IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15194

ShopEase: A MERN Stack Based E-Commerce Web Application with Rule-Based Chatbot Assistance

L M Veena [1], K Sharath [2]

Department of MCA, BIT, K.R. Road, V.V. Pura, Bangalore, India[1]

Assistant Professor, Department of MCA, BIT, K.R. Road, V.V. Pura, Bangalore, India[2]

Abstract: The rapid growth of online shopping has increased the demand for scalable, secure, and user-friendly e-commerce platforms. This paper presents ShopEase, a full-stack e-commerce web application developed using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The system enables users to browse products, manage shopping carts, and complete secure transactions through an integrated backend architecture. A key innovation of this project is the integration of a rule-based live chatbot, implemented using pure JavaScript, which provides real-time assistance to users by answering frequently asked questions, guiding navigation, and improving user engagement without relying on external AI services. Security is ensured using JSON Web Token (JWT) based authentication for safe user sessions and role-based access control. The results demonstrate that ShopEase offers an efficient, modular, and scalable solution suitable for modern e-commerce applications.

Keywords: E-Commerce, MERN Stack, React.js, Node.js, MongoDB, JWT Authentication, Rule-Based Chatbot

I. INTRODUCTION

E-commerce platforms play a vital role in the digital economy by providing convenient access to products and services. With increasing user expectations, modern e-commerce systems must ensure high performance, security, and ease of use. Traditional web applications often face challenges related to scalability, maintainability, and user support.

ShopEase is designed to address these challenges by adopting a modern full-stack architecture using the MERN stack. The project integrates frontend, backend, and database layers efficiently while introducing a chatbot system to enhance customer interaction. This application demonstrates how modern web technologies can be combined to create a robust and user-friendly online shopping platform.

1.1 Project Description

ShopEase is a web-based e-commerce application that allows users to register and log in securely, browse available products, add items to a shopping cart, and place orders. The system also includes an administrative module that enables authorized users to manage products and view customer orders.The application follows a client-server architecture where the frontend is developed using React.js, the backend is implemented using Node.js and Express.js, and MongoDB is used for data storage. Communication between the frontend and backend is achieved through RESTful APIs.

1.2 Motivation

The motivation behind developing ShopEase is to gain hands-on experience in building a real-world full-stack web application while understanding how frontend, backend, and database systems work together. Another important motivation is to enhance user experience by providing instant assistance through a chatbot, reducing the need for manual customer support and improving usability.

II. RELATED WORK

Paper[1], Presents the design and implementation of a web-based e-commerce application developed using the MERN stack. The study focuses on the use of React.js for creating interactive and responsive user interfaces, demonstrating how component-based architecture improves user experience and application maintainability.

Paper [2], Explores the role of Node.js and Express.js in building scalable and efficient backend services for e-commerce platforms. The authors highlight the importance of RESTful APIs in managing user authentication, product listings, shopping cart operations, and order processing while ensuring smooth communication between frontend and backend layers.

DOI: 10.17148/IJARCCE.2026.15194

Paper [3], Discusses the use of MongoDB as a NoSQL database for storing and managing e-commerce data. The study emphasizes the flexibility of document-oriented databases in handling dynamic data such as user profiles, product catalogs, and order histories, which is essential for modern online shopping applications.

Paper [4], Reviews security mechanisms adopted in web applications, with particular focus on JSON Web Token (JWT) based authentication. The paper explains how JWT improves security by enabling stateless authentication, protecting API endpoints, and ensuring secure access control in e-commerce systems.

Paper [5]**,** Examines the integration of rule-based chatbot systems in web applications to enhance user interaction and support. The authors highlight how chatbot-based assistance helps users navigate platforms, resolve common queries, and improve overall user engagement without relying on complex artificial intelligence models.

III. METHODOLOGY

A. Overall Architecture

The ShopEase application follows a three-tier architecture consisting of the presentation layer, application layer, and data layer. The frontend is developed using React.js, the backend is implemented using Node.js with Express.js, and MongoDB is used as the database. In addition to these layers, a rule-based chatbot module is integrated to provide real-time user assistance and improve interaction

B. .Frontend Design

React.js is used to create a responsive and interactive user interface. Components are designed for product listing, shopping cart, checkout process, user authentication, and chatbot interaction. State management techniques ensure real-time updates and smooth navigation across the application.

