FOOD ORDERING SYSTEM

A PROJECT REPORT

Submitted by

BOOBALAN M (920422205020)

DARIN VIDHU A (920422205022)

SOUNDRA PANDIAN K (920422205105)

KARTHICK M (920422205304)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution - Affiliated to Anna University, Chennai)

K.VELLAKULAM, VIRUDHUNAGAR - 625 701

NOVEMBER 2024

KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution- Affiliated to Anna University, Chennai)

K.VELLAKULAM, VIRUDHUNAGAR - 625 701

BONAFIDE CERTIFICATE

Certified that the project report "**FOOD ORDERING SYSTEM USING MERN STACK (MONGO DB, EXPRESS JS, REACT, NODE.JS)**" is the bonafide work **of "BOOBALAN M (920422205020) DARIN VIDHU A (920422205022) SOUNDRA PANDIAN K (920422205105) & KARTHICK M (920422205305)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. E. VAKAIMALAR	**Dr. R. ARTHY**
Head of the Department,	**SUPERVISOR,**
Associate Professor,	Assistant Professor,
Dept. of Information Technology,	Dept. of Information Technology,
Kamaraj College of Engg & Tech,	Kamaraj College of Engg &Tech,
K.Vellakulam,	K.Vellakulam,
Virudhunagar - 625 701.	Virudhunagar - 625701.

INTERNAL EXAMINER **EXTERNAL** **EXAMINER**

ABSTRACT

This project presents the design and development of a full-stack web application using the MERN stack, an innovative combination of MongoDB, Express.js, React, and Node.js. The project, titled **FOOD ORDERING SYSTEM**, focuses on creating a digital platform that facilitates the process of ordering meals from restaurants or food vendors. It typically includes a user-friendly interface where customers can browse menus, customize orders, and make payments online. Key features often include real-time order tracking, user accounts for order history, and customer reviews to enhance decision-making. For restaurants, the system provides order management tools, inventory tracking, and analytics to optimize operations. The integration of mobile applications and delivery services has further streamlined the ordering process, improving customer convenience and expanding market reach. Overall, a food ordering system enhances the dining experience by making it faster, easier, and more accessible. The use of the MERN stack provides a unified development environment, with JavaScript powering both the frontend and backend. MongoDB, a NoSQL database, stores dynamic product and user data, while Express.js and Node.js handle server-side logic and API requests. React.js delivers a seamless, responsive, and interactive user experience on the frontend.

Acknowledgment

First of all, We would like to thank Lord Almighty for his abundant grace and countless blessings in making this work a great success.

We Wish to express our thanks and sincere gratitude to our Head of the Department **Dr. E. VAKAIMALAR**, for giving this opportunity to do this project as a team.

We would like to thank **Dr. R. ARTHY**, for her valuable advice and guidance in making the project possible

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. HTML

In a food ordering system built using HTML, the structure typically includes sections for user registration, login, menu display, a shopping cart, and an order placement form. The registration and login forms collect user details like email, username, and password to manage user authentication. The homepage features a menu display where food items, such as pizza or burgers, are listed with descriptions, prices, and buttons to add them to the cart. The cart section displays the selected items, their quantities, and the total price, allowing users to review their order before proceeding.

The "Place Order" section collects additional information such as delivery address and payment method through a form. These elements work together to create a streamlined, user-friendly interface for browsing food items and making purchases, with the data dynamically handled through backend systems connected to a database for storing user, product, and order information. Additional features could include real-time order tracking and a notification system to inform users when their order is being processed or delivered, enhancing the overall user experience.

1.2. CSS

In a food ordering system, CSS plays a crucial role in creating a visually appealing and responsive user interface. It helps style the layout by organizing sections such as the menu, cart, and order forms, ensuring a clean and user-friendly design. CSS is used to define consistent typography, button styles, and color schemes that match the brand’s identity. Flexbox or CSS Grid layouts allow for easy alignment of items, making the menu display or cart items responsive across different screen sizes.

