A
Practical Activity Report
Submitted for

UI & UX SPECIALIST-(UCS542)

END-Semester Lab Evaluation

Ecommerce Website

Submitted to-

Dr.Vibha Jain

BE Third Year

Submitted by-

Rachit Bansal (102203155)

Abhiram Epuru (102203173)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

TIET, Patiala

August-December 2024

INDEX

INTRODUCTION

E-commerce has transformed the way businesses and consumers interact, enabling seamless transactions across geographical boundaries. With the rapid advancement in technology, there is a growing need for user-friendly, efficient, and scalable platforms that cater to both buyers and administrators. This project, titled "Full Stack E-Commerce Website Using React.js, MongoDB, Express, and Node.js," aims to bridge the gap between consumer demands and business capabilities by leveraging modern web technologies.

Built using the MERN stack, the project provides a feature-rich platform that simplifies online shopping. The front end, powered by React.js, ensures a dynamic and responsive user interface, delivering an intuitive shopping experience. The back end, developed with Node.js and Express, manages server-side operations, ensuring efficient data handling and seamless communication between the client and server. MongoDB serves as the database, providing a scalable and flexible solution for storing and retrieving product, user, and order data.

The platform incorporates essential e-commerce functionalities such as user authentication, product browsing, shopping cart management, and secure checkout processes. Additionally, it includes an admin panel for inventory and order management, making it a comprehensive solution for businesses. Designed with scalability and performance in mind, the application ensures compatibility across devices and provides a secure environment for handling user data and transactions.

By addressing the challenges faced by small and medium-sized businesses in adopting digital solutions, this project serves as a model for developing efficient and affordable e-commerce platforms. It highlights the integration of modern technologies to create a seamless and robust application, tailored to meet the needs of both users and administrators.

PROBLEM STATEMENT

The increasing digitization of businesses has made e-commerce an essential component for reaching a wider audience. However, many small and medium-sized enterprises (SMEs) face significant challenges in transitioning to digital platforms. The primary hurdles include the high cost of proprietary e-commerce solutions, technical complexities of deployment, and the lack of customization options to meet specific business needs. Existing platforms are often either too rigid, offering limited flexibility for growth, or overly complex, requiring specialized skills to manage and maintain.

Moreover, customers today expect seamless, responsive, and intuitive online shopping experiences. A poorly designed e-commerce platform can lead to user dissatisfaction, abandoned carts, and ultimately, loss of sales. Key areas like real-time product availability, secure payment gateways, and efficient data management are often overlooked in many solutions. This creates a gap between user expectations and the services businesses can deliver.

Another critical challenge lies in the management of back-end operations. Admins need efficient tools to monitor inventory, track orders, and process transactions. Without a robust and scalable system, businesses struggle to adapt to increasing demands, especially during peak times such as seasonal sales or promotions.

This project aims to address these challenges by developing a scalable, user-friendly, and customizable ecommerce platform. Using the MERN (MongoDB, Express, React.js, and Node.js) stack, the solution integrates modern technologies to provide a seamless experience for end-users while simplifying administrative tasks. By focusing on affordability, ease of use, and scalability, this project bridges the gap between traditional businesses and the demands of a digital marketplace.

SPECIFIC REQUIREMENTS

a) Functional Requirements

i) **User Authentication and Authorization:**

Allows users to register and log in using secure credentials.Providing role-based access, such as admin access for managing products and orders**.**

ii) **Product Management:**

Admins are able to add, update, and delete products.Support for product categories, descriptions, images, and stock availability.

iii) **Search and Filter Options:**

Enabling users to search for products by name or category.Providing filtering options such as price range, ratings, and product availability.

iv) **Shopping Cart:**

Allowing users to add products to a cart and adjust quantities.Cart data is persist for logged-in users even after page refresh.

v) **Order Placement and Tracking:**

Users are able to place orders securely and view order history.Admins have the ability to update order statuses (e.g., "Processing," "Shipped," "Delivered").

vi) **Payment Integration:**

Integrating a secure payment gateway to handle transactions.Including validation to ensure accurate billing and transaction success messages.

vii) **Responsive Design:**

The application is fully responsive, providing a seamless experience across devices (mobile, tablet, and desktop).

viii) **Admin Dashboard:**

A dedicated interface for administrators to manage inventory, monitor user activity, and view order statistics.

b) Non-Functional Requirements

i) Scalability:

The system support increasing numbers of users and products without performance degradation.

ii) Performance:

Page load times are minimal, ensuring a smooth user experience even during high traffic.

iii) Security:

Implementing best practices for user data protection, including encrypted storage for passwords and secure payment processing.

iv) Reliability and Availability:

Ensuring the platform is operational 99.9% of the time, with robust error-handling mechanisms to manage unexpected issues.

v) Maintainability:

Using clean and modular code to facilitate easy updates and feature additions in the future.

vi) Cross-Browser Compatibility:

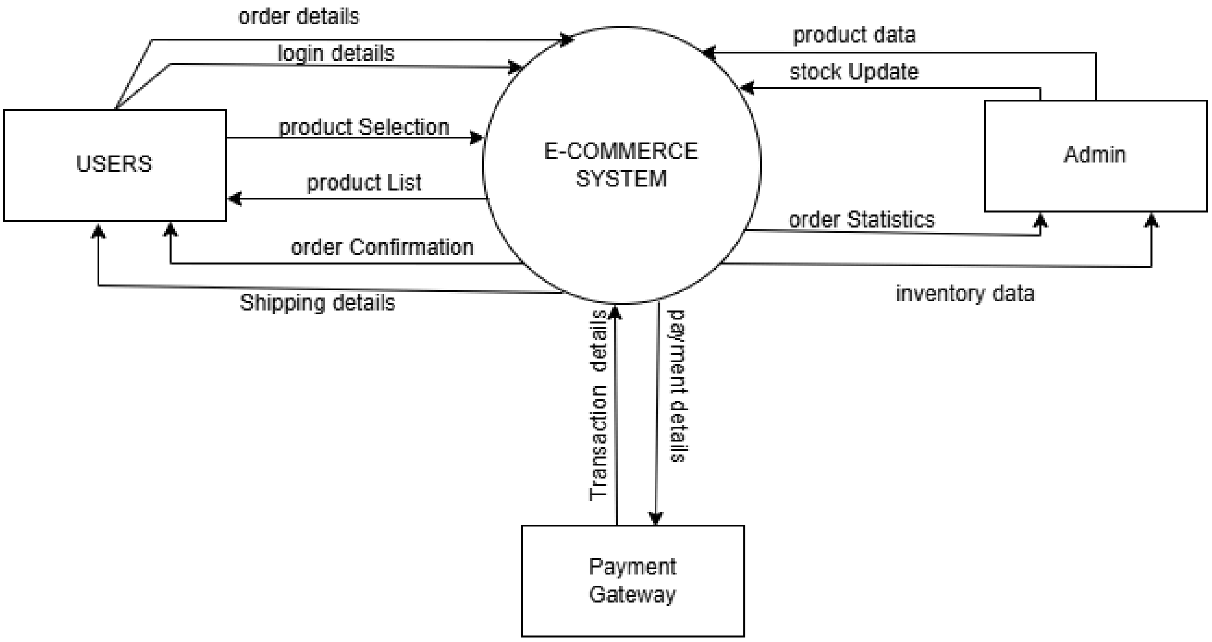
Ensuring the application functions consistently across major browsers like Chrome, Firefox, Safari, and Edge.

vii) Ease of Use:

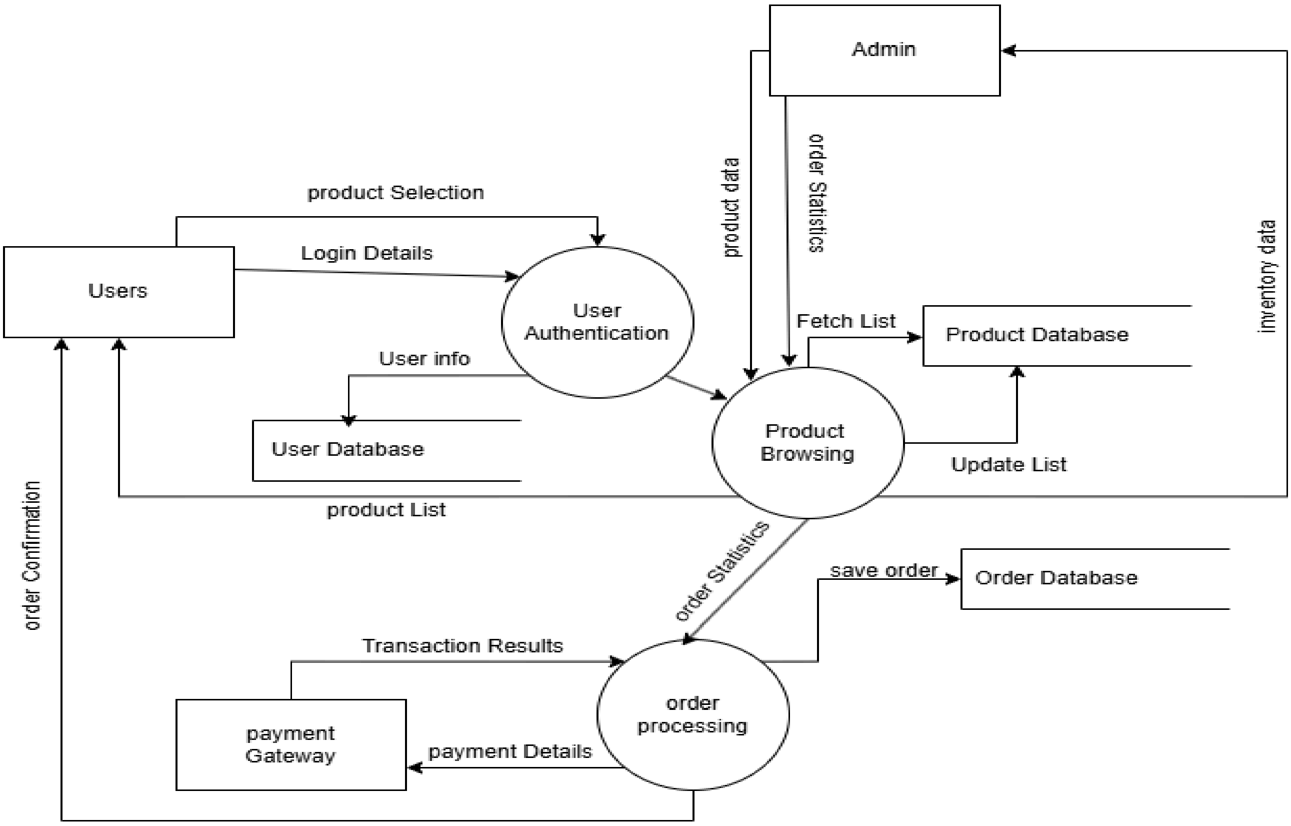
Designing the interface to be intuitive for both end-users and administrators, minimizing the learning curve.

CONTEXT LEVEL AND DATA FLOW DIAGRAM

Context Level Diagram -(Level 0)



Data Flow Diagram -(Level 1)



SYSTEM SPECIFICATION

1. Hardware Specifications

Development Environment:

- **Processor**: Intel Core i5/i7 or AMD Ryzen 5/7 (or equivalent).
- **RAM:** 8 GB (minimum); 16 GB (recommended for smoother development).
- **Storage:** 256 GB SSD (minimum); 512 GB SSD or higher (recommended).
- **Graphics:** Integrated GPU (sufficient for web development).
- **Display**: Full HD (1920x1080) resolution monitor.

2. Software Specifications:

Development Tools:

- **Operating System**: Windows 10/11.
- **IDE/Editor**: Visual Studio Code (preferred) or any text editor.
- **Version Control**: Git with GitHub/GitLab for code management.
- **Package Manager**: npm (Node Package Manager).

Frameworks and Libraries:

- **Frontend**: React.js with Material-UI or Bootstrap for styling.
- **Backend**: Node.js with Express.js for API development.
- **Database**: MongoDB (NoSQL database for efficient data handling).

Browsers:

- Google Chrome, Firefox, or Edge (latest versions for development and testing).

Additional Tools:

- **Postman**: For API testing.
- **MongoDB Compass**: GUI for managing MongoDB.

TOOLS USED

The successful implementation of the Full Stack E-Commerce Website was made possible by leveraging a variety of tools for development, testing, and deployment. These tools were chosen to ensure efficiency, scalability, and maintainability. Below is a detailed explanation of the tools used and their roles in the project:

1. MERN Stack

- **MongoDB**: A NoSQL database used to store and retrieve product details, user information, and order data efficiently. Its schema-less nature made it adaptable to changing data structures.
- **Express.js**: A lightweight and flexible framework for building server-side applications and RESTful APIs. It helped in managing user requests and performing CRUD operations.
- **React.js**: A powerful front-end library used to create a dynamic and responsive user interface. It ensured seamless navigation and improved user experience with reusable components.
- **Node.js**: A runtime environment that allowed us to execute JavaScript on the server side. It facilitated efficient handling of back-end operations and server-side scripting.