C. Backend Implementation

The backend is developed using Node.js and Express.js, which handle API requests, business logic, and authentication. RESTful APIs are implemented to manage users, products, shopping carts, and orders. JSON Web Tokens (JWT) are used to secure user sessions. Backend logic also supports chatbot request handling based on predefined rules.

D. Database Management

MongoDB is used as a NoSQL database to store user information, product details, order records, and chatbot-related interaction data if required. Its document-based structure allows flexible schema design and efficient data retrieval.

E. Chatbot Design and Functionality

A rule-based chatbot is implemented using JavaScript to provide instant responses to user queries. The chatbot operates on predefined conditions and keyword matching to assist users with navigation, product inquiries, and common questions. This approach improves user engagement without relying on external artificial intelligence services.

F. Payment Integration

The system integrates online payment functionality using a secure payment gateway in sandbox mode. This enables users to complete transactions safely while maintaining data confidentiality.

DOI: 10.17148/IJARCCE.2026.15194

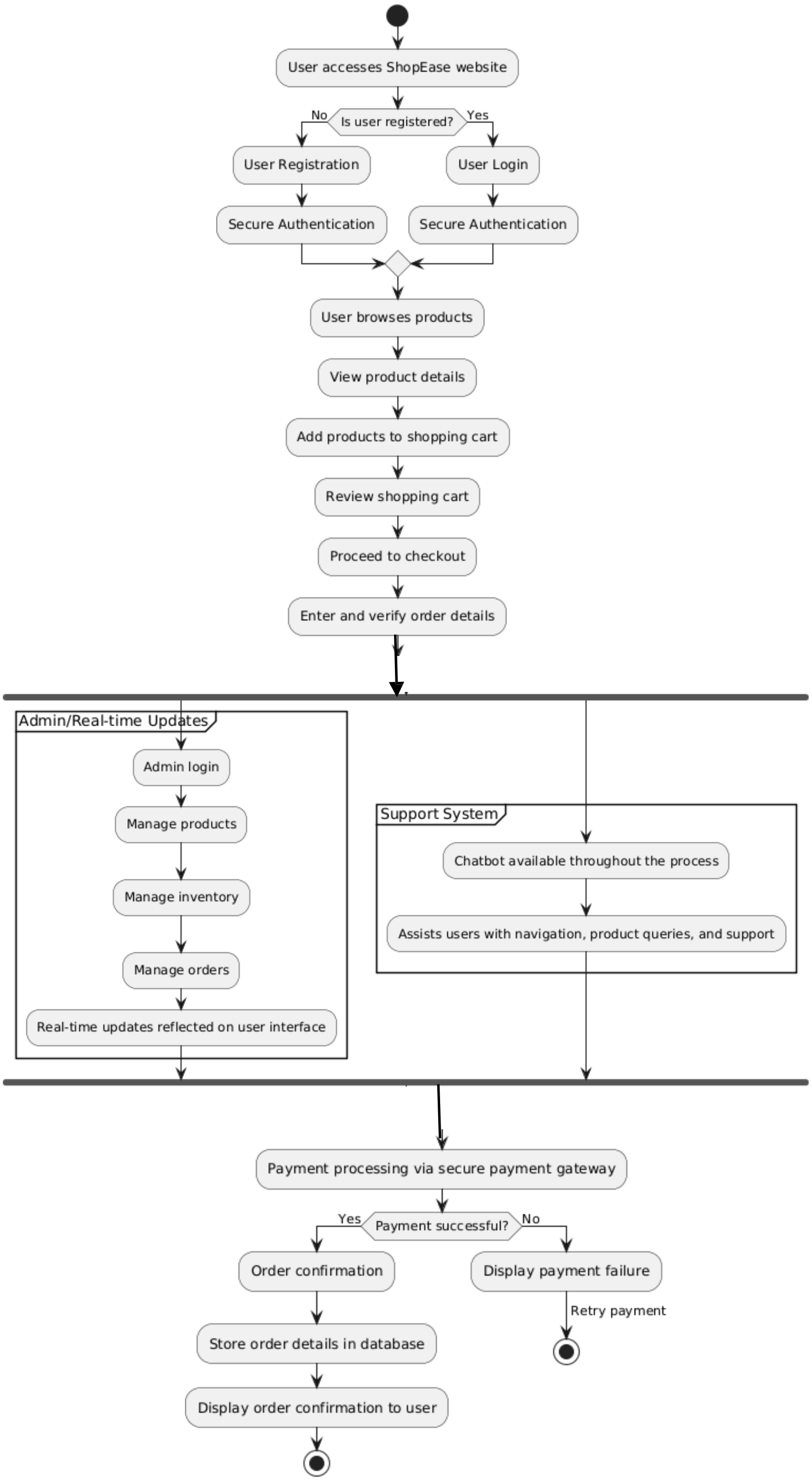

User accesses ShopEase website
Is user registered?
No
Yes
User Registration
User Login
Secure Authentication
Secure Authentication
User browses products
View product details
Add products to shopping cart
Review shopping cart
Proceed to checkout
Enter and verify order details
Admin/Real-time Updates
Admin login
Manage products
Manage inventory
Manage orders
Real-time updates reflected on user interface
Support System
Chatbot available throughout the process
Assists users with navigation, product queries, and support
Payment processing via secure payment gateway
Payment successful?
Yes
No
Order confirmation
Display payment failure
Retry payment
Store order details in database
Display order confirmation to user

Fig.1. Flowchart of methodology

G. Implementation Flow

1. Load the ShopEase web application, including frontend components developed using React.js and backend services implemented using Node.js and Express.js.
2. Initialize required modules and configurations such as user authentication, product management, shopping cart functionality, chatbot interaction, and database connectivity.
3. At each user interaction:

DOI: 10.17148/IJARCCE.2026.15194

o Render dynamic user interface components such as product listings, product details, cart view, and checkout pages using React.js.
o Authenticate users using JSON Web Tokens (JWT) to ensure secure access to protected features.
o Handle API requests through RESTful services for user authentication, product retrieval, cart operations, and order processing.
o Enable the rule-based chatbot to assist users with navigation, product-related queries, and support throughout the shopping process.
