlighting how different functionalities are accessed and performed. The primary actor in the system is the **Customer**, who can perform actions such as registering an account, logging in, browsing products, filtering and searching for specific shoes, adding items to the cart, placing orders, and making secure payments. The customer can also view their order history and track current orders. Another important actor is the **Admin**, who has access to the admin panel andcan manage product listings, update inventory, view and process customer orders, and handle user management. The **Payment Gateway** may be considered an external system that interacts with the platform to process transactions securely.



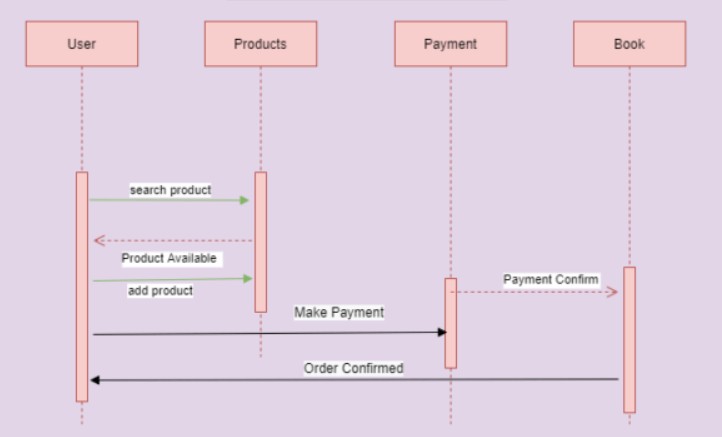
### 3.5.3 CLASS DIAGRAM

The class diagram for the Online Shoe Store represents the object-oriented structure of the system, showing the classes involved, their attributes, methods, and the relationships between them. The core classes include User, Product, Cart, Order, OrderDetail, Admin, and Payment. The User class includes attributes such as user ID, name, email, password, and address, along with methods for registration, login, and profile management. The Product class holds attributes like product ID, name, category, brand, price, size, and stock quantity, and includes methods for display ing and updating product details. The Cart class is associated with the user and contains a list of CartItem objects, each representing a product added to the cart with quantity. The Order class includes details like order ID, date, total amount, and status, and is linked to both User and OrderDetail classes. The OrderDetail class captures individual product details within an order, such as quantity and unit price. The Payment class manages payment-related data, including payment ID, method, status, and date. The Admin class includes functionalities for managing users, products, and orders. Relationships between classes are typically one-to-many or one-to-one, such as one user placing many orders, or one order having multiple order details. This class diagram ensures a well-structured and modular design that supports easy develop- ment, testing, and maintenance of the application.



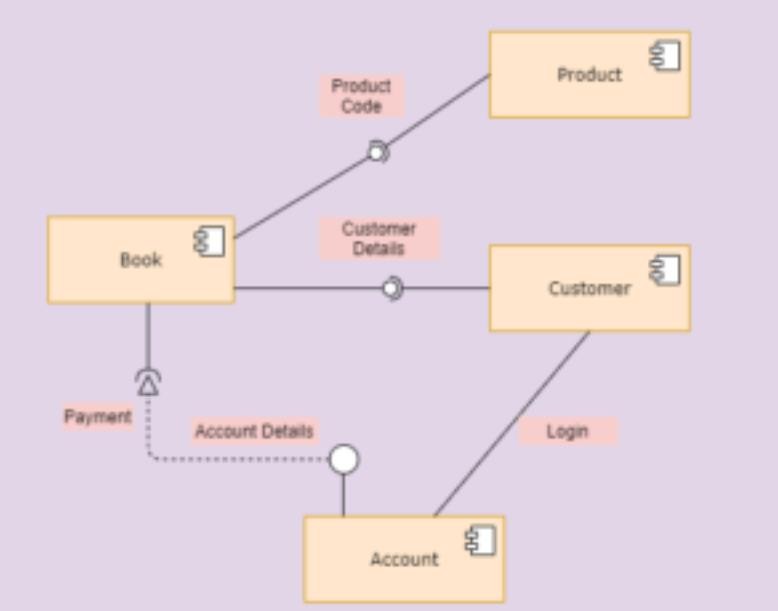
### 3.5.4 SEQUENCE DIAGRAM

A Sequence Diagram for the Online Shoe Store illustrates the flow of interactions between different system components over time for specific scenarios. It captures how entities like User, PC, Booking System, and Payment Gateway interact to complete processes such as booking a PC or processing a payment. For instance, when a user initiates a booking, the diagram shows the sequence of messages exchanged between the user interface, booking system, and payment gateway, detailing steps such as verifying availability, confirming the booking, and handling the payment transaction. This diagram helps in visualizing the dynamic behavior of the system, ensuring that all necessary interactions and their order are clearly defined for accurate implementation and troubleshooting.



### 3.5.5 PACKAGE DIAGRAM

A Package Diagram for the Online Shoe Store organizes the system into modular components or packages, each encapsulating related classes and functionalities. This diagram visually represents the logical grouping of the system’s components, such as the User Management Package, Booking Management Package, and Event Management Package. Each package contains classes and interfaces pertinent to its function, such as user authentication and profile management within the User Management Package, or booking scheduling and cost calculation within the Booking Management Package. The Package Diagram helps in understanding the system’s structure and dependencies, promoting a modular design approach that enhances maintainability and scalability by clearly delineating responsibilities and interactions among different parts of the system.



### 3.5.6 DEPLOYMENT

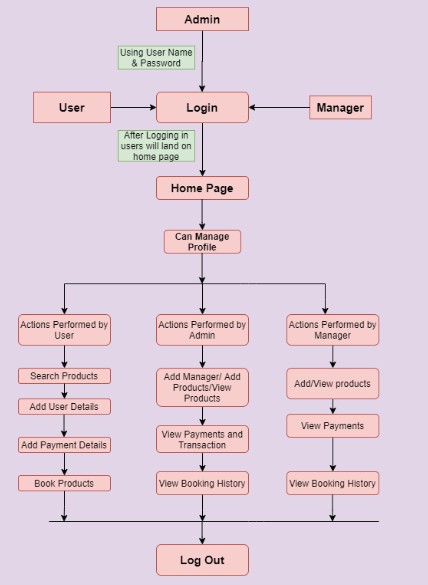
The deployment for the Online Shoe Store illustrates the physical architecture of the system, showing how software components are deployed on different hardware or infrastructure nodes. The system is primarily built using PHP for backend logic and MySQL for database management, hosted on a Local Server (e.g., XAMPP/WAMP) during development, with the option to move to a Web Hosting Server for live deployment. The main nodes in the diagram include the Client Device, Web Server, and Database Server.

The Client Device (user’s computer or mobile phone) runs a web browser to access the website

It sends requests to the Web Server, which hosts the application code (PHP, HTML, CSS, JavaScript). The Web Server processes these requests, interacts with the Database Server to fetch or update data, and sends the response back to the client. The Database Server stores all the critical data such as user information, product listings, orders, payments, and inventory details. An Admin Interface is also accessible via a browser and connects to the same web server, allowing admins to manage the system.

### 3.5.7 SYSTEM FLOW CHART

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction. Different flow chart symbols have different meanings.



