A Customizable E-commerce Platform Tailored for Seamless User Experiences Using the MERN Stack

BY

Md. Abdur Rahim
ID: 221-15-4674

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Zahid Hassan
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised By

Md. Sazzadur Ahamed
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2024

APPROVAL

This Project titled **"A Customizable E-commerce Platform Tailored for seamless user experiences using the MERN stack"**, submitted by Md. Abdur Rahim, ID No: **221-15-4674** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **13 January, 2025.**

BOARD OF EXAMINERS

Dr. Md. Zahid Hasan — **Chairman**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mohammad Monirul Islam — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

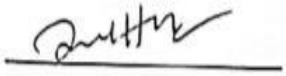
Mr. Afjal Hossan Sarower — **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Ahmed Wasif Reza — **External Examiner**
Professor
Department of Computer Science and Engineering
East West University

DECLARATION

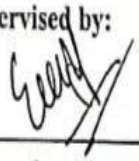
I hereby declare that, this project has been done by me under the supervision of **Md. Zahid Hasan, Associate Professor, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Md. Zahid Hasan
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised by:



Md. Sazzadur Ahamed
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Abdur Rahim
ID: 221-15-4674
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First I express My heartiest thanks and gratefulness to almighty God for His divine blessing makes possible to complete the final year project successfully.

I am really grateful and wish my profound indebtedness to **Md. Zahid Hasa**n, Associate Professor, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Web Application, Database, Mobile Application (Traditional), Machine Learning (ML), Artificial Intelligence (AI), Image Processing, Deep learning, Artificial Neural Network, Cryptography and Information Security" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

This project showcases a modern E-commerce platform built using the MERN stack (MongoDB, Express.js, React.js, Node.js), designed to deliver a seamless online shopping experience for users and comprehensive management tools for administrators. Key features include a fully functional shopping cart enabling users to add, remove, and manage products, along with the ability to leave reviews and ratings, enhancing user engagement and trust. A top products carousel highlights featured or highly-rated items, while product pagination and an efficient search feature ensure smooth navigation and accessibility. Registered users benefit from personalized profiles to view and track their order history. The platform includes a robust admin dashboard offering complete control over admin accounts, products, users, and orders. Administrative tools include product management (add, edit, delete), user management, order details tracking, and the ability to mark orders as delivered. A detailed order management system is accessible for both users and administrators, fostering transparency. The streamlined checkout process incorporates shipping and payment options, ensuring a hassle-free purchasing experience. Additionally, a database seeder simplifies testing by populating the platform with sample users and products. Combining efficiency, scalability, and user-centric design, this platform exemplifies a reliable solution for modern E-commerce, leveraging the MERN stack's capabilities to meet the dynamic demands of online retail.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The rapid growth of e-commerce has transformed customer shopping habits, creating web-based services which offer an interactive, intuitive, and feature-laden experience. Yet several current e-commerce websites often lack in accessibility and operational efficiency. This project aims to bridge these gaps by creating an entire, scalable, and user-friendly e-commerce platform using the MERN stack (MongoDB, Express.js, React.js, Node.js). The platform seeks to improve the consumer shopping experience while offering general capabilities for administrators to efficiently manage items, users, and orders. The MERN stack guarantees superior speed, scalability, and seamless integration, which are crucial for a dynamic e-commerce landscape.

1.2 Motivation

This project is motivated by the increasing need for minimized and scalable e-commerce systems that serve consumers and admins. Although various e-commerce systems exist, many do not provide an overall visitor experience that integrates user-friendly navigation for consumers with robust administration capabilities for administrators. The project intends to address prevalent issues, like sluggish performance, complex navigation, and insufficient administrative control, by using the MERN stack. This platform aims to provide a dynamic solution that facilitates smooth shopping experiences, effective product administration, and real-time changes, while also being scalable to accommodate rising user traffic and data.

1.3 Objective

The primary aim of this project is to create a fully operational and scalable e-commerce platform that delivers an easy shopping experience for users and equips administrators with robust tools for management and optimization. The specified aims consist of:

1. **Develop a Robust E-Commerce Platform:** To develop and deliver a scalable, flexible e-commerce platform using the MERN stack.
2. **Enhance User Experience:** To set up a simple and responsive experience for users, allowing smooth navigation and interaction.
3. **Streamline Administrative Operations:** To provide administrators with complete tools for the effective management of items, users, and orders.
4. **Ensure Future Scalability and Integration:** To design a system that integrates sophisticated technologies such as AI customization and blockchain payments.
5. **Implement Secure and Efficient Operations:** To ensure secure data handling, user authentication, and reliable performance under heavy traffic.

1.4 Expected Outcome

The objective of this project is to develop a fully functional e-commerce platform, including a simpler UI and comprehensive administrative functionalities. The platform will enhance the customer experience with features such as a shopping cart, product search, and user reviews while allowing administrators to effectively manage goods, orders, and users through an intuitive dashboard. The system will have a quick checkout process, secure payment options, and a sample data creation tool for streamlined testing and presentation. The system will eventually be scalable, efficient, and poised for future development, addressing the needs of both users and administrators in the e-commerce sector.

Chapter 2

LITERATURE REVIEW

2.1 Comparison with Existing E-Commerce Platforms

MERN stack-based e-commerce solutions provide significant advantages over traditional platforms such as Shopify and Magento. Shopify provides ease of use. Yet, it limits customisation and scalability without incurring additional costs. Magento, despite its flexibility, requires significant technical expertise and resources for effective growth. The MERN stack provides comprehensive flexibility in modification and scalability, enabling developers to address complex requirements. Studies demonstrate that MongoDB's NoSQL database design is particularly proficient at handling large datasets, such as product inventories and customer reviews, often seen in e-commerce environments [1], [2].

2.2 Key Technologies in the MERN Stack

The MERN stack combines MongoDB, Express.js, React.js, and Node.js into a unified development framework. MongoDB's schema-less design facilitates flexible data storage, making it suitable for e-commerce systems where product qualities may differ considerably. The non-blocking, I/O operations of Node.js provide increased concurrency and responsiveness, essential for systems experiencing significant user traffic. React.js enables the creation of dynamic, interactive user interfaces, markedly increasing user experience via decreased load times and enhanced interaction [1], [3].

A comparative study underscores the efficacy of using a single programming language (JavaScript) across the stack. This minimizes context switching for developers, expedites development cycles, and decreases maintenance costs relative to legacy stacks such as LAMP (Linux, Apache, MySQL, PHP) [2], [3].

2.3 User Experience, Scalability, and E-Commerce Trends

User experience (UX) is fundamental to the success of e-commerce systems. Studies indicate that dynamic frontends, such as those developed using React.js, enhance user engagement via responsive designs and expedited page loading times. Contemporary customers want intuitive, mobile-optimized interfaces, and MERN's modular design enables fast adaptation to these trends [1, 4].

The scalability of MERN-based solutions is another significant advantage. MongoDB facilitates horizontal scalability by distributing data across numerous servers, which is crucial for accommodating rising user demands and extensive product catalogs. The event-driven design of Node.js guarantees rapid handling of concurrent queries.

MERN platforms can effectively integrate current e-commerce advancements like AI-driven customisation and blockchain-based payment methods. This presents MERN as a future-proof option for enterprises seeking to implement cutting-edge technologies while preserving a strong foundational infrastructure. [1], [4].

2.4 Comparative analysis

In the creation of e-commerce platforms, several frameworks, technologies, and database architectures have been extensively implemented to enhance performance, scalability, and user experience. A comparative examination of this project highlights its capabilities and situates it within the framework of current e-commerce solutions.

2.4.1. E-commerce Platform Database Design

Many studies highlight the significance of well-organized databases in e-commerce platforms. An effective Entity-Relationship (ER) diagram underpins systems like as Amazon and Shopify, where user, product, and order entities are interlinked for optimal efficiency. This project employs a comprehensive ER model that includes multi-valued features like product photos and optimized connections across administrators, users, goods, orders, and payments. This approach guarantees data integrity, enhances query efficiency,

and accommodates real-world complications such as numerous product photos or user orders [6].

2.4.2. Scalability and MERN Stack Adoption

Current e-commerce platforms emphasize scalability and quicker loading speed. A comparative analysis of systems developed using the MERN stack vs conventional server-based technologies (e.g., PHP, MySQL) underscores the enhanced flexibility and scalability offered by MongoDB, Node.js, and React [7]. The schema-less architecture of MongoDB facilitates swift alterations, making it suitable for managing unstructured data, such customer evaluations and product information, without necessitating comprehensive redesign.

2.4.3. Enhancing User Experience

User experience (UX) is crucial to the success of an e-commerce business. Studies demonstrate that functionalities like integrated shopping carts, order tracking, and product search significantly enhance customer retention and satisfaction [9]. This project encompasses critical capabilities such as a comprehensive shopping cart, product search, and premier product carousels, according to optimal UX design principles. Furthermore, an administrative interface streamlines backend administration, enabling efficient changes for items, orders, and users [10].

2.4.4. Multi-valued Data Handling

Multi-valued features, such as product photographs, are a significant factor in e-commerce systems. Research highlights the use of complex database models to meet these needs without redundancy [6]. This project enhances digital media content management in current e-commerce systems by integrating multi-valued product picture storage into the database.

2.4.5. Comparative Platform Benchmarking

Established systems like as Shopify, WooCommerce, and Magento are esteemed for their

adaptability and user-focused designs. The elements of this project, including administrative product administration, user purchase history, and product ratings, closely conform to industry norms [7]. The modularity provided by the MERN stack facilitates modification and improvements over time.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Hardware Requirements

The successful development and deployment of this e-commerce platform using the MERN stack require the following hardware specifications:

3.2 Development Environment:

Processor: Quad-Core CPU (e.g., Intel i5 or AMD Ryzen 5) or higher.

RAM: Minimum 8 GB (16 GB recommended for smoother multitasking and testing environments).

