International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 5 May. 2026, pp: 123-129 www.ijaem.net ISSN: 2395-5252

A Web-Based Food Ordering System Using Mern Stack with Secure Payment Integration

[1]Ayush Singh, [2]Abdullah Khan
[12]*Department of Computer Science and Engineering, Dronacharya Group of Institutions, Greater Noida, India*

Date of Submission: 25-04-2026 Date of Acceptance: 04-05-2026

ABSTRACT
The rapid growth of digital technologies has significantly transformed the food service industry, increasing the demand for efficient and user-friendly online food ordering systems. However, many small and local food businesses still rely on traditional ordering methods, which often lead to inefficiencies, order inaccuracies, and delayed service. To address these challenges, this research presents a web-based food ordering system developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js).
The proposed system allows users to register, browse food categories, add items to a cart, and place orders through an intuitive web interface. It supports both secure online payments using Stripe and Cash on Delivery (COD), ensuring flexibility for users. A robust authentication mechanism is implemented to manage user access, while an administrative dashboard enables efficient management of food items, orders, and user data.
The system is designed with a modular and scalable architecture, allowing future extension into a multi-vendor platform without major structural changes. Experimental evaluation shows that the application delivers fast response times, reliable order processing, and improved user experience compared to traditional methods. The proposed system provides a practical and cost-effective solution for small-scale food businesses aiming to adopt digital ordering systems.

Keywords: User Authentication, MERN Stack, Web Application, Payment Gateway, Cart Management, Online Ordering.

I. Introduction

The rapid expansion of internet connectivity and the widespread use of smartphones have significantly transformed consumer behaviour, particularly in the food service industry.

Online food ordering systems have become an integral part of modern lifestyles, offering users the convenience of browsing menus, placing orders, and making payments through digital platforms. These systems reduce dependency on traditional ordering methods such as phone calls and in-person visits, thereby improving efficiency and customer satisfaction. As a result, web-based food ordering applications are increasingly being adopted by restaurants and food businesses.

Despite this growth, many small and local food vendors still face challenges in adopting digital platforms due to high development costs, lack of technical expertise, and dependence on third-party services that charge significant commissions. Traditional ordering methods are often prone to errors, delays, and inefficient order management, which negatively impacts both customer experience and business operations. Furthermore, the absence of secure and integrated payment systems limits the flexibility and reliability of such services.

To address these issues, this research proposes a web-based food ordering system developed using the MERN stack. The system provides a seamless interface for users to register, browse categorized food items, add items to a cart, and place orders efficiently. It integrates secure online payment using Stripe along with Cash on Delivery (COD), ensuring flexibility and convenience for users. An administrative dashboard is also implemented to manage food items, track orders, and monitor user activity in real time.

The proposed system is designed using a modular and scalable architecture that supports future enhancements. Although the current implementation focuses on a single-vendor model, the system is structured in a way that allows easy expansion into a multi-vendor platform by extending database schemas and backend functionalities. This ensures long-term usability and adaptability of the system in real-world applications.

The key objectives of this research are:

1. To design and develop a web-based food ordering system using the MERN stack;
2. To implement a user-friendly interface for browsing food items and placing orders efficiently;
3. To integrate secure online payment using

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 5 May. 2026, pp: 123-129 www.ijaem.net ISSN: 2395-5252

Stripe along with Cash on Delivery options;
4. To develop an administrative dashboard for managing orders, users, and food items;
5. To ensure scalability of the system for future multi-vendor integration;
6. To improve order accuracy, reduce processing time, and enhance overall user experience.

II. Literature Review

The growth of online food delivery systems has significantly improved accessibility and convenience for users, while also transforming restaurant operations. However, several studies highlight that small and local vendors face challenges such as high commission fees, limited digital infrastructure, and reduced market visibility when using existing platforms.

Ray et al. (2019) analyzed user adoption behavior in food delivery applications and identified convenience and trust as key influencing factors. Kapoor and Vij (2018) examined platform-based food delivery systems and highlighted the economic impact of aggregator models, particularly the reduced profitability for small vendors. Similarly, Pigatto et al. (2017) studied operational models in food delivery services and emphasized the importance of logistics and system efficiency.

