Journal of Integrated
SCIENCE & TECHNOLOGY

Harnessing the MERN stack for scalable E-commerce website design: A full-stack approach with MongoDB, Node.js, Express.js, and React.js

Aparna Patil, Mangal Patil,* Sharada Tondare, Prajakta Naregalkar, Anuradha Nigade, Sonali Pawar, Jyoti Morbale

Department of Electronics & communication Engineering, Bharati Vidyapeeth(Deemed to be University) College of Engineering, Pune, Maharashtra, India.

| Article

ABSTRACT

In today's digital era, global internet usage is rising, driving a surge in online transactions and making e-commerce a dominant market force. Consumers increasingly prefer online shopping for its convenience and accessibility. This paper explores the development of a comprehensive e-commerce platform using the MERN stack: MongoDB, Express.js, React.js, and Node.js is one of the most robust technology stacks for full-stack web applications. Our design presents a feature-rich e-commerce platform offering a seamless user experience, fast loading, multiple payment options, and high-resolution visuals. Users can effortlessly browse, select, and purchase products. Key features include a shopping cart, wish list, integrated payment gateway, and distinct user and admin interfaces for efficient product, category, and order management. By leveraging the MERN stack, we demonstrate its potential in building scalable, secure, and high-performance e-commerce platforms. This research systematically reviews existing literature on technological implementations, feature integration, performance evaluation, and future advancements in e-commerce development. We analyse prior studies to showcase how the MERN stack effectively addresses challenges like scalability, security, and user experience. Our findings underscore its role as a vital enabler in e-commerce, offering valuable insights for researchers, developers, and industry professionals exploring innovative solutions in online retail technology.

Keywords: MERN Stack, Express.js, Node.js, MongoDB, React.js, Digital Transformation, Framework, E-Commerce, Online shopping.

1. INTRODUCTION

With the rapid development of e-commerce as more businesses move their operations online, the e-commerce sector has become more competitive, pushing companies to look for innovative ways to provide their clients with a more efficient, secure, exclusive, and fascinating shopping experience[1,2,3.] Recently, online platforms are growing into popular options for buying and trading goods and services, but their ease of use, assurance, and scalability remain hampered, requiring users to create separate accounts and manage different payment systems, which can be complicated and time-consuming[4,5]. In order to get over these challenges, we developed an e-commerce platform utilizing MERN stack technology with the aim of delivering easier-to-use, more secure, and better-performing web applications that meet evolving business and customer needs[6,7,8.]. Fig.1 illustrates the significant rise in e-commerce usage over the past few years[9,10].

Our web platform uses the following two entities: There will be a registered user and an administrator present. The administrator has access to order details, product sales tracking, modification of existing products, uploading of new things, and removal of products that aren't appropriate. The other entity is the user interface[11]. Here, logged-in customers can browse and purchase generic products by applying various filters, and they can complete transactions by choosing the best possible payment method. We also included voice search, allowing users to effortlessly navigate through products using voice commands, catering to a diverse range of consumer choices. Moreover, the direct messaging capabilities facilitate seamless communication between sellers and buyers, fostering trust and personalized interactions.

The e-commerce website is designed with the MVC (Model-View-Controller) structure, which is an effective design paradigm in software development[12]. The main element of MVC is the model, which maintains information about the application and its business

*Corresponding Author: Mangal Patil
Tel: 9226418220; mvpatil@bvucoep.edu.in

reasoning distinct from the user interface. By collecting user input and transforming it into commands for the system, the controller functions as an interface between the model and the system.

We organised a research paper as follows: Sec. 2 describes the literature review on related works. Sec. 3 discusses the proposed design. Sec. 4 illustrates the proposed methodology for web design. Sec. 5 describes MVC architectural pattern. The implementation and experimental results are explained in Sec.6 followed by the conclusion and future scope in Sec.7. The research will focus on the usability, security, and scalability of the platform, drawing on relevant literature and best practices in software development to inform our evaluation. Ultimately, our goal is to provide a more efficient and user-friendly e-commerce platform that can benefit both buyers and sellers in the online marketplace.
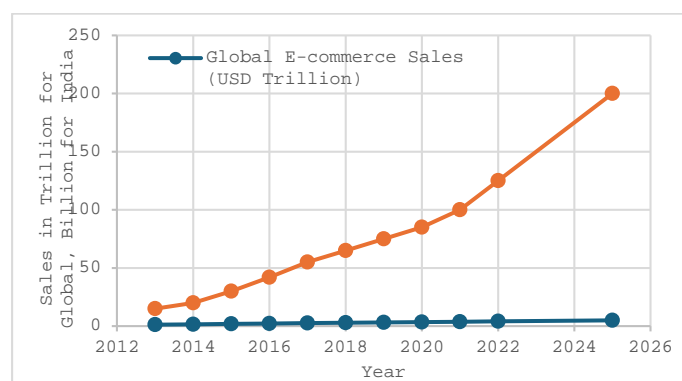
Figure 1. E-Commerce Growth Global vs India – Past 10 Year and Forecast.

