E - COMMERCE WEBSITE

A PROJECT REPORT
for
Major Project (CA301P)
Session (2025-26)

Submitted by
HARSH GUPTA
(202410116100082)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. Rabi.N Panda
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATION
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(Nov -2025)

CERTIFICATE

Certified that **Harsh Gupta (202410116100082)** have carried out the project work having **"E – COMMERCE WEBSITE."** (Major Project, CA301P) for **Master of Computer Application** from **Dr. A.P.J. Abdul Kalam Technical University (AKTU)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself, and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Rabi.N Panda   **Dr. Sachin Malhotra**
Associate Professor   **Dean**
Department of Computer Applications   **Department of Computer Applications**
KIET Group of Institutions, Ghaziabad   **KIET Group of Institutions, Ghaziabad**

ACKNOWLEDGEMENTS

HARSH GUPTA

(202410116100082)

E - COMMERCE WEBSITE

ABSTRACT

The rapid growth of digital technologies has significantly transformed the way consumers shop and businesses operate. E-commerce has become an integral part of the global economy, providing convenience, accessibility, and a wide range of products to customers across geographical boundaries. This project aims to design and develop a full-stack e-commerce website that offers a seamless and secure online shopping experience. The system is implemented using **React.js** and **Tailwind CSS** for the frontend to ensure an interactive, responsive, and visually appealing user interface. The **backend** is developed using **Node.js** and **Express.js**, which provide a robust and scalable server-side architecture for handling various client requests and managing data flow efficiently.

For data storage and management, **MongoDB** is utilized as a NoSQL database to maintain flexibility and scalability in handling dynamic datasets such as user information, product details, and transaction records. The project also integrates **JWT (JSON Web Token)** and **OAuth** authentication mechanisms to ensure secure user access, prevent unauthorized data manipulation, and safeguard user credentials. The application supports core functionalities such as user registration and login, product browsing, cart management, order placement, and payment gateway integration.

This e-commerce platform emphasizes modern web development principles, including component-based architecture, RESTful API design, and responsive UI/UX practices. The outcome of this project is a fully functional and secure online shopping platform that demonstrates the integration of frontend and backend technologies in a real-world application. The system not only provides a user-friendly interface but also ensures performance, reliability, and security, making it a practical and scalable solution for modern online retail businesses.

TABLE OF CONTENTS

INTRODUCTION

Overview

In the past decade, the internet has revolutionized the way people communicate, work, and conduct business. One of the most significant transformations brought about by digital innovation is the emergence of electronic commerce (e-commerce). E-commerce refers to the process of buying and selling goods or services over the internet. With the rapid growth of online platforms and digital payment systems, consumers now have the convenience of shopping from anywhere at any time, while businesses can reach a wider audience beyond geographical limitations.

This project focuses on the development of a **full-stack e-commerce website** that provides an efficient and secure platform for online shopping. The system enables users to browse a catalog of products, add items to their cart, make secure payments, and manage their orders. The platform also includes administrative functionalities to manage products, users, and orders efficiently. The website is built using **React.js** and **Tailwind CSS** for the frontend, **Node.js** and **Express.js** for the backend, and **MongoDB** as the database. Additionally, **JWT (JSON Web Token)** and **OAuth** are implemented to ensure secure authentication and authorization processes.

Motivation

With the increasing reliance on digital platforms, the demand for efficient and user-friendly e-commerce systems has grown exponentially. Many small and medium-scale businesses face challenges in developing customized online stores that meet modern usability and security standards. This project was motivated by the need to create a scalable, secure, and easy-to-use e-commerce solution that could serve as a foundation for real-world business applications.

The project also serves as an opportunity to apply theoretical knowledge of full-stack development into practice. By combining technologies like React, Node.js, and MongoDB, the goal is to demonstrate the integration of modern web technologies into a cohesive system that emphasizes scalability, performance, and user experience.