Hover effects and transitions on buttons provide an interactive experience, while form fields for login, registration, and checkout are styled for clarity and accessibility. Media queries ensure the system works seamlessly on both desktop and mobile devices, improving usability. CSS also enhances the user experience by maintaining a balance between aesthetics and functionality, allowing users to easily browse and interact with the system. Lastly, CSS animations can be added to highlight important actions like order confirmations or cart updates.

1.3. JAVASCRIPT

JavaScript is essential in a food ordering system for handling dynamic interactions and enhancing the overall user experience. It enables real-time updates, such as adding or removing items from the cart without refreshing the page, making the ordering process smoother. JavaScript also manages form validation for registration, login, and checkout, ensuring users input valid data before submission. Through event listeners, users can interact with buttons and menus, with JavaScript handling the logic behind placing orders, updating quantities, and calculating total prices dynamically.

AJAX or Fetch API is often used to communicate with the backend server, enabling seamless interactions like fetching menu items from a database or submitting orders. Additionally, JavaScript can store cart data temporarily using local storage or session storage, allowing users to navigate the site without losing their selections. It also powers user feedback mechanisms, such as displaying order status, success messages, or error prompts in real-time. Integrating JavaScript with a payment gateway further enables secure and efficient payment processing, creating a responsive, interactive, and user-friendly food ordering platform.

1.4. MERN Stack

- **MongoDB**
  - A NoSQL database that stores data in flexible, JSON-like documents.
  - Provides high scalability and performance for handling large datasets.
  - Ideal for dynamic applications requiring frequent updates to data structures.

- **Express.js**
  - A minimal and flexible Node.js web application framework.
  - Facilitates the creation of RESTful APIs to manage client-server communication.
  - Supports middleware functions for streamlined request handling and routing.

- **React**
  - A JavaScript library for building user interfaces, developed by Facebook.
  - Utilizes a component-based architecture for better code organization and reusability.
  - Features a virtual DOM for efficient rendering and improved performance.
- **Node.js**
  - A JavaScript runtime built on Chrome's V8 engine for server-side programming.
  - Enables the creation of scalable network applications that handle multiple requests simultaneously.
  - Allows for the use of JavaScript across both frontend and backend, promoting efficiency.

CHAPTER 2

METHODOLOGY

2.1 Objective

The main objective is to build a fully functional food ordering system using the MERN stack. The application must allow users to browse menu items, manage their orders, and complete purchases seamlessly.

2.2 Problem Statement

The problem statement given to us is to develop an food ordering system website using the MERN stack, focused on selling and managing products online.