o Validate user inputs and update application state in real time.

4. During checkout:

o Process payment requests through a secure payment gateway operating in sandbox mode.
o Verify transaction status and generate order confirmation upon successful payment.
o Store order details, user information, and transaction records in the MongoDB database.

5. After order completion:

o Display order confirmation details to the user.
o Allow administrators to manage products, inventory, and orders through the admin interface.
o Reflect real-time updates on the user interface based on backend changes.

H. Hardware and Software Requirements

- Hardware: Standard desktop or laptop computer with at least 8GB RAM, quad-core processor, and stable internet connectivity.
- Software: Windows 10 or later operating system, Visual Studio Code or any modern code editor, Node.js (v16 or later), MongoDB (local or MongoDB Atlas), web browser (Google Chrome or Mozilla Firefox), and Git/GitHub for version control.

IV. SYSTEM DESIGN AND IMPLEMENTATION FRAMEWORK

This section describes the overall system design, implementation process, and evaluation strategy adopted for the ShopEase e-commerce application. The system is developed using the MERN stack, which integrates React.js for frontend development, Node.js with Express.js for backend services, and MongoDB for database management. The application follows a client-server architecture and is designed to ensure secure, scalable, and user-friendly online shopping.

A. System Architecture and Workflow

The architecture of ShopEase is designed to support efficient interaction between users, administrators, and system services. The major components of the system are described below:

React.js Frontend:
The frontend serves as the presentation layer and is responsible for rendering user interfaces such as product listings, shopping cart, checkout pages, user authentication screens, and chatbot interaction. It ensures a responsive and interactive user experience.

Node.js and Express.js Backend:
The backend handles application logic, authentication, and communication between the frontend and database. RESTful APIs are used to manage users, products, carts, orders, and payments.

MongoDB Database:
MongoDB is used as a NoSQL database to store user details, product information, order records, and transaction data. Its flexible schema design supports dynamic e-commerce data management.

JWT-Based Authentication:
JSON Web Tokens (JWT) are used to authenticate users and authorize access to protected resources, ensuring secure session management.
Rule-Based Chatbot Module:
A rule-based chatbot is integrated into the system to provide real-time assistance to users. The chatbot responds to

DOI: 10.17148/IJARCCE.2026.15194

predefined user queries related to navigation, product information, and support, improving user engagement.