Storage: SSD with at least 256 GB free space for software tools, databases, and project files.

Display: Monitor with a resolution of 1920x1080 or higher for UI/UX design and development.

3.3 Deployment Server:

Processor: Multi-core server-grade CPU (e.g., Intel Xeon or AMD EPYC).

RAM: Minimum 16 GB (32 GB or higher for handling increased traffic).

Storage: SSD with a minimum of 500 GB for hosting the application and database.

Network: High-speed internet connection with at least 100 Mbps upload/download speed.

Operating System: Linux (Ubuntu 20.04 LTS or equivalent), Windows Server.

3.4 Software Requirements

The following software tools are essential for the development, testing, and deployment of the platform:

1. **Code Editor:** Visual Studio Code (preferred) or JetBrains WebStorm.
2. **Version Control:** Git and GitHub for source code management.

3. **Backend Framework:** Node.js (v16 or higher) with Express.js.
4. **Frontend Framework:** React.js (v18 or higher).
5. **Database:** MongoDB
6. **Package Manager:** npm (Node Package Manager) or Yarn.
7. **Testing Tools:** Postman for API testing, Jest for unit testing.

3.5 Functional Requirements

The functional specifications of the core features and capabilities of the platform:

3.5.1. User Features:

- Full featured shopping cart for adding, removing, and managing products.
- Product reviews and ratings.
- Top products carousel to highlight featured items.
- Efficient product pagination.
- Search feature to locate products by keywords.
- User profiles for managing order history and preferences.

3.5.2. Admin Features:

- Dashboard for managing products, users, and orders.
- CRUD operations (Create, Read, Update, Delete) for admins, users, and products.
- Detailed order management and delivery status updates.

3.5.3. Core Functionalities:

- Seamless checkout process integrating shipping and payment options.
- Database seeding for initializing sample data.

3.6 Nonfunctional Requirements

The nonfunctional requirements ensure the system meets performance, security, and usability standards:

3.6.1. Performance:

- Response time should be under 2 seconds for all user operations.
- Capable of handling up to 1000 concurrent users during peak hours

3.6.2. Scalability:

- Horizontally scalable to accommodate increased traffic and data volume.
- Support for adding new features and modules with minimal impact on existing code.

3.6.3. Security:

- Implementation of encryption (SSL/TLS) for secure data transmission.
- Authentication and authorization using JSON Web Tokens (JWT).
- Compliance with industry standards like PCI DSS for secure payment processing.

3.6.4. Usability:

- Intuitive user interface with responsive design for the desktop.
- Accessibility compliance (WCAG 2.1 standards) for inclusive design.

3.6.5. Reliability:

- Uptime of 99.9% with failover mechanisms for critical operations.

3.7 Technology Stack Overview

This project utilizes the MERN stack for its strong, scalable, and developer-friendly environment.

1. MongoDB:

- NoSQL database for flexible and scalable data storage.
- Efficient indexing and querying for high performance data operations.

2. Express.js:

- Lightweight and modular backend framework.
- Simplifies REST API creation and middleware integration.

3. React.js:

- Component based frontend framework for dynamic user interfaces.
- Improves development speed through reusable components and virtual DOM.

4. Node.js:

- Event driven, nonblocking runtime environment for building scalable applications.
- Ideal for handling real-time operations like order tracking.

3.8 Software Development Life Cycle (SDLC)

This project utilizes the Agile Software Development Life Cycle technique due of its iterative and adaptable nature in software development. The main stages include the following:

1. **Plan:** Understanding user and system requirements, including hardware, software, and functional needs.
2. **Design:** Creating system architecture, database schema, and diagrams to structure the platform.
3. **Develop:** Implementing the frontend using React.js, backend using Node.js and Express.js, and database using MongoDB.
4. **Test:** Ensuring functionality through unit tests, integration tests, and user acceptance testing (UAT).
5. **Deploy:** Deploying the platform to a production environment using services like AWS or Digital Ocean.
6. **Review:** Regular updates and scalability enhancements to meet evolving needs.

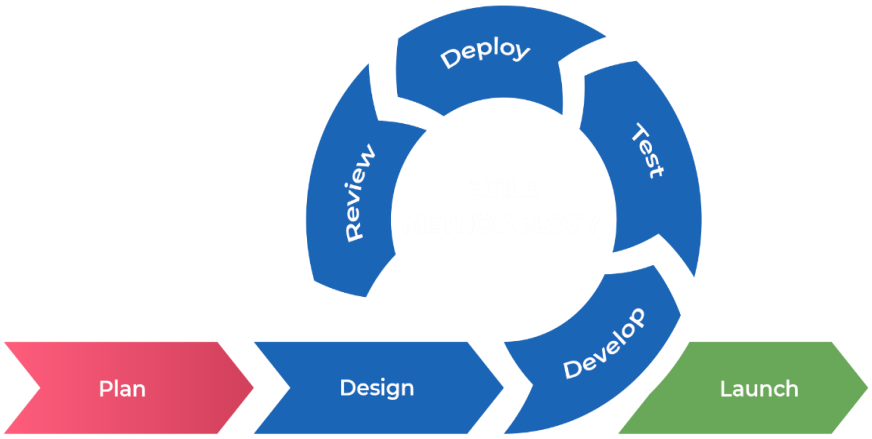


Figure 1 Agile Method

3.9 Design Requirement

- Modular architecture to ensure scalability.
- Responsive UI design for cross-platform usability.
- Secure authentication and data encryption.
- Fast and reliable data access with indexed queries in MongoDB.

3.10 Basic Use-Case Diagram

This graphic illustrates a simplified summary of interactions among users, administrators, and the system concerning fundamental operations like as browsing, product management, and order processing.

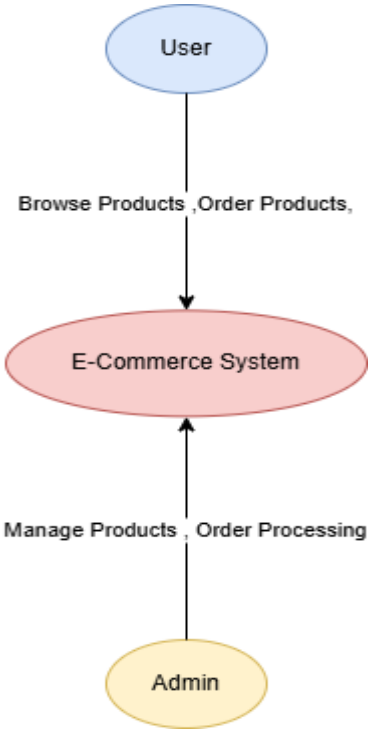


Figure 2 Basic Use case Diagram

3.11 Overall Use-Case Diagram

This diagram captures all major interactions in the system, including browsing, managing orders, user profiles, and admin actions.

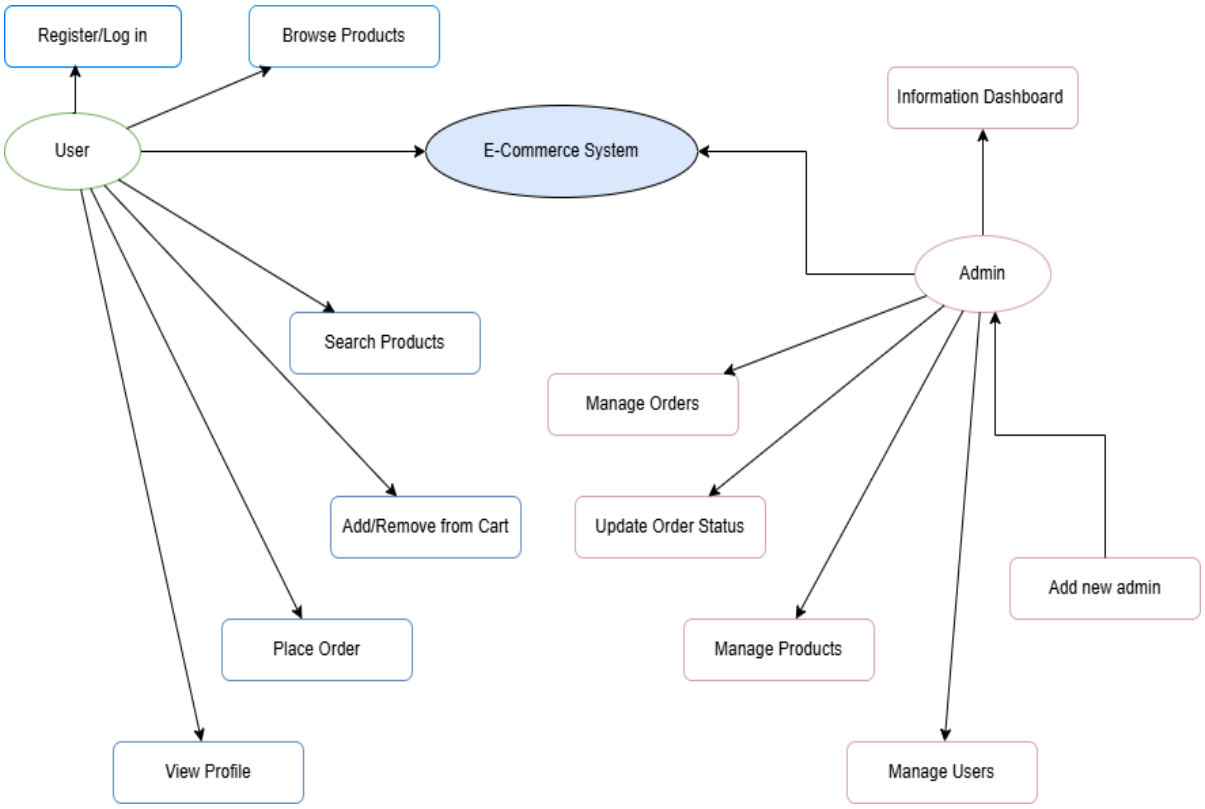


Figure 3 Overall Use case Diagram

3.12 ER Diagram

The ER Diagram illustrates the relationships between users, products, orders, and admins.