Varshney and Sharma (2020) demonstrated that web-based food ordering systems improve efficiency compared to traditional methods but lack scalability and modern architectural design. With the advancement of web technologies, the MERN stack has emerged as a reliable framework for building scalable and modular applications. Official documentation of MongoDB, Node.js, and React highlights their capability to support real-time data handling and dynamic user interfaces. Additionally, Stripe provides secure and efficient online payment integration, ensuring reliable transaction processing.

Study	Contribution	Limitation
Ray et al. (2019)	Identified key factors influencing app adoption	No vendor-focused analysis
Kapoor & Vij (2018)	Analyzed impact of aggregator platforms	Reduced vendor profitability
Pigatto et al. (2017)	Explained operational models in delivery systems	Limited technical implementation
Varshney & Sharma (2020)	Improved ordering efficiency over manual systems	Lack of scalability
MERN Stack Docs	Supports scalable web application development	Not domain-specific
Stripe (2023)	Enables secure online payments	Requires internet dependency
Proposed System	Combines scalability and flexible payments	Single-vendor (current)

Table 2.1: Comparative Analysis of Existing Food Ordering Systems

From the above studies, it is evident that while existing systems provide convenience and technological advancements, they often fail to address affordability and flexibility for small vendors. The proposed system aims to bridge this gap by providing a scalable, efficient, and cost-effective web-based food ordering platform with integrated payment options.

III. Methodology

The proposed web-based food ordering system is developed using a structured client–server architecture to ensure efficient communication between the frontend, backend, and database. The system is implemented using the MERN stack, enabling scalable performance, real-time data handling, and secure transaction processing. The methodology focuses on providing a streamlined workflow from user interaction to order management.

3.1 User Interaction and Authentication

The system provides a responsive user interface where customers can register and log in securely. Authentication is handled through encrypted credential validation and session management. Once authenticated, users can browse categorized food items, view details, and interact with the application through a dynamic interface developed using React.js.

3.2 Workflow Description

The workflow begins when a user selects food items and adds them to the cart. The cart dynamically updates based on item selection and quantity. Users proceed to checkout by entering delivery details and selecting a payment method. The system supports both Stripe-based online payment and Cash on

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 5 May. 2026, pp: 123-129 www.ijaem.net ISSN: 2395-5252

Delivery (COD). After successful payment or confirmation, the order is recorded in the database and processed by the system.

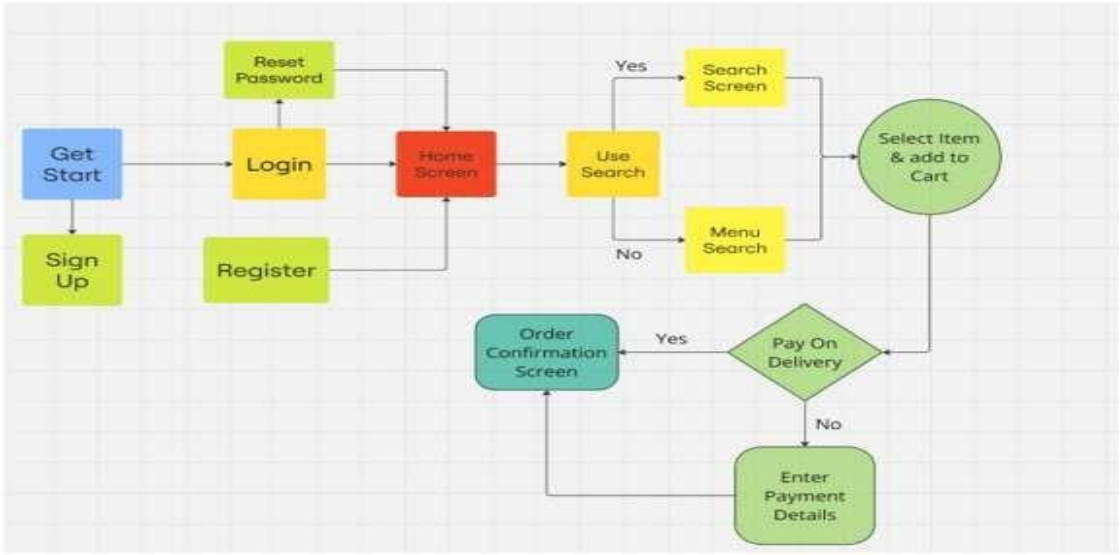

Figure 3.1: System Workflow

3.3 Use Case Modelling

The system consists of two primary actors: User and Admin.

