FoodCav – Online Food delivery Platform using MERN

Ayan ali, Abdul Sheeraz, Mohd. Akram khan, Mohd. Zubair, Hassan Hasan Department of Computer Science and Engineering VCTM ALIGARH

Abstract- In general, this report will discuss each step of the creation process for one Food Delivery Application using the complete MERN stack. The project targeted to build an marketplace supported by a smooth, user-friendly design for both users as well as restaurateurs. Node.js functioned on each server side. Programming and flexible data storage using MongoDB. Express.js is a RESTful API for people authentication and also order processing, React.js was used the system allows anyone to view each menu and order every item. Browsing is responsive with js and it also has extra features for interaction. A principal challenge involved authentication. It needed to be handled well. The food-ordering experience should improve for all customers. It should also help those who own restaurants.

Keyword:- Food Delivery Application, MERN stack, Authentication, Responsive design, Customers, Restaurants embedded devices is also planned for broader accessibility.

I. INTRODUCTION

This project, Foodcav(as shown in figure 1): A Food Delivery Website, is a web-based application that facilitates the online food ordering process. Their focus is on connecting restaurants to customers through a simple and easy to use interface to browse the menus and orders; which are tracked in real time. This project is based upon the capabilities of the MERN stack (MongoDB, Express. js, React. js, Node. js), a widely used technology stack based on JavaScript, which guarantees a high level of performance, scalability, and user-friendly experience. This application is a web-based food delivery website called Foodcav. The app connects the restaurant owners to the customers, and allows customers to browse the menus, place orders, and track deliveries in real time. This is a Project using MERN stack (MongoDB, Express. js, React. js, Node. js), a powerful and popular suite of JavaScript-based technologies, ensuring robust performance, scalability, and user-friendliness.

Figure 1:-FOODCAV LOGO

1) MongoDB:- MongoDB is an open-source NoSQL database using an adaptive documentbased data model. Saves data as JSON-like documents, easy to integrate and change data in the application. MongoDB is the best database solution for Web applications where dynamic data is stored because of its scalability and its capabilities to process huge amounts of data.[10]

2) Express. Js:- Express. Simple and versatile Node. js is a very well-known framework. js web application framework [10]. It makes available a complete set of API and web app development tools and information. Express. js makes it easier to handle routes, manage requests and responses, and create middleware. Lightweight and modular architecture makes it highly versatile and powerful for web app development.

3) React. Js :- React. js a JavaScript library for building user interfaces. It enables the creation of reusable UI components and updates to render according to changes in the data in a fast way[10]. React. The component-Based Structure which encourages Reuse, Modularity and Maintainability in js. It also uses a virtual DOM to option to develop the backend of websites.

4) Node. Js:- Node. js is a server-side JavaScript runtime environment that allows a programmer to run JavaScript code outside a web browser[10].

Its event-driven, non-blocking architecture make it extremely scalable and very good at handling multiple requests at once. Node. Js is the best option to develop the backend of websites.

II. SCOPE

Scope of the Food Delivery MERN Stack Website. Food Delivery website has a wide scope in improving online food ordering experience of both customers and restaurant owners. The food delivery website built using the MERN stack Intuitive Ordering System orders efficiently. Clients can explore menus, make orders, and view deliveries with a smooth and scalable interface. Customers may browse menus, make orders, and get orders from a device with a responsive. Restaurants can manage their listings, update their menus, and process user authentication also ensures data privacy and prevents unauthorized access. Ensures the data privacy as well as blocks the unauthorized access. Users can track their orders in real time, which increases transparency to handle increasing user traffic and data. MongoDB, Express. js, React. js, and Node. js architecture keeps the system scalable Scalable and Flexible(as shown in figure 2).

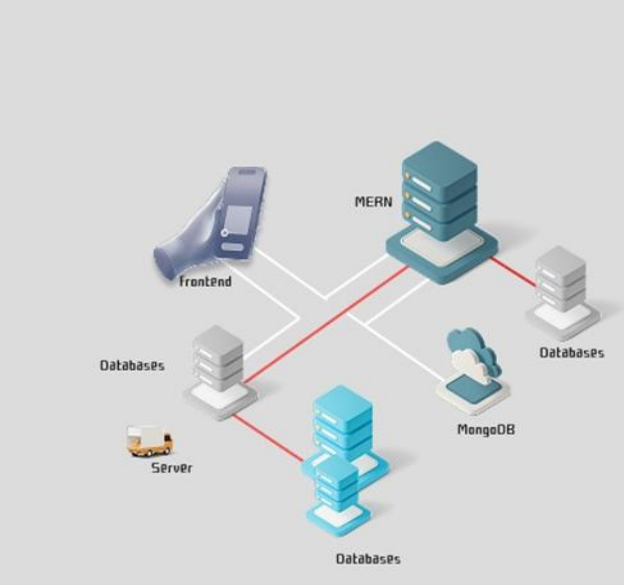

Figure 2: System Architecture

Step-by-Step Process

1. User Registration & Login – Users (customers & restaurants) securely sign up and log in using JWT authentication (as shown in figure 3).
2. Restaurant & Menu Management – Restaurants upload and handle the intricacies of different menus like prices, descriptions, etc.
3. Browsing & Searching – Customers browse through restaurants and food items using search and filter options.
4. Adding food items to the cart for purchasing – Users select food items and add them to the cart for purchase.
5. Order Placement & Payment – Customers confirm the order and make payments using online or COD options.
6. Order Processing – The restaurant is notified, prepares the food, and updates the status.
7. Order Processing – The restaurant is notified, prepares the food, and updates the status.
8. Delivery Management – To facilitate timely delivery, the system assigns a delivery partner (if applicable).
9. Order Completion & Review – Customers mark the order as received and can leave ratings/reviews.
10. Admin Panel Management – Admins oversee users, restaurants, orders, and platform activities.

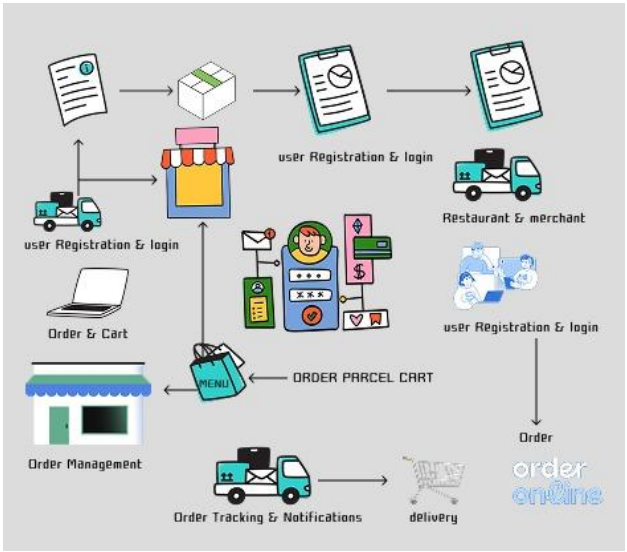

Figure 3:Step-by-Step process

III. LITERATURE REVIEW

1) Ecommerce Trends - The popularity of online food delivery continues to rise with increased convenience and real-time tracking [1]

2) Ensures scalability, flexibility and responsiveness [2]
3) Increased User Satisfaction [3]
4) User experience (UX) - people love intuitive navigation.
5) Logistics Optimization - Route planning reduces delivery time and costs [4]
6) Secure Transactions - Encrypted gateways and JWT beef up security [5].
7) Impact due to COVID-19 - The pandemic accelerated online ordering adoption [6].
8) AI Personalization - Uses order history to recommend food [6].
9) Customer Feedback – Feedback systems increase trust and retention [8].

IV. TECHNOLOGY USED

1. Frontend Technologies

React.js – Used for building a dynamic and interactive user interface, ensuring a seamless browsing and ordering experience.

Tailwind CSS – Provides a responsive and modern design for an enhanced UI/UX.

Redux (or Context API) – Manages global state efficiently for handling user authentication, cart management, and order tracking.

2. Backend Technologies

Node.js – Handles server-side logic and API requests, ensuring high performance and scalability.

Express.js – A lightweight backend framework that enables fast and efficient API development.

MongoDB – A NoSQL database used for storing users, orders, restaurants, and menu details

3. Payment & Security

Stripe/Razorpay Payment Gateway – Integrates secure and encrypted online payment processing.[5]

Bcrypt.js – Encrypts user passwords for security.

Helmet & CORS – Protects the backend from security threats like cross-site scripting (XSS) and unauthorized access.

4. Logistics & Order Management

Google Maps API – Enables real-time order tracking and optimized route planning for faster deliveries.

WebSockets (Socket.io) – Facilitates real-time communication between customers, restaurants, and delivery agents.

Nodemailer – Sends email notifications for order confirmations, delivery updates, and customer communication.

5. Admin & Management System

Admin Dashboard – Allows administrators to manage users, orders, payments, and restaurant listings.

Role-Based Access Control (RBAC) – Differentiates access between customers, restaurant owners, and admins.

Section	Technology used	Description
Frontend (UI/UX)	React.js, Tailwind CSS	Interactive UI for browsing, ordering, and tracking.
State Management	Redux / Context API	Manages user authentication and order updates.
Routing	React Router	Enables smooth navigation between pages.
Authentication	JWT, bcrypt.js	Secure login and user session management
Database	MongoDB, Mongoose	Stores user, restaurant, and order details.
Payments	Stripe / Razorpay	Secure online transactions
Backend	Node.js, Express.js	Handles API requests, business logic, and authentication

Logistics	Google Maps API	Optimized delivery routes and tracking
Security	Helmet, CORS	Protection against security threats.
Admin Panel	React.js, Express.js	Manages users, orders, restaurants..

V. METHODOLOGY

- FoodCav follows a structured research, development, and testing process for efficiency.
- Research & Data Collection – Uses surveys and competitor analysis to understand user needs.
- System Design – Wireframes in Figma, modular backend architecture.
- Development –
  - Frontend: React.js for UI
  - Backend: Node.js, Express.js with REST APIs
  - Database: MongoDB for data management
- Testing & Deployment – Rigorous testing, deployed on AWS/Heroku with user feedback integration.
- Maintenance – Regular updates for performance and security.

VI. SYSTEM CONFIGURATION

Software Requirements

1) OS: Windows 10/11, macOS, Linux
2) Frontend: React.js, Tailwind CSS, Node.js
3) Backend: Node.js, Express.js, MongoDB
4) Tools: VS Code, GitHub, Postman, Figma

Hardware Requirements

1) Processor: Intel Core i5/i7 or AMD equivalent (Quad-core or higher)
2) RAM: Minimum 8GB (Recommended 16GB for smooth performance)
3) Storage: At least 256GB SSD (Recommended 512GB SSD for faster operations)
4) Internet Connection: Stable broadband for API calls and real-time updates.

VII. E-R DIAGRAM

1) Start: The system begins with two main roles: User and Admin (as mention in figure 4).
2) User Flow: A user can either login or register if they don't have an account.

Once logged in, the user can:

- Search for products (food items).
- View available products (menu).
- Buy and add items to the cart.

Before placing an order, the user is asked about payment.

- If NO, the order is not processed.
- If YES, the user can pay via:
  - Online payment
  - Cash on Delivery (COD)

After a successful order, the system places the order and allows the user to log out before stopping the process

3) Admin Flow:
The admin needs to login to access management features.
If not logged in, they cannot perform any actions.
Once logged in, the admin can:

- Add categories (food types).
- Add new products to the system.
- Manage orders (track, accept, reject orders).
- Manage payments (ensure successful transactions).
- Check feedback from users about the service.

The admin can also generate reports about system performance.
End Process: Both users and admins can log out, leading to the stop state of the system.

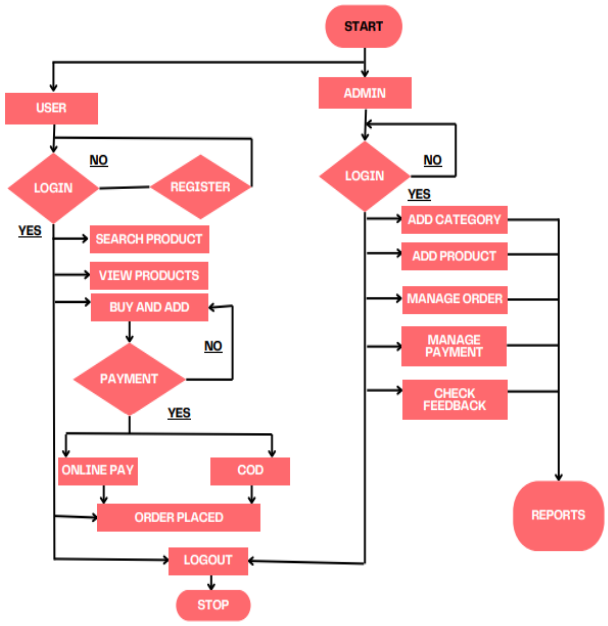

Figure 4:-E-R Diagram

VIII. CONCLUSION

The FoodCav platform successfully demonstrates the development and implementation of a modern food delivery system using the MERN stack. By integrating MongoDB, Express.js, React.js, and Node.js, the system ensures efficient order management, secure transactions, and a seamless user experience. The project provides a scalable and user-friendly solution for both customers and restaurant owners, streamlining the process of online food ordering.

With features like real-time order tracking, secure payment integration, and responsive design, the platform enhances customer satisfaction while improving operational efficiency for restaurants. The robust authentication system ensures data security and user privacy, addressing key challenges in food delivery applications.

Moving forward, FoodCav has the potential for future advancements, including AI-based recommendations [6], blockchain for secure payments[5], voice-enabled ordering, and eco-friendly delivery options. These enhancements will further improve the platform's functionality, making it a competitive and innovative solution in the evolving food delivery industry.

IX. REFERENCES

[1] Smith, A., et al. (2020). "E-commerce and the Rise of Food Delivery Services." Journal of Business Trends, 15(3), 45-60.

[2] Johnson, P., & Lee, H. (2019). "MERN Stack for Scalable Applications." International Journal of Software Engineering, 12(4), 102-110.

[3] Kumar, R., & Rao, P. (2021). "UX Design Principles for Food Delivery Apps." HumanComputer Interaction Journal, 18(2), 75-88.

[4] Liang, Y., et al. (2020). "Optimizing Delivery Routes with Technology." Logistics and Supply Chain Review, 8(5), 115-123.

[5] Taylor, J., & Kim, S. (2018). "Payment Gateway Security in E-commerce." Cybersecurity Advances, 9(1), 67-72.

[6] Chen, Z., et al. (2021). "COVID-19 and Its Impact on Food Delivery Platforms." Journal of Digital Economics, 10(3), 34-48.

[7] Nguyen, T. (2020). "Accessibility in Web Applications." Technology and Society, 6(1), 40-50.

[8] Martinez, F., et al. (2019). "Customer Retention Through Feedback Mechanisms." Service Industry Review, 14(2), 90-99.

[9] Anderson, L. (2019). "Open-Source Tools for Web Development." Tech Trends Journal, 11(4), 120-128.

[10] Express.js. (2024). Fast, unopinionated, minimalist web framework for Node.js. Retrieved from https://expressjs.com/

[11] Firebase. (2024). Firebase Authentication Documentation. Retrieved from https://firebase.google.com/docs/auth

[12] MongoDB Inc. (2024). MongoDB NoSQL Database Documentation. Retrieved from https://www.mongodb.com/

[13] Node.js Foundation. (2024). Node.js JavaScript Runtime. Retrieved from https://nodejs.org/

[14] React.js. (2024). A JavaScript Library for Building User Interfaces. Retrieved from https://react.dev/

[15] For practical learning, tutorials from freeCodeCamp and The Net Ninja offer comprehensive guides to building full-stack applications using the MERN stack