RELATED WORK

The MERN stack technology has recently become fairly popular in the development of the e-commerce arena. Judging by the amount of effort required in building an e-commerce web store using it, Over the past few years, many different research studies and other works have been published, covering an array of topics associated with e-commerce development. The approaches, technologies, resources, and results have been covered in detail in the documents.

M.M. V. Sai Nikhil and Aniket Sarrin's study aims to design the backend of an e-commerce application, using diagrams such as flowcharts and data flow diagrams[13]. Their design covers user login, item selection, cart addition, and order placement, along with payment gateway and return policy integration. They use a pushdown automaton to reverse the order flow. A database can be implemented to store user inputs. Their challenge was building a user-friendly and developer-efficient backend, and they concluded that multiple databases for different functions was the most effective approach.

In order to facilitate real-time communication through instant messaging and brief message transmission, Kiatruangkrai P. and Phusayangkul P. designed a content management system for e-commerce[14]. Both content management system and e-commerce features are covered. Two front office and back office styles, as well as seven subsystems, make up the system. The Ajax approach

is used in conjunction with PHP, CSS, and HTMP to construct it. We have ten subjects testing this system.

Using a client-server architecture and Bootstrap 4.0 for the UI and Django for backend operations and APIs, Shivanshu Tyagi and Shashwat Yadav created a basic web application[15]. In their design, by connecting to the interface, users can request products from the relevant APIs, which then add the items to their shopping cart and complete the checkout process. Secure transactions are made over a PayPal payment gateway. Package data can be checked using an order- ID, and updates regarding package arrival are sent via a tracker. TimeMejs is a JavaScript library that monitors user expenditure and grants credit points upon checkout. With the help of this study, they concluded that businesses can flourish in large companies by learning how to create an e-commerce web application from the ground up[16].
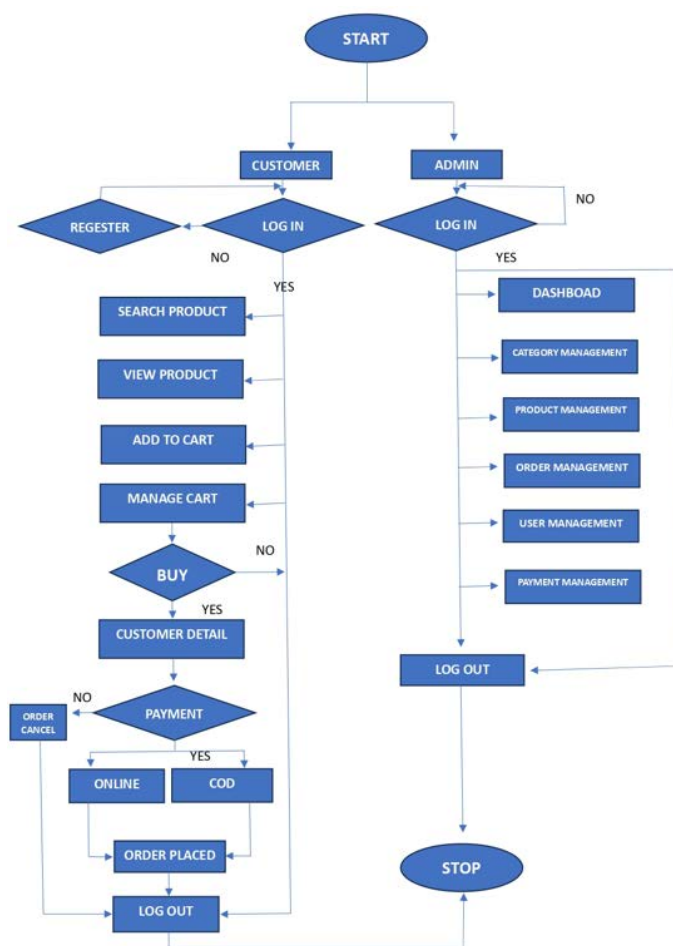
The research of A. B. N. Rahma Putra and A. Mukhadis focuses on creating e-commerce with superior regional items, evaluating its feasibility and appeal, and employing R&D techniques[17]. Development of the product is the first step, and then website and design specialists test its usability and appeal. The product is visually appealing, has demonstrated efficacy in enhancing business acumen in rural communities, and can be used as a benchmark for future studies, according to the findings.