2.3 Block Diagram

2.3.1 Frontend

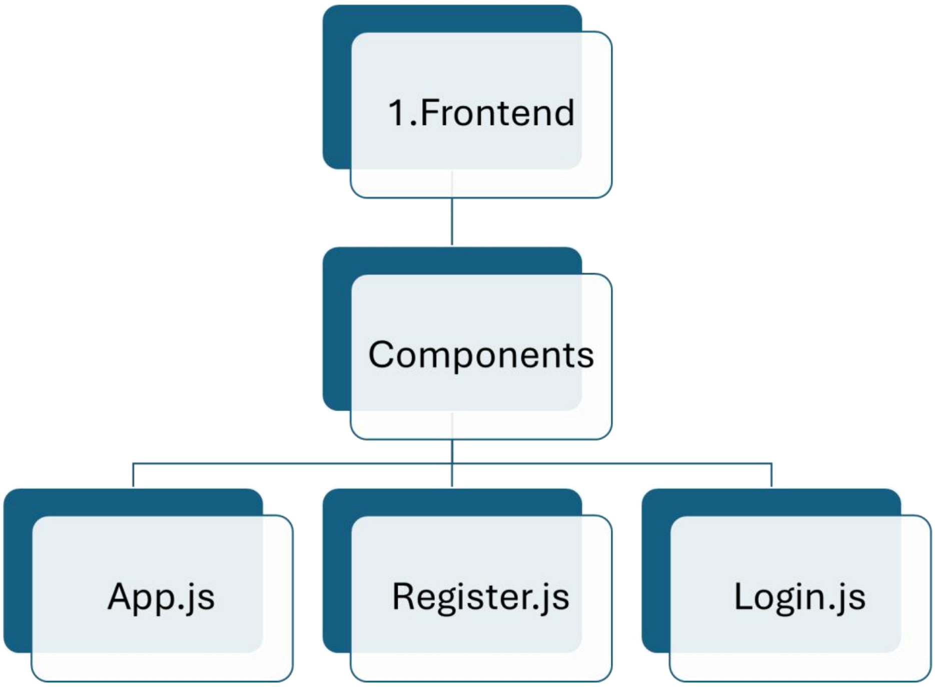


Figure 2.1 Block Diagram for frontend

2.3.2 Bakend And Routes

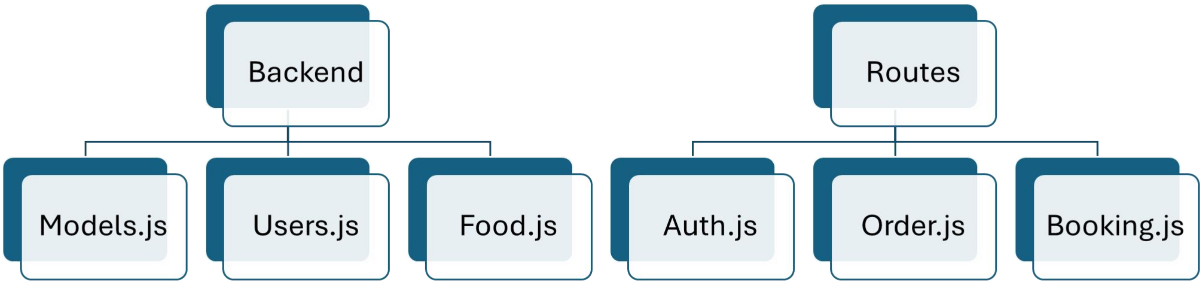


Figure 2.2 Block Diagram for backend and routes

2.4 Module Explanation

1. Login page

2. Signup page

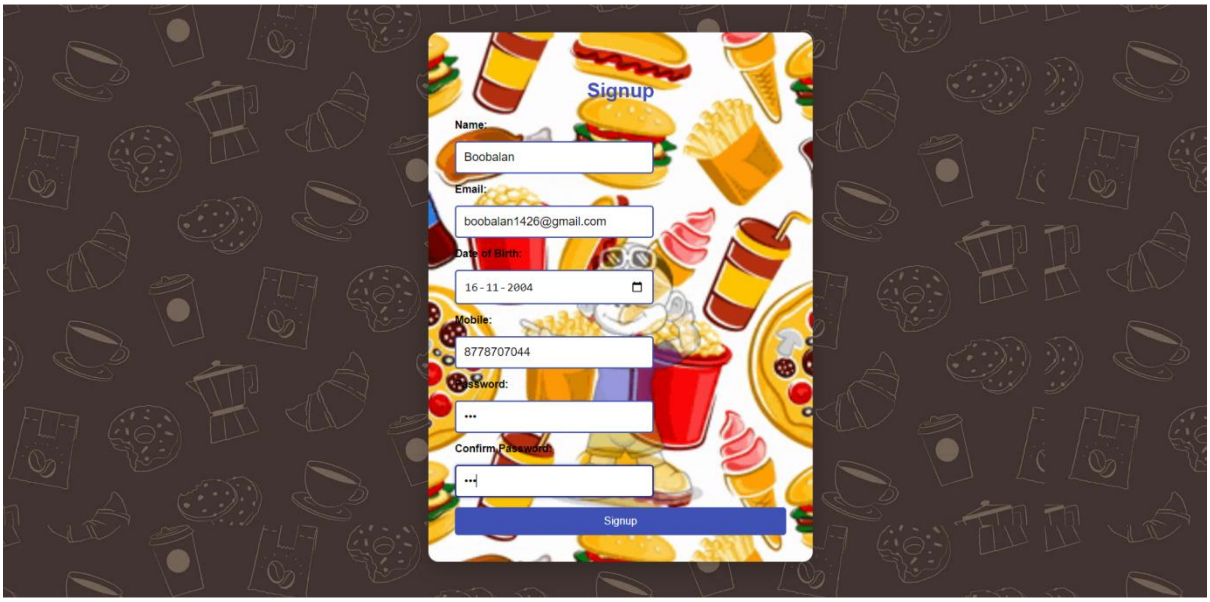
3. Add to cart page

4. Setup page

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Screenshots



Name: Boobalan

Email: boobalan1426@gmail.com

Date of Birth: 16-11-2004

Mobile: 8778707044

Password: •••

Confirm Password: •••

Figure 3.1 Sign Up

The sign-up page (**Figure 3.1**) enables new customers to create an account by providing essential information, such as name, email, and password. It ensures a straightforward registration process, allowing users to access the system and place orders easily.

