International Journal of
AI ELECTRICAL CIVIL AND MECHANICAL ENGINEERING
Peer Reviewed, Referred & Indexed Journal
)70-0434 www.zesterapublications.com Original Research Paper

E-Commerce Website: A Full-Stack Web Application

Mr. Aditya Anhuman Sahoo
Student, Dept. of CSE,
GIFT Autonomous, Bhubaneswar
Mr. Sonu Swain
Student, Dept. of CSE,
GIFT Autonomous, Bhubaneswar
Prof. Dr. PB Manoj
Professor & Dean Exams, Dept. of CSE, GIFT Autonomous, Bhubaneswar

***Abstract*—** The rapid growth of internet technology has significantly transformed the way businesses operate, leading to the widespread adoption of electronic commerce (e-commerce). This paper presents the design and development of a fully functional E-Commerce website that enables users to browse products, add items to a cart, and securely complete purchases online. The system is developed with the objective of providing a seamless, user-friendly, and efficient online shopping experience. The proposed platform includes user authentication with OTP-based email verification, product management, shopping cart functionality, order processing, and PayPal payment integration. The application is built using the MERN Stack — MongoDB, Express.js, React.js, and Node.js — ensuring scalability, security, and maintainability. All 18 test cases passed successfully and performance metrics confirmed API response times under 150ms with a Lighthouse score of 92/100.

Keywords *— E-Commerce, MERN Stack, OTP Authentication, PayPal Integration, React.js, Node.js, MongoDB, JWT, Web Application.*

I. INTRODUCTION

E-commerce (Electronic Commerce) refers to the process of buying and selling products or services over the internet using electronic systems such as websites, mobile applications, and digital payment platforms. In recent years, online shopping has become an essential and integral part of daily life due to its unmatched convenience, global accessibility, and wide range of product availability.

Traditional shopping methods present numerous challenges for both customers and businesses. Customers must physically visit stores, which is time-consuming and inconvenient. Limited product availability, long queues, restricted store hours, and the inability to compare products reduce the efficiency of traditional shopping.

The developed E-Commerce website provides a complete digital shopping platform where users can register with OTP verification, browse products organized by categories and brands, add items to a cart, and place orders with secure online payment through PayPal. The system also includes a comprehensive admin panel for product management with cloud image upload and order monitoring.

A. Problem Statement

Existing e-commerce solutions for small and medium-scale businesses often lack: (1) a proper modern UI built with current frontend technologies, (2) secure end-to-end payment integration with trusted gateways, (3) efficient cloud-based media management, (4) robust multi-factor authentication such as OTP, and (5) a comprehensive administrative dashboard for real-time monitoring.

B. Objectives

- To design and develop a complete full-stack E-Commerce platform using the MERN Stack.
- To implement secure OTP-based email verification for user registration.
- To build a comprehensive admin panel with cloud image upload and order management.
- To integrate PayPal as the payment gateway for secure online transactions.
- To develop a Forgot Password feature using OTP verification.
- To create a fully responsive and mobile-friendly user interface.

II. LITERATURE REVIEW

E-commerce platforms have fundamentally revolutionized the global retail industry. Research by Laudon and Traver (2019) highlights that successful online shopping systems must prioritize user experience design, system performance, and transaction security. Their study shows that even a one-second delay in page loading can result in a 7% reduction in conversions [1].

The MERN Stack has emerged as one of the most popular choices for building modern web-based shopping systems, providing a unified JavaScript environment across the entire application stack. Studies on single-page application (SPA) architectures show that React.js-based SPAs provide significantly better user experiences

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 04-06-2026

International Journal of
AI ELECTRICAL CIVIL AND MECHANICAL ENGINEERING
Peer Reviewed, Referred & Indexed Journal
www.zesterapublications.com
Original Research Paper

by eliminating full page reloads [2].

Research by the Payment Card Industry (PCI) Security Standards Council emphasizes that all e-commerce systems handling financial transactions must comply with PCI DSS requirements. PayPal handles this compliance on behalf of merchants, reducing the security burden on application developers [3]. OTP-based verification, as studied by Google's security research team, can prevent 99.9% of automated bot attacks [4].

III. SYSTEM DESIGN

A. System Architecture

The E-Commerce system follows a Three-Tier Architecture separating the application into three distinct logical layers. This separation ensures that each layer can be developed, maintained, and scaled independently.

- Presentation Layer (Frontend): Built using React.js with Redux Toolkit for state management. Communicates with the backend exclusively through RESTful HTTP API calls.
- Application Layer (Backend): Built using Node.js and Express.js. Contains all business logic, enforces security through JWT authentication middleware, and integrates with external services (PayPal, Cloudinary, Brevo SMTP).
- Data Layer (Database): MongoDB stores all user accounts, product information, and order records in BSON format. Mongoose provides schema validation and a rich query interface.

B. Software Requirements

Frontend technologies include React.js (v18+), Redux Toolkit, React Router DOM, Tailwind CSS, ShadCN UI, and Axios. Backend technologies include Node.js (v18+), Express.js, Mongoose, JWT, Bcryptjs, Multer, and Nodemailer. Cloud services include MongoDB Atlas, Cloudinary for image storage, PayPal REST SDK, and Brevo SMTP for email delivery.

C. Database Design

The system uses three primary MongoDB collections:

- Users Collection: Stores user credentials, role (user/admin), isVerified flag, and OTP fields with expiry timestamps.
- Products Collection: Stores title, description, category, brand, price, salePrice, totalStock, and Cloudinary image URL.
- Orders Collection: Stores userId reference, cartItems array, addressInfo, orderStatus, paymentMethod, paymentStatus, totalAmount, and orderDate.

IV. MODULE DESCRIPTION

A. User Module

The User Module provides all functionalities for customer interaction. The registration process validates input, hashes the password using bcrypt with a salt factor of 12, generates a cryptographically secure six-digit OTP using Node.js crypto module, and saves the user with isVerified set to false. An OTP email is sent via Nodemailer with Brevo SMTP. Upon successful OTP verification within 10 minutes, the isVerified flag is set to true.

Login authenticates credentials using bcrypt comparison and generates a JWT token stored in an HttpOnly cookie containing the user's ID, email, role, and username.

B. Admin Module

The Admin Module provides role-based access control for administrators. Administrators can perform complete CRUD operations on products, upload images to Cloudinary via Multer middleware, manage all orders, and update order statuses. API endpoints are protected with admin JWT middleware.

C. Cart and Order Module

The shopping cart is implemented as a combination of Redux client-side state and a database-backed model, ensuring persistence across sessions. The checkout process creates a PayPal order through the backend, redirects the user to PayPal's hosted payment page, and upon successful payment confirmation, creates an order record and clears the user's cart.

D. OTP Authentication Module

Six-digit numeric OTPs are generated using Node.js's built-in crypto.randomInt() function for cryptographic security. OTPs have a 10-minute expiry period. The module supports two flows: (1) Registration OTP verification and (2) Forgot Password OTP reset. Each new OTP request overwrites the previous one, preventing accumulation of valid OTPs.

V. IMPLEMENTATION

A. Frontend Development

The frontend follows a component-based architecture with logical directory structure: pages for top-level components, components for reusable UI elements, store for Redux Toolkit slices, and config for constants. Redux state is divided into slices: authSlice, adminProductsSlice, shoppingProductsSlice, shoppingCartSlice, and shoppingOrderSlice. Each slice uses createAsyncThunk for handling asynchronous API calls with proper loading and error states.

B. Backend Development

The Express.js server configures CORS middleware allowing the frontend origin with credentials, cookie-parser for HttpOnly JWT cookies, and JSON body parser. The authentication controller uses bcryptjs with a cost factor of 12 for password hashing and signs

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 04-06-2026

International Journal of
AI ELECTRICAL CIVIL AND MECHANICAL ENGINEERING
Peer Reviewed, Referred & Indexed Journal
)70-0434 www.zesterapublications.com Original Research Paper

JWT tokens with 60-minute expiry. The product controller handles Cloudinary image upload by converting Multer memory buffer to base64 and uploading via the Cloudinary Node.js SDK.

C. API Design

The system exposes RESTful API endpoints returning JSON responses with consistent structure { success: boolean, data/message: value }. Key endpoint groups include: /api/auth/* for authentication, /api/shop/* for customer operations, and /api/admin/* for administrative operations. All sensitive routes are protected with JWT middleware.

VI. TESTING AND RESULTS

A. Testing Overview

The system underwent unit testing of individual controller functions, integration testing of frontend-backend-database communication, system testing through complete end-to-end user workflows, and browser compatibility testing across Chrome, Firefox, Edge, and Safari. Responsive design was verified from 320px to 1920px screen widths.

B. Functional Test Results

A total of 18 test cases were designed and executed covering registration, OTP verification, login, forgot password, product management, cart operations, payment, and logout. All 18 test cases passed successfully with no failures detected.

TC ID	Module	Test Description	Result
TC01	Registration	Register with valid credentials, OTP sent	Pass
TC02	Registration	Register with existing email, error shown	Pass
TC03	OTP Verify	Enter correct OTP within 10 minutes	Pass
TC04	OTP Verify	Enter incorrect OTP, error displayed	Pass
TC05	OTP Verify	Enter expired OTP after 10 minutes	Pass
TC06	Login	Login with correct credentials	Pass
TC07	Login	Login with unverified account	Pass
TC08	Login	Login with wrong password	Pass
TC09	Forgot Pwd	Request OTP with registered email	Pass
TC12	Product	Admin adds product with image upload	Pass
TC14	Cart	Add product to cart, count updated	Pass
TC17	Payment	Complete PayPal sandbox payment	Pass
TC18	Logout	Cookie cleared, redirect to login	Pass

Table I. Test Case Results

C. Performance Analysis

The system was analyzed for performance under various conditions. Frontend performance scored 92/100 on Chrome DevTools Lighthouse for desktop. First Contentful Paint was 0.8 seconds, well within the 1.8-second threshold. Largest Contentful Paint was 1.6 seconds within the good 2.5-second threshold.

Metric	Measured Value	Target
API Avg Response Time	< 150 ms	< 200 ms
Frontend FCP (Desktop)	0.8 sec	< 1.8 sec
Frontend LCP (Desktop)	1.6 sec	< 2.5 sec
Lighthouse Score	92/100	> 85/100
OTP Email Delivery	2-5 sec	< 10 sec
Image Upload Time	1.2 sec	< 3 sec
Concurrent Users	20+	> 10

Table II. Performance Analysis

VII. CONCLUSION

This paper successfully demonstrated the complete design, development, and testing of a full-stack E-Commerce website using the MERN Stack. The system provides a comprehensive digital shopping platform that addresses key limitations of traditional shopping methods and existing basic e-commerce solutions.

A secure OTP-based email verification system was implemented for user registration and password recovery. A fully functional admin panel supports product catalogue management with Cloudinary cloud image storage. The PayPal sandbox payment integration provides a realistic online checkout experience. The frontend built with React.js, Tailwind CSS, and ShadCN UI provides a clean, modern, and fully responsive design.

All 18 test cases passed and all performance metrics met or exceeded defined targets, confirming the system is well-optimized and ready for production deployment with appropriate server-side scaling.

Future Work

- AI-based product recommendation engine using collaborative filtering algorithms.
- React Native mobile application for Android and iOS platforms.
- Real-time order tracking using WebSockets (Socket.IO).
- Multi-vendor marketplace support with seller authentication and commission management.
- Cloud deployment with Docker containerization and CI/CD pipelines.
- Progressive Web App (PWA) support for offline browsing and push notifications.

REFERENCES

Received: 20-04-2026 | Accepted: 28-05-2026 | Published: 04-06-2026

International Journal of
AI ELECTRICAL CIVIL AND MECHANICAL ENGINEERING
Peer Reviewed, Referred & Indexed Journal
)70-0434 www.zesterapublications.com Original Research Paper

[1] K. C. Laudon and C. G. Traver, E-commerce: Business, Technology, Society, 15th ed. Pearson Education, 2019.

[2] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Univ. of California, Irvine, 2000.

[3] Payment Card Industry Security Standards Council, PCI DSS Requirements and Security Assessment Procedures, 2021.

[4] Y. Li, S. Miltchev, and J. Beck, "New research: How effective is basic account hygiene at preventing hijacking," Google Security Blog, 2019.

[5] React Documentation, Meta Open Source, 2024. [Online]. Available: https://reactjs.org/docs/

[6] Node.js Documentation, OpenJS Foundation, 2024. [Online]. Available: https://nodejs.org/en/docs/

[7] MongoDB Documentation, MongoDB Inc., 2024. [Online]. Available: https://www.mongodb.com/docs/manual/

[8] OWASP Foundation, OWASP Top Ten Web Application Security Risks, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/

[9] PayPal Developer Documentation, PayPal Holdings Inc., 2024. [Online]. Available: https://developer.paypal.com/api/rest/

[10] Cloudinary Documentation, Cloudinary Ltd., 2024. [Online]. Available: https://cloudinary.com/documentation/