Objectives

The main objective of this project is to design and implement a **full-stack e-commerce web application** with a secure and responsive user interface. The specific goals include:

1. To design a responsive and user-friendly frontend using **React.js** and **Tailwind CSS**.
2. To develop a robust backend system using **Node.js** and **Express.js** that can efficiently manage data flow between the client and the server.
3. To use **MongoDB** as the database system for storing user, product, and order data in a flexible, document-oriented structure.
4. To implement **secure authentication** using **JWT and OAuth** for login, registration, and user access control.
5. To integrate core e-commerce functionalities, including product browsing, cart management, order placement, and payment integration.
6. To ensure scalability, maintainability, and modular design for future enhancements.

Scope of the Project

The scope of this project is limited to building a fully functional e-commerce website that provides a secure and interactive environment for both customers and administrators. The system includes the following modules:

- **User Module:** Allows users to register, log in, browse products, add items to the cart, and place orders.
- **Admin Module:** Enables administrators to add, update, or delete products and manage users and orders.
- **Product Module:** Manages product details such as name, price, description, category, and stock availability.
- **Order Module:** Handles the order process, including order placement, status updates, and history tracking.
- **Authentication Module:** Ensures secure access using JWT and OAuth-based login.

While the project focuses primarily on the web application, it can be extended in the future with features such as a mobile application, AI-driven recommendations, and advanced analytics.

Literature Review

The evolution of e-commerce has drastically transformed how people shop, interact, and conduct business online. Over the years, several technological advancements have contributed to the growth and sophistication of e-commerce systems. This chapter presents a comprehensive review of the literature and technologies relevant to the development of an e-commerce platform. It explores existing e-commerce systems, their architecture, features, and challenges, as well as the various technologies that have emerged to support secure, scalable, and user-friendly online shopping environments.

The literature review also provides insights into the tools and technologies used in this project—**React.js**, **Tailwind CSS**, **Node.js**, **Express.js**, **MongoDB**, and authentication mechanisms such as **JWT** and **OAuth**—and discusses their significance in modern web development.

Overview of E-Commerce Systems

E-commerce systems are online platforms that facilitate the buying and selling of goods and services through the internet. They typically consist of a **frontend interface** for users, a **backend system** that handles business logic and data processing, and a **database** that stores product, order, and user information.

Over the years, many e-commerce models have evolved, including:

- **B2C (Business-to-Consumer):** Businesses sell products directly to customers (e.g., Amazon, Flipkart).
- **B2B (Business-to-Business):** Transactions occur between businesses (e.g., Alibaba).
- **C2C (Consumer-to-Consumer):** Users sell products to other users via a platform (e.g., eBay).
- **C2B (Consumer-to-Business):** Individuals offer products or services to companies (e.g., Upwork).

The key features of an effective e-commerce platform include product catalog management, secure payment gateways, user authentication, responsive design, and efficient order management.

Review of Existing E-Commerce Platforms

Several major e-commerce platforms have set benchmarks for design, usability, and performance:

1. **Amazon:**
   Launched in 1995, Amazon is one of the largest global e-commerce platforms. It

offers advanced personalization, AI-driven product recommendations, and efficient logistics. Its architecture relies heavily on microservices and cloud infrastructure.

2. **Flipkart:**
Founded in India, Flipkart focuses on a localized approach to e-commerce, providing features such as cash-on-delivery, regional language support, and personalized deals. The system uses technologies like React, Node.js, and Elasticsearch for scalability and speed.

3. **Shopify:**
Shopify provides a platform-as-a-service (PaaS) solution for businesses to build their online stores. It includes a built-in CMS, payment gateway, and analytics tools. However, it lacks full customization freedom due to its hosted nature.

4. **WooCommerce & Magento:**
Both are open-source platforms that integrate with existing CMS systems like WordPress. They provide flexibility for developers but require technical expertise to manage performance and security effectively.

Despite their advantages, these platforms often have limitations such as high costs, limited customization, or dependency on third-party services. Therefore, developing a **custom full-stack e-commerce system** offers greater flexibility, security, and learning opportunities.