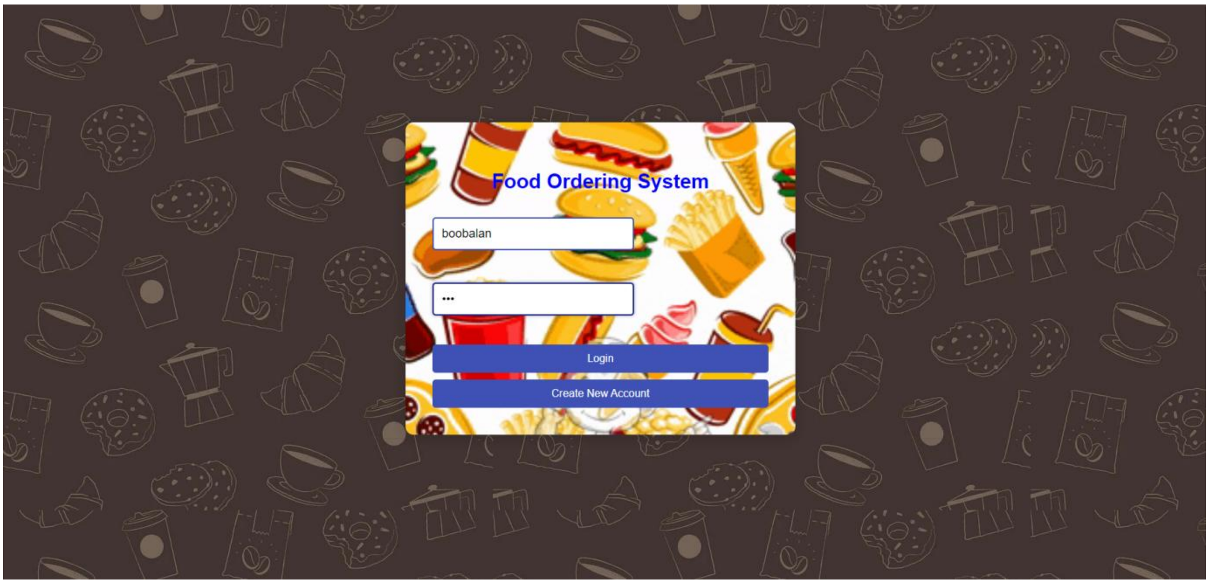


Figure 3.2 Login Page

The login page (**Figure 3.3)** Menu allows customers to securely access their accounts by entering their credentials, such as email and password. It provides options for password recovery and links to the registration page for new users.

