VaaRee

[Vit Kathiritya]

Department of Computer Science and Engineering
[Parul Institute Of Technology], [Vadodara, India]
[2203051050789l@paruluniversity.ac.in]

How to Cite this Article:

Kathiritya, V. (2026). VaaRee. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.662

https://doi.org/10.55041/ijcope.v2i4.662

Abstract

With the rapid growth of online shopping, developing a robust and scalable platform has become essential for modern businesses. This project focuses on building a dynamic and user-friendly e-commerce website using the MERN Stack (MongoDB, Express.js, React.js, Node.js). The platform provides a seamless online shopping experience, allowing users to browse products, add them to their cart, and securely complete purchases. interface. The frontend, built with React.js, ensures an interactive and responsive user The backend, powered by Node.js and Express.js, efficiently manages user authentication, order processing, and business logic. MongoDB, a NoSQL database, is used to store product details, user information, and transaction records. Authentication is secured using JWT (JSON Web Token), and payments are integrated using payment gateways like Stripe or PayPal. Key features include user authentication, product search and filtering, order management, secure payment processing, and admin dashboard for inventory management. The platform follows RESTful principles and is optimized for scalability, performance, and security. API This project highlights the efficiency of the MERN Stack in developing full-stack web applications and demonstrates its potential in creating modern, feature-rich e-commerce solutions.

1. Introduction

E-commerce has revolutionized the way businesses operate by enabling online digital buying and selling of products. With the increasing demand for shopping experiences, building a robust and scalable e-commerce platform is essential. This project focuses on developing a full-stack e- commerce website using the MEAN Stack (MongoDB, Express.js, Angular, Node.js). The platform provides an intuitive user interface, a secure authentication system, product integration. By management, leveraging the order tracking, capabilities of and seamless payment MongoDB for database management, Express.js for server-side operations, Angular for a dynamic frontend, and Node.js for backend processing, the system ensures high performance, scalability, and efficiency. Traditional retail businesses face several challenges, including limited reach, inventory mismanagement, and high operational costs. Customers also experience inconveniences such as time-consuming shopping, product unavailability, and restricted payment options. The primary issues include: Lack of an easily accessible and user-friendly online shopping platform. Difficulty in effectively. managing product inventory, orders, and customer data Security concerns related to user authentication and payment transactions. Inability to provide a personalized shopping experience to customers. To address these challenges, this project aims to develop a fully functional and feature-rich e-commerce platform that provides a seamless shopping experience for users while ensuring efficient management for administrators.

literature survey is crucial in understanding the existing technologies, frameworks, and methodologies used in e-commerce website development. This explores the background of e-commerce platforms, the advantages of using the MEAN stack, and how similar technologies have been implemented in the industry. Although the MEAN stack traditionally includes React.js Angular, due to modern its implementations often replace Angular with flexibility and performance. This utilizes MongoDB, Express.js, Node.js, and React.js as the primary technology stack.

2. Related Work

Existing E-Commerce Solutions

Several e-commerce platforms have been developed using various technologies, including monolithic architectures and microservices-based solutions. Some of the widely used platforms include: Shopify — A hosted e-commerce solution with a built-in CMS. Magento — A PHP-based open-source e-commerce platform. WooCommerce functionality. Custom lack Solutions platforms — A WordPress plugin that enables e-commerce (MERN/MEAN-based) — Many businesses develop using full-stack JavaScript frameworks for better performance and scalability. While these platforms provide essential e-commerce functionalities, they may customization flexibility, performance optimization, and full control over business logic, making custom development a preferred choice for scalable applications.

MEAN Stack for E-Commerce Development

The MEAN Stack (MongoDB, Express.js, Angular/React.js, and Node.js) is widely used for modern web applications, including e-commerce systems. Its advantages include full-stack JavaScript development, high performance, and scalability. Each component of the stack plays a crucial role.

2.1Frontend (React.js)

React.js is a component-based frontend library used to build dynamic user interfaces. Although Angular is traditionally used in the MEAN stack, React.js is Case often preferred due to its performance, reusability, and SEO-friendliness. Advantage: Virtual DOM for fast rendering, reusable components, and state management using Redux or Context APL Use in E-Commerce: Provides an interactive shopping experience with features like product filters, cart updates, and order history.

2.2Backend (Node.js + Express.js)

Express.js is a lightweight Nodejs web framework that simplifies API development and server-side logic. Advantage: Middleware support, routing capabilities, and REST API creation. Use Case in E-Commerce: Handles user authentication, order processing, and business logic.

Node.js is a JavaScript runtime built on Chrome's developers to execute JavaScript on the server side. V8 engine, allowing Advantage: Non-blocking /O operations, high concurrency, and performance efficiency. Use Case in E-Commerce: Manages server-side operations, real-time updates, and order processing.