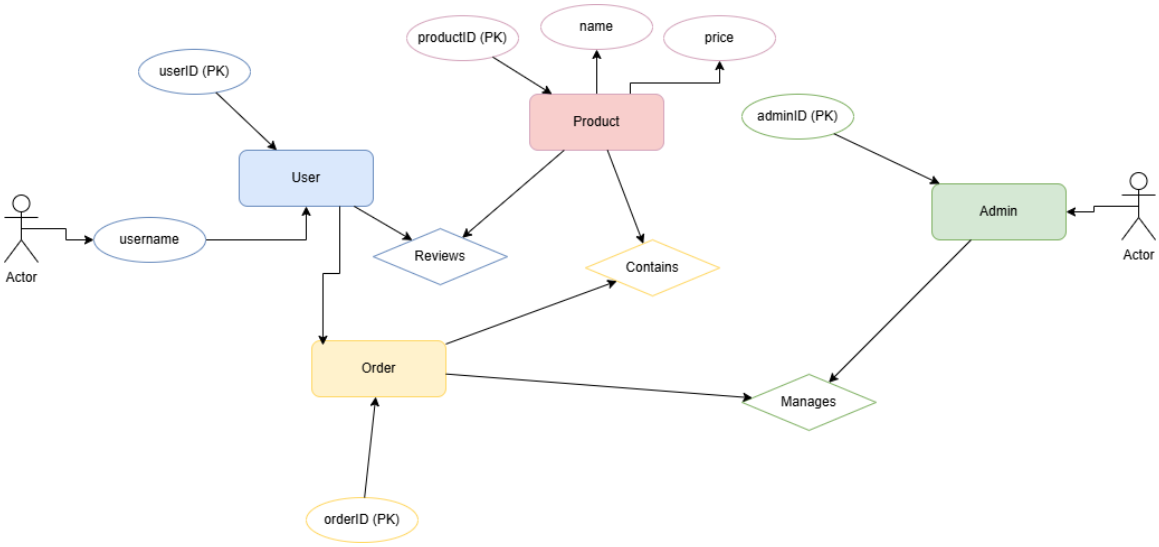


Figure 4 ER Diagram

3.13 ER Diagram Description

User: Stores user details such as name, email, and password.

Product: Maintains product attributes like name, price, and description.

Order: Tracks details about user purchases, such as order ID, status, and delivery date.

Admin: Manages products, users, and orders.

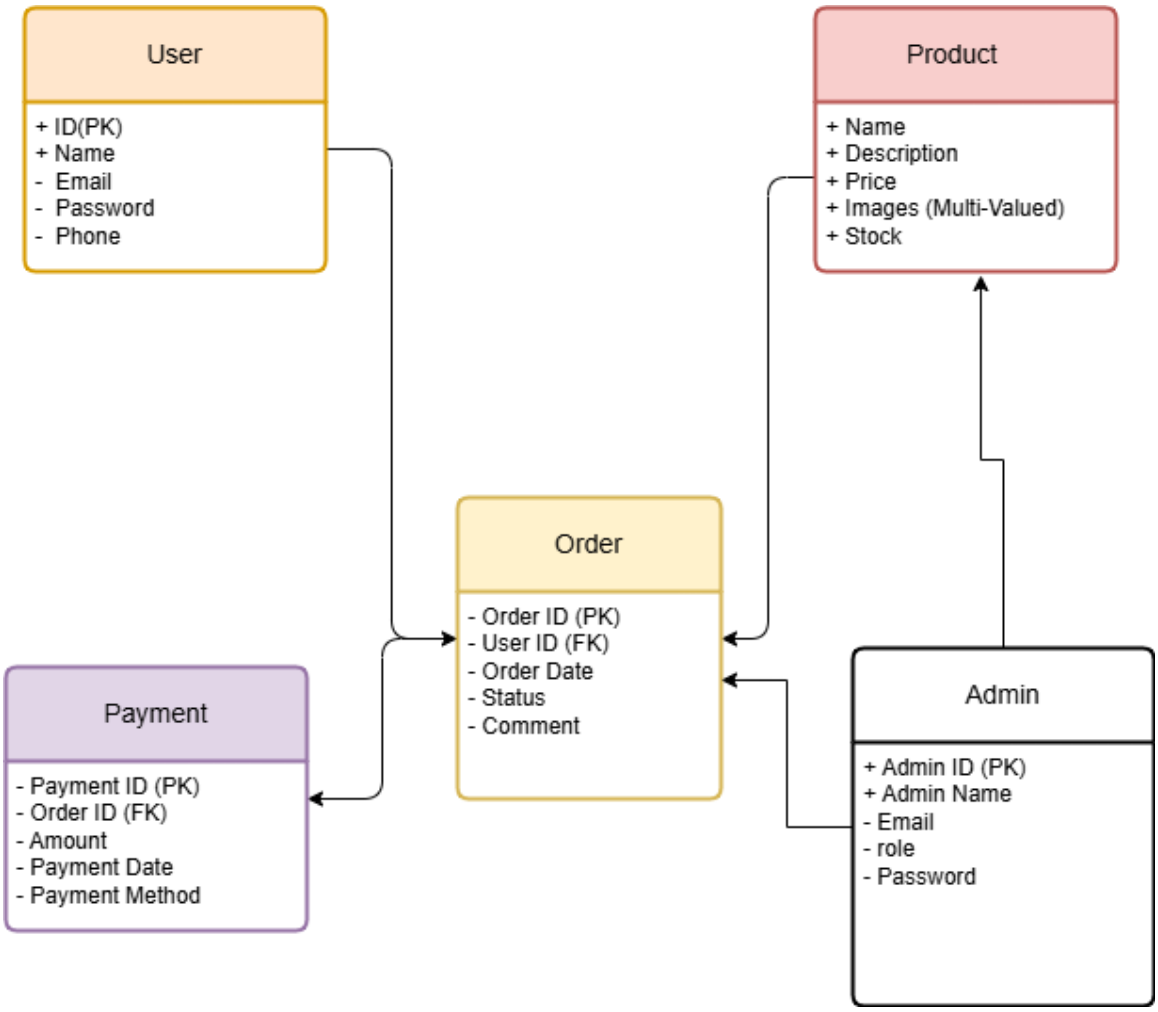


Figure 5 Logical Data Model

3.14 Logical Data Model

- **User Table:** ID (Primary Key), Name, Email, Password.
- **Product Table**: ID (Primary Key), Name, Price, Stock.
- **Order Table:** ID (Primary Key), UserID (Foreign Key), ProductID (Foreign Key), Status.
- **Admin Table:** ID (Primary Key), Name, Email, Role.

3.15 Logical Diagram Description

3.15.1 User Entity

Represents customers who interact with the system.

Attributes:

1. **ID (PK)**: Primary key, unique identifier for each user.
2. **Name**: User's full name.
3. **Email**: Email address used for communication and authentication.
4. **Password**: Encrypted password for secure login.
5. **Phone**: Contact number of the user.

3.15.2 Product Entity

Represents products available in the e-commerce store.

Attributes:

1. **Name**: Product name.
2. **Description**: Brief details about the product.
3. **Price**: Cost of the product.
4. **Images**: Multi-valued attribute that holds multiple images for each product.
5. **Stock**: Quantity of the product available in inventory.

3.15.3 Order Entity

Represents orders placed by users.

Attributes:

1. **Order ID (PK)**: Unique identifier for each order.
2. **User ID (FK)**: Foreign key that connects to the **User** entity.
3. **Order Date**: Date when the order was placed.
4. **Status**: Current status of the order (e.g., pending, shipped, delivered).
5. **Comment**: Optional remarks related to the order.

Relationships:

1. Connected to the **User** entity through **User ID**.
2. Linked to the **Admin** entity for order management.
3. Associated with the **Payment** entity for payment details.

3.15.4 Payment Entity:

Represents payment details associated with orders.

Attributes:

1. **Payment ID (PK)**: Unique identifier for each payment.
2. **Order ID (FK)**: Foreign key linking to the **Order** entity.
3. **Amount**: Payment amount.
4. **Payment Date**: Date the payment was made.
5. **Payment Method**: Mode of payment (e.g., credit card, PayPal).

Relationships:

1. Linked to the **Order** entity via **Order ID**.

3.15.5 Admin Entity:

Represents administrative users who manage the e-commerce system.

Attributes:

1. **Admin ID (PK)**: Primary key, unique identifier for each admin.
2. **Admin Name**: Full name of the admin.
3. **Email**: Admin's email address for communication.
4. **Role**: Specifies the admin's permission level (e.g., Super Admin, Product Manager).
5. **Password**: Encrypted password for secure access.

Relationships:

1. Connected to the **Product** entity for managing product inventory.
2. Linked to the **Order** entity for monitoring and updating order statuses.

3.16 Relationships Between Entities

1. **User → Order**:
   - **One-to-Many Relationship**: A user can place multiple orders.
2. **Order → Payment**:
   - **One-to-One Relationship**: Each order has one payment record.
3. **Product → Order**:
   - **Many-to-Many Relationship** (not directly shown but implied): Multiple products can belong to an order.
4. **Admin → Product**:
   - **One-to-Many Relationship**: Admins manage multiple products.
5. **Admin → Order**:
   - **One-to-Many Relationship**: Admins oversee and update multiple orders.