An e-commerce-based online grocery store with freshly packaged food products has been developed by Khan M. M. and Shams-E-Mofiz M. Admin and user panels, along with Gmail login credentials, are elements included in the system design[18]. Especially in the midst of the COVID-19 pandemic, this user-friendly technology is meant to make life simpler, safer, and faster. Nithin Linga and Kartikeya Vinay's paper review about an e-commerce website using MERN stack offering free user registration, advanced product search, and multiple payment options, including "Cash on Delivery," "PayPal," and various card types[19]. The administrator exclusively manages product listings, while customers can update personal data and make repeat orders without re-entering billing details. Products are categorized by brands, and detailed views are accessible by hovering over product images.

Maguire, M.'s study discusses ways for enhancing the usability and conversion rate of e-commerce websites[20]. Clear calls to action, navigation support, effective layout, streamlined checkout, responsive design, accessibility, credibility, user trust, and customer review management are among the most important recommendations. Designers must strike a balance between usability and business goals.

PROPOSED DESIGN

The whole range of obligations and duties necessary for administrating a website that sells goods are included in the activities of e-commerce operations. This embrace branding, placing orders, customer service, managing inventory, and logistics planning, all of those are focused on providing a wonderfull shopping encounter[21]. Figure 2 depicts the flow chart diagram of the designed system.

Journal of Integrated Science and Technology

J. Integr. Sci. Technol., 2025, 13(5), 1116   Pg 2

Figure 2. E-commerce Operational Flow

E-commerce websites enables users to connect with them on a wide range of devices, but in order to take advantage of more features, users must register. Users can quickly search for their preferred goods using precise parameters after logging in. After being found, a large variety will appear along with choices to narrow down your search according to tastes. Conveniently add purchases to the cart with preservation. Before checking out, buyers check the details of the products and choose how they want to pay. Users are able to reassured by the seamless interactions made possible by integrated gateways for payment. Ecommerce mediums, in general, simplify the buying and selling experience for


Figure 3. Ecommerce Operations

consumers while providing safety and convenience[22]. The developed e-commerce operations in proposed system are depicted in figure 3.

Procedure from Placing an Order to receiving the order

In the process of e-commerce, products are first received and then packaged before being placed on shelves. When an order is made, the items are selected from their designated shelves and undergo a comprehensive inspection prior to packing. Once packed, they are labeled for shipment and dispatched. Ultimately, the product is delivered to the customer, thereby concluding a smooth and efficient process that guarantees customer satisfaction.

Shipping Process

Once an order gets placed by the customer, the shipping process begins. The purchased items are picked from inventory; these goods are carefully wrapped and put into packets, then precisely weighted to determine delivery cost. Following this, shipping labels are generated, and the packages undergo a final quality check to ensure compliance with standards. The meticulously prepared packages are then taken to the loading dock, where they are loaded onto trucks. At the same time, the order status is promptly updated to "shipped," enabling real-time tracking for both the customer and the system. Furthermore, stock levels are adjusted to reflect the shipped items, maintaining precise inventory records. The carrier is given the necessary delivery instructions to facilitate a smooth handover to complete the order journey. Figure 4 shows entire shipping logistics and transportation protocol.
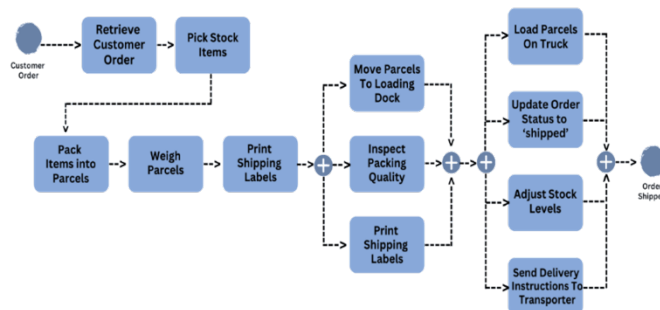
**Figure 4.** Shipping logistics and transportation protocol