### CHAPTER 4 SYSTEM DESIGN

#### 4.1 Basic Modules

In this section, the core modules of the "Online Shoes Storet" project will be outlined. Each module represents a distinct functionality or service provided by the system. These modules work together to ensure smooth operation and management of the Shoes Website

**1. Functional Requirements**

These are the features the system needs to support:

**Customer-facing features:**

* User registration/login (with email/social login)
* Product browsing and search (by size, category, brand, etc.)
* Product detail pages with images, sizes, and reviews
* Shopping cart and wishlist
* Checkout process (with shipping details, payment, promo codes)
* Order tracking
* Reviews and ratings
* Notifications (email, SMS)  **Admin panel:**
* Manage products (CRUD)
* Manage orders and returns
* Inventory tracking
* Manage discounts and coupons
* Customer support tools

**Non-Functional Requirements**

* Scalability (handle spikes in traffic, e.g., during sales)
* High availability
* Fault tolerance
* Security (especially for payments and user data)
* Performance (fast response, low latency)
* SEO friendly frontend
* Analytics tracking

**3.Component Design**

**A. Frontend (Web/Mobile)**

* React.js, Next.js (for SSR & SEO), or Vue.js
* Flutter/React Native for mobile
* Responsive, intuitive UI

**B. API Gateway**

* Nginx, Kong, or AWS API Gateway
* Handles routing, authentication, caching

**C. Backend Services (Microservices)**

* **User Service**

Handles login, signup, profile, auth (JWT/OAuth2)

* **Product Catalog Service**

Categories, filters, images, descriptions, availability

Use Elasticsearch for full-text search and filtering

* **Cart Service**

In-memory or Redis-based store for speed

Cart linked to user session/account

* **Order Service**

Order creation, status, payment handling, returns

* **Payment Service**

Integrates with Stripe, Razorpay, or PayPal

Tokenize card data, avoid storing sensitive info

* **Notification Service**

Sends emails (via SendGrid, SES), SMS (Twilio)

* **Review Service**

Users can rate and review purchased products

#### 4.2 Data Design, Data Integrity, and Constraints

1. Data Design

The data design for the "Online Shoes Store" system includes multiple interconnected tables in a relational database managed using MySQL with PHP. Below are the key database tables:

2.Data Validation:

Input validation ensures correctness, such as proper email format and correct date entries.

3.Normalization:

The database follows normalization principles to eliminate redundancy and enhance load

4.Unique Constraints:

Fields like username and email must be unique to prevent duplicate user registrations

5.Data Consistency:

Transaction handling ensures that either all operations are committed, maintaining data integrity.

User Table :

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| user\_id | UUID / INT | PK, AUTO\_INCREMENT or UUID |
| Name | VARCHAR | NOT NULL |
| Email | VARCHAR | UNIQUE, NOT NULL |
| password\_hash | VARCHAR | NOT NULL |
| phone\_number | VARCHAR | OPTIONAL |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

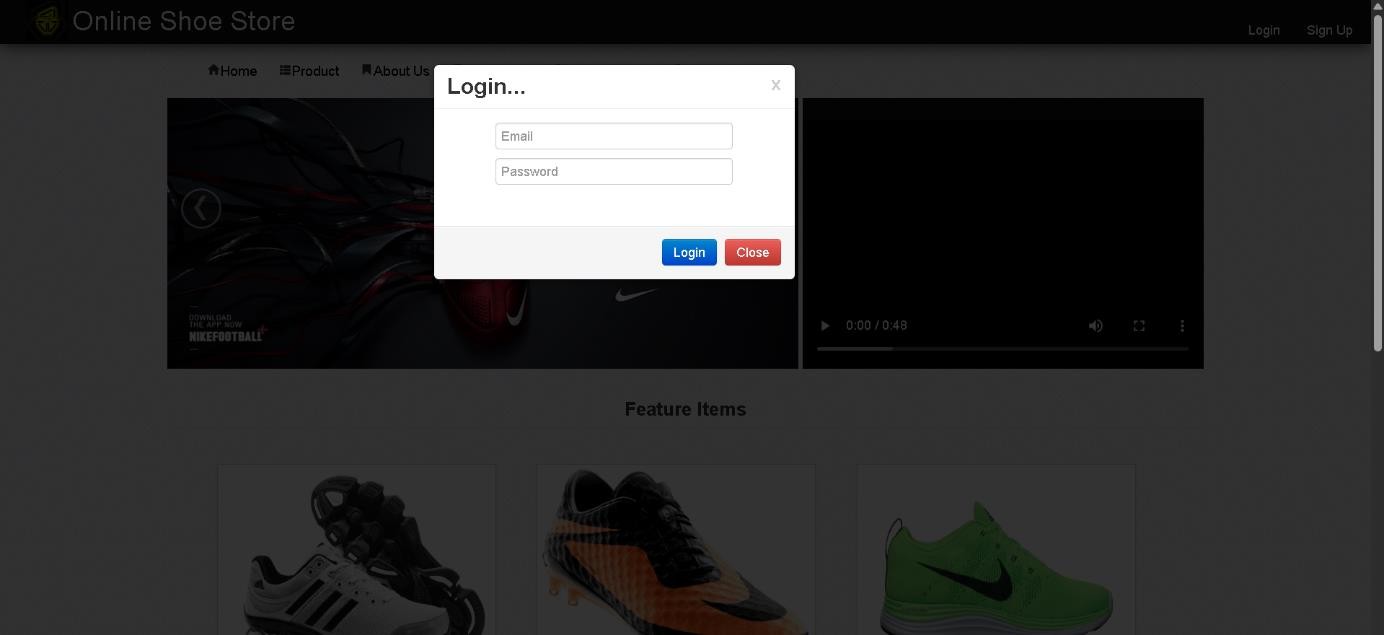
Product Table:

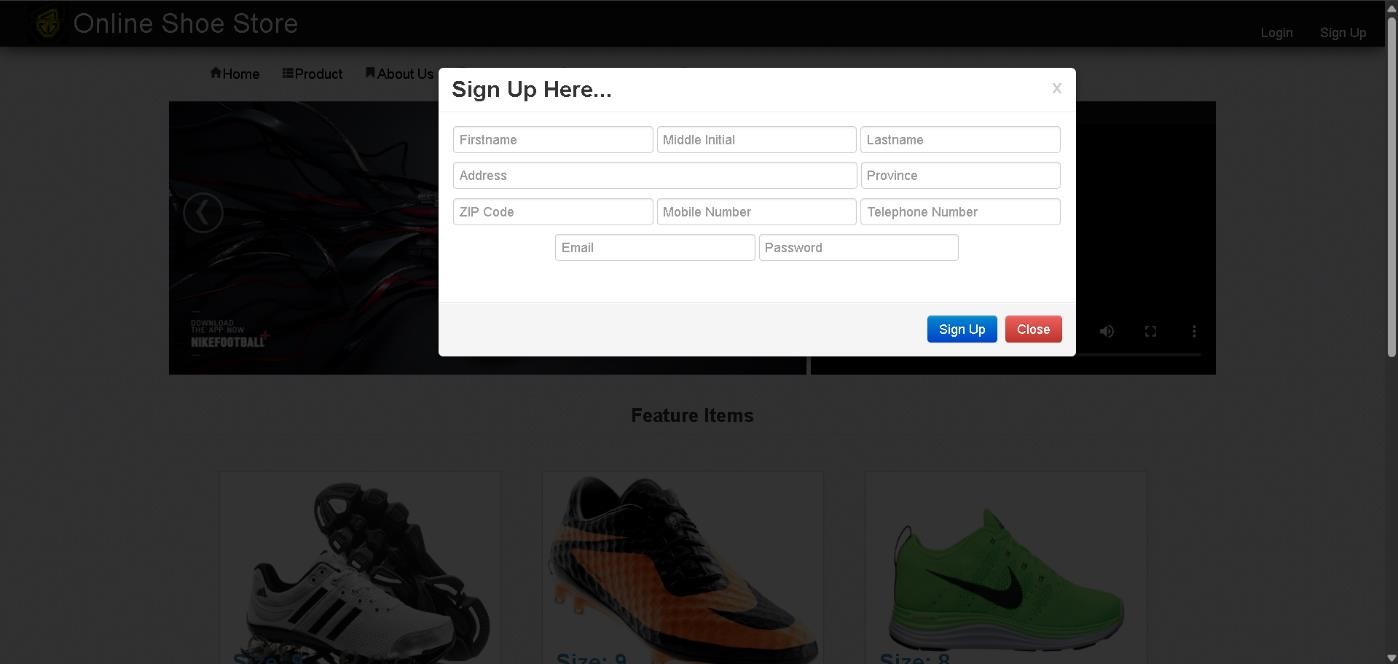
|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| product\_id | UUID / INT | PK, AUTO\_INCREMENT |
| name | VARCHAR | NOT NULL |
| description | TEXT |  |
| brand | VARCHAR |  |
| category | VARCHAR | ENUM (Men, Women, Kids, etc.) |
| **Column** | **Type** | **Constraints** |
| price | DECIMAL | NOT NULL |
| discount | DECIMAL | DEFAULT 0 |
| created\_at | TIMESTAMP |  |