Chapter 4
DESIGN SPECIFICATION

4.1 Front-End Design

The front-end design demonstrates the development of an intuitive and user-friendly interface for customers and administrators alike. The client front-end is designed for responsiveness and accessibility, facilitating effortless navigation across devices. The platform includes an interactive homepage, product search capabilities, a shopping cart, and login and sign-up pages, all built using React.js for dynamic rendering and seamless user experiences. Focus is directed on aesthetically pleasing layouts and comprehensible components that enhance client involvement. Conversely, the administrative front-end offers a comprehensive and effective interface for overseeing platform activities. It encompasses vital sections such the dashboard, product management, order management, user administration, administrative account control, and profile management. These components are meticulously organized to enable administrators to effectively oversee and control the system. Utilizing contemporary UI frameworks such as Bootstrap or Material-UI, the front-end design conforms to prevailing web development trends to provide a sophisticated and effective user experience.

4.2 Back-End Design

The back-end architecture functions as the system's foundation, facilitating seamless interaction between the client interface and the database while effectively managing business logic. The back end, constructed using Node.js and Express.js, offers a powerful, scalable, and secure server-side architecture for the e-commerce platform. RESTful APIs are designed to provide efficient data interchange between the front-end and back-end, enabling functions such as product administration, user authentication, order processing, and payment processing. The back-end manages procedures like product exploration, cart administration, order submission, and user verification for client operations. On the

administrative side, it facilitates functions such as adding, modifying, or removing goods, managing users and orders, and overseeing system operations. Database activities are conducted using MongoDB, facilitating efficient storage, retrieval, and modification of data, including collections for users, goods, orders, and administrative organizations. Data validation, error handling, and security protocols, including JWT-based authentication and input sanitization, are implemented to safeguard against unauthorized access and data breaches. This meticulously organized back-end architecture guarantees dependability, scalability, and optimal performance for the whole e-commerce system.

4.3 Project Layout

- **Customer Panel**
  1. Home Page
  2. Cart
  3. Searching
  4. Login
  5. Sign Up

- **Admin Panel**
  1. Admin Dashboard
  2. Products
  3. Orders
  4. Users
  5. Admins
  6. Profile
  7. Logout

4.4 Home Page

The homepage of the website showcases a dynamic and flexible design, with a big product banner that emphasizes a highlighted item along with its characteristics, price, and major selling aspects. Under the header, a "Latest Products" section showcases a grid-based product listing system, including several goods along with critical information such as price, user ratings, and reviews. Each product card has a "Add To Cart" button, which facilitates the addition of products to the shopping cart, embodying essential characteristics of E-commerce systems. The upper navigation bar has a search bar for quick product filtering, a shopping cart symbol for cart management, and a sign-in option for user verification. The pagination feature is located at the bottom, signifying support for extensive datasets and improving scalability. The design integrates intuitive navigation with fundamental E-commerce functionalities, exemplifying the use of contemporary technology in creating a practical, dynamic, and responsive online platform.

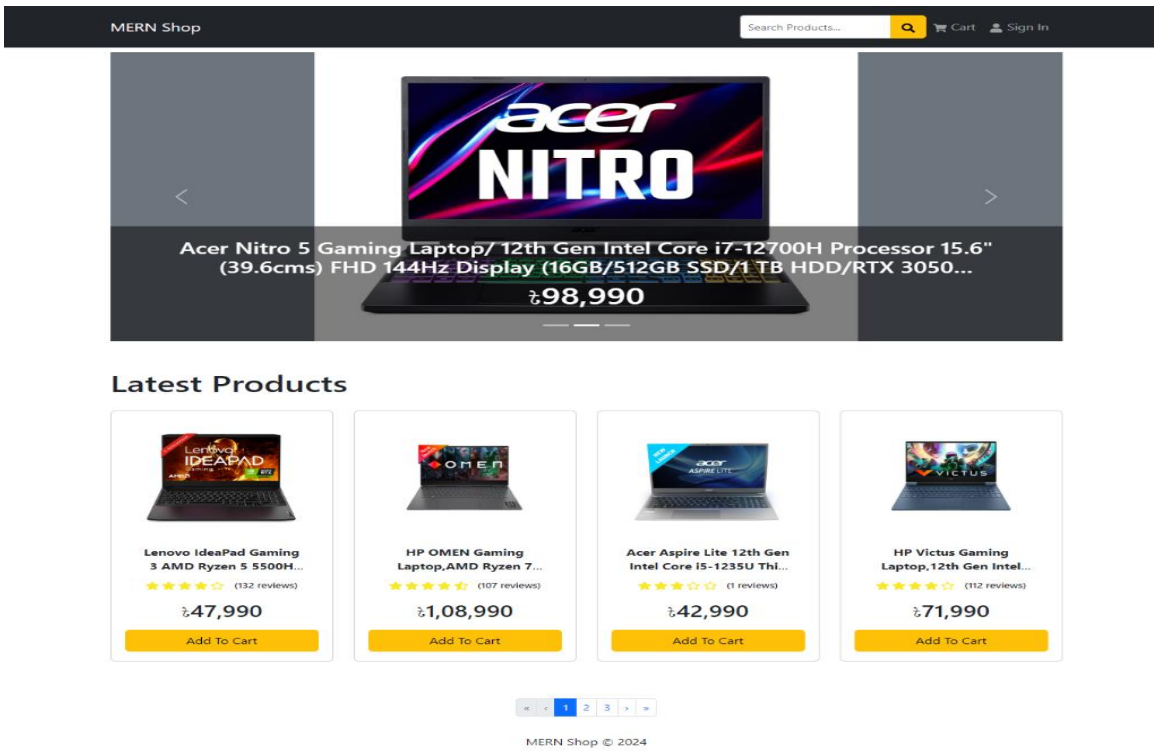


Figure 6 Home Page

4.5 Product view page:

The page emphasizes the execution of a dynamic product display system, including a comprehensive product description, price, and inventory status. The layout offers a product picture illustrating the dynamic management and presentation of detailed data. User engagement features include a "Write a Customer Review" area, allowing registered users to post reviews and facilitating customer feedback functions. The website incorporates stock management with real-time availability, a quantity selection, and a "Add to Cart" button for efficient cart operations. The reviews area facilitates consumer ratings and comments, improving user experience via interactive involvement. This page illustrates fundamental E-commerce features, including comprehensive product presentations, user-generated feedback mechanisms, and inventory oversight, highlighting the practical implementation of full-stack web development.

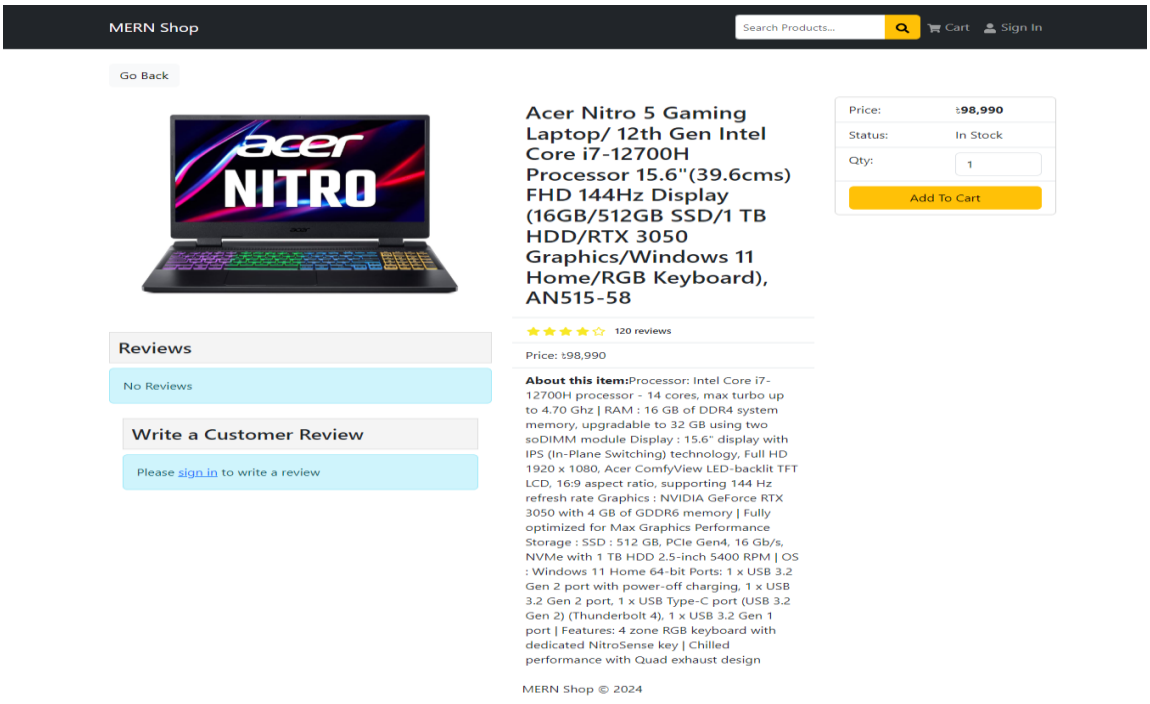


Figure 7 Product View Page

4.6 User Sign-in:

In keeping with industry standards for user identification, the page has a simple and clean design and requests for an email address and password. In addition, there is a "Keep me signed in" option to keep user sessions alive and a "Forgot password?" link to show that password recovery is an option. Users may easily log in using the "Sign In" button, while new users can access the account creation page with the "Register" link. Essential user management functions, such as account authentication, session persistence, and user onboarding, are highlighted by these characteristics. Following current standards in web construction, this page provides a solid basis for interactive and secure e-commerce platforms.