Review of Existing E-Commerce Platforms

Several major e-commerce platforms have set benchmarks for design, usability, and performance:

1. **Amazon:**
Launched in 1995, Amazon is one of the largest global e-commerce platforms. It offers advanced personalization, AI-driven product recommendations, and efficient logistics. Its architecture relies heavily on microservices and cloud infrastructure.

2. **Flipkart:**
Founded in India, Flipkart focuses on a localized approach to e-commerce, providing features such as cash-on-delivery, regional language support, and personalized deals. The system uses technologies like React, Node.js, and Elasticsearch for scalability and speed.

3. **Shopify:**
Shopify provides a platform-as-a-service (PaaS) solution for businesses to build their online stores. It includes a built-in CMS, payment gateway, and analytics tools. However, it lacks full customization freedom due to its hosted nature.

4. **WooCommerce & Magento:**
Both are open-source platforms that integrate with existing CMS systems like WordPress. They provide flexibility for developers but require technical expertise to manage performance and security effectively.

Despite their advantages, these platforms often have limitations such as high costs, limited customization, or dependency on third-party services. Therefore, developing a **custom full-stack e-commerce system** offers greater flexibility, security, and learning opportunities.

OBJECTIVES

The primary objective of this project is to design and develop a **full-stack e-commerce web application** that provides users with a secure, efficient, and user-friendly platform for online shopping. The system aims to integrate modern web technologies to deliver a seamless experience for both customers and administrators, ensuring scalability, performance, and data security.

The specific objectives of the project are as follows:

1. **To design a responsive and interactive user interface** using **React.js** and **Tailwind CSS**, ensuring a visually appealing and consistent experience across various devices and screen sizes.
2. **To develop a robust backend server** using **Node.js** and **Express.js** that handles data transactions, manages requests, and ensures reliable communication between the frontend and the database.
3. **To implement a flexible and scalable database structure** using **MongoDB**, capable of storing and retrieving large amounts of data related to users, products, and orders efficiently.
4. **To provide secure user authentication and authorization** using **JWT (JSON Web Token)** and **OAuth**, ensuring that user data is protected and access to system resources is properly managed.
5. **To enable core e-commerce functionalities** such as product listing, product search, shopping cart management, checkout, and order processing.
6. **To integrate payment gateway and order tracking features** for completing transactions and monitoring order status.
7. **To ensure data security, integrity, and privacy** through proper encryption, validation, and secure communication channels.
8. **To conduct comprehensive testing and deployment**, ensuring that the system performs efficiently, meets user requirements, and can scale as the number of users and transactions increases.
9. **To develop an easily maintainable and extendable architecture** that allows future enhancements such as product recommendations, analytics dashboards, or AI-driven personalization.

HARDWARE AND SOFTWARE REQUIRMENT

4.1 Hardware Requirements

a) Workstations
- Processor: Intel Core i5/i7 or AMD Ryzen equivalent
- RAM: 8GB or higher
- Storage: 256GB SSD or higher
- Display: Full HD (1920x1080) resolution

b) Network Infrastructure
- High-speed Wi-Fi or Ethernet connection
- Optional cloud storage support for data backup

c) External Devices (Optional)
- External monitor for multi-display support
- Printer for generating products sell reports

4.2 Software Requirements

a) Operating Systems
- Windows 10/11, macOS, or Linux

b) Programming Languages & Frameworks
- Frontend: HTML, CSS, JavaScript, React.js, Tailwind CSS, Axios
- Backend: Node.js, Express.js, PostMan

c) Database Management System (DBMS)
- To store products records, user authentication, and purchase, orders
- Examples: MongoDB

d) Development Tools
- Visual Studio Code, GitHub, Postman (for API testing)

e) Security & Backup
- SSL encryption for secure data transmission
- Cloud storage for backups (Google Drive, AWS, or Azure)
- User authentication and role-based access control to ensure data security

PROJECT FLOW

The project flow defines the logical sequence of operations that occur within the e-commerce system. It describes how users interact with the application, how data travels between the client (frontend) and the server (backend), and how different modules work together to achieve the desired functionality.