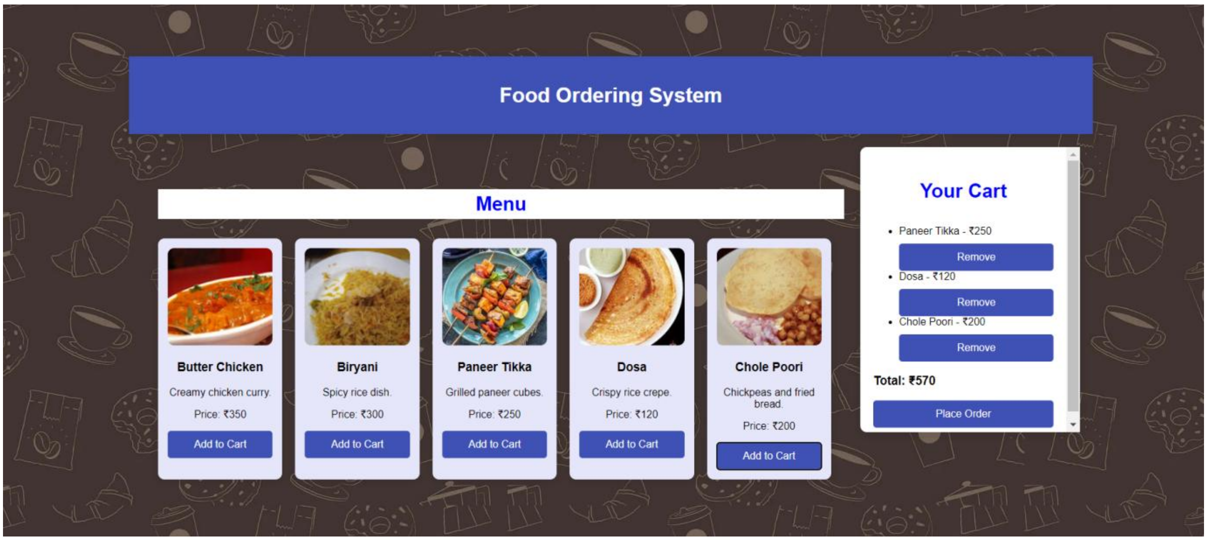
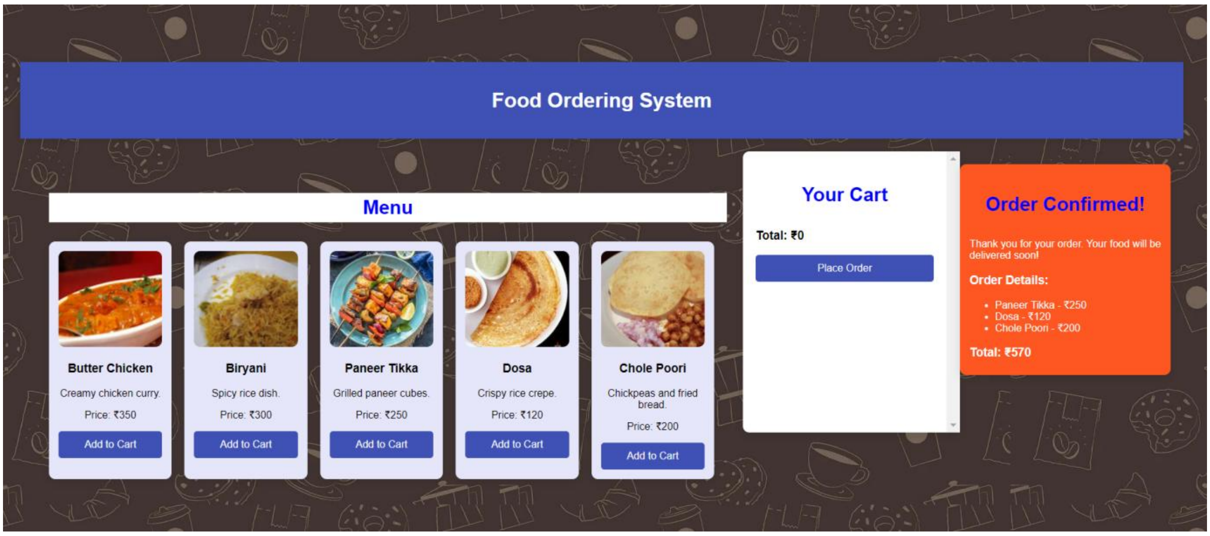


Figure 3.3 Menu

The menu page (**Figure 3.3)** Menu displays a comprehensive list of available food items, complete with descriptions, prices, and options to customize orders. Users can easily browse through categories and add their desired items to the cart for convenient online ordering.



Butter Chicken
Creamy chicken curry.
Price: ₹350

Biryani
Spicy rice dish.
Price: ₹300

Paneer Tikka
Grilled paneer cubes.
Price: ₹250

Dosa
Crispy rice crepe.
Price: ₹120

Chole Poori
Chickpeas and fried bread.
Price: ₹200

Your Cart

Total: ₹0

Order Confirmed!

Thank you for your order. Your food will be delivered soon!