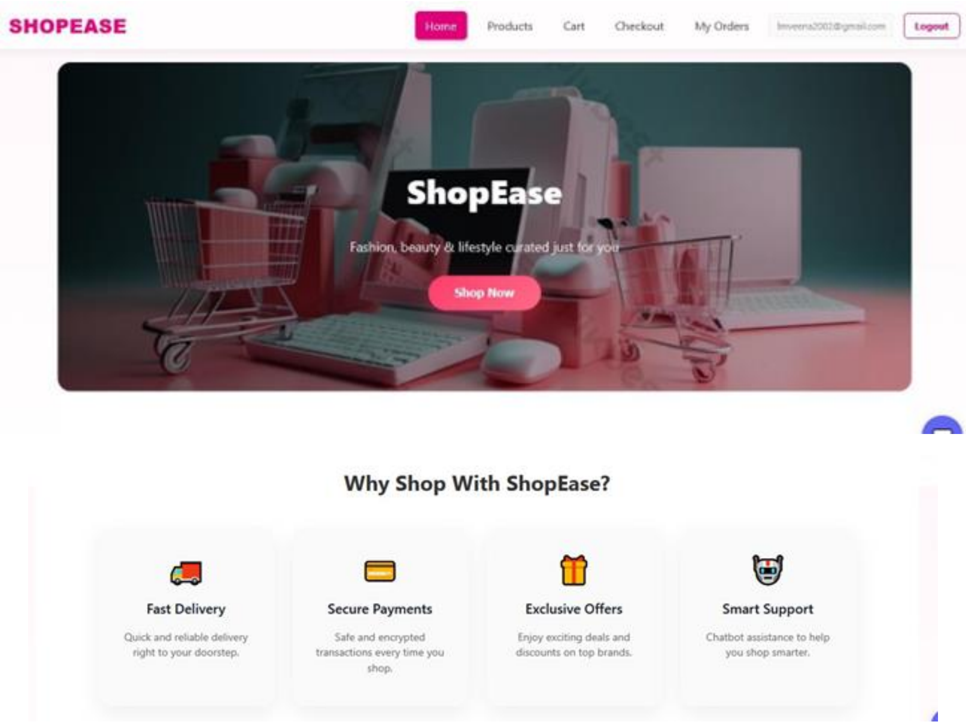
B. Application Setup and Execution

The ShopEase application is executed as a single-page application. Users access the system through a web browser and interact with the frontend, which communicates with the backend APIs. The backend processes requests, performs authentication and validation, and interacts with the database to retrieve or store data.

C. Evaluation Strategy

The system is evaluated based on functional correctness, user experience, and performance. Key evaluation parameters include secure user authentication, smooth navigation, accurate product and order management, chatbot responsiveness, and successful payment processing. Testing confirms that the system operates efficiently under normal usage conditions and provides a seamless shopping experience.

*D.*Results



Why Shop With ShopEase?

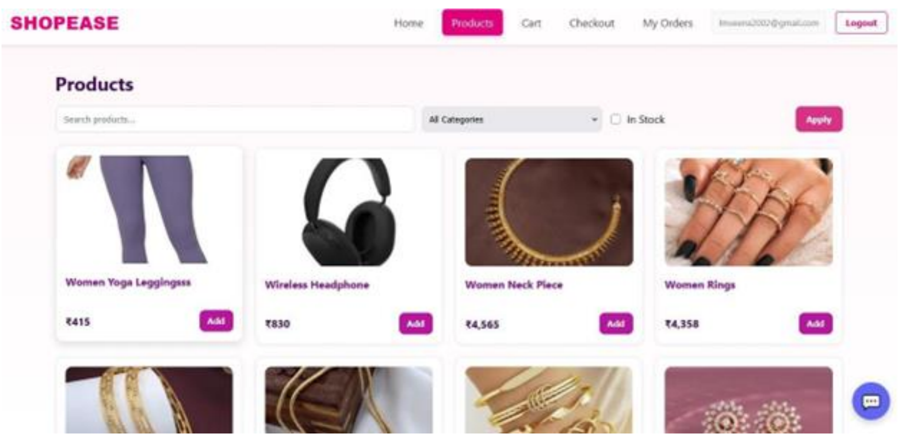
Fast Delivery
Quick and reliable delivery right to your doorstep.

Secure Payments
Safe and encrypted transactions every time you shop.