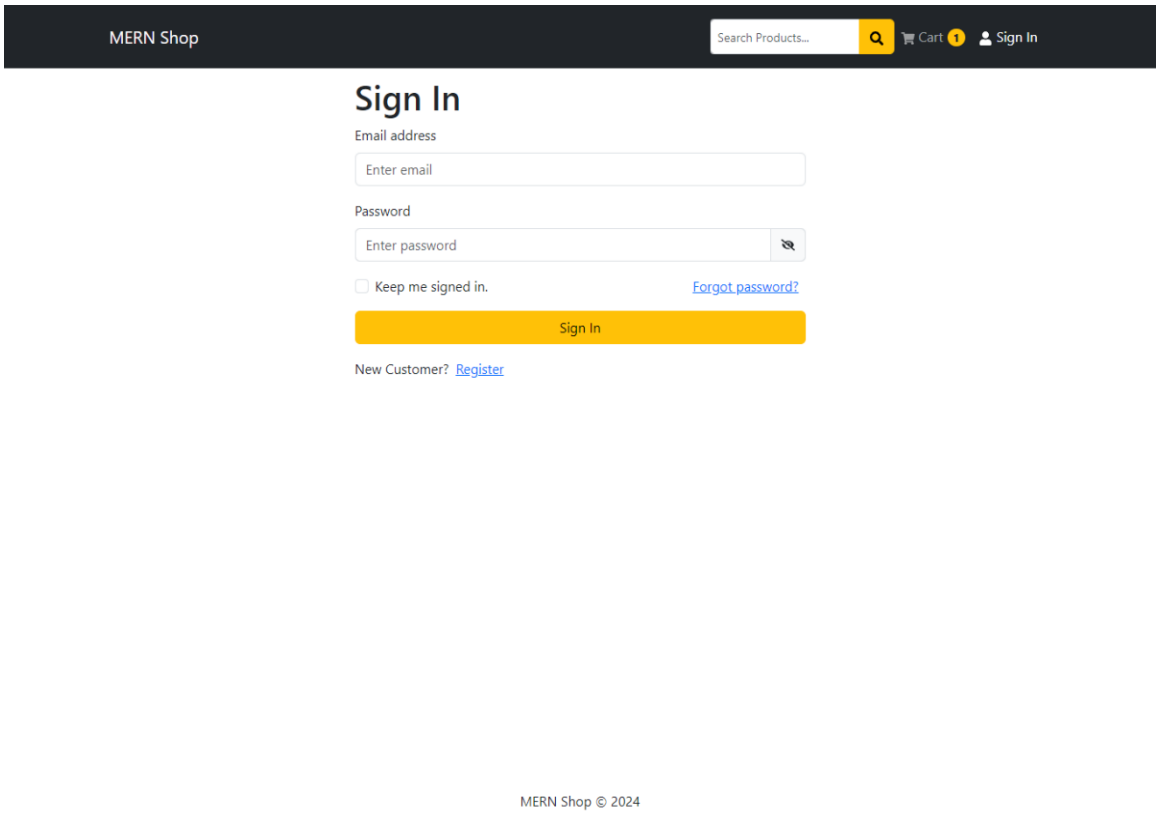


Figure 8 User Sign in

4.7 Register new user page:

The feature for new user registration is built into the Register page of the website. For reliable and organized user registration, it has a form with fields for Name, Email Address, Password, and Confirm Password. Users may enter their information by clicking the "Register" button, and the form validation checks to see if the password and confirm password boxes are same, so the data stays intact. In order for new customers to be able to establish accounts, access personalized features, and engage with the platform, this page is vital for user initial setup.

Combined with the Sign-In page, it establishes a complete user authentication system, aligning with industry standards for secure and user-friendly E-commerce platforms.

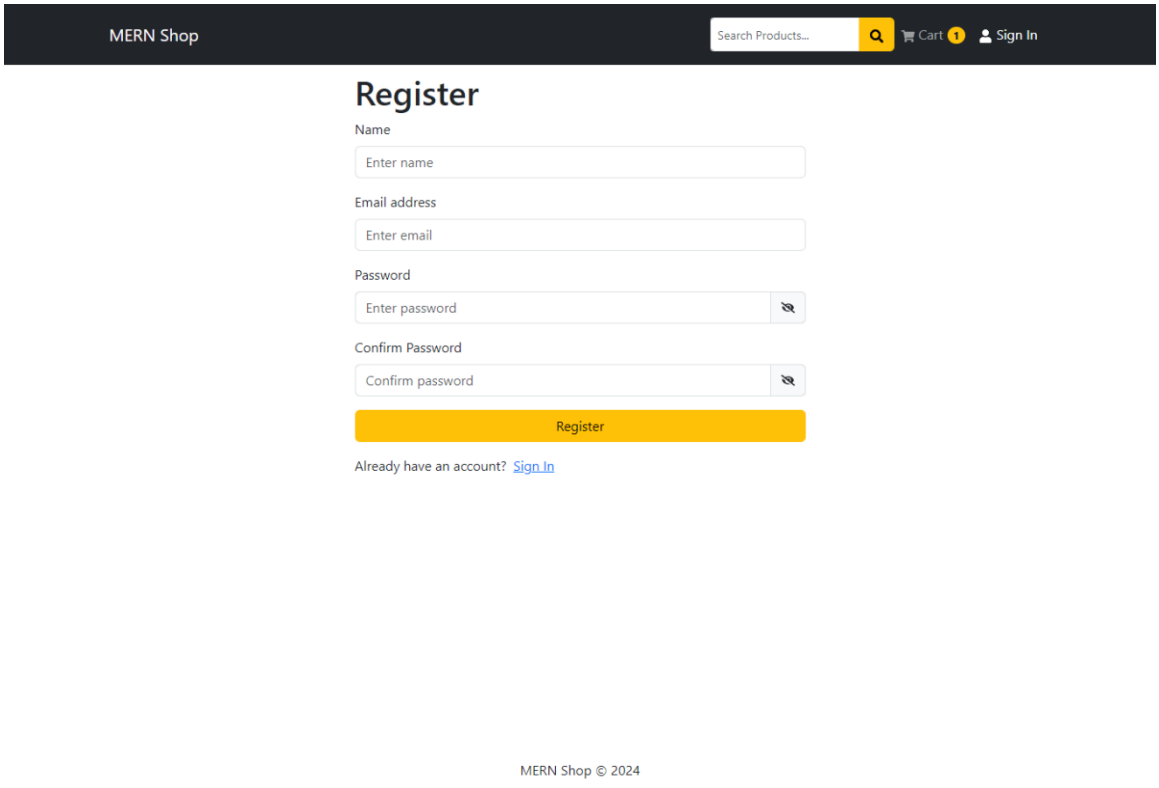


Figure 9 Register Page

4.8 Shopping Cart Page:

Users may access the basic features for managing their shopping carts on the Shopping Cart page. To make sure users know exactly how much everything costs, it shows them a list of the things they've chosen along with information like name, quantity, and price. Based on the quantity of items picked, the total amount is dynamically calculated and updated by the cart. The ability to remove things from the shopping cart gives users more flexibility and control. Customers may easily complete their purchases by clicking the "Proceed to Checkout" button. This page successfully illustrates the incorporation of fundamental E-commerce functionalities, such as dynamic pricing calculation, product elimination, and transaction processes, enhancing an easy and efficient user experience.