2. Postman

- Used for testing APIs during development. It allowed us to validate the functionality of routes and endpoints, ensuring smooth communication between the client and server.

3. MongoDB Compass

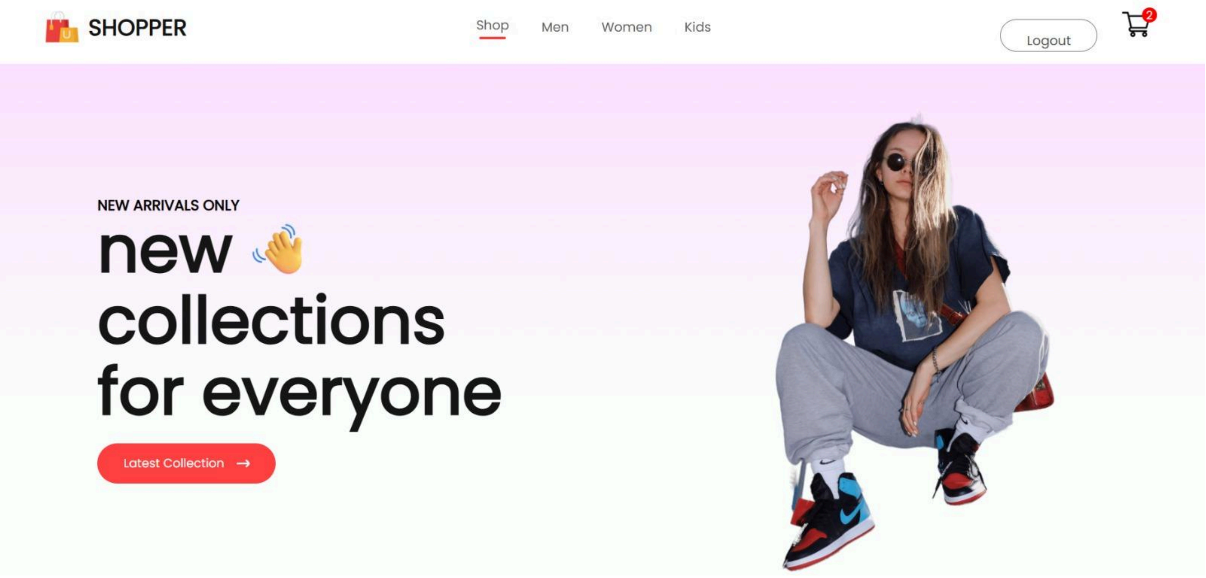
- A GUI tool for visualizing and managing the MongoDB database. It simplified querying and testing database operations during development.

4. Git and GitHub

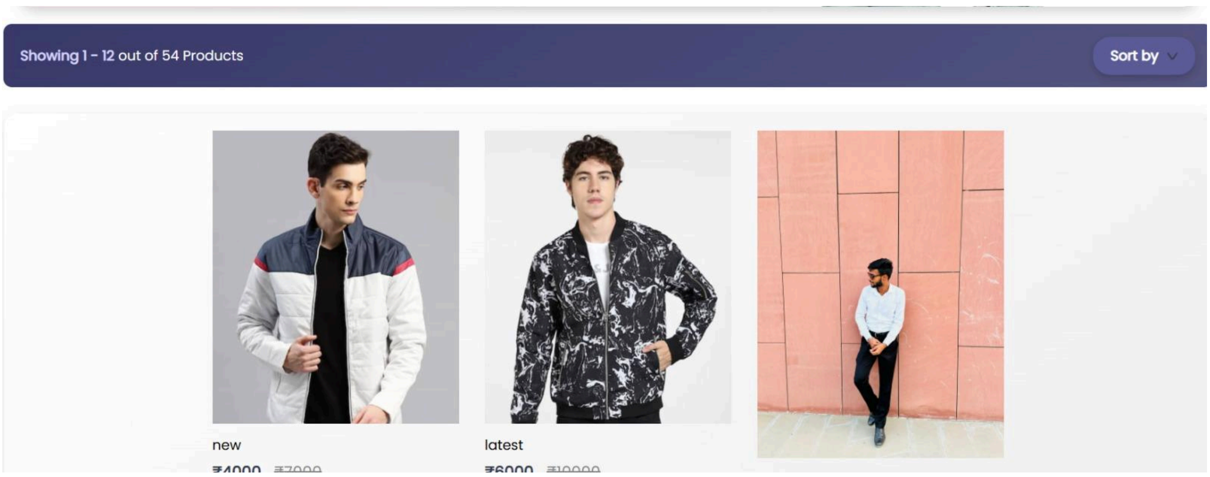
- Version control systems were employed to track changes, manage the codebase, and enable collaboration. GitHub repositories were used to store and share the project securely.