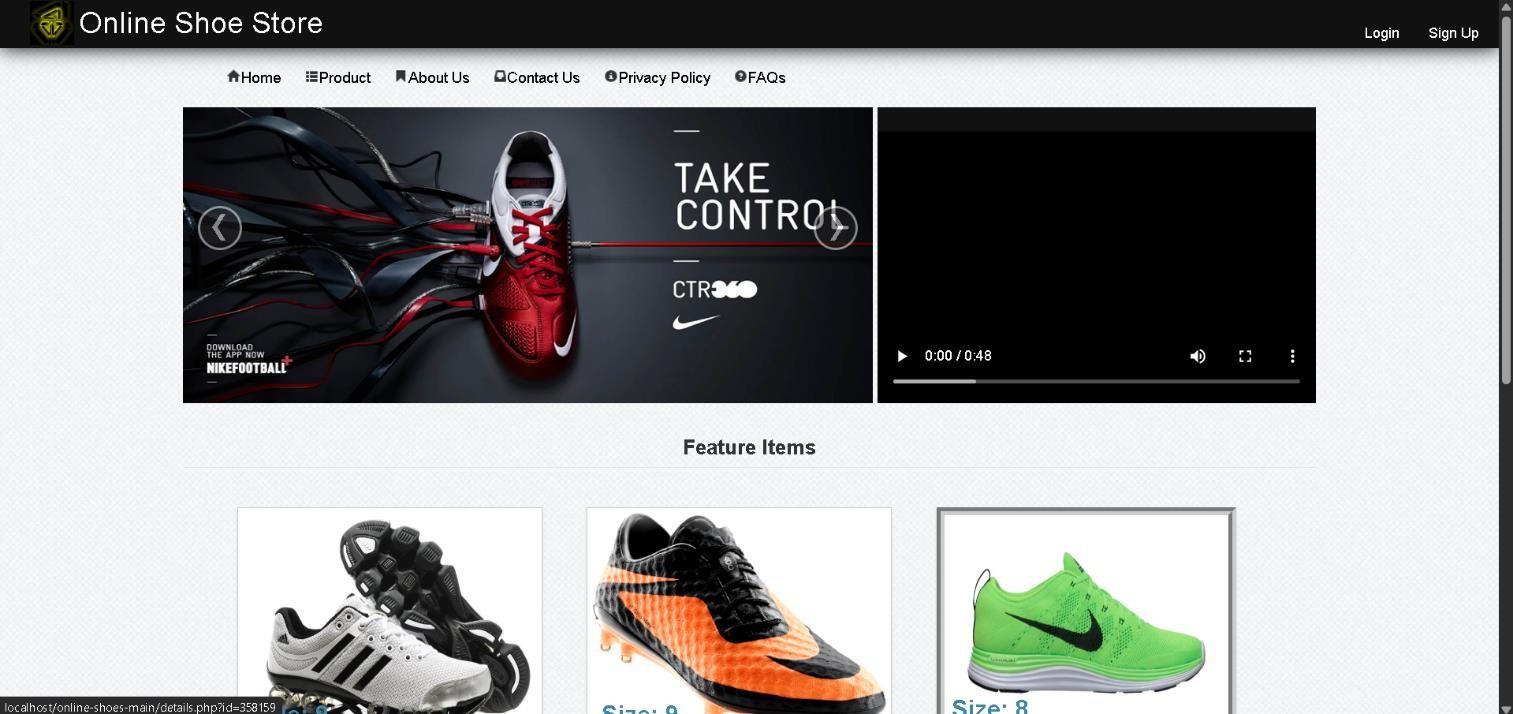
Inventory Table:

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| inventory\_id | UUID / INT | PK |
| product\_id | FK (Products) | NOT NULL |
| size | VARCHAR | e.g. '7', '8', '9.5' |
| quantity | INT | CHECK (quantity >= 0) |