The proposed e-commerce website follows a **modular and layered architecture** built using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**. This architecture ensures smooth communication between system components, enabling scalability, maintainability, and efficient data handling.

The project flow can be divided into several stages: **User Interaction**, **Frontend Processing**, **Backend Communication**, **Database Operations**, and **Response Rendering**. Each stage plays a crucial role in ensuring that the system provides a seamless and secure shopping experience.

The overall flow of the e-commerce platform is as follows:

1. **User Registration and Login**
   - New users register by providing personal details (name, email, password, etc.).
   - The backend validates the data and stores it securely in the **MongoDB** database after encrypting the password.
   - Returning users can log in using their credentials.
   - On successful login, a **JWT token** is generated and sent to the client for authentication in future requests.
   - Users may also log in via **OAuth providers** such as Google or Facebook for convenience.
2. **Product Browsing and Search**
   - Once authenticated, users can browse through various product categories.
   - The frontend fetches product data from the backend API using **Axios** or **Fetch API**.
   - The backend retrieves the requested data from **MongoDB** and sends it back in JSON format.
   - The frontend dynamically renders product cards using **React components** and **Tailwind CSS** for responsiveness.
   - Users can search for specific items using keywords, categories, or filters.
3. **Product Details and Cart Management**
   - Clicking on a product opens a detailed view showing images, price, description, and stock status.
   - Users can add items to their **shopping cart**.

o The cart data is maintained locally in the frontend state (using React Context or Redux) and also synchronized with the backend for logged-in users.

o Users can update quantities or remove products from their cart before checkout.

4. **Order Placement and Checkout**

o During checkout, the system verifies user authentication using JWT.

o The user provides shipping details and selects a payment method.

o The backend calculates the total amount, applies any discount codes, and generates an order summary.

o Payment can be simulated (for demo purposes) or integrated using a payment gateway (e.g., Razorpay, Stripe, or PayPal API).

o Once payment is successful, the order details are stored in MongoDB, and the user receives a confirmation email or message.

5. **Order Management**

o The **Admin Panel** allows administrators to view, update, or cancel orders.

o Admins can add, edit, or remove products, manage inventory, and monitor customer activities.

o The system maintains detailed records of each transaction for analytics and business tracking.

6. **Logout and Session Management**

o Users can log out at any time.

o The JWT token is invalidated, and session data is cleared from local storage to ensure security.

o OAuth sessions are managed via token revocation or expiration.

Module-wise Flow

1. User Module

Handles registration, login, profile updates, and authentication.
Flow:
User → Input Validation → Authentication → Token Generation → Session Management.

2. Product Module

Responsible for displaying product catalogs, managing inventory, and enabling search filters.
Flow:
Frontend Request → API Call → Database Query → Product Display.

3. Cart Module

Manages selected items, quantity updates, and total cost calculation.
Flow:
User Adds to Cart → Local/Backend Storage Update → Total Calculation → Checkout Redirect.

4. Order Module

Handles checkout, payment processing, and order tracking.
Flow:
Checkout Form → Payment Gateway → Order Confirmation → Database Storage →
Notification.

5. Admin Module

Accessible only by administrators for system management.
Flow:
Admin Login → Access Dashboard → Manage Products/Orders → Generate Reports.

PROJECT OUTCOME

The development and implementation of the Employee Management System (EMS) have resulted in significant improvements in HR efficiency, data management, and decision-making. The project outcomes align with the defined objectives and address challenges faced by organizations in managing employee data. Below is a detailed overview of the project outcomes:

5.1 Streamlined Employee Management
- Centralized digital records have reduced reliance on manual paperwork.
- Employee data is now organized systematically, making retrieval, updates, and record-keeping faster and more accurate.

5.2 Improved Data Accuracy
- Secure database storage minimizes errors in employee records, payroll, and performance tracking.
- Automated calculations and validations ensure consistency in salary, attendance, and role assignments.