Payment Process

To speed up the payment process, the payer starts a transaction with the business during the checkout phasee[23]. After obtaining the transaction information from the merchant, the payment processing gateway safely transfers them to the acquiring bank for approval. The acquiring bank and the issuing bank, where the payer has their account, are connected by the payment system. Payment is only accepted when the issuer bank has thoroughly reviewed the transaction and verified that sufficient funds are available. To ensure a stable and secure payment system for the customer and seller, the payment system employs the payment gateway to send the confirmation back to the seller with authorization.

METHODOLOGY

In the development of an e-commerce website, we used the MERN stack because of its efficient development workflow, scalability, and performance[7]. Figure 5 shows MERN Stack architecture, which comprises the open-source elements, namely

Journal of Integrated Science and Technology

J. Integr. Sci. Technol., 2025, 13(5), 1116   Pg 3

MongoDB, Express.JS, React.JS/Redux, and Node.JS. In this the frontend is assigned by React.JS, HTML, and CSS. For the backend runtime environment, Node.JS and web framework Express.JS, which gives and provides various libraries[24]. MongoDB is used as a data-based system in which data is stored in documents, and a group of documents is known as collections. We used Windows operating system, along with Git, GitHub, and VS Code tools. The MERN stack significantly boosts the e-commerce platform's usability and overall functionality. Its ability to scale ensures the platform can handle growth without compromising performance. Node.js powers fast server-side processing, resulting in quick response times, while React.js creates a smooth, interactive user interface with rapid load speeds. MongoDB's flexible data management efficiently supports large product inventories, and Express.js strengthens backend security. Together, these technologies create a responsive, secure, and high-performing platform, offering an outstanding shopping experience that evolves with user demands.
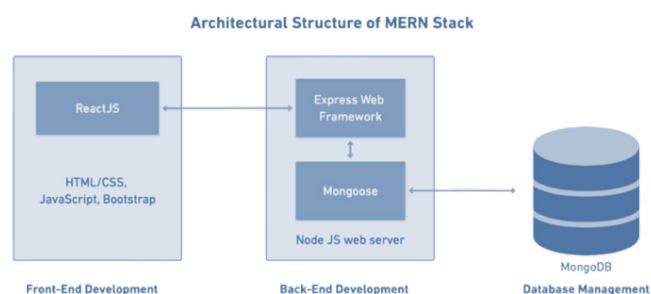
Figure 5. MERN Stack Architecture

MONGODB

For our project, we employed MongoDB, a document-oriented database. In this database, data is stored in documents, and a group of documents is known as a collection[25]. MongoDB, the most well-known NoSQL database, offers a substitute for relational databases, which store all of their data in tables with rows and columns. JSON data is actually transformed by MongoDB into a binary version on the server, which is essentially stored and queried more effectively. BSON is the query language used by MongoDB. We cannot conceive of MongoDB as a JSON database, however, as it saves BSON format both internally and over the network. Any data can be represented in JSON format and saved natively in MongoDB. It can also be simply accessed in JSON format. Thus, from our study and implementation of MongoDB, we can conclude that it is flexible and enables users to design schema, databases, tables, and other objects. We had the choice to use Mongo Shell after installing MongoDB because it provides a JavaScript interface for users to communicate and do any query-related activities. Since MongoDB is a document-oriented database, indexing documents is simple. and for that reason, it processes responses more quickly. Scalability Characterizes MongoDB. The enormous data is divided in the MongoDB database into a nested described structure. Multiple databases can be operated on MongoDB, a database server.