Exclusive Offers
Enjoy exciting deals and discounts on top brands.

Smart Support
Chatbot assistance to help you shop smarter.

Fig.2.Home Page

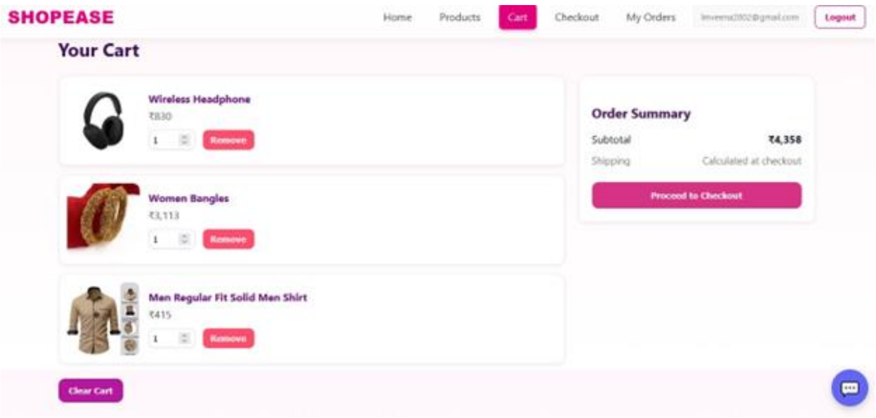


Products

Product	Price
Women Yoga Leggingsss	₹415
Wireless Headphone	₹830
Women Neck Piece	₹4,565
Women Rings	₹4,358