5.3 Enhanced User Engagement
- User-friendly dashboards and interfaces encourage HR personnel and managers to efficiently manage employee data.
- Features like search, filtering, and report generation allow users to interact with the system effectively.

5.4 Efficient HR Processes
- Administrative tasks such as onboarding, role assignment, and employee updates are automated.
- Reporting modules provide instant summaries for payroll, department statistics, and performance evaluations.

5.5 Scalability and Extensibility
- The system is designed to accommodate an increasing number of employee records.
- Modular design allows additional features, such as leave management, attendance tracking, or performance analytics, to be integrated seamlessly in future versions.

5.6 Improved Decision-Making
- Managers can use dashboards and reports to make informed HR decisions.
- Data-driven insights help identify workforce trends, performance gaps, and resource allocation needs.

5.7 Data-Driven Insights
- The system collects usage data, such as frequently accessed modules and report generation patterns.
- Analytics help optimize workflows, identify bottlenecks, and improve overall system efficiency.

Data Flow Diagram (DFD)
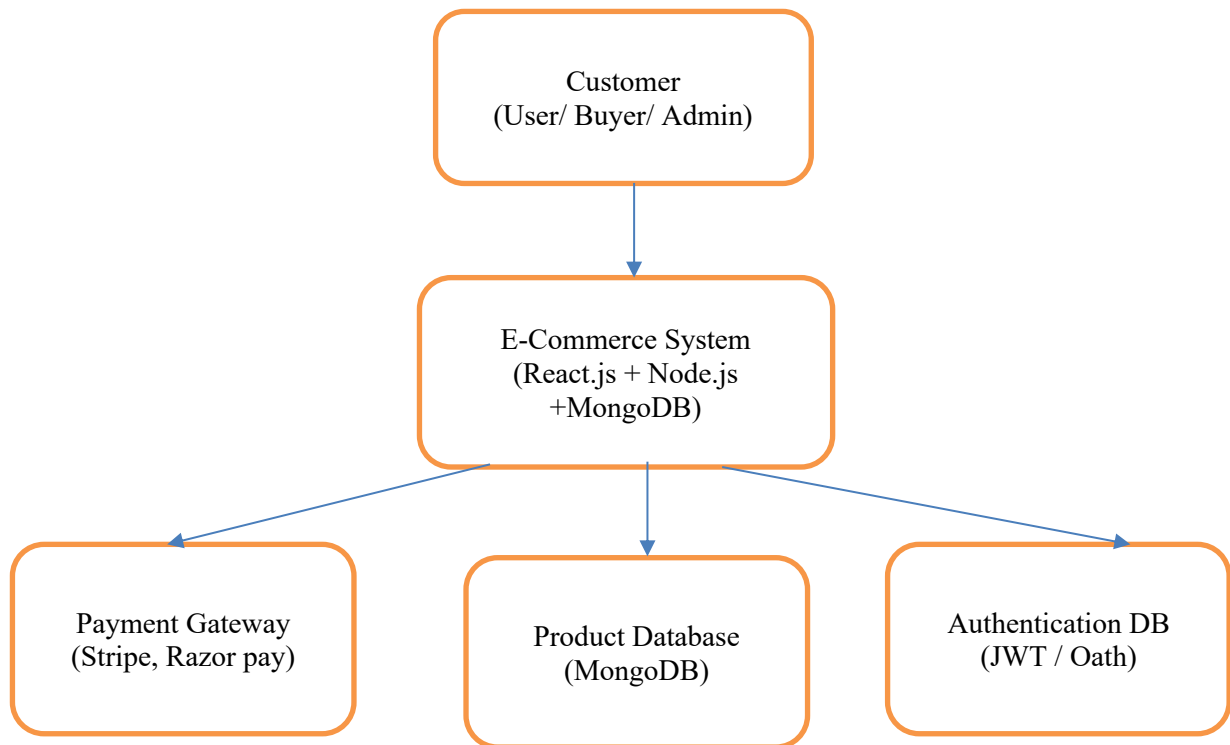
Context Diagram (Overall System View)

Data Flow Explanation:

- **Customer → System:** Sends requests (register, login, search products, add to cart, place orders).
- **System → Customer:** Sends responses (product details, confirmation, order updates).
- **System ↔ Database:** Stores/retrieves user, product, and order data.
- **System ↔ Payment Gateway:** Handles secure payment processing.