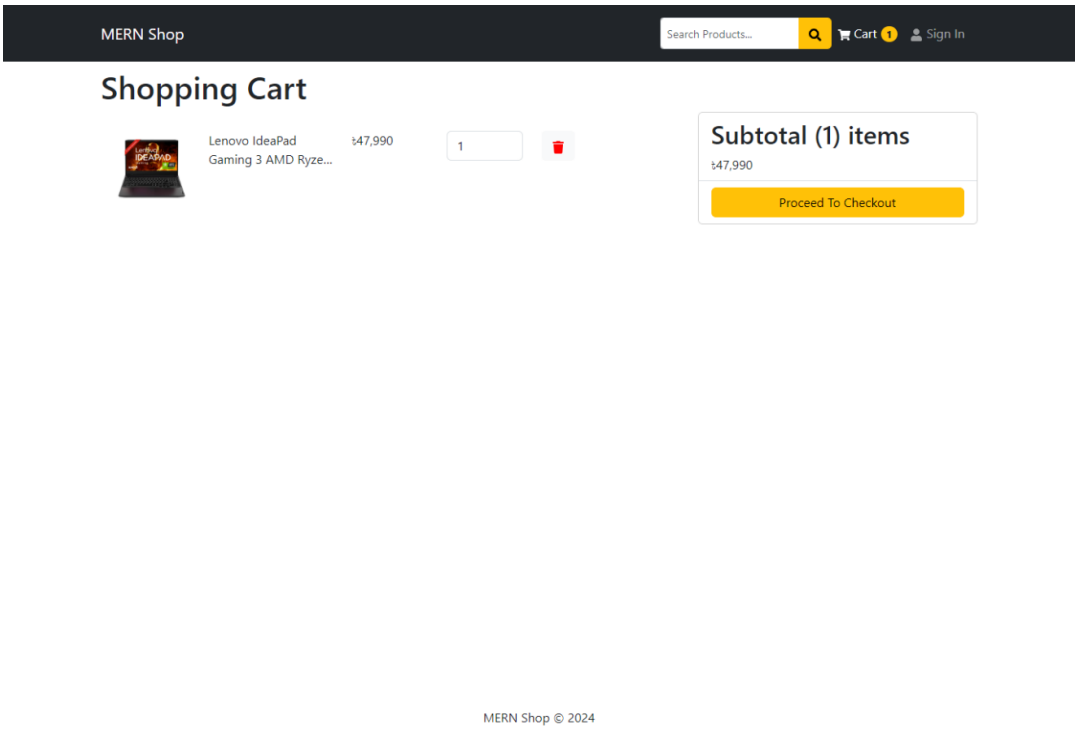


Figure 10 Shopping Cart

4.9 Admin Dashboard:

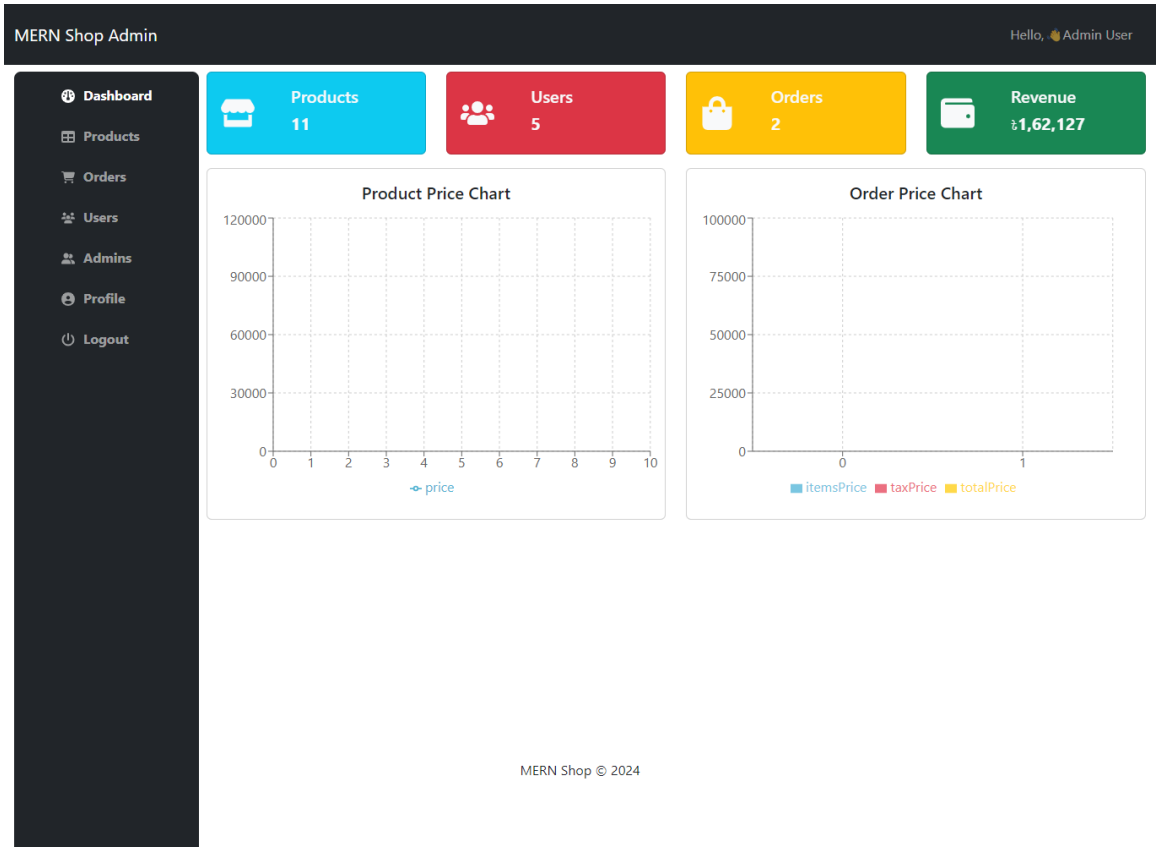


Figure 11 Admin Dashboard

The Admin Dashboard of the E-commerce website is organized to offer administrators with an extensive overview and management of platform activities. The dashboard includes summary cards that present essential indicators, such as the overall count of items, users, orders, and revenue, facilitating rapid insights into the system's operation. Interactive charts show product price data and order pricing breakdowns, providing an analytical perspective on trends and financial performance. The side navigation menu offers links for managing items, orders, users, and administrators, as well as access to profile settings and a logout option, facilitating efficient navigation of administrative functions. This dashboard demonstrates an effective use of administrative control, integrating data, consequently serving as an essential element of the platform's backend operations.

4.10 Product management: The Product Management Page enables the administrator to effectively manage the inventory of the e-commerce site. This page presents a tabular format that displays vital product information, including ID, name, price, category, and brand, assuring clarity and accessibility to crucial data.

Each product row contains action buttons for editing or removing a product, facilitating efficient changes or removal of items as required. The pagination function at the bottom enables effortless exploration across extensive product catalogues. The "Add Product" button, strategically located at the top-right corner, enables administrators to efficiently incorporate new goods into the inventory.

This page is essential for sustaining an accurate and current product inventory, facilitating the efficient functioning of the e-commerce platform.

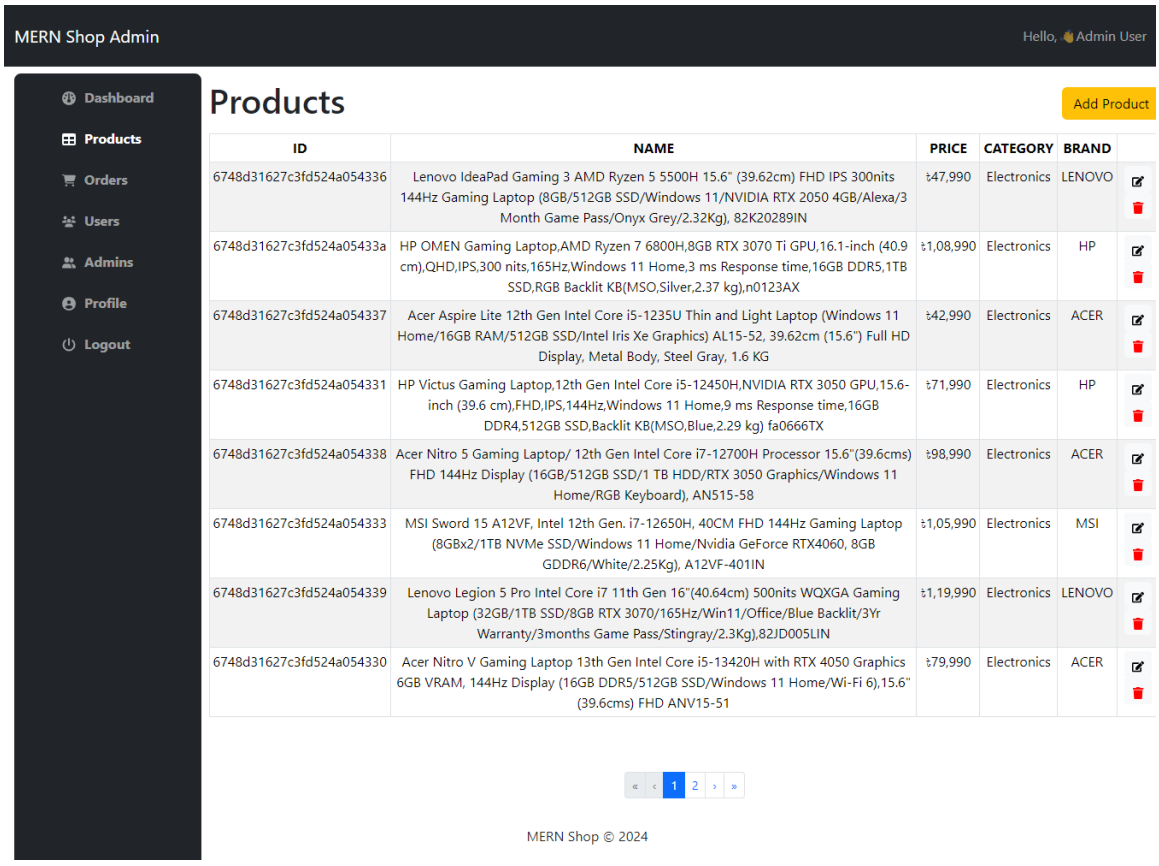


Figure 12 Products Management Page

4.11 Customer Order Page:

The "Orders" page in the MERN Shop Admin interface displays a tabular summary of recent orders. Each row presents important order information, comprising a unique identifier, the customer's name, the order date, the total amount, payment status (paid or unpaid), delivery status (delivered or undelivered), and a "Details" button for further information. This website functions as a single repository for order management, enabling administrators to monitor order status, detect possible problems, and promptly address client enquiries.

.

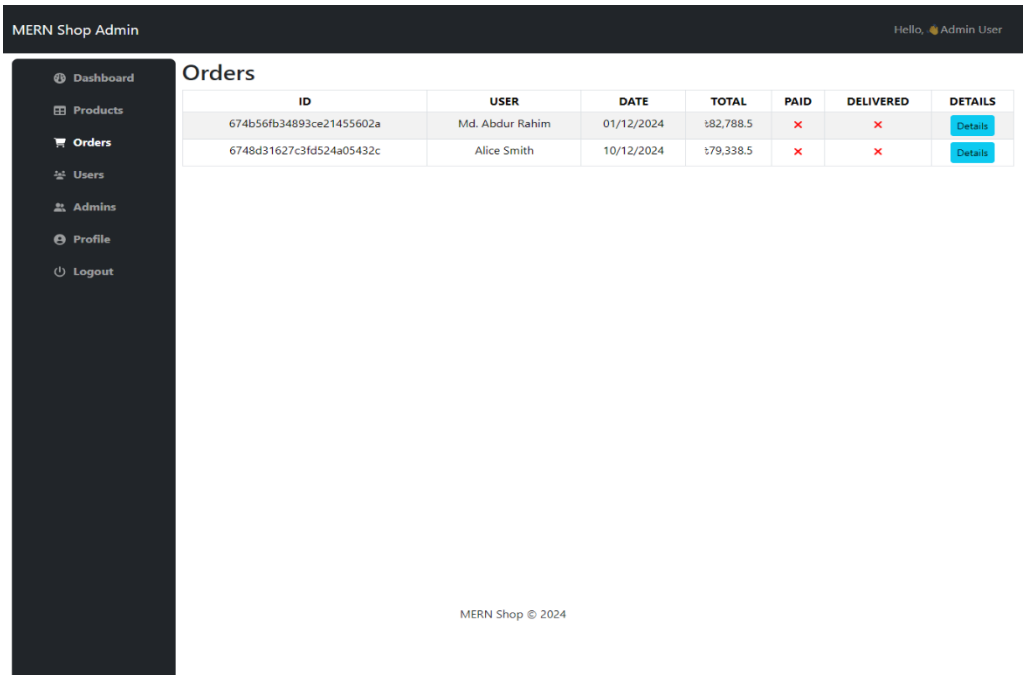


Figure 13 Order Management

4.12 User management Page:

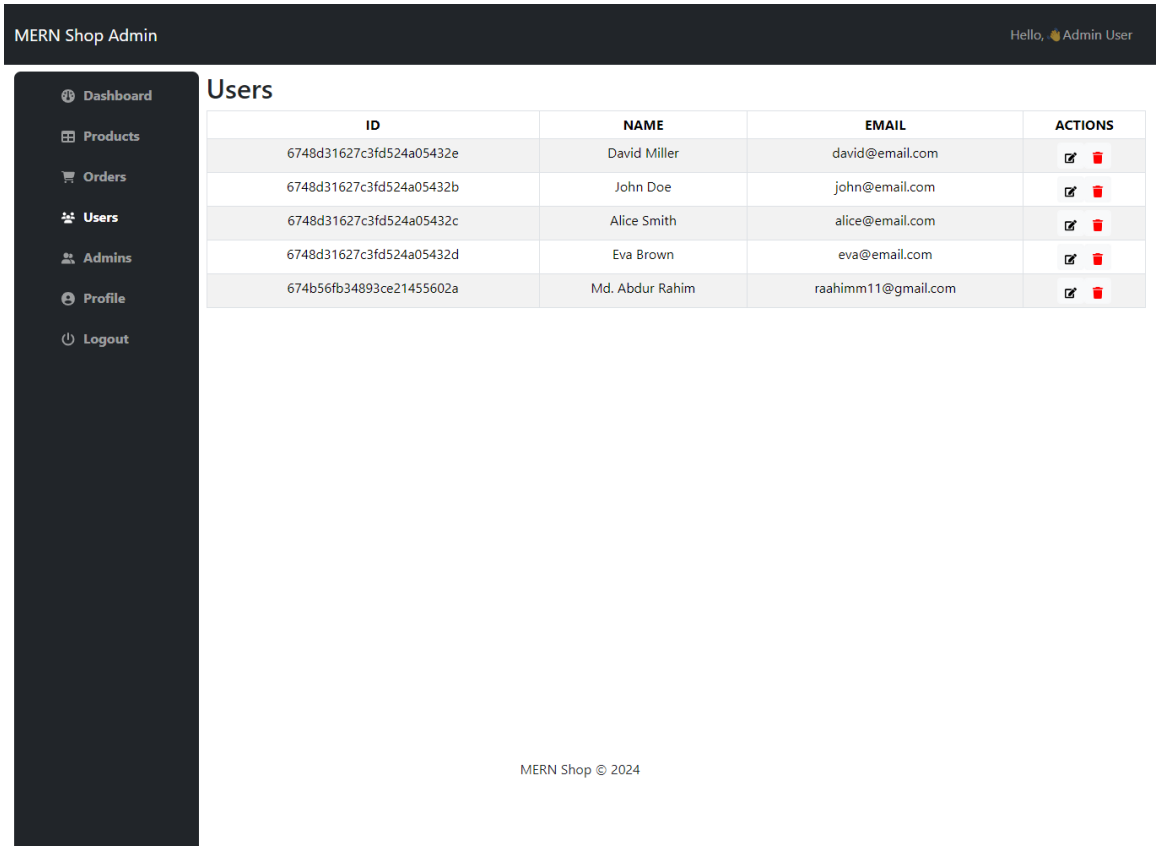


Figure 14 User Management Page

The "Users" area operates as the primary Centre for overseeing registered users on the website. This page features a tabular collection of users, with each row presenting their unique ID, complete name, email address, and a series of options available for their account, including amending or deleting. This feature enables administrators to effectively manage user accounts, equipping them with the capabilities to add, alter, or delete users as required.

4.13 Admin List Page: The "Admins management page" serves as a single point for the management of administrator users. This page presents a list of current administrators, with each entry including their unique ID, name, email address, and a series of actions available for their accounts, including modification or deletion. A noticeable "Add Admin" button allows for the establishment of new administrative accounts, empowering administrators to augment their team and assign administrative duties as required. This page enables administrators to regulate access and uphold a safe and efficient administrative environment for the online store.

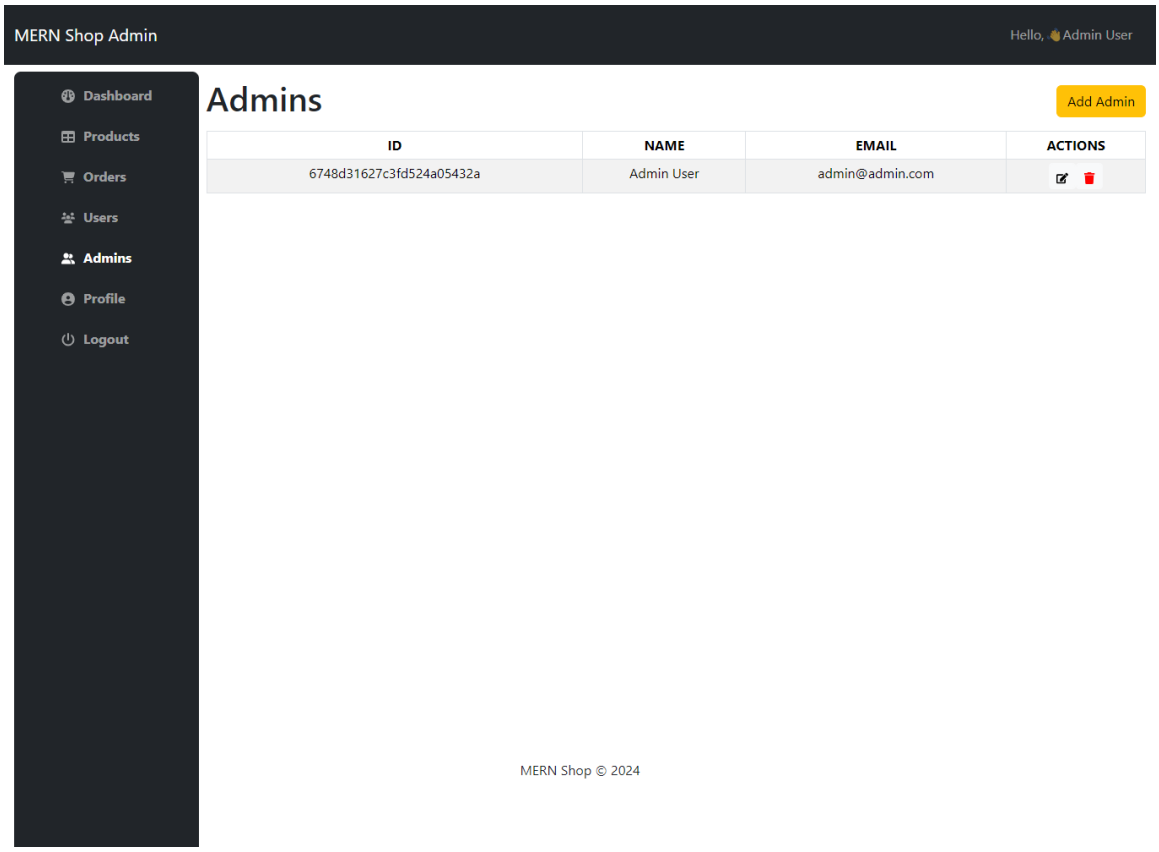


Figure 15 Admin Management Page

4.14 Admin Profile Page:

The "Admin Profile" tab provides a specific platform for monitoring the current administrator's account information. This page generally features input forms for modifying the administrator's name and email address, as well as password management fields which often contain toggles for password visibility for enhanced security. The "Update" button enables the submission of modifications, initiating a request to the backend server to revise the administrator's information in the database. This page allows administrators to uphold precise and current account information while guaranteeing a secure and user-friendly interface for controlling their administrative access.

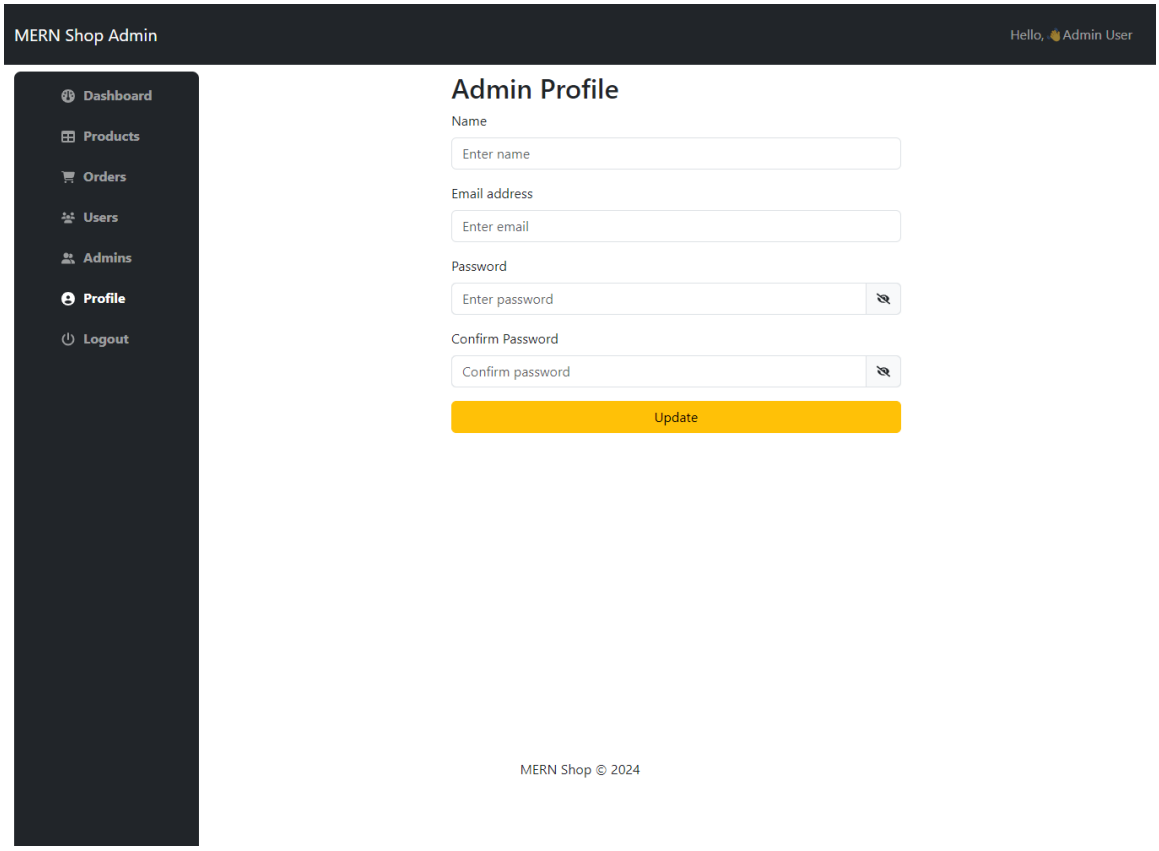


Figure 16 Admin Profile

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Discussion and Conclusion

The creation of the MERN stack-based e-commerce platform proficiently resolves the difficulties associated with developing a contemporary, user-centric, and efficient online purchasing system. The platform includes comprehensive features, such as product exploration, user identification, shopping cart administration, order processing, and efficient administrative tools for overseeing users, orders, and inventories. The project utilises the MERN stack (MongoDB, Express.js, React.js, and Node.js) to guarantee scalability, speed, and efficient data management across components. The use of flexible design concepts enhances user experience across several devices, while security protocols, such as JWT authentication, protect user data and transactions. The project demonstrates the capacity to leverage contemporary technology to address the increasing requirements of e-commerce enterprises, providing an easy and efficient solution for both users and administrators. The platform fulfils its basic objectives while establishing a robust basis for further enhancements and extensions.

5.2 Scope for Future Development

The e-commerce platform may be developed and improved in various ways to address changing technology and commercial requirements. Potential possibilities for future expansion encompass, such as:

Advanced Analytics and Reporting: Utilizing data analytics techniques can provide administrators with insights into sales patterns, user behavior, and inventory management.

Recommendation Engine: Applying machine learning algorithms to provide personalized product suggestions to consumers based on their browsing and purchase history.

Real-Time Order Tracking: Adding live order tracking features for customers, improving transparency and user satisfaction.

Integration with Third-Party APIs: Expanding the platform through the incorporation of payment gateways, shipping services, and other markets to improve functionality.

Mobile Application: Creating a dedicated mobile application to enhance the online platform and expand consumer reach.

Multilingual and Multi-Currency Support: Facilitating support for several languages and currencies to accommodate a worldwide audience.

AI-Powered Chatbots: Incorporating AI-driven customer care solutions to aid consumers with enquiries, grievances, and suggestions.

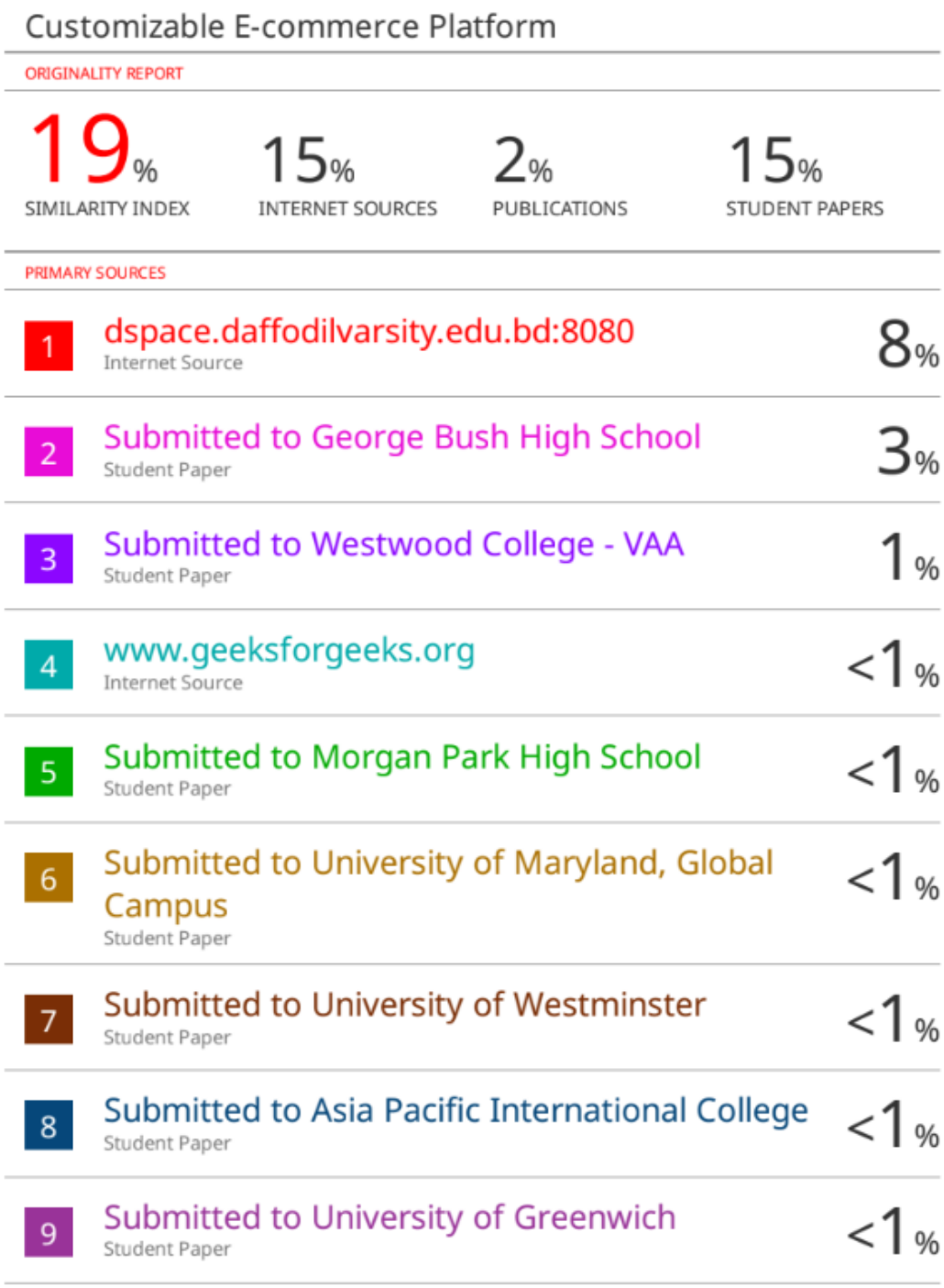
Enhanced Security Measures: Consistently enhancing safety measures to address growing threats, including fraud detection systems and sophisticated encryption methods.

These afterwards advancements will not only improve the platform's functioning but also guarantee its relevance and competitiveness in the evolving e-commerce environment. Through persistent adaptation and innovation, the platform can successfully address the evolving requirements of businesses and users.

REFERENCE

1. A. Shrivas, A. Pawar, P. Mishra, and S. Chadokar, “E-commerce Website using MERN Stack,” *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, vol. 9, no. 3, pp. 1–10, 2024.
2. D. Sai Tharun et al., “E-commerce Platform Development using MERN Stack,” *International Journal of Advanced Research in Science, Communication and Technology*, vol. 4, no. 1, pp. 112–119, 2024.
3. "Comparison of E-commerce Technologies: MERN vs. Alternatives," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 10, no. 2, pp. 587–591, 2023. Available: www.ijsrset.com.
4. “7 Reasons to Choose MERN Stack for E-Commerce Development,” Biz4Commerce. Available: www.biz4commerce.com.
5. GeeksforGeeks. (n.d.). *How to design ER diagrams for e-commerce websites*. GeeksforGeeks. Retrieved June 2024, from https://www.geeksforgeeks.org/how-to-design-er-diagrams-for-e-commerce-website
6. Vertabelo. (n.d.). *ER diagram for an online shop*. Vertabelo. Retrieved June 2024, from https://vertabelo.com/blog/er-diagram-for-online-shop
7. Giri, B. (2022). *Designing the database schema for a new e-commerce platform and considering factors like scalability*. Medium. Retrieved June 2024, from https://bgiri-gcloud.medium.com/designing-the-database-schema-for-a-new-e-commerce-platform-and-considering-factors-like-ec28d4fb81db
8. Fabric Inc. (n.d.). *E-commerce database design example*. Fabric. Retrieved June 2024, from https://fabric.inc/blog/commerce/ecommerce-database-design-example
9. CIN-UFPE. (n.d.). *A comparative analysis of entity-relationship diagrams*. CIN-UFPE. Retrieved June 2024, from https://www.cin.ufpe.br/~in1008/aulas/A%20Comparative%20Analysis%20of%20Entity-Relationship%20Diagrams.pdf
10. Reddit. (n.d.). *Tips for creating an e-commerce database*. Reddit. Retrieved June 2024, from https://www.reddit.com/r/Database/comments/1cbbxwq/tips_for_creating_an_ecommerce_database

APPENDIX



Customizable E-commerce Platform

ORIGINALITY REPORT

19%	15%	2%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to George Bush High School Student Paper	3%
3	Submitted to Westwood College - VAA Student Paper	1%
4	www.geeksforgeeks.org Internet Source	<1%
5	Submitted to Morgan Park High School Student Paper	<1%
6	Submitted to University of Maryland, Global Campus Student Paper	<1%
7	Submitted to University of Westminster Student Paper	<1%
8	Submitted to Asia Pacific International College Student Paper	<1%
9	Submitted to University of Greenwich Student Paper	<1%