SAMPLE SCREENSHOTS

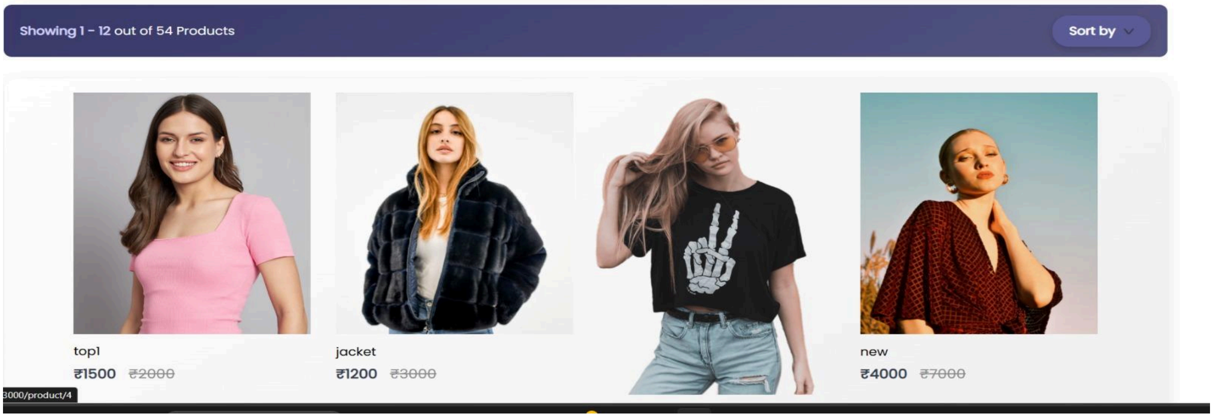
1) Main page



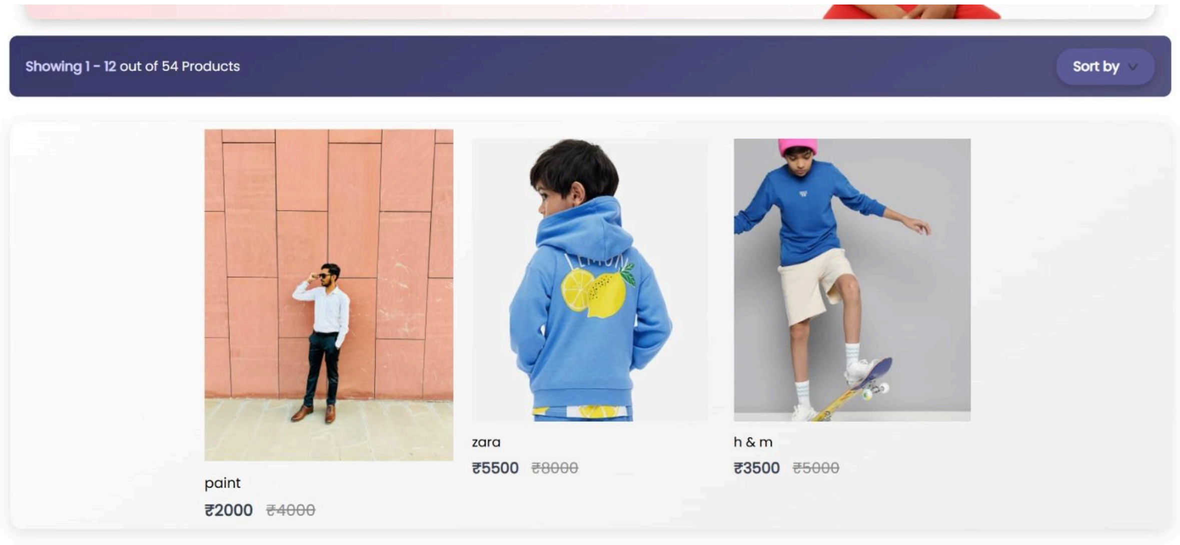
2) Men page -



3) Women page -



4) **Kids page -**



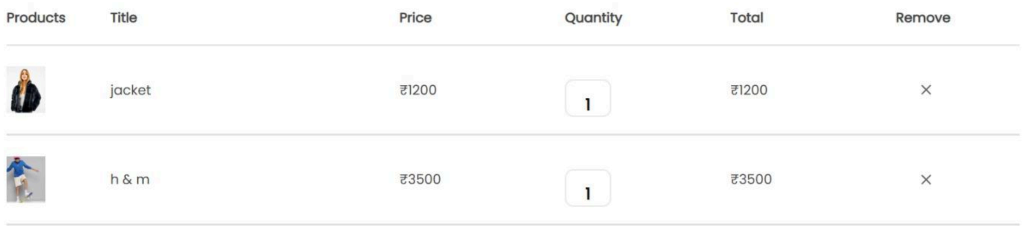
Showing 1 - 12 out of 54 Products

