WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

E-Commerce Store with an Admin Dashboard

Mr.P.Logiyan[1], Dommeti Venkata kiran[2]

[1]Assistant Professor, Department of Computer Applications, Sri Manakula Vinayagar Engineering College
[2]PG Student, Department of Computer Applications, Sri Manakula Vinayagar Engineering College

ABSTRACT

This project presents the design and implementation of a modern, full-stack E-Commerce Store with an Admin Dashboard, developed using the MERN stack—MongoDB, Express.js, React.js, and Node.js—integrated with Stripe for secure payment processing and Redis for high-performance caching. The primary goal of the system is to deliver a seamless online shopping experience for customers while providing store administrators with powerful tools to manage products, track orders, and analyze sales. The application supports secure user authentication using JWT access and refresh tokens, enabling smooth and safe login/logout flows. Customers can browse products by categories, add items to their shopping cart, apply coupon codes, and complete purchases through Stripe's trusted payment gateway. Meanwhile, the admin interface allows for real-time management of inventory, user roles, and performance metrics through intuitive dashboards and visual analytics. Built with TailwindCSS, the frontend offers a clean, responsive design that works across all devices. On the backend, Redis ensures optimized performance through caching, and MongoDB handles complex data storage. Security measures, including input validation, password hashing, and route protection, are incorporated throughout the system to protect user data and prevent unauthorized access. This project showcases how scalable, secure, and efficient e-commerce platforms can be developed using modern web technologies, offering practical insights into full-stack architecture, cloud integration, and production-grade software engineering.

INTRODUCTION

In the modern digital era, e-commerce platforms have revolutionized the way consumers and businesses interact, offering convenience, scalability, and global reach. This paper presents the development of a full-stack E-Commerce Store with an Admin Dashboard, built using the MERN stack (MongoDB, Express.js, React.js, Node.js), tailored to simulate real-world commercial functionality. The system provides two primary interfaces: one for end-users to browse, purchase, and manage their orders, and another for administrators to control product listings, handle orders, apply discounts, and monitor real-time analytics. To enhance the performance and usability of the application, the system integrates Stripe for secure payment processing and Redis for caching frequently accessed data. Security features such as JWT-based authentication and role-based access control ensure data protection and privacy. Unlike conventional e-commerce platforms, this project emphasizes customization, scalability, and educational value, making it ideal for both deployment and academic exploration. The proposed solution addresses the limitations of existing platforms—such as high cost, limited backend control, and lack of development insight—by offering a robust, open-source, and modular alternative.

PROBLEM STATEMENT

Despite the rapid growth of online commerce, many existing e-commerce platforms like Shopify, WooCommerce, and Magento present significant limitations for developers and small businesses. These include high subscription costs, restricted backend customization, limited scalability, and lack of transparency in system architecture. Additionally, most platforms do not offer real-time analytics, efficient caching mechanisms, or developer-friendly codebases for academic or training purposes. There is a pressing need for a cost-effective, customizable, and scalable e-commerce solution that offers end-to-end functionality, including secure user authentication, product and order management, payment processing, and admin-level analytics. Moreover, integrating modern technologies such as the MERN stack, Stripe, and Redis can provide a better learning experience for developers while solving real-world problems. This project aims to address these challenges by developing a full-stack e-commerce application with an intuitive admin dashboard, built entirely using open-source technologies.

LITERATURE SURVEY

1. Sharma, K., & Desai, P. (2016). *Design and Implementation of a Web-Based E-commerce Platform*. International Journal of Computer Applications, 145(9), 1–5.
2. Patil, R., & Jadhav, A. (2019). *MERN Stack E-Commerce Web Application Development*. International Research Journal of Engineering and Technology (IRJET), 6(5), 1425–1429.
3. Gupta, S., & Rathod, D. (2020). *Design of Admin Dashboard for Small Scale E-commerce Platforms*. Journal of Web Engineering and Technology, 7(2), 93–100.

4. Verma, T., & Kapoor, A. (2021). *Secure Online Transaction System using Stripe and Node.js*. International Journal of Advanced Computer Science and Applications (IJACSA), 12(6), 411–417.
5. Sinha, M., & Thomas, L. (2020). *Performance Optimization in Web Applications using Redis Caching*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(1), 45–51.
6. Nair, J., & George, B. (2021). *Comparative Study of Traditional vs. Headless E-commerce Platforms*. International Journal of Digital Technologies, 5(3), 67–73.
7. Mehta, A., & Banerjee, R. (2022). *Full-Stack JavaScript for Scalable Web Applications: A Case Study*. Journal of Software Engineering and Development, 10(1), 28–36.

PROPOSED SYSTEM

The proposed system is a **full-stack e-commerce web application** built using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**, designed to offer a scalable, secure, and customizable online shopping experience. It includes two key interfaces:

- A user-facing storefront for browsing products, managing carts, applying coupons, placing orders, and making payments.
- An admin dashboard for managing product inventory, tracking orders, issuing coupons, viewing sales analytics, and controlling system settings.

The system adopts modular architecture and modern development practices to address the limitations of traditional platforms. It uses JWT-based authentication with role-based access control to ensure secure user interactions. Stripe API integration enables seamless and secure payment processing, while Redis caching boosts system performance by reducing response times for frequently accessed data. Furthermore, the system is optimized for mobile responsiveness, API-first communication, and real-time data tracking using admin dashboards. It promotes developer learning and business readiness by using only open-source, production-ready technologies and offering complete control over the backend.

Modules

1. User Authentication Module

- Implements secure login and registration using JWT (access and refresh tokens).
- Includes password hashing, role-based access control, and protected routes for users and admins.

2. Product and Category Management

- Admins can create, update, and delete products and categories.
- Includes fields like product name, price, description, stock, image, and category association.

3. Shopping Cart System

- Allows users to add, update, and remove items from the cart.
- Auto-updates total cost and validates available stock in real-time.

4. Coupon Module

- Admins can generate and manage discount coupons (percentage or fixed).
- Customers can apply coupons at checkout to receive automatic discounts.

5. Order Management System

- Handles order creation, status updates (pending, shipped, delivered), and invoice generation.
- Admins can view and update order statuses through the dashboard.

6. Admin Dashboard Module

- Admins have access to a visual control panel with real-time analytics.
- Includes charts, user management, product reports, and order tracking tools.

7. Stripe Payment Integration

- Integrates with **Stripe API** to process secure online payments.
- Supports webhook listeners to handle payment confirmation and transaction

DIAGRAM

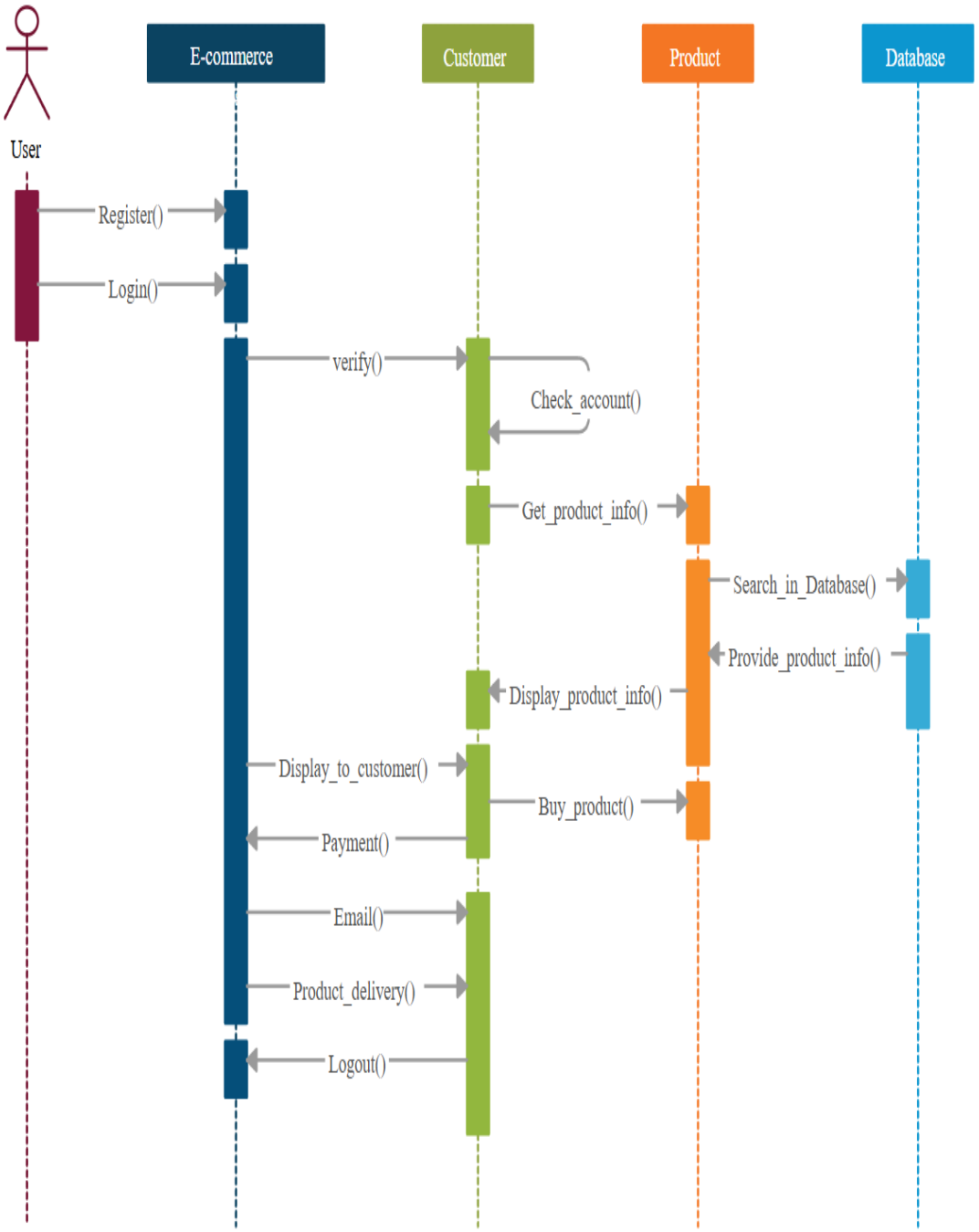

Fig 1: USE CASE DIAGRAM

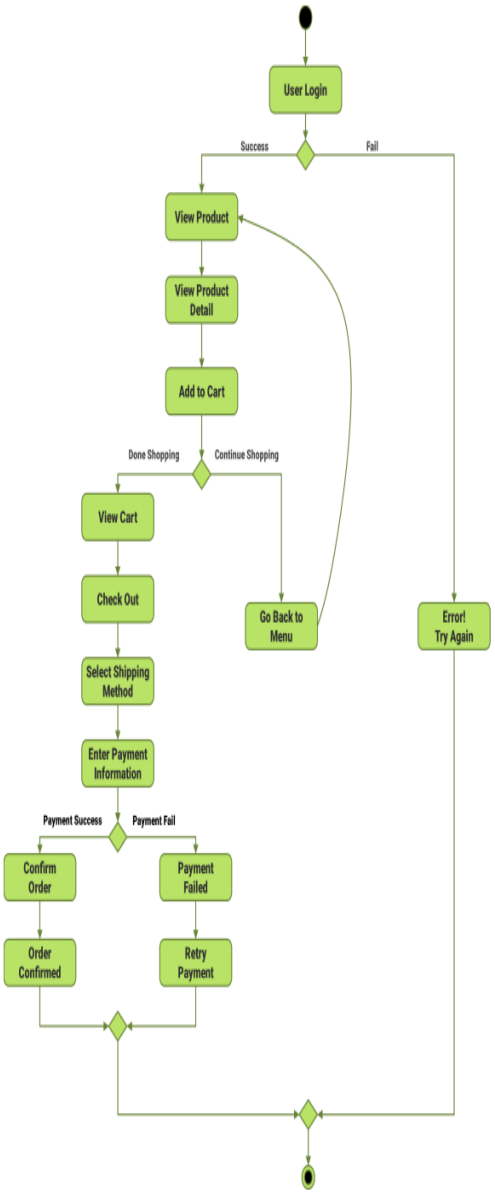

Fig 2: ACTIVITY DIAGRAM

Methodology

The development of this E-Commerce platform followed a structured, modular, and iterative approach grounded in modern web development practices. The methodology was divided into key stages, as outlined below:

1. Requirement Analysis

The process began with collecting and analyzing functional and non-functional requirements. This included identifying key features such as user authentication, product management, cart and order systems, payment integration, and admin dashboards. The system was designed to support both user and admin roles with clearly defined privileges.

2. Technology Stack Selection

The MERN stack was chosen for its efficiency, scalability, and use of a single programming language (JavaScript) across the entire application. Additional tools and services used include:

- **Stripe API** for secure online payments
- **Redis** for performance optimization via caching
- **Tailwind CSS** for responsive and modern UI
- **Git and GitHub** for version control and collaboration

3. System Design

- **Frontend (React.js)**: Component-based architecture was adopted for reusability and responsiveness. React Router was used for client-side navigation.
- **Backend (Node.js & Express.js)**: RESTful APIs were created to handle authentication, product, and order management. Middleware was implemented for validation and error handling.
- **Database (MongoDB)**: NoSQL schema design using Mongoose models allowed for flexible storage of users, products, orders, and coupons.
- **Authentication**: JWT (access & refresh tokens) provided secure login and role-based access.
- **Admin Dashboard**: Built with protected routes and integrated real-time sales analytics using chart libraries.

4. Implementation

The development followed agile principles, with each module implemented and tested in iterations:

- User Authentication
- Product & Category Management
- Cart and Coupon System
- Order Processing
- Stripe Payment Gateway
- Redis Caching Layer
- Admin Control Panel

5. Testing and Validation

Manual and automated testing were performed to validate:

- API endpoints (using Postman)
- UI responsiveness (mobile and desktop)
- Authentication flows and role access
- Payment processing and webhook verification
- Caching and database response times

6. Deployment

- Frontend was deployed on Vercel
- Backend APIs were hosted using Render/Heroku
- Environment variables and secure credentials were managed using .env files
- GitHub was used for CI/CD version tracking

This methodology ensured a smooth development cycle, resulting in a scalable, secure, and production-ready e-commerce solution.

CONCLUSION

The development of the E-Commerce Store with Admin Dashboard using the MERN stack has successfully demonstrated the creation of a scalable, modular, and secure online shopping platform. The system integrates modern technologies like JWT for authentication, Stripe for secure payment processing, and Redis for caching, ensuring both performance and reliability. It features complete customer-side functionality alongside a powerful admin dashboard that allows product management, order tracking, coupon generation, and real-time sales analytics. This project not only replicates real-world e-commerce features but also provides a valuable learning experience in full-stack web development, API design, security implementation, and deployment practices.

FUTURE SCOPE

- **Product Review and Rating System**: Allow users to post reviews and rate products, improving customer engagement.
- **Wishlist and Favorites**: Enable customers to save products for later purchase.
- **Shipping & Delivery Tracking**: Integrate APIs to provide real-time shipping updates.
- **Multi-vendor Support**: Allow multiple sellers to manage their inventories on the same platform.
- **Mobile App Integration**: Develop a cross-platform mobile app using React Native for enhanced user access.
- **AI-Based Product Recommendations**: Use machine learning to recommend products based on user behavior and history.

- **Advanced Admin Analytics**: Implement charts and predictive analytics for sales trends and customer behavior.

REFERENCES

1. Brad Traversy – *MERN Stack Front To Back*, Udemy Course.
2. **MDN Web Docs** – JavaScript, HTML, CSS Documentation. Available at: https://developer.mozilla.org
3. **React.js Documentation** – Official React Docs. Available at: https://reactjs.org
4. **Node.js Documentation** – Node.js API and Modules. Available at: https://nodejs.org
5. **Express.js Guide** – Express Documentation. Available at: https://expressjs.com
6. **MongoDB Documentation** – MongoDB Reference Manual. Available at: https://docs.mongodb.com
7. **Stack Overflow** – Developer Discussions and Code Solutions.
8. **GeeksforGeeks** – MERN Stack Tutorials and Web Dev Guides.
9. **W3Schools** – Full Stack Web Development Reference.
10. **GitHub** – Repositories and Open-Source Contributions for MERN projects.