NODEJS

C++ is used in the creation of Node.js, a JavaScript operating system. For optimal performance, it makes use of the Google Chrome V8 engine. Node.js is built with a single-thread architecture and employs event-driven, asynchronous programming callback functions. Our ability to create websites using node.js for back-end development has been aided by the abundance of event-based and asynchronous APIs that are made possible by the essential core notion of node.js design: event-drivenness. In line with our web business logic, Node.js employed the appropriate callback function[26]. Using a single-threaded architecture is the primary benefit of event-driven and asynchronous programming.Without waiting for a specific code to finish, the callback function code is executed, and the limited resources were used for other tasks that were part of our web business logic. This design was appropriate for our back-end development, which was also our system's intended purpose. Managing synchronous requests was a significant task in server development, because blocking resulted in either inefficient or wasteful use of the resources. The resource usage is increased and enhanced website speed with single-threaded architecture and asynchronous callback routines, which also produced the expected testing outcomes. It is evident from the Node.js supported module that many functions, including file operations, are performed asynchronously, in contrast to other programming languages. Node.js uses particularly big network components, such as HTTP, DNS, NET, UDP, HTTPS, TLS, etc., to make server building easier. These network modules allow developers to set up a Web server.

EXPRESSJS

Express is a Node.js framework that we used. While constructing the website, we learned that Express made it simpler and easier to write the back-end code and implement it in an organized fashion, as opposed to writing the code with NodeJS and producing a ton of node modules. Express's support for numerous middlewares, which results in shorter and easier-to-write code, assisted us in the creation of the web and APIs needed for the project[26]. Our design benefits greatly from Express's single-threaded architecture and asynchronous programming. Robust API is essential for our design. In order to begin our Express project, we made a new folder. The first step involved adding a command to the command prompt in order to populate the package.json file. Following that, we had to proceed with accepting the default settings. The command to begin is npm init.

REACTJS

Programming in JavaScript uses the front-end package React.JS. We built our web design user interface using React.JS because of its fast rendering of dynamically changing data. With React, developers can write JS code and produce User Interface elements. We looked at the virtual DOM objects in React.JS, the framework we used for our project. The entire user interface would re-render the virtual DOM whenever we made modifications to our e-commerce design. This lets us assess how the DOM object and virtual DOM might differ from one another. We employed JSX. It made writing our code in the React application more straightforward and easier[18]. The components with React.JS are used while each component has a logic connected to our e-commerce design and adds to the overall user interface of the

project. Components are the building blocks of the user interface. It was easier to understand our site design code and improve overall efficiency when components could be reused.

MVC architectural pattern

Computer scientist Trygve Mikkjel Heyerdahl Reenskaug created the Model-View-Controller (MVC) paradigm in 1979 [27]. It breaks down the complexity of web applications into three logical components: the controller, view, and model. This paradigm is followed by well-known web frameworks including Angular, Ruby on Rails, and Laravel. React is a JavaScript, HTML, and CSS client-side view that uses ReactJS [28]. Mongoose is the server-side model. The controllers in question are Node and Express.

EXPERIMENTAL RESULTS

Graphical results enhance the reader's comprehension and appreciation of the research findings by providing illustrations of user interfaces, product displays, website design, and customer interactions.

Signup and Login

Our platform has accessible signup and login pages that make it easy for users to browse and interact with our wide choice of items and services. Ecommerce User SignUp and Login page is depicted in Fig.6 (a) and Fig.6 (b)

Home Page

Our online shopping website's homepage greets users with an eye-catching design that highlights featured goods and special deals as shown in Fig.7. Users may quickly peruse our vast inventory because to simple navigation and separate categories, making their buying experience enjoyable right away[29,30].

Figure 6 (a). Ecommerce User SignUp Page. **(b).** Ecommerce User Login page.
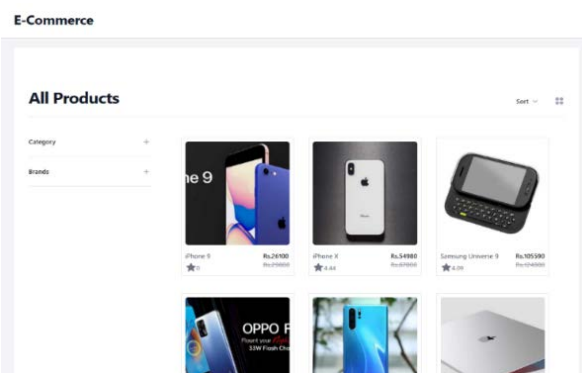
Figure 7. Ecommerce Home Page for every user.

User Profile

The e-commerce website's user profile page enhances the buying experience by offering customized ideas, order history, and preferences. To ensure a smooth and custom experience, users may track deliveries, modify their account information, and interact with customized recommendations as shown in Figure 8.