paint
₹2000 ~~₹4000~~

zara
₹5500 ~~₹8000~~

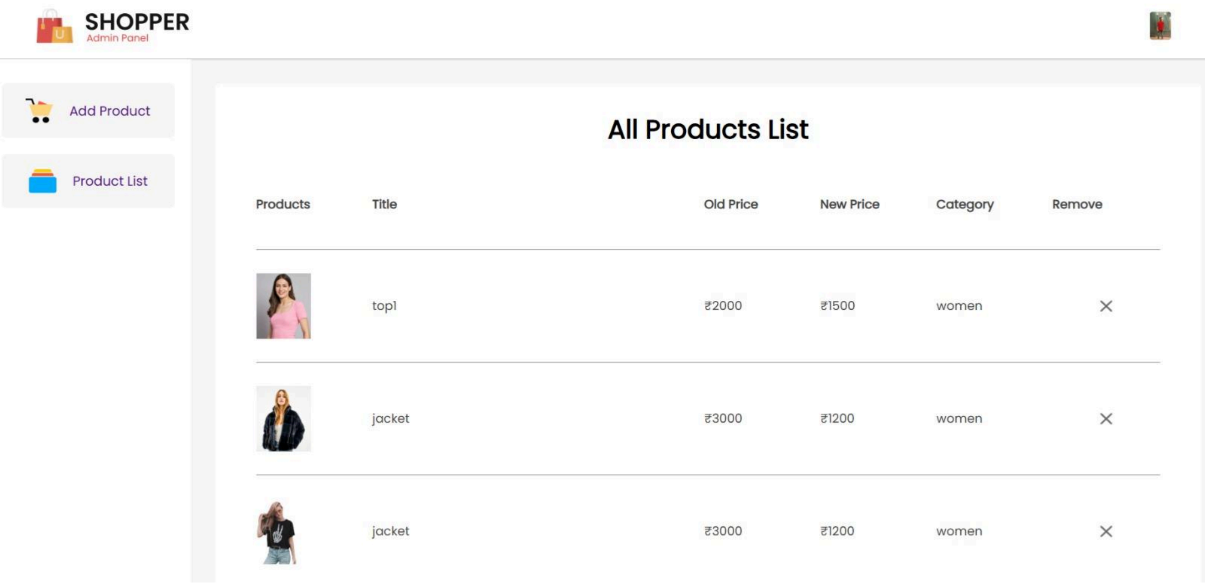
h & m
₹3500 ~~₹5000~~

5) **Cart image -**



Products	Title	Price	Quantity	Total	Remove
	jacket	₹1200	1	₹1200	×
	h & m	₹3500	1	₹3500	×