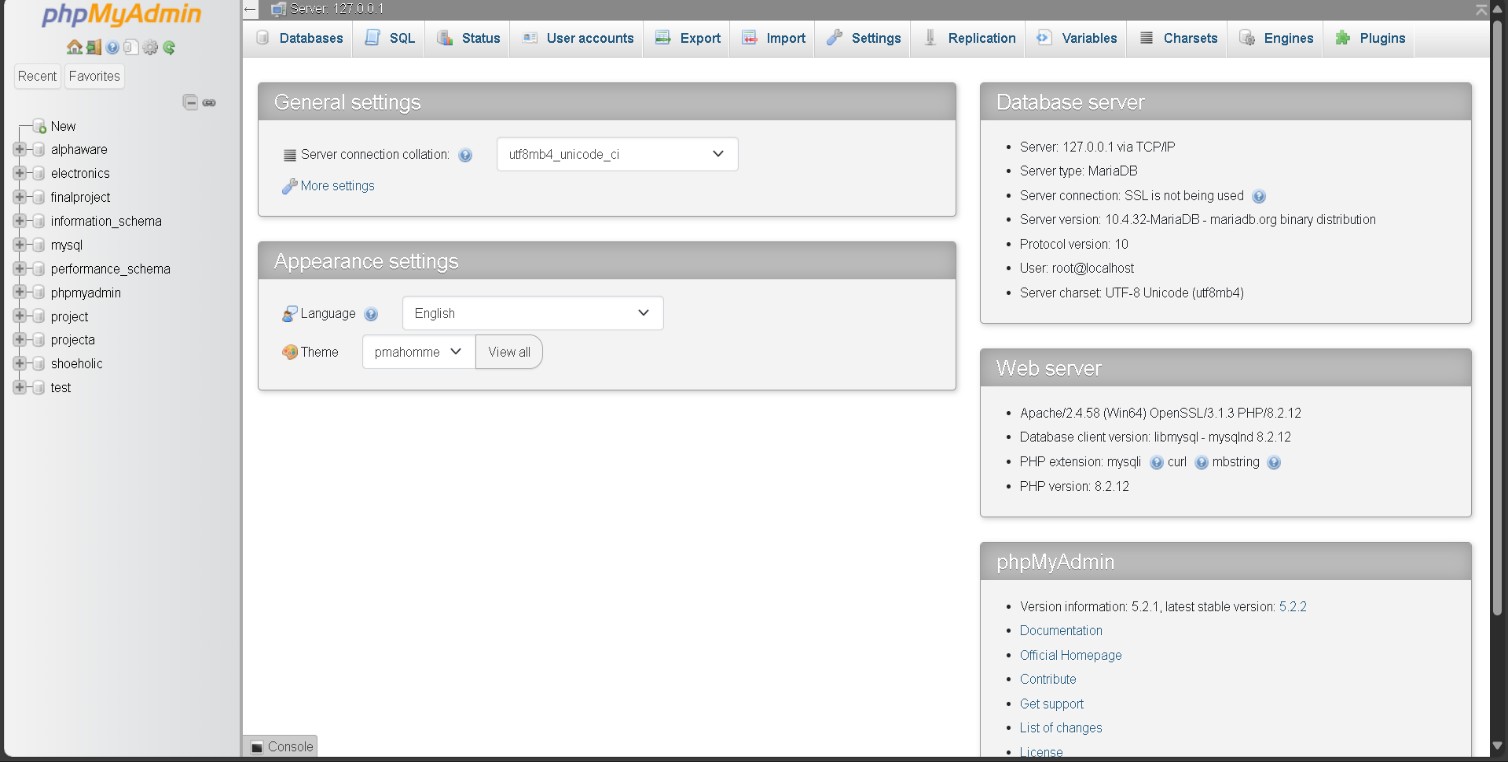
### 4.3 USER INTERFACE AND DESIGN



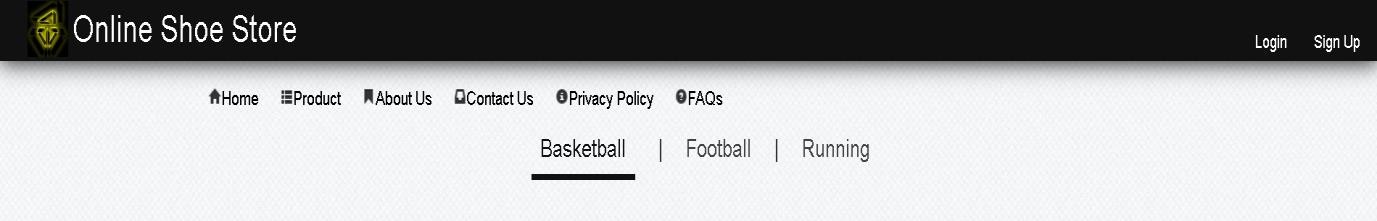


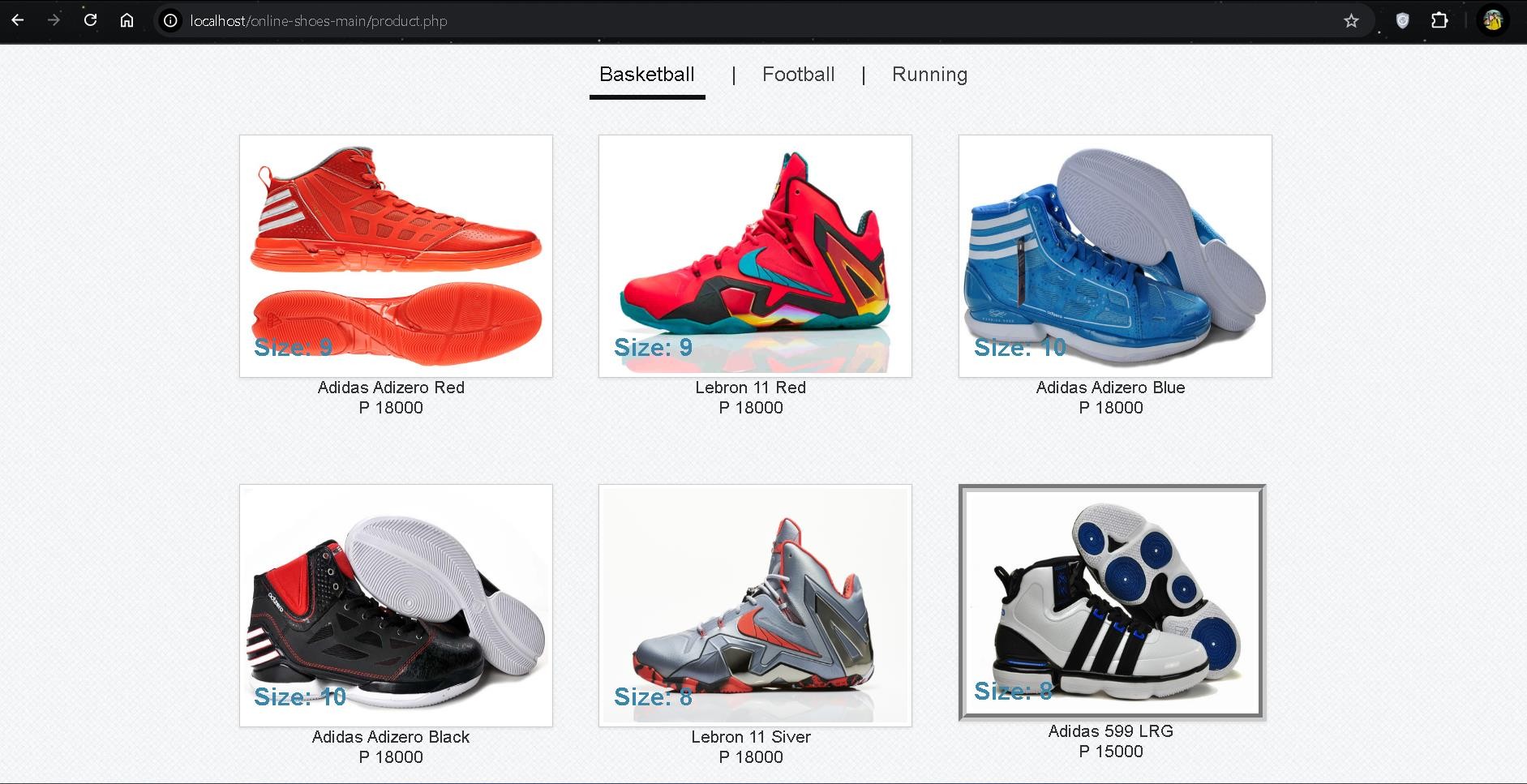


#### Data base



#### DashBoard





### 4.4 SECURITY ISSUES

4.4.1 Authentication and Authorization

Weak password policies and insufficient user authentication methods may lead to unauthorized access to user accounts or the admin panel.

4.4.2 SQL Injection

Improper handling of user input, such as login forms, search fields, and other data entry points, may expose the system to SQL injection attacks.

4.4.3 Cross-Site Scripting (XSS)

XSS vulnerabilities occur when user-generated content, such as comments on blog posts, is not properly sanitized, allowing malicious scripts to be executed in other users' browsers.

4.4.4 Cross-Site Request Forgery (CSRF)

CSRF attacks trick authenticated users into performing unintended actions on the system, such as making changes to their profile, billing details, or even performing admin actions.

4.4.5 Data Leakage and Privacy Issues

Sensitive user information such as personal details, billing records, and loyalty points could be exposed if proper data protection mechanisms are not in place.

4.4.6 Session Management Vulnerabilities

Improper session management can lead to session hijacking, where attackers gain control of a user's session after authentication.

4.4.7 Insecure APIs

The system may expose APIs for user registration, booking, payment processing, etc. If not secured properly, these APIs can be exploited by attackers to bypass the frontend or extract information directly.

4.4.8 Payment Fraud

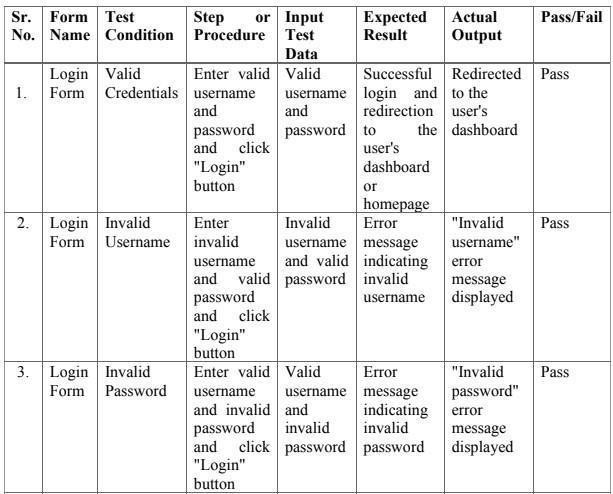
Since the system handles financial transactions (billing, memberships, etc.), there is a risk of fraudulent payments or unauthorized access to payment details

4.3.9 Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attackers may attempt to overwhelm the system with traffic or resource-intensive requests, causing service disruption.

4.3.10 Insecure Hosting and Deployment

If the server hosting environment is not secured, attackers can gain unauthorized access to the application, its database, or its codebase

#### 4.5 Test Cases



#### Chapter 5 System Coding, Implementation and Testing

**5.1 Coding Details.**

**5.2 Coding Efficiency**

**1. Dashboard.php**

<?php

require "function/session.php"; require "function/dbconn.php";

?>

<!DOCTYPE html>

<html>

<head>

<title>Online Shoe Store</title>

<link rel = "stylesheet" type = "text/css" href="css/style.css" media="all">

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">

<script src="js/bootstrap.js"></script>

<script src="js/jquery-1.7.2.min.js"></script>

<script src="js/carousel.js"></script>

<script src="js/button.js"></script>

<script src="js/dropdown.js"></script>

<script src="js/tab.js"></script>

<script src="js/tooltip.js"></script>

<script src="js/popover.js"></script>

<script src="js/collapse.js"></script>

<script src="js/modal.js"></script>

<script src="js/scrollspy.js"></script>

<script src="js/alert.js"></script>

<script src="js/transition.js"></script>

<script src="js/bootstrap.min.js"></script>

</head>

<body>

<div id="header">

<img src="img/logo.jpg">

<label>Online Shoe Store</label>

<?php

$id = (int) $\_SESSION['id'];