2.3Database (MongoDB)

MongoDB is a NoSQL document-oriented database that stores data in JSON-like BSON format. It is highly scalable and suitable for managing product catalogues, user data, and order details. Advantage: Flexible schema, fast read/write operations, and easy scalability. Use Case in E-Commerce: Stores product listings, customer orders, and user authentication details.

3. Prompt Engineering and Constraint Design

The development of an E-Commerce Website using the MERN Stack (MongoDB, Express.js, Angular, Node.js) follows an agile methodology to ensure scalability, performance, and security. The approach includes structured data gathering, backend and frontend development, and security integration to provide an efficient online shopping experience. The methodology is divided into multiple stages, including data processing, misinformation categorization (for product authenticity workflow management. checks), result interpretation, frontend integration, and.

Data Gathering

Data collection is a crucial step in developing an efficient e-commerce platform. The gathered data includes: « e « * Product Data: Collected from manufacturers, suppliers, and user-generated content. User Data: Includes transaction history. registration details, shopping preferences, Transaction Data: Tracks orders, payments, and shipping details. and Reviews & Ratings: Helps analyze user feedback and improve product recommendations. Sources of Data: Direct database population by admin or vendors. User-generated data like product reviews, queries, and feedback. APIs for fetching external product details, currency exchange rates, and payment transactions. 3.3 Text Preprocessing Text preprocessing ensures that product descriptions, reviews, and user-generated content are filtered and structured for better search functionality and analysis. '

It includes: »

Tokenization: individual words. A Breaking down product descriptions and reviews into Stopword Removal: Removing common words (e.g., “and,” “the”) to improve search accuracy. Stemming & Lemmatization: Converting words to their root forms (e.g., "running" — "run"). Sentiment Analysis: Categorizing reviews into positive, neutral, or negative sentiments for fraud detection and product authenticity checks.

Project Platforms Used in Development

The is including: developed using modern development tools and platforms, Frontend Technologies:

Angular — Dynamic Ul rendering and component-based architecture.

HTML, CSS, Bootstrap/Tailwind CSS — Styling and responsiveness.

RxJS & Angular Services — State management and asynchronous operations. Backend Technologies:

Node.js & Express.js — Server-side logic and API development.

MongoDB — NoSQL database for storing user and product data.

JWT Authentication — Secure login and authorization. RESTful API — Communication between frontend and backend.

4. Psychological Framework

Raahi's design is grounded in three established psychological frameworks that have been adapted for AI implementation.

4.1Person-Cantered Therapy (PCT)

Carl Rogers proposed that the therapeutic relationship — characterized by unconditional positive regard, empathic understanding, and congruence — creates the conditions under which individuals naturally move toward growth and self-actualization. Raahi operationalizes PCT's core conditions: it maintains a consistently non-judgmental stance (unconditional positive regard), mirrors the user's internal frame of reference (empathic understanding), and avoids performing emotional states it is not expressing (a form of congruence appropriate to AI systems).

4.2Motivational Interviewing (MI)

Motivational Interviewing, developed by Miller and Rollnick, is a goal-oriented counselling style that elicits and strengthens a person's own motivation for change. Its core techniques — open-ended questions, affirmations, reflective listening, and summarizing (OARS) — map directly onto Raahi's conversational constraints. Raahi's exclusive use of open-ended questions (Constraint C3) and its mirroring of user language (Constraint C4) are direct implementations of MI's OARS framework.

4.3Socratic Questioning

The Socratic method uses systematic questioning to help individuals examine their assumptions and reach deeper understanding through their own reasoning. Raahi applies Socratic questioning not in an educational or dialectical context, but in a psychological one — using questions to invite users to examine the assumptions embedded in their daily language about concepts like love, success, identity, and belonging.

5. Evaluation

Evaluating a non-directive reflective AI system presents unique methodological challenges, as standard chatbot evaluation metrics (task completion rate, answer accuracy, BLEU score) are explicitly inapplicable — Raahi's goal is not to complete tasks or provide correct answers.

5.1Constraint Adherence Evaluation

We propose evaluating constraint adherence through expert annotation of conversation transcripts. Annotators assess each Raahi response against the seven constraints in Table 1 using a binary (compliant / non-compliant) rating. Inter-annotator agreement is measured using Cohen's Kappa. Initial testing indicates high constraint adherence rates, particularly for constraints C1 (no advice) and C2 (no prescriptions), which are the most critical to the system's identity.

5.2Reflective Quality Assessment

A qualitative evaluation framework is proposed in which participants engage with Raahi for a fixed session duration (15–20 minutes) and then complete a post-session questionnaire assessing: (1) the degree to which they felt heard and understood; (2) whether the conversation prompted new personal insights; (3) perceived judgment or prescription from the system. A Likert scale instrument adapted from validated therapeutic alliance measures is used.