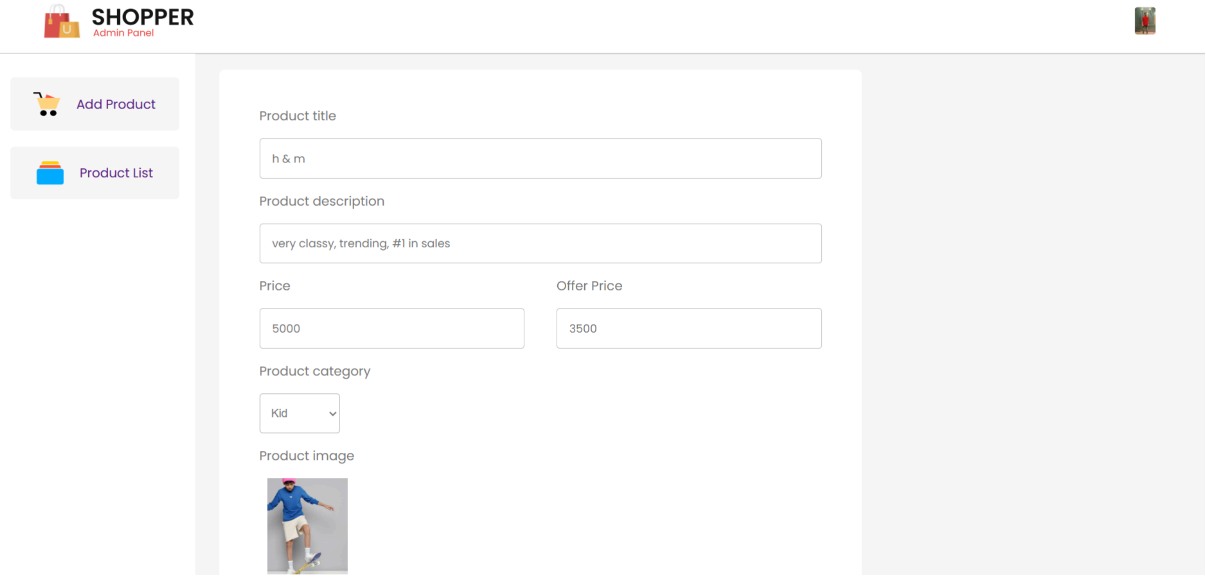
6) **Product List image -**



All Products List

Products	Title	Old Price	New Price	Category	Remove
	top1	₹2000	₹1500	women	×
	jacket	₹3000	₹1200	women	×
	jacket	₹3000	₹1200	women	×

7) **Admin page image -**



SHOPPER
Admin Panel

Add Product

Product List

Product title

h & m

Product description

very classy, trending, #1 in sales

Price

5000

Offer Price

3500

Product category

Kid

Product image

OUTPUT REPORTS

This section outlines the key reports generated during the development and testing phases of the Full Stack E-Commerce Website. Although the actual documents are not included, the following details summarize the process and key insights.

a. API Testing Report

API testing was conducted using Postman to validate the functionality and reliability of various endpoints. The following key endpoints were tested:

- **User Login Endpoint**: Ensured successful authentication and token generation for valid credentials.
- **Product Search Endpoint**: Verified that search queries return accurate product details based on filters and categories.
- **Cart Update Endpoint**: Confirmed the addition and removal of items from the user cart, along with proper error handling for invalid requests.

b. Debugging Logs

During development, debugging was performed extensively using the following methods:

- **Console Logs**: Key application states, such as database connection status and API request payloads, were logged.
- **Error Handling Middleware**: Error messages and stack traces were captured to identify and resolve issues effectively.

CONCLUSION

The development of the Full Stack E-Commerce Website using the MERN stack has been a comprehensive journey encompassing multiple aspects of modern web application development. This project was undertaken with the aim of creating a user-friendly, efficient, and scalable platform to streamline the online shopping experience for both customers and administrators. By integrating technologies like MongoDB, Express.js, React.js, and Node.js, the system ensures seamless interactions between the front-end, back-end, and database layers.

One of the major accomplishments of this project is the dynamic and responsive interface built with React.js, enabling customers to browse and interact with the platform effortlessly. The use of MongoDB as a NoSQL database has ensured efficient storage and retrieval of data, making the system suitable for handling real-time operations like user authentication, product browsing, and order processing. Additionally, the back-end, powered by Node.js and Express.js, has been optimized for smooth API communication, ensuring that the data flow between the client and the server remains robust and secure.

The project also emphasized scalability and maintainability, making it adaptable for future enhancements such as integrating advanced features like AI-driven product recommendations or multiple payment gateways. Security has been given priority through practices such as token-based authentication (JWT), ensuring data integrity and user trust.

In conclusion, this e-commerce platform successfully demonstrates how modern web technologies can be leveraged to create a reliable, feature-rich solution. While this project has met its initial objectives, it lays a strong foundation for further development, addressing the evolving needs of both businesses and customers in the digital age. This achievement reflects not only technical proficiency but also the ability to design and implement solutions that align with real-world requirements.