Below Diagram Steps

D1 – User Data Stores user credentials, profiles, and JWT tokens.

D2 – Product Data Stores all product information (name, price, stock, etc.).

D3 – Cart Data Temporary data for user shopping cart.

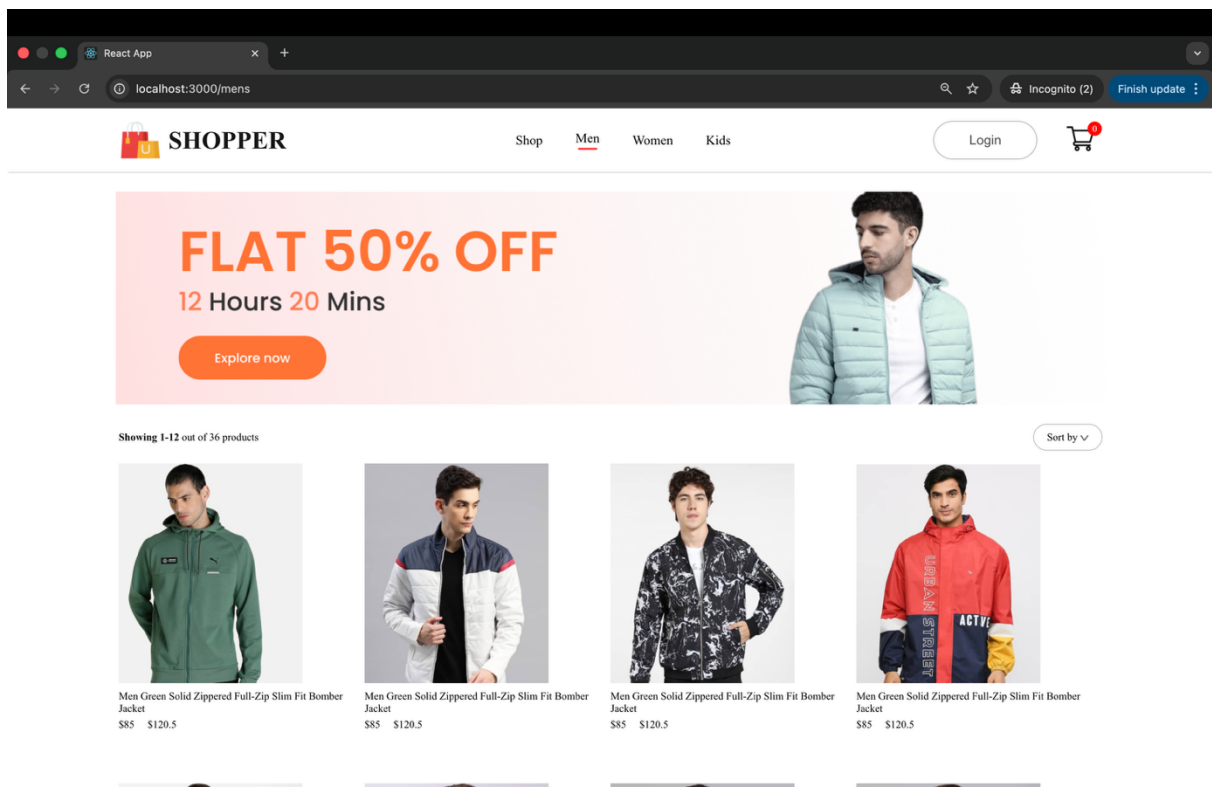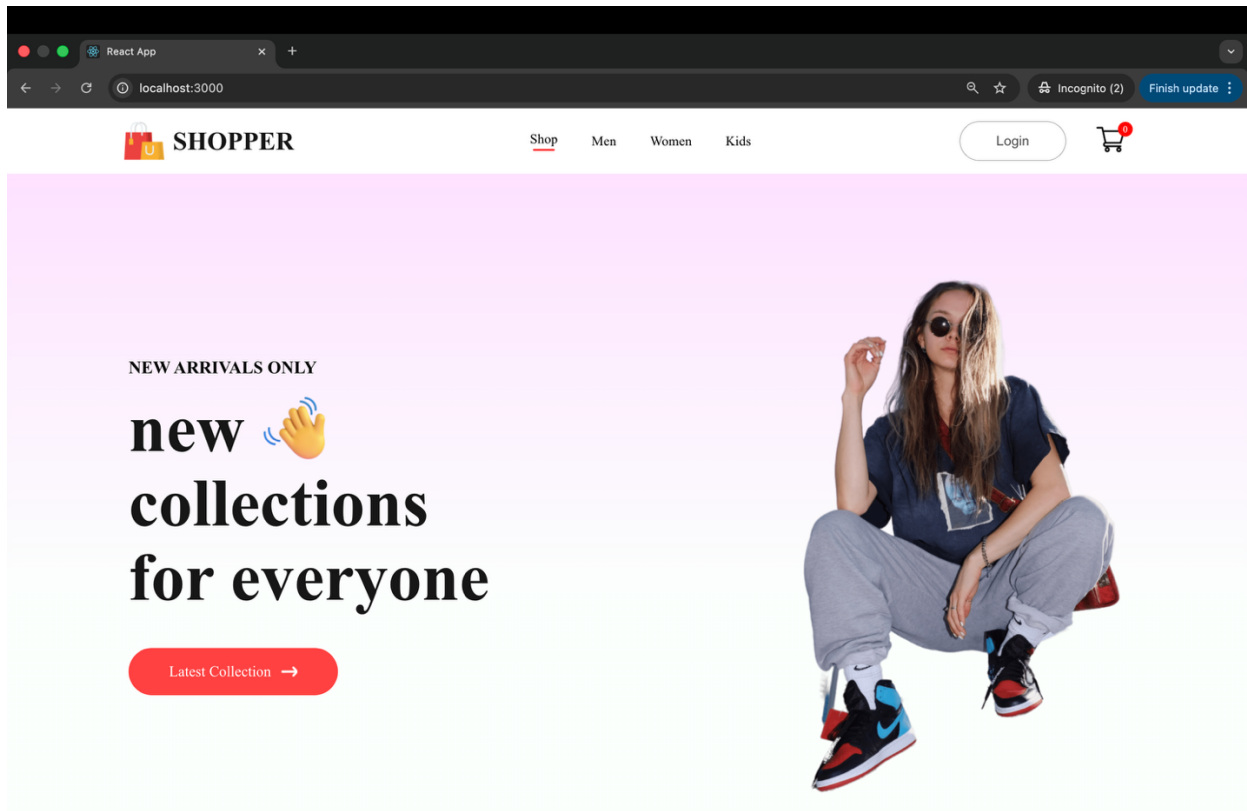**D4 – Order Data** Stores order history, payment status, and delivery details.