Figure 8. User Profile

Order Details

An online shopping platform's order detail page provides thorough insights into previous transactions, including order status, tracking details, and itemised summaries as depicted in Fig. 9. Transparency and convenience are ensured throughout the buying process by the ease with which users can review transaction data,print invoices, and easily do refunds or exchanges.

Figure 9 Order Details Page.

Checkout Page

Our e-commerce platform's checkout page simplifies the ordering process by leading customers through the payment, shipping, and order confirmation steps. Customers may evaluate their items, enter delivery information, and complete their orders quickly and easily as shown in figure 10 with the help of clear prompts along with safe transaction options, assuring a smooth and enjoyable shopping experience.[31]

Journal of Integrated Science and Technology

J. Integr. Sci. Technol., 2025, 13(5), 1116   Pg 5

Figure 10. Checkout Page

Order Successful page

Our e-commerce platform's order successful page provides the user with a confirmation that their order has been placed as shown in Fig. 11. Users are also given their Order number for future tracking. An option for returning to the Home Page is also available in case any user wants to return to the home page.
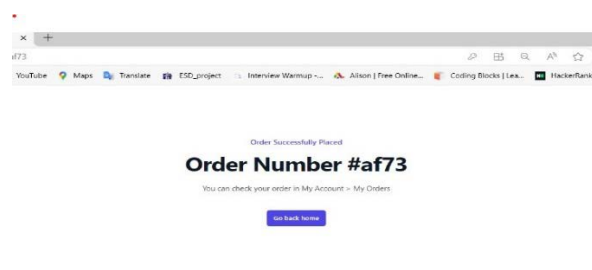

Figure 11. Order Successful Page

The MERN stack enhances the e-commerce platform by providing fast performance with Node.js and a responsive, dynamic user interface with React.js. MongoDB's flexible data handling ensures the platform can easily scale as traffic and product catalogs increase. The modular architecture makes adding new features simple, and Express.js ensures robust security for user data. Overall, MERN delivers a faster, more scalable, and secure platform, making it a superior choice over traditional e-commerce websites.

This research introduces an innovative e-commerce platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), which ensures high scalability, robust security, and exceptional performance. Key developments include a tailored shopping experience with features such as a smart shopping cart, wish lists, and an AI-driven recommendation system. Additionally, the platform provides separate interfaces for users and admins, facilitating efficient management. It also supports secure payment integrations, fast loading speeds, and high-resolution visuals.

CONCLUSION

The internet-based trading platform functions as an innovative hub for worldwide commerce, promoting firms and propelling economic growth. Leveraging cutting-edge tools such as MongoDB, Node.js, Express.js, and React.js, it promises scalability, flexibility, and flawless user experiences.[26] By putting an emphasis on user-focused design, the platform tackles obstacles like safety risks and provides reliable payments together with intuitive interfaces. It alters offerings, improves consumer interactions, and increases vendor gains by deploying big data analytics. By using MERN stack, this research offers a cutting-edge, scalable solution that overcomes common e-commerce challenges and provides valuable insights for future progress in online retail technology. Following that, innovations aim to improve data security, accuracy, and availability in order to foster prosperity and concurrence.

ACKNOWLEDGMENTS

We are grateful to all who supported us throughout this research. My heartfelt thanks for their invaluable guidance, time, and mentorship.

CONFLICT OF INTEREST STATEMENT

The authors have no conflicts of interest to declare. All co authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