$query = $conn->query ("SELECT \* FROM customer WHERE customerid = '$id' ") or die

(mysqli\_error());

$fetch = $query->fetch\_array ();

?>

<ul>

<li><a href="function/logout.php"><i class="icon-off icon-white"></i>logout</a></li>

<li>Welcome:&nbsp;&nbsp;&nbsp;<a href="#profile" href data-toggle="modal"><i class="icon-user icon-white"></i><?php echo $fetch['firstname']; ?>&nbsp;<?php echo $fetch['lastname'];?></a></li> </ul>

</div>

<div id="profile" class="modal hide fade" tabindex="-1" role="dialog" arialabelledby="myModalLabel" aria-hidden="true" style="width:700px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button>

<h3 id="myModalLabel">My Account</h3>

</div>

<div class="modal-body">

<?php

$id = (int) $\_SESSION['id'];

$query = $conn->query ("SELECT \* FROM customer WHERE customerid = '$id' ") or die (mysqli\_error());

$fetch = $query->fetch\_array ();

?>

<center>

<form method="post">

<center>

<table>

<tr>

<td class="profile">Name:</td><td class="profile"><?php echo $fetch['firstname'];?>&nbsp;<?php echo $fetch['mi'];?>&nbsp;<?php echo $fetch['lastname'];?></td> </tr>

<tr>

<td class="profile">Address:</td><td class="profile"><?php echo $fetch['address'];?></td> </tr>

<tr>

<td class="profile">Country:</td><td class="profile"><?php echo $fetch['country'];?></td> </tr>

<tr>

<td class="profile">ZIP Code:</td><td class="profile"><?php echo $fetch['zipcode'];?></td> </tr>

<tr>

<td class="profile">Mobile Number:</td><td class="profile"><?php echo $fetch['mobile'];?></td> </tr>

<tr>

<td class="profile">Telephone Number:</td><td class="profile"><?php echo $fetch['telephone'];?></td>

</tr>

<tr>

<td class="profile">Email:</td><td class="profile"><?php echo $fetch['email'];?></td> </tr>

</table>

</center>

</div>

<div class="modal-footer">

<a href="account.php?id=<?php echo $fetch['customerid']; ?>"><input type="button" class="btn btnsuccess" name="edit" value="Edit Account"></a>

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button>

</div>

</form>

</div>

<br>

<div id="container">

<div id="content">

<div class="nav">

<ul>

<li><a href="home.php"><i class="icon-home"></i>Home</a></li>

<li><a href="product1.php"><i class="icon-th-list"></i>Product</a>

<li><a href="aboutus1.php"><i class="icon-bookmark"></i>About Us</a></li>

<li><a href="contactus1.php"><i class="icon-inbox"></i>Contact Us</a></li>

<li><a href="privacy1.php"><i class="icon-info-sign"></i>Privacy Policy</a></li> <li><a href="faqs1.php"><i class="icon-question-sign"></i>FAQs</a></li>

</ul>

</div>

<div id="carousel">

<div id="myCarousel" class="carousel slide">

<div class="carousel-inner">

<div class="active item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner1.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner2.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner3.jpg" class="carousel"></div>

</div>

<a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>

<a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>

</div>

</div>

<div id="video">

<video controls autoplay width="445px" height="300px">

<source src="video/commercial.mp4" type="video/mp4">

</video>

</div>

<div id="product" style="position:relative; margin-top:30%;">

<center><h2><legend>Featured Items</legend></h2></center>

<br />

<?php

$query = $conn->query("SELECT \*FROM product WHERE category='feature' ORDER BY product\_id DESC") or die (mysqli\_error());

while($fetch = $query->fetch\_array())

{

$pid = $fetch['product\_id'];

$query1 = $conn->query("SELECT \* FROM stock WHERE product\_id = '$pid'") or die (mysqli\_error());

$rows = $query1->fetch\_array();

$qty = $rows['qty']; if($qty <= 5){

}else{

echo "<div class='float'>";

echo "<center>"; echo "<a href='details.php?id=".$fetch['product\_id']."'><img class='img-polaroid' src='photo/".$fetch['product\_image']."' height = '300px' width = '300px'></a>"; echo "".$fetch['product\_name'].""; echo "<br />";

echo "P ".$fetch['product\_price'].""; echo "<br />"; echo "<h3 class='text-info' style='position:absolute; margin-top:-90px; text-indent:15px;'> Size:

".$fetch['product\_size']."</h3>"; echo "</center>"; echo "</div>";

}

}

?>

</div>

</div>

<br />

</div>

<br />

<br />

<div id="footer">

<div id="foot">

<h4>Links</h4>

<ul>

<a href="http://www.facebook.com/OnlineShoeStore"><li>Facebook</li></a>

<a href="http://www.twitter.com/OnlineShoeStore"><li>Twitter</li></a>

<a href="http://www.pinterest.com/OnlineShoeStore"><li>Pinterest</li></a>

<a href="http://www.tumblr.com/OnlineShoeStore"><li>Tumblr</li></a> </ul>

</div>

</div>

</body>

</html>

**Login Page** <?php

require "function/login.php";

require "function/customer\_signup.php";

?>

<!DOCTYPE html>

<html>

<head>

<title>Online Shoe Store</title>

<link rel="icon" href="img/logo.jpg" />

<link rel = "stylesheet" type = "text/css" href="css/style.css" media="all">

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">

<script src="js/bootstrap.js"></script>

<script src="js/jquery-1.7.2.min.js"></script>

<script src="js/carousel.js"></script>

<script src="js/button.js"></script>

<script src="js/dropdown.js"></script>

<script src="js/tab.js"></script>

<script src="js/tooltip.js"></script>

<script src="js/popover.js"></script>

<script src="js/collapse.js"></script>

<script src="js/modal.js"></script>

<script src="js/scrollspy.js"></script>

<script src="js/alert.js"></script>

<script src="js/transition.js"></script>

<script src="js/bootstrap.min.js"></script>

</head>

<body>

<div id="header">

<img src="img/logo.jpg">

<label>Online Shoe Store</label>

<ul>

<li><a href="#signup" data-toggle="modal">Sign Up</a></li>

<li><a href="#login" data-toggle="modal">Login</a></li> </ul>

</div>

<div id="login" class="modal hide fade" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true" style="width:400px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button> <h3 id="myModalLabel">Login...</h3>

</div>

<div class="modal-body">

<form method="post">

<center>

<input type="email" name="email" placeholder="Email" style="width:250px;">

<input type="password" name="password" placeholder="Password" style="width:250px;">

</center>

</div>

<div class="modal-footer">

<input class="btn btn-primary" type="submit" name="login" value="Login">

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button> </form>

</div>

</div>

<div id="signup" class="modal hide fade" tabindex="-1" role="dialog" arialabelledby="myModalLabel" aria-hidden="true" style="width:700px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button>

<h3 id="myModalLabel">Sign Up Here...</h3>

</div>

<div class="modal-body">

<center>

<form method="post">

<input type="text" name="firstname" placeholder="Firstname" required>

<input type="text" name="mi" placeholder="Middle Initial" maxlength="1" required>

<input type="text" name="lastname" placeholder="Lastname" required>

<input type="text" name="address" placeholder="Address" style="width:430px;"required>

<input type="text" name="country" placeholder="Province" required>

<input type="text" name="zipcode" placeholder="ZIP Code" required maxlength="4">

<input type="text" name="mobile" placeholder="Mobile Number" maxlength="11">

<input type="text" name="telephone" placeholder="Telephone Number" maxlength="8">

<input type="email" name="email" placeholder="Email" required>

<input type="password" name="password" placeholder="Password" required> </center>

</div>

<div class="modal-footer">

<input type="submit" class="btn btn-primary" name="signup" value="Sign Up">

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button> </div>

</form>

</div>

<br>

<div id="container">

<div class="nav">

<ul>

<li><a href="index.php"><i class="icon-home"></i>Home</a></li>

<li><a href="product.php"><i class="icon-th-list"></i>Product</a>

<li><a href="aboutus.php"><i class="icon-bookmark"></i>About Us</a></li>

<li><a href="contactus.php"><i class="icon-inbox"></i>Contact Us</a></li>

<li><a href="privacy.php"><i class="icon-info-sign"></i>Privacy Policy</a></li> <li><a href="faqs.php"><i class="icon-question-sign"></i>FAQs</a></li> </ul>