5.3Comparative Evaluation

A comparative study design is proposed in which participants are randomly assigned to interact with either Raahi or a standard solution-oriented chatbot (e.g., baseline GPT-4 with no system prompt constraints) following the same initial

prompt. Post-session measurements of self-reported insight, emotional clarity, and perceived autonomy are compared across conditions. This evaluation is identified as a primary direction for future work.

6. Discussion

The development of the E-Commerce Website using the MERN (MongoDB, Express.js, Node.js) stack is expected to achieve the following outcomes:

6.1Fully Functional E-Commerce Platform

A user-friendly, responsive, and dynamic shopping website the Smooth browsing, searching, filtering, and purchasing experience .Integration of shopping cart and secure checkout process.

Secure & Scalable Backend

Node.js & Express.js-based RESTful API to handle user requests. MongoDB for scalable and efficient data management. JWT authentication & role-based access control (RBAC) for security.

Robust User Management System

User Registration/Login (JWT Authentication) for secure access. Admin Dashboard to manage users, products, orders, and reports. Role-based access (Admin, Customer) with different privileges.

Performance Optimization & Scalability

Optimized database queries for faster data retrieval. Lazy loading & caching mechanisms for improved speed. Scalability for handling increased traffic & products.

Mobile-Responsive Ul

Adaptive Ul frameworks. for desktop, tablet, and Fast loading and optimized performance.

Deployment & Cloud Integration

Hosting on cloud platforms (AWS, Firebase, Digital Ocean, or Heroku).CI/CD pipeline for automated deployment & updates.

Real-Time Notifications & Communication

Email & SMS alerts for order confirmation & status updates. Live chat support for customer queries (optional).

Analytics & Reporting

Admin dashboard with sales reports, revenue insights, and user analytics. CSS

6.2Limitations

Performance Challenges at Scale

MERN apps (especially using React and Node.js) can struggle under heavy traffic if not optimized.

Node.js is single-threaded → can bottleneck with CPU-heavy tasks

Large product catalogs or many users → slower response if queries aren't optimized

SEO Limitations

React-based frontends often rely on client-side rendering.

Search engines may not fully index dynamic pages

Needs extra setup like server-side rendering (SSR) or frameworks (e.g., Next.js)

Security Concerns

MERN requires careful handling of security.

Vulnerable to attacks like XSS, CSRF, injection if poorly coded

Payment and user data protection needs strong backend practices

Database Limitations (MongoDB)

Not ideal for highly relational data (orders, users, payments)

Complex joins are harder compared to SQL databases

Maintenance & Debugging Difficulty

Full-stack JavaScript means bugs can exist anywhere (frontend + backend)

Harder to debug when application grows large

7. Future Work

The project has the potential for further enhancement and expansion to improve user engagement, performance, and business opportunities. Below are some future improvements:

Al-Based Product Recommendation System Implementing machine learning algorithms to analyse user behaviour and provide personalized product recommendations. Utilizing Al-driven chatbots for customer support and queries. Integration with Multiple Payment Gateways Adding more payment methods like cryptocurrency, UPI, and digital wallets for broader customer reach.

Multi-Vendor Marketplace Support Expanding the platform into a multi-vendor e-commerce system where multiple sellers can list their products. Implementing commission-based selling models for vendors. Advanced Data Analytics & Business Intelligence Implementing real-time data analytics dashboards for better business insights. Al-powered sales forecasting for inventory and revenue predictions. Subscription-Based Services Introducing subscription models for premium users, offering discounts, early access, and personalized deals. Voice Search & Smart Assistants Integrating voice search functionality to enhance accessibility. Connecting with Al-powered assistants (Alexa, Google Assistant, Siri) for hands-free shopping. Global Expansion with Multi-Language & Multi-Currency Support Enabling multi-language support for better localization. Dynamic currency conversion for international customers.

8. Conclusion

The development of an E-Commerce Website using the MERN (MongoDB, Express.js, Node.js) Stack has successfully provided scalable, efficient, and user-friendly platform for online shopping. The system integrates secure user authentication, product management, cart a functionality, order processing, and payment gateway integration, ensuring a

seamless shopping experience. By leveraging the MERN stack, the platform benefits from: Fast and scalable backend operations with Node.js & Express.js. Efficient database management with MongoDB for handling large datasets. Dynamic and responsive UI with Angular for an enhanced user experience. Seamless API communication between frontend and backend. Security features such as JWT authentication and role-based access control. The project demonstrates how modern web technologies can be combined to build a fully functional, high-performance e-commerce platform that meets the demands of today's digital economy.

References

[1] Amazon
Best for: search, recommendations, reviews, fast checkout

Conversion-focused

[2] Apple

Minimal design, premium feel, product storytelling

Brand stoytelling

[3] Mynta
Strong visuals, smooth filtering, mobile

Fashion discovery

[4] Shopify → easiest, beginner-friendly
[5] WooCommerce → flexible, low cost
[6] Magento → powerful but complex
[7] BigCommerce → scalable