- User: registers, logs in, browses food items, adds items to cart, places orders, and selects payment methods.
- Admin: manages food items, monitors orders, and updates system data through the admin dashboard.

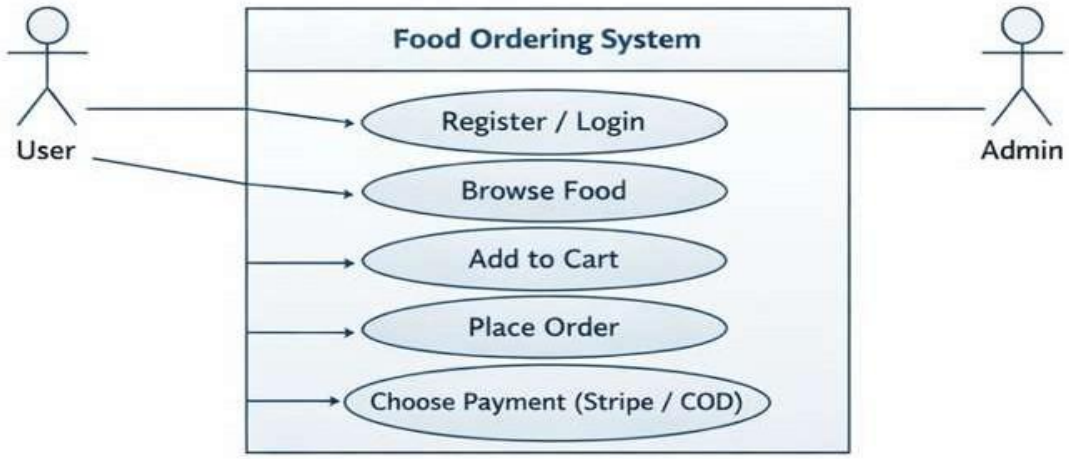

Figure 3.2: Use Case Diagram

3.4 Backend Processing

The backend is developed using Node.js and Express.js, which handle API requests, authentication, and order processing. MongoDB is used as the database to store user data, food items, and order details. RESTful APIs ensure smooth communication between frontend and backend. Stripe integration is used for secure online transactions, while COD orders are processed directly through the system without external dependency.

3.5 Admin Dashboard Functionality

The admin dashboard provides centralized control over the system. It allows administrators to add, update, or remove food items, view incoming orders, and manage user-related data. This functionality ensures efficient system management and improves operational workflow.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 5 May. 2026, pp: 123-129 www.ijaem.net ISSN: 2395-5252

IV. Implementation

The implementation of the proposed food ordering system is carried out using the MERN stack, integrating frontend, backend, and database components to ensure efficient system performance. The system is designed to handle user interactions, order processing, and payment integration in a structured and reliable manner.

4.1 Order Processing Algorithm

The order processing mechanism follows a sequential flow to ensure accurate order placement and tracking:

Step 1: User registers or logs into the system
Step 2: User browses food items and selects desired products Step 3: Selected items are added to the cart
Step 4: Total cost is calculated based on item price and quantity Step 5: User proceeds to checkout and enters delivery details Step 6: User selects payment method (Stripe or Cash on Delivery) Step 7: If Stripe is selected, payment is processed securely
Step 8: Upon successful payment or confirmation (COD), order is stored in database Step 9: Order status is initialized as "Pending"
Step 10: Admin updates order status to "Delivered" after completion

4.2 Mathematical Representation

The total cost of an order is calculated using the following equation: Total Cost = Σ (Price × Quantity) + Delivery Fee

This ensures dynamic calculation of the order value based on user selection.

4.3 Dataset Representation

The system stores order-related data in a structured format within MongoDB. A sample dataset is shown below:

Order ID	User ID	Item Name	Quantity	Total Price	Payment Method	Status
101	U01	Pizza	2	500	Stripe	Delivered
102	U02	Burger	1	150	COD	Pending
103	U03	Pasta	3	450	Stripe	Delivered

This structured data enables efficient tracking of orders, user activity, and payment status. The inclusion of order status (Pending/Delivered) ensures proper order lifecycle management and enhances system reliability.

V. Results and Discussion

The proposed food ordering system was evaluated across its core functionalities, including user interaction, order processing, payment integration, and admin management. The

system demonstrated reliable performance and smooth operation under normal usage conditions.

5.1 Functional Results

The application was tested end-to-end, and all modules performed as expected:

- User Authentication: Users were able to register and log in securely without errors.
- Food Browsing and Cart Management: Items could be added, updated, and removed from the cart dynamically.
- Order Placement: Orders were successfully placed using both Stripe and Cash on Delivery (COD).
- Payment Integration: Online payments via Stripe were processed securely and confirmed instantly.
- Order Management: Orders were stored in the database with accurate details and status tracking.
- Admin Dashboard: Admin users were able to manage food items and update order status efficiently.

These results confirm that the system provides a complete and functional food ordering workflow.

5.2 Performance Summary

The system performance was evaluated based on response time and processing efficiency. The observed results are summarized below:

Operation	Performance
User Login Response Time	1–2 seconds
Food Data Fetching	< 1 second
Cart Update Time	Instant
Stripe Payment Processing	2–4 seconds
Order Storage in Database	< 500 ms
Order Status Update	< 1 second

The results indicate that the system delivers fast response times and efficient data handling, ensuring a smooth user experience.

5.3 User Experience and Benefits

User interaction testing indicated a positive experience across all functionalities:

- Users found the interface simple and easy to navigate.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 5 May. 2026, pp: 123-129 www.ijaem.net ISSN: 2395-5252

- The ordering process was quick and required minimal steps.
- Multiple payment options (Stripe and COD) increased usability and flexibility.
- Admin controls enabled efficient monitoring and management of orders.

Overall, the system significantly reduces manual effort and improves order accuracy compared to traditional ordering methods.

5.4 Comparative Analysis

To further evaluate the effectiveness of the proposed system, a comparative analysis was conducted against traditional food ordering methods and existing platforms.

Aspect	Traditional System	Existing Platforms	Proposed System
Architecture	Manual / Semi-digital	Monolithic / Cloud-based	MERN (Modular Client-Server)
Data Handling	Manual / Spreadsheet	Centralized	MongoDB (Real-time NoSQL)
Frontend	None / Basic UI	App-based	React (Dynamic UI)
Backend	Minimal	Proprietary APIs	Node.js + Express (REST APIs)
Scalability	Low	High	High (Modular Design)

Table 5.1: System Architecture Comparison

This Table visually represents architectural differences across systems, highlighting the modular and scalable design advantages of the proposed MERN-based system.

5.5 : Performance Behaviour Comparison

Performance behaviour comparison evaluates how different systems perform in terms of efficiency, responsiveness, and accuracy during operation. It provides a qualitative assessment of how effectively each system handles user interactions, processes data, and delivers results under typical usage conditions.

Metric	Traditional System	Existing Platforms	Proposed System
Response Time	High Delay	Moderate	Low
Order Accuracy	Error-prone	Improved	High (Validated)
User Interaction	Manual	App-based	Dynamic Web Interface
Processing Efficiency	Low	Moderate	High

Figure 5.2: Performance Behaviour Comparison

The graph below highlights improved system performance in terms of response time, accuracy, and processing efficiency compared to traditional and existing systems.

Figure 5.3: Graph depicting performance behaviour comparison

5.6 Discussion

The results demonstrate that the MERN-based architecture provides a reliable and scalable solution for online food ordering. The integration of secure payment systems and efficient database handling ensures system stability and performance. Compared to traditional manual systems, the proposed solution offers faster processing, improved accuracy, and better user experience.

Although the current implementation supports a single-vendor model, the modular system design allows easy extension to multi-vendor functionality in future. This makes the system adaptable and suitable for real-world deployment.

VI. Conclusion and Future Work

The proposed web-based food ordering system successfully demonstrates an efficient and user-friendly solution for digital food ordering using the MERN stack. By integrating core functionalities such as user authentication, food browsing, cart management, and secure payment processing, the system provides a complete and reliable platform for handling online food orders. The inclusion of both Stripe-based online payment and Cash on Delivery ensures flexibility and convenience for users.

The system architecture, based on a modular client–server model, enables smooth interaction between the frontend, backend, and database components. The use of MongoDB for data storage and Node.js with Express.js for backend processing ensures efficient handling of user requests and order management. The admin dashboard further enhances system usability by providing centralized control over food items, orders, and user data.

The results indicate that the system delivers fast response times, accurate order processing, and improved user experience compared to traditional manual ordering methods. The implementation also ensures secure transactions and reliable data management, making it suitable for real-world applications.

Although the current system is limited to a single-vendor model, it is designed with scalability in mind. The modular architecture allows easy extension to a multi-vendor platform by incorporating additional modules such as vendor registration, vendor dashboards, and order routing mechanisms.

Future work can focus on enhancing the system by integrating real-time order tracking, mobile application support, and AI-based food recommendation systems. Additional features such as advanced analytics, user behavior tracking, and multi-language support can further improve system usability. These enhancements will enable the system to evolve into a comprehensive and scalable food delivery platform capable of supporting diverse business requirements.

REFERENCES

[1] A. Ray, A. Dhir, P. K. Bala, and P. Kaur, "Why do people use food delivery apps (FDA)? A uses and gratification theory perspective," *Journal of Retailing and Consumer Services*, vol. 51, pp. 221–230, 2019.

[2] A. P. Kapoor and M. Vij, "Technology at the dinner table: Ordering food online through mobile apps," *Journal of Retailing and Consumer Services*, vol. 43, pp. 342–351, 2018.

[3] G. Pigatto, J. G. Machado, A. S. Negreti, and L. M. Machado, "Have you chosen your request? Analysis of online food delivery companies in Brazil," *British Food Journal*, vol. 119, no. 3, pp. 639–657, 2017.

[4] S. Varshney and R. Sharma, "Design and implementation of an online food ordering system," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 6, pp. 120–124, 2020.

[5] MongoDB Inc., "MongoDB Documentation," 2023. [Online]. Available: https://www.mongodb.com/docs/

[6] Node.js Foundation, "Node.js Documentation," 2022. [Online]. Available: https://nodejs.org/

[7] React, "React Official Documentation," 2023. [Online]. Available: https://reactjs.org/

[8] Express.js, "Express.js Documentation," 2023. [Online]. Available: https://expressjs.com/

[9] Stripe Inc., "Stripe API Documentation," 2023. [Online]. Available: https://stripe.com/docs/

[10] A. Bhardwaj and M. Sharma, "Building scalable web applications using MERN stack: Best practices and case studies," *International Journal of Software Engineering and Applications*, vol. 13, no. 2, pp. 45–58, 2022.

[11] H. Zhao and Q. Bian, "Research on online food delivery platform based on user experience," *IEEE Access*, vol. 8, pp. 123456–123465, 2020.

[12] S. Kumar, M. Kumar, and R. Singh, "Online food ordering system using web-based application," *International Journal of*

Scientific Research in Computer Science, vol. 8, no. 3, 2020.

[13] A. Alalwan, "Mobile food ordering apps: An empirical study of the factors affecting customer satisfaction," *International Journal of Information Management*, vol. 50, pp. 28–44, 2020.

[14] M. Verma and A. Sinha, "E-commerce based food delivery systems and their impact," *Procedia Computer Science*, vol. 132, pp. 1102–1109, 2018.

[15] R. Gupta and S. Jain, "Cloud-based food ordering system," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, 2020.

[16] P. Kotler and K. Keller, *Marketing Management*, 15th ed., Pearson, 2016.

[17] A. Osterwalder and Y. Pigneur, *Business Model Generation*, Wiley, 2010.

[18] M. Porter, "Competitive advantage in digital platforms," *Harvard Business Review*, 2019.

[19] J. Nielsen, "Usability engineering principles in web applications," Nielsen Norman Group, 2012.

[20] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed., Pearson, 2004.

[21] D. Evans, *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*, Cisco, 2011.

[22] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[23] T. Erl, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, 2013.

[24] K. Schwaber and J. Sutherland, *The Scrum Guide*, 2020.

[25] Google Developers, "Web Performance Optimization Guide," 2022.