REFERENCES AND NOTES

1. Chen Y, Li M, Song J. A study of cross-border E-commerce research trends: Based on knowledge mapping and literature analysis. *Front Psychol.* **2022**.
2. P. M. Alamdari, N. J. Navimipour. A Systematic Study on the Recommender Systems in E-Commerce, *IEEE Access*, **2020**, 8, 115694-115716.
3. Xun Xu, Charles L.The impact of e-service offerings on the demand of online customers*; Int. J. Production Economics*, **2017**, 184, **231-244**.
4. S. E. Cebeci, K. Nari and E. Ozdemir,Secure E-Commerce Scheme, *IEEE Access,* **2022**, 10, 10359-10370.
5. H. Hans, Tomas Falk. eTransQual: A transaction process-based approach for capturing service quality in online shopping, *J. Business Research,* **2006**, 59, 866-875.
6. Cebeci, S. E., Nari, K., & Ozdemir, E. Secure E-Commerce Scheme. *IEEE Access*, **2022**, 10, 10359-10370.
7. Subramanian, V. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. *Springer Science Business Media,* **2017**.
8. Soleimani, Marzieh. Buyers' trust and mistrust in e-commerce platforms: a synthesizing literature review. *Info. Systems and e-Business Management 20*, **2022** ,1, 57-78.
9. Kalkha, H., Khiat. The rising trends of smart e-commerce logistics. *IEEE Access,* **2023**, 11, 33839-33857.
10. K. Satish, Weng Marc Lim. 20 years of electronic commerce research. *Electronic Commerce Res.,* **2021**, 21, 1-40.
11. Wasilewski, Adam. Multi-variant User Interfaces in E-commerce, A Practical Approach to UI Personalization. *Progress in IS*, **2024.**
12. Ahmad, S. I., Rana, T. A Model-Driven framework for the development of MVC-Based (Web) application. *Arabian J. Sci. Engin.,* **2021**, 47(2), 1733–1747.
13. M. M. V. Sai Nikhil. Design and Implementation of E-commerce Website using Automata Theory. *ICOEI*, **2022**, 957-963.
14. P. Kiatruangkrai. Design and Development of Real-Time Communication Content Management System for E-Commerce, *International Symposium on Data, Privacy, and E-Commerce,* **2010**, 111-116.

15. S. Tyagi, S. Yadav. Analysis and Development of E-Commerce Web Application, *CCICT*, **2022**, 65-72.
16. Behl, Abhishek, Pankaj Dutta. A conceptual framework for the adoption of big data analytics by e-commerce startups: a case-based approach. *Information systems and e-business management* .**2019,** 17, 285-318.
17. A.B.N.R. Putra et al. ,The innovation of E-Commerce Based on Local Culture Products to Improve Business Skills to Rural Communities in the Era of Industrial Revolution 4.0. *Int Conf. on Electrical, Electronics and Information Engineering* ,**2019**, 206-210.
18. Khan, M. M. Development of E-Commerce-Based On-line Web Application for COVID-19 Pan demic. *iBusiness,* **2020,** 12, 113-126.
19. Nithin Linga, Kartikeya Vinay. eCommerce Product Showcase using MERN Stack, *Advancements in Communication and Systems, SCRS, India,* **2024**, pp. 253-272.
20. Maguire, M. A Review of Usability Guidelines for E-Commerce Website Design. In: Marcus, A., Rosenzweig, E., Soares, M.M. (eds) Design, User Experience, and Usability. *Computer Science,* **2023**, vol. 14032. Springer, Cham.
21. Zennaro, Ilenia. Implementing E-commerce from logistic perspective: Literature review and methodological framework. *Sustainability*, **2022**, 14 (2), 911.
22. P. M. Alamdari."A Systematic Study on the Recommender Systems in the E-Commerce. *IEEE Access*, **2020**, 8, 115694-115716.
23. Palese, Antonio Usai. The relative importance of service quality dimensions in E-commerce experiences. *Int. J. Information Management,* **2018,** 40, 132-140.
24. Shukla, Santosh Kumar.Application using MERN Stack. *Int. J. Modern Trends in Science Technology,* **2022**, 8, 102-105.
25. Mehra, Monika. MERN Stack Web Development. *Annals of the Romanian Society for Cell Biology,* **2021**, 25, 6.
26. Hoque, Shama. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.JS, *Packt Publishing Ltd*, **2020**.
27. Juneau, Josh. The MVC Framework. Java EE 8 Recipes: A Problem-Solution Approach. Berkeley, CA: Apress, **2018**, 313-344.
28. Hasan, S. S. An integrated approach of MAS-CommonKADS, Model–View–Controller and web application optimization strategies for web-based expert system development. *Expert Systems with Applications*, **2011,** 38(1), 417-428.
29. Song, Qinbao, M. Shepperd. Mining web browsing patterns for E-commerce. *Computers in Industry*, **2006**, 6, 622-630.
30. Hong, Weiyin. Designing product listing pages on e-commerce websites: an examination of presentation mode and information format. *Int. Jour. of Human-Computer Studies*, **2004**, 61, 481-503.
31. Deng, Liqiong, M. Scott Poole. Aesthetic design of e-commerce web pages–Webpage Complexity, Order and preference. *Electronic Commerce Res. Appl.*, **2012**, 11(4), 420-440.

Journal of Integrated Science and Technology

J. Integr. Sci. Technol., 2025, 13(5), 1116   Pg 7