D5 – Admin Data Stores admin credentials and privileges.

Main Process Breakdown

CUSTOMER

USER AUTHENTICATION
(Login, Register, JWT/OAuth Verification)

PRODUCT MANAGEMENT
(Browse, Search, View Product Details)

CART MANAGEMENT
(Add to Cart, Update, Remove, Checkout)

ORDER PROCESSING
(Confirm Order, Update Order Status)

PAYMENT MODULE
(Secure Payment Gateway, Confirmation)

ADMIN MANAGEMENT
(Manage Products, Orders, Users)
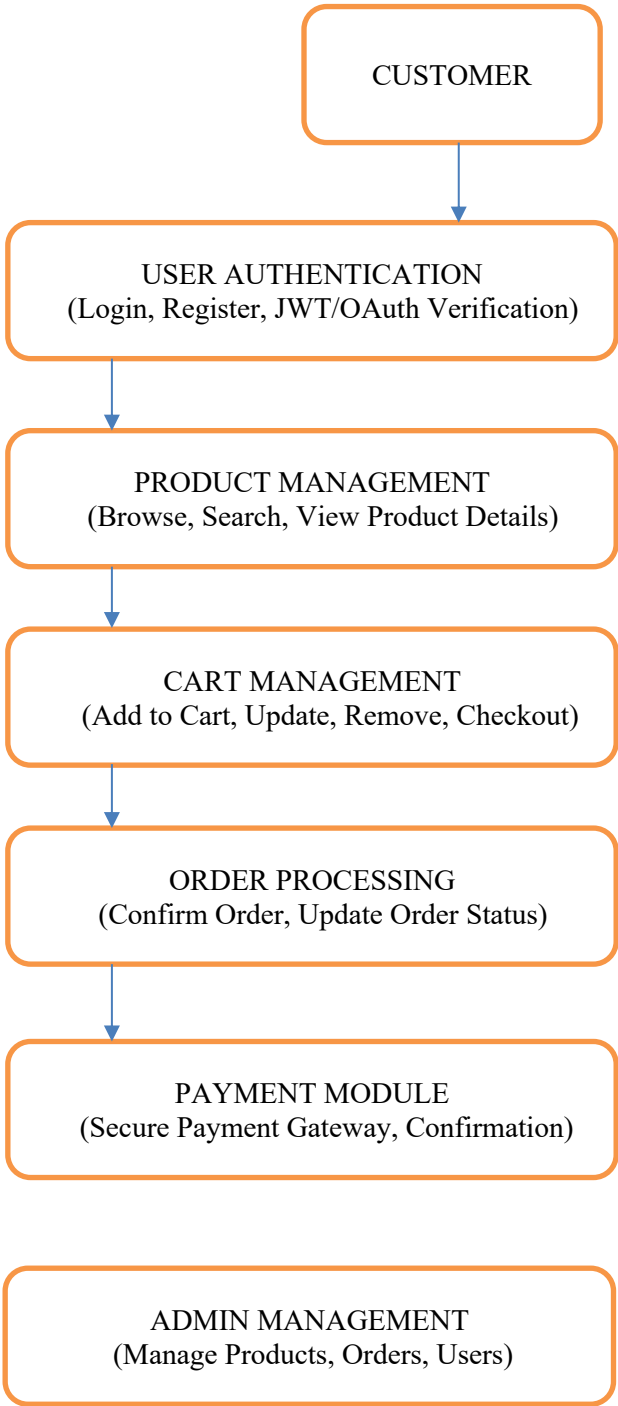
E-Commerce System

Screenshots:

SHOPPER

Shop   Men   Women   Kids

Login

Products	Title	Price	Quantity	Total	Remove
	Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse	$85	1	$85	✕

cart Totals

Subtotal	$85
Shipping Fee	Free
Total	**$85**

Proceed to Checkout

If you have a promo code, Enter it here

| Promo code | Submit |

SHOPPER

Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$50 $80.5

Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$85 $120.5

Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$60 $100.5

Boys Orange Colourblocked Hooded Sweatshirt
$100 $150

Get Exclusive Offers on Your Email

Subscribe to our newsletter and stay updated.

| Your Email id | Subscibe |

SHOPPER

Company   Products   Offices   About   Contact

References:

- Welling, L., & Thomson, L. (2017). **PHP and MySQL Web Development** (5th Edition). Addison-Wesley.

- Flanagan, D. (2020). **JavaScript: The Definitive Guide** (7th Edition). O'Reilly Media.

- Osmani, A. (2018). **Learning React: Functional Web Development with React and Redux**. O'Reilly Media.

- Hunt, A., & Thomas, D. (2000). **The Pragmatic Programmer: Your Journey to Mastery** (2nd Edition). Addison-Wesley.

- Node.js Documentation. (2025). Retrieved from https://nodejs.org/en/docs/

- Express.js Documentation. (2025). Retrieved from https://expressjs.com/