Fig.3.Product Page

DOI: 10.17148/IJARCCE.2026.15194



Your Cart

Wireless Headphone
₹830
1

Women Bangles
₹3,113
1

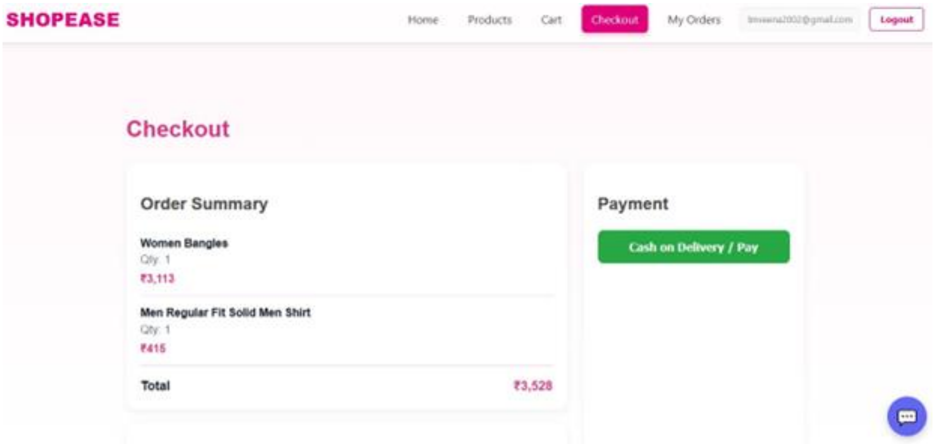
Men Regular Fit Solid Men Shirt
₹415
1

Order Summary

Subtotal ₹4,358

Shipping Calculated at checkout

Fig.4.Cart Page



Checkout

Order Summary

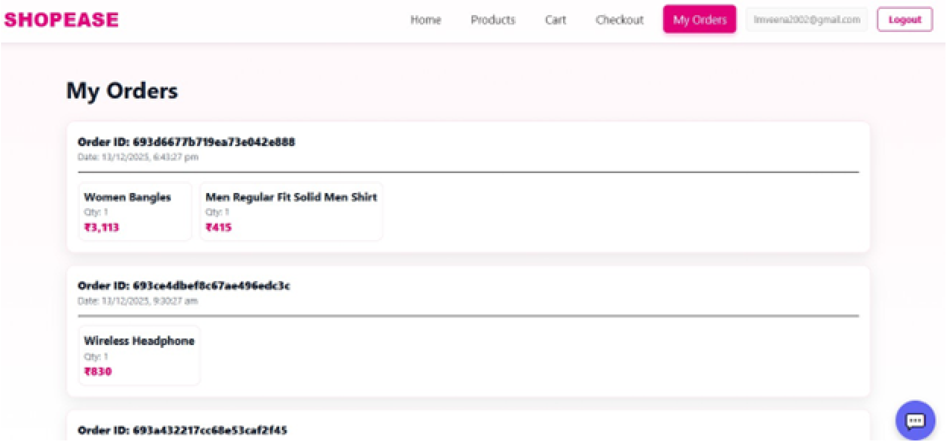
Women Bangles
Qty: 1
₹3,113

Men Regular Fit Solid Men Shirt
Qty: 1
₹415

Total ₹3,528

Payment

Fig.5.Checkout Page



My Orders

Order ID: 693d6677b719ea73e042e888
Date: 13/12/2025, 6:43:27 pm

Women Bangles
Qty: 1
₹3,113

Men Regular Fit Solid Men Shirt
Qty: 1
₹415

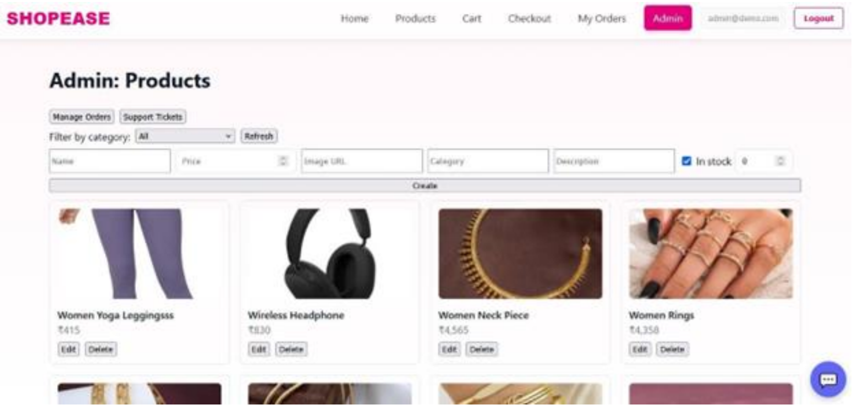
Order ID: 693ce4dbef8c67ae496edc3c
Date: 13/12/2025, 9:30:27 am

Wireless Headphone
Qty: 1
₹830

Order ID: 693a432217cc68e53caf2f45

Fig.6.MyOrders Page

DOI: 10.17148/IJARCCE.2026.15194



Admin: Products

Filter by category: All

Name | Price | Image URL | Category | Description | ☑ In stock | 0

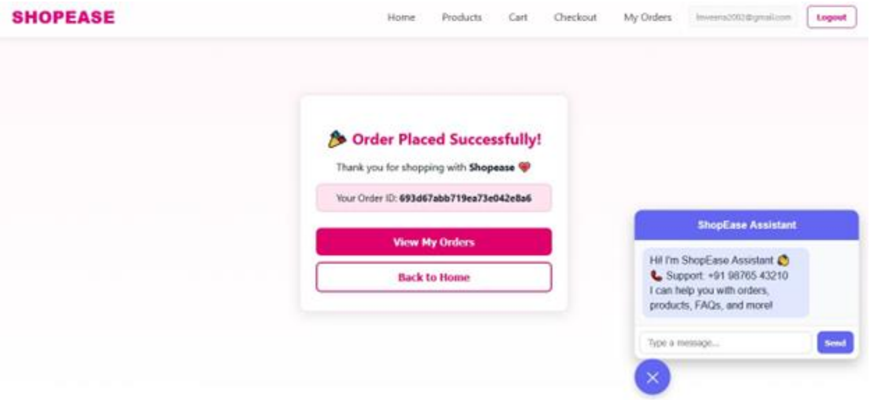
Women Yoga Leggingsss
₹415

Wireless Headphone
₹830

Women Neck Piece
₹4,565

Women Rings
₹4,358

Fig.7.Admin Page



Order Placed Successfully!

Thank you for shopping with **Shopease**

Your Order ID: **693d67abb719ea73e042e8a6**

Hi! I'm ShopEase Assistant
☎ Support: +91 98765 43210
I can help you with orders, products, FAQs, and more!

Type a message...

Fig.8.Order & Chatbot page

V.RESULTS AND DISCUSSION

The implementation and testing of the ShopEase e-commerce web application demonstrate that the system effectively supports core online shopping functionalities with secure and smooth user interaction. The application was tested under normal usage scenarios to evaluate authentication, product management, order processing, payment handling, and chatbot performance.