</div>

<div id="carousel">

<div id="myCarousel" class="carousel slide">

<div class="carousel-inner">

<div class="active item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner1.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner2.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner3.jpg" class="carousel"></div>

</div>

<a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>

<a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>

</div>

</div>

<div id="video">

<video controls autoplay width="445px" height="300px">

<source src="video/commercial.mp4" type="video/mp4">

</video>

</div>

<div id="content">

<div id="product" style="position:relative; margin-top:30%;">

<center><h2><legend>Feature Items</legend></h2></center> <br />

<?php

$query = $conn->query("SELECT \*FROM product WHERE category='feature' ORDER BY product\_id DESC") or die (mysqli\_error()); while($fetch = $query->fetch\_array())

{

$pid = $fetch['product\_id'];

$query1 = $conn->query("SELECT \* FROM stock WHERE product\_id = '$pid'") or die (mysqli\_error());

$rows = $query1->fetch\_array();

$qty = $rows['qty'];

if($qty <= 5){

}else{

echo "<div class='float'>"; echo "<center>";

echo "<a href='details.php?id=".$fetch['product\_id']."'><img class='img-polaroid' src='photo/".$fetch['product\_image']."' height = '300px' width = '300px'></a>"; echo " ".$fetch['product\_name'].""; echo "<br />";

echo "P ".$fetch['product\_price']."";

echo "<br />"; echo "<h3 class='text-info' style='position:absolute; margin-top:-90px; text-indent:15px;'> Size:

".$fetch['product\_size']."</h3>"; echo "</center>"; echo "</div>";

}

}

?>

</div>

</div>

<br />

</div>

<br />

<div id="footer">ss

<div id="foot">

<h4>Links</h4>

<ul>

<a href="http://www.facebook.com/OnlineShoeStore"><li>Facebook</li></a>

<a href="http://www.twitter.com/OnlineShoeStore"><li>Twitter</li></a>

<a href="http://www.pinterest.com/OnlineShoeStore"><li>Pinterest</li></a>

<a href="http://www.tumblr.com/OnlineShoeStore"><li>Tumblr</li></a> </ul>

</div>

</div>

</body>

</html>

#### Register page

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Sign Up Modal</title> <link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"> </head>

<body>

<!-- Sign Up Button (to open modal) -->

<button type="button" class="btn btn-primary" data-toggle="modal" datatarget="#signupModal">

Sign Up

</button>

<!-- Sign Up Modal -->

<div class="modal fade" id="signupModal" tabindex="-1" arialabelledby="signupModalLabel" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="signupModalLabel">Sign Up Here...</h5> <button type="button" class="close" data-dismiss="modal" arialabel="Close">&times;</button>

</div>

<div class="modal-body">

<form>

<div class="form-row">

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="Firstname">

</div>

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="Middle Initial"> </div>

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="Lastname">

</div>

</div>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" placeholder="Address">

</div>

<div class="form-group col-md-6">

<input type="text" class="form-control" placeholder="Province">

</div>

</div>

<div class="form-row">

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="ZIP Code">

</div>

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="Mobile Number">

</div>

<div class="form-group col-md-4">

<input type="text" class="form-control" placeholder="Telephone Number">

</div>

</div>

<div class="form-row">

<div class="form-group col-md-6">

<input type="email" class="form-control" placeholder="Email">

</div>

<div class="form-group col-md-6">

<input type="password" class="form-control" placeholder="Password">

</div>

</div>

</form>

</div>

<div class="modal-footer">

<button class="btn btn-primary">Sign Up</button>

<button class="btn btn-danger" data-dismiss="modal">Close</button> </div>

</div>

</div>

</div>

<!-- Bootstrap JS + jQuery -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script >

</body>

</html>

#### Admin Feature

<?php

include("../function/session.php"); include("../db/dbconn.php");

?>

<!DOCTYPE html>

<html>

<head>

<title>Online Shoe Store</title>

<link rel = "stylesheet" type = "text/css" href="../css/style.css" media="all">

<link rel="stylesheet" type="text/css" href="../css/bootstrap.css">

<script src="../js/bootstrap.js"></script>

<script src="../js/jquery-1.7.2.min.js"></script>

<script src="../js/carousel.js"></script>

<script src="../js/button.js"></script>

<script src="../js/dropdown.js"></script>

<script src="../js/tab.js"></script>

<script src="../js/tooltip.js"></script>

<script src="../js/popover.js"></script> <script src="../js/collapse.js"></script>

<script src="../js/modal.js"></script>

<script src="../js/scrollspy.js"></script>

<script src="../js/alert.js"></script>

<script src="../js/transition.js"></script>

<script src="../js/bootstrap.min.js"></script>

<script src="../javascripts/filter.js" type="text/javascript" charset="utf-8"></script> <script src="../jscript/jquery-1.9.1.js" type="text/javascript"></script>

<!--Le Facebox-->

<link href="../facefiles/facebox.css" media="screen" rel="stylesheet" type="text/css" />

<script src="../facefiles/jquery-1.9.js" type="text/javascript"></script>

<script src="../facefiles/jquery-1.2.2.pack.js" type="text/javascript"></script>

<script src="../facefiles/facebox.js" type="text/javascript"></script>

<script type="text/javascript"> jQuery(document).ready(function($) {

$('a[rel\*=facebox]').facebox()

})

</script>

</head>

<body>

<div id="header" style="position:fixed;">

<img src="../img/logo.jpg">

<label>Online Shoe Store</label>

<?php

$id = (int) $\_SESSION['id'];

$query = $conn->query ("SELECT \* FROM admin WHERE adminid = '$id' ") or die (mysqli\_error());

$fetch = $query->fetch\_array (); ?>

<ul>

<li><a href="../function/admin\_logout.php"><i class="icon-off iconwhite"></i>logout</a></li>

<li>Welcome:&nbsp;&nbsp;&nbsp;<i class="icon-user icon-white"></i><?php echo $fetch['username']; ?></a></li>

</ul>

</div>

<br>

<a href="#add" role="button" class="btn btn-info" data-toggle="modal"

style="position:absolute;margin-left:222px; margin-top:140px; z-index:-1000;"><i class="icon-plus-sign icon-white"></i>Add Product</a>

<div id="add" class="modal hide fade" tabindex="-1" role="dialog" arialabelledby="myModalLabel" aria-hidden="true" style="width:400px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button> <h3 id="myModalLabel">Add Product...</h3>

</div>

<div class="modal-body">

<form method="post" enctype="multipart/form-data">

<center>

<table>

<tr>

<td><input type="file" name="product\_image" required></td>

</tr>

<?php include("random\_id.php"); echo '<tr>

<td><input type="hidden" name="product\_code" value="'.$code.'" required></td>

<tr/>';

?>

<tr>

<td><input type="text" name="product\_name" placeholder="Product Name" style="width:250px;" required></td>

<tr/> <tr>

<td><input type="text" name="product\_price" placeholder="Price" style="width:250px;" required></td>

</tr> <tr>

<td><input type="text" name="product\_size" placeholder="Size" style="width:250px;" maxLength="2" required></td>

</tr>

<tr>

<td><input type="text" name="brand" placeholder="Brand Name "

style="width:250px;" required></td>

</tr> <tr>

<td><input type="number" name="qty" placeholder="No. of Stock" style="width:250px;" required></td>

</tr> <tr>

<td><input type="hidden" name="category" value="feature"></td>

</tr>

</table>

</center>

</div>

<div class="modal-footer">

<input class="btn btn-primary" type="submit" name="add" value="Add">

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button>

</form>

</div>

</div>

<?php

if (isset($\_POST['add']))

{

$product\_code = $\_POST['product\_code'];

$product\_name = $\_POST['product\_name'];

$product\_price = $\_POST['product\_price'];

$product\_size = $\_POST['product\_size'];

$brand = $\_POST['brand'];

$category = $\_POST['category'];

$qty = $\_POST['qty'];

$code = rand(0,98987787866533499);

$name = $code.$\_FILES["product\_image"] ["name"];

$type = $\_FILES["product\_image"] ["type"];

$size = $\_FILES["product\_image"] ["size"];

$temp = $\_FILES["product\_image"] ["tmp\_name"];

$error = $\_FILES["product\_image"] ["error"];

if ($error > 0){

die("Error uploading file! Code $error.");} else

{

if($size > 30000000000) //conditions for the file

{

die("Format is not allowed or file size is too big!");

} else

{

move\_uploaded\_file($temp,"../photo/".$name);

$q1 = $conn->query("INSERT INTO product ( product\_id,product\_name, product\_price, product\_size, product\_image, brand, category)

VALUES ('$product\_code','$product\_name','$product\_price','$product\_size','$name', '$brand', '$category')");

$q2 = $conn->query("INSERT INTO stock ( product\_id, qty) VALUES ('$product\_code','$qty')");

header ("location:admin\_feature.php");

}}

}

?>

<div id="leftnav">

<ul>

<li><a href="admin\_home.php" style="color:#333;">Dashboard</a></li> <li><a href="admin\_home.php">Products</a>

<ul>

<li><a href="admin\_feature.php "style="font-size:15px; marginleft:15px;">Features</a></li>

<li><a href="admin\_product.php "style="font-size:15px; marginleft:15px;">Basketball</a></li>

<li><a href="admin\_football.php" style="font-size:15px; marginleft:15px;">Football</a></li>

<li><a href="admin\_running.php"style="font-size:15px; marginleft:15px;">Running</a></li>

</ul> </li>

<li><a href="transaction.php">Transactions</a></li>

<li><a href="customer.php">Customers</a></li>

<li><a href="message.php">Messages</a></li>

<li><a href="order.php">Orders</a></li>

</ul>

</div>

<div id="rightcontent" style="position:absolute; top:10%;">

<div class="alert alert-info"><center><h2>Features</h2></center></div>

<br />

<label style="padding:5px; float:right;"><input type="text" name="filter" placeholder="Search Product here..." id="filter"></label>

<br />

<div class="alert alert-info">

<table class="table table-hover" style="background-color:;"> <thead>

<tr style="font-size:20px;">

<th>Product Image</th>

<th>Product Name</th>

<th>Product Price</th>

<th>Product Sizes</th>

<th>No. of Stock</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<?php

$query = $conn->query("SELECT \* FROM `product` WHERE category='feature' ORDER

BY product\_id DESC") or die(mysqli\_error());

while($fetch = $query->fetch\_array())

{

$id = $fetch['product\_id'];

?>

<tr class="del<?php echo $id?>">

<td><img class="img-polaroid" src = "../photo/<?php echo $fetch['product\_image']?>" height = "70px" width = "80px"></td>

<td><?php echo $fetch['product\_name']?></td>

<td><?php echo $fetch['product\_price']?></td>

<td><?php echo $fetch['product\_size']?></td>

<?php

$query1 = $conn->query("SELECT \* FROM `stock` WHERE product\_id='$id'") or die(mysqli\_error());

$fetch1 = $query1->fetch\_array();

$qty = $fetch1['qty'];

?>

<td><?php echo $fetch1['qty']?></td>

<td>

<?php

echo "<a href='stockin.php?id=".$id."' class='btn btn-success' rel='facebox'><i class='iconplus-sign icon-white'></i> Stock In</a> ";

echo "<a href='stockout.php?id=".$id."' class='btn btn-danger' rel='facebox'><i class='iconminus-sign icon-white'></i> Stock Out</a>";

?>

</td>

</tr>

<?php

}

?>

</tbody>

</table>

</div>

</div>

<?php

/\* stock in \*/

if(isset($\_POST['stockin'])){

$pid = $\_POST['pid'];

$result = $conn->query("SELECT \* FROM `stock` WHERE product\_id='$pid'") or die(mysqli\_error()); $row = $result->fetch\_array();

$old\_stck = $row['qty'];

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

$total = $old\_stck + $new\_stck;

$que = $conn->query("UPDATE `stock` SET `qty` = '$total' WHERE `product\_id`='$pid'") or die(mysqli\_error());

echo "<script>window.location = 'admin\_feature.php'</script>"; //header("Location:admin\_feature.php");

}

/\* stock out \*/

if(isset($\_POST['stockout'])){

$pid = $\_POST['pid'];

$result = $conn->query("SELECT \* FROM `stock` WHERE product\_id='$pid'") or die(mysqli\_error()); $row = $result->fetch\_array();

$old\_stck = $row['qty'];

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

$total = $old\_stck - $new\_stck;

$que = $conn->query("UPDATE `stock` SET `qty` = '$total' WHERE `product\_id`='$pid'") or die(mysqli\_error());

echo "<script>window.location = 'admin\_feature.php'</script>"; //header("location:admin\_feature.php");

}

?>

</body>

</html>

<script type="text/javascript">

$(document).ready( function() {

$('.remove').click( function() { var id = $(this).attr("id"); if(confirm("Are you sure you want to delete this product?")){

$.ajax({ type: "POST", url: "../function/remove.php",

data: ({id: id}), cache: false,

success: function(html){

$(".del"+id).fadeOut(2000, function(){ $(this).remove();});

}

}); }else{ return false;}

});

});

</script>

#### Data base connection

<?php

$conn = new mysqli('localhost', 'root', '', 'alphaware'); if(!$conn){

die("Fatal Error: Connection Error!");

}

?>

#### Login Process .php

<?php

require "dbconn.php";

if (isset($\_POST['login']))

{

$email=$\_POST['email'];

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

$result=$conn->query("SELECT \* FROM customer WHERE email='$email' AND password='$password' ")

or die ('cannot login' . mysqli\_error()); $row=$result->fetch\_array ();

$run\_num\_rows = $result->num\_rows;

if ($run\_num\_rows > 0 )

{

session\_start ();

$\_SESSION['id'] = $row['customerid']; header ("location:home.php");

} else

{

echo "<script>alert('Invalid Email or Password')</script>"; header("location:home.php");

}

}

?>

#### Register process.php

<?php

require "dbconn.php"; if (isset($\_POST['signup']))

{

$firstname=$\_POST['firstname'];

$mi=$\_POST['mi'];

$lastname=$\_POST['lastname'];

$address=$\_POST['address'];

$country=$\_POST['country'];

$zipcode=$\_POST['zipcode'];

$mobile=$\_POST['mobile'];

$telephone=$\_POST['telephone'];

$email=$\_POST['email'];

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

$query = $conn->query("SELECT \* FROM `customer` WHERE `email` = '$email'");

$check = $query->num\_rows; if($check == 1)

{

echo "<script>alert('EMAIL ALREADY EXIST')</script>";

} else

{

$conn->query ("INSERT INTO customer (firstname, mi, lastname, address, country, zipcode, mobile, telephone, email, password)

VALUES ('$firstname', '$mi', '$lastname', '$address', '$country', '$zipcode', '$mobile',

'$telephone', '$email', '$password')")

or die(mysqli\_error());

}

}

?>

#### session.php

<?php

session\_start();

if(!ISSET($\_SESSION['id']))

{

echo "<script>window.location = 'index.php';</script>";

}

?>

#### addcart .php

<?php

if (isset($\_POST['add']))

{

$prod\_id =$\_POST['product\_id'];

$cust\_id =$\_POST['customerid'];

$conn->query ("INSERT INTO cart (prod\_id,cust\_id) VALUES ('$prod\_id', '$cust\_id') ") or die(mysqli\_error());

header("location: product1.php");

}

?>

#### Index.php

<?php

require "function/login.php";

require "function/customer\_signup.php";

?>

<!DOCTYPE html>

<html>

<head>

<title>Online Shoe Store</title>

<link rel="icon" href="img/logo.jpg" />

<link rel = "stylesheet" type = "text/css" href="css/style.css" media="all">

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">

<script src="js/bootstrap.js"></script>

<script src="js/jquery-1.7.2.min.js"></script>

<script src="js/carousel.js"></script>

<script src="js/button.js"></script>

<script src="js/dropdown.js"></script>

<script src="js/tab.js"></script>

<script src="js/tooltip.js"></script>

<script src="js/popover.js"></script> <script src="js/collapse.js"></script>

<script src="js/modal.js"></script>

<script src="js/scrollspy.js"></script>

<script src="js/alert.js"></script>

<script src="js/transition.js"></script>

<script src="js/bootstrap.min.js"></script>

</head>

<body>

<div id="header">

<img src="img/logo.jpg">

<label>Online Shoe Store</label>

<ul>

<li><a href="#signup" data-toggle="modal">Sign Up</a></li> <li><a href="#login" data-toggle="modal">Login</a></li>

</ul>

</div>

<div id="login" class="modal hide fade" tabindex="-1" role="dialog" arialabelledby="myModalLabel" aria-hidden="true" style="width:400px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button> <h3 id="myModalLabel">Login...</h3>

</div>

<div class="modal-body">

<form method="post">

<center>

<input type="email" name="email" placeholder="Email" style="width:250px;">

<input type="password" name="password" placeholder="Password" style="width:250px;">

</center>

</div>

<div class="modal-footer">

<input class="btn btn-primary" type="submit" name="login" value="Login">

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button>

</form>

</div>

</div>

<div id="signup" class="modal hide fade" tabindex="-1" role="dialog" arialabelledby="myModalLabel" aria-hidden="true" style="width:700px;">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">x</button> <h3 id="myModalLabel">Sign Up Here...</h3>

</div>

<div class="modal-body">

<center>

<form method="post">

<input type="text" name="firstname" placeholder="Firstname" required>

<input type="text" name="mi" placeholder="Middle Initial" maxlength="1" required>

<input type="text" name="lastname" placeholder="Lastname" required>

<input type="text" name="address" placeholder="Address" style="width:430px;"required>

<input type="text" name="country" placeholder="Province" required>

<input type="text" name="zipcode" placeholder="ZIP Code" required maxlength="4">

<input type="text" name="mobile" placeholder="Mobile Number" maxlength="11">

<input type="text" name="telephone" placeholder="Telephone Number" maxlength="8">

<input type="email" name="email" placeholder="Email" required>

<input type="password" name="password" placeholder="Password" required>

</center>

</div>

<div class="modal-footer">

<input type="submit" class="btn btn-primary" name="signup" value="Sign Up">

<button class="btn btn-danger" data-dismiss="modal" aria-hidden="true">Close</button>

</div>

</form>

</div>

<br>

<div id="container">

<div class="nav">

<ul>

<li><a href="index.php"><i class="icon-home"></i>Home</a></li>

<li><a href="product.php"><i class="icon-th-list"></i>Product</a>

<li><a href="aboutus.php"><i class="icon-bookmark"></i>About Us</a></li>

<li><a href="contactus.php"><i class="icon-inbox"></i>Contact Us</a></li>

<li><a href="privacy.php"><i class="icon-info-sign"></i>Privacy Policy</a></li>

<li><a href="faqs.php"><i class="icon-question-sign"></i>FAQs</a></li>

</ul>

</div>

<div id="carousel">

<div id="myCarousel" class="carousel slide">

<div class="carousel-inner">

<div class="active item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner1.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner2.jpg" class="carousel"></div>

<div class="item" style="padding:0; border-bottom:0 solid #111;"><img src="img/banner3.jpg" class="carousel"></div>

</div>

<a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>

<a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a> </div>

</div>

<div id="video">

<video controls autoplay width="445px" height="300px">

<source src="video/commercial.mp4" type="video/mp4"> </video>

</div>

<div id="content">

<div id="product" style="position:relative; margin-top:30%;">

<center><h2><legend>Feature Items</legend></h2></center>

<br />

<?php

$query = $conn->query("SELECT \*FROM product WHERE category='feature' ORDER BY product\_id DESC") or die (mysqli\_error()); while($fetch = $query->fetch\_array())

{

$pid = $fetch['product\_id'];

$query1 = $conn->query("SELECT \* FROM stock WHERE product\_id = '$pid'") or die (mysqli\_error());

$rows = $query1->fetch\_array();

$qty = $rows['qty']; if($qty <= 5){

}else{

echo "<div class='float'>"; echo "<center>";

echo "<a href='details.php?id=".$fetch['product\_id']."'><img class='img-polaroid'

src='photo/".$fetch['product\_image']."' height = '300px' width = '300px'></a>"; echo " ".$fetch['product\_name'].""; echo "<br />";

echo "P ".$fetch['product\_price'].""; echo "<br />";

echo "<h3 class='text-info' style='position:absolute; margin-top:-90px; textindent:15px;'> Size: ".$fetch['product\_size']."</h3>";

echo "</center>"; echo "</div>";

}

}

?>

</div>

</div>

<br />

</div>

<br />

<div id="footer">ss

<div id="foot">

<h4>Links</h4>

<ul>

<a href="http://www.facebook.com/OnlineShoeStore"><li>Facebook</li></a>

<a href="http://www.twitter.com/OnlineShoeStore"><li>Twitter</li></a>

<a href="http://www.pinterest.com/OnlineShoeStore"><li>Pinterest</li></a> <a href="http://www.tumblr.com/OnlineShoeStore"><li>Tumblr</li></a>

</ul>

</div>

</div>

</body>

</html>

#### edit\_customer .php

<?php

include ("../db/dbconn.php"); include ("session.php"); if(ISSET($\_POST['edit']));

{

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

$firstname=$\_POST['firstname'];

$mi=$\_POST['mi'];

$lastname=$\_POST['lastname'];

$address=$\_POST['address'];

$country=$\_POST['country'];

$zipcode=$\_POST['zipcode'];

$mobile=$\_POST['mobile'];

$telephone=$\_POST['telephone'];

$email=$\_POST['email'];

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

$conn->query("UPDATE customer SET firstname='$firstname', mi='$mi', lastname='$lastname', address='$address', country='$country', zipcode='$zipcode', mobile='$mobile', telephone='$telephone', email='$email', password='$password' WHERE customerid='$id' ") or die (mysqli\_error());

header("location:../home.php");

}

?>

#### Logout page .php

<?php

session\_start();

session\_destroy();

header("location:../index.php");

?> mysqli\_query($con, "DELETE FROM tblusers WHERE ID='$id'"); header("Location: users.php");

}

?>

#### Testing Approach

##### 5.3.1 Unit Testing

Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but do confine the test within one module.

In our application we tested each module particularly, to see whether there are any bugs or error while executing the module. Changes were made if any error was found. We went through a pattern of testing which was defined in system testing chapter.

##### 5.3.2 Integrated Testing

Integrated Testing brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. After completing unit testing, we combined all the modules together for testing the entire application. Through this technique we came to know about the application limits and features are tested here. .

#### Chapter 6 Conclusion and Future Work

This project successfully developed an online shoe shop platform designed to streamline

the buying and selling of footwear through a digital interface. The system includes key functionalities such as user registration and authentication, product browsing, cart management, order placement, and administrative controls for inventory and user management. The implementation utilized standard web technologies, including HTML, CSS, JavaScript, and a backend framework with database connectivity, ensuring reliable performance and data integrity.

Future work will focus on expanding the platform's capabilities to improve user experience and operational efficiency. Planned enhancements include the integration of a secure online payment gateway, implementation of advanced product search and filtering features, and the deployment of machine learning algorithms for personalized product recommendations. Additionally, mobile application development is under consideration to increase accessibility. Ongoing security assessments and scalability improvements will be prioritized to support increased user demand and ensure compliance with data protection standards.

#### Chapter 7 References

1. PHP Documentation - https://[www.php.net/docs.php](http://www.php.net/docs.php)

1. MySQL Documentation - https://dev.mysql.com/doc/

1. XAMPP Official Site - https://[www.apachefriends.org/index.html](http://www.apachefriends.org/index.html)
2. HTML & CSS Guide - https://developer.mozilla.org/en-US/docs/Web

1. JavaScript Reference - https://developer.mozilla.org/en-US/docs/Web/JavaScript