The results show that the React.js frontend provides responsive and dynamic user interfaces, enabling users to browse products, manage shopping carts, and complete checkout operations efficiently. Integration with the Node.js and Express.js backend through RESTful APIs ensured reliable communication between the frontend and server-side logic. JWT-based authentication successfully secured user sessions and restricted access to protected features.

The rule-based chatbot proved effective in assisting users with navigation, product-related queries, and common support questions. During testing, the chatbot responded instantly to predefined user inputs, reducing user effort and improving overall engagement. This feature contributed positively to user experience without introducing additional system complexity.

MongoDB efficiently handled data storage and retrieval for users, products, and orders. The system maintained data consistency during concurrent operations such as multiple users browsing products and placing orders. Payment transactions processed through the sandbox payment gateway were completed successfully, and order details were accurately stored and displayed.

DOI: 10.17148/IJARCCE.2026.15194

Overall, the evaluation confirms that ShopEase operates as a reliable, secure, and user-friendly e-commerce platform. The modular MERN stack architecture supports scalability and future enhancement, while the chatbot serves as a valuable innovation that enhances usability.

VI. CONCLUSION

This paper presented ShopEase, a MERN stack based e-commerce web application that provides a secure and user-friendly online shopping experience. The system integrates React.js, Node.js with Express.js, and MongoDB to demonstrate effective full-stack web application development.The application supports core e-commerce functionalities such as user authentication, product browsing, cart management, order processing, and secure payment handling. JSON Web Tokens (JWT) are used to ensure secure user sessions, while RESTful APIs enable smooth communication between system components.A key innovation of ShopEase is the integration of a rule-based chatbot that assists users with navigation and product-related queries, improving usability and user interaction without relying on complex artificial intelligence techniques.Overall, the project demonstrates the suitability of the MERN stack for building scalable and reliable e-commerce applications with enhanced user experience.

VII. FUTURE WORK

Although ShopEase meets the current functional requirements of an e-commerce application, there are several opportunities for future enhancement. One major improvement involves integrating artificial intelligence and natural language processing (NLP) techniques into the chatbot to enable more intelligent, conversational, and context-aware user interactions.Future versions of the application may include personalized product recommendation **sys**tems based on user behavior and purchase history, which can further improve customer engagement. Deploying the application on cloud platforms such as AWS or Azure can enhance scalability, availability, and performance.Additional enhancements may involve implementing server-side rendering (SSR) to improve page load times and search engine optimization (SEO). Features such as real-time order tracking, advanced analytics for administrators, multi-language support, and mobile application integration can also be incorporated to extend the functionality and reach of the system.

REFERENCES

[1] M. Grinberg, *Full Stack Web Development with React, Node.js, and MongoDB*, Packt Publishing, 2021.

[2] Facebook Inc., “React – A JavaScript library for building user interfaces,” [Online]. Available: https://react.dev

[3] OpenJS Foundation, “Node.js Documentation,” [Online]. Available: https://nodejs.org

[4] Express.js Foundation, “Express – Fast, unopinionated, minimalist web framework for Node.js,” [Online]. Available: https://expressjs.com

[5] MongoDB Inc., “MongoDB Documentation,” [Online]. Available: https://www.mongodb.com/docs

[6] K. Pan, “RESTful API Design Best Practices,” *International Journal of Web Engineering*, vol. 9, no. 2, pp. 45–52, 2020.

[7] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),” *RFC 7519*, Internet Engineering Task Force (IETF), 2015.

[8] A. Kumar and S. Sharma, “Design and Implementation of Secure E-Commerce Applications Using MERN Stack,” *International Journal of Computer Applications*, vol. 182, no. 30, pp. 15–21, 2019.

[9] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, University of California, Irvine, 2000.

[10] J. Nielsen, “Usability Engineering for Web Applications,” *IEEE Computer*, vol. 33, no. 1, pp. 75–82, 2018.

[11] S. Shaw and M. Carter, “Rule-Based Chatbot Design for Web Applications,” *International Journal of Artificial Intelligence Applications*, vol. 12, no. 3, pp. 23–30, 2021.

[12] PayPal Inc., “PayPal Developer Documentation,” [Online]. Available: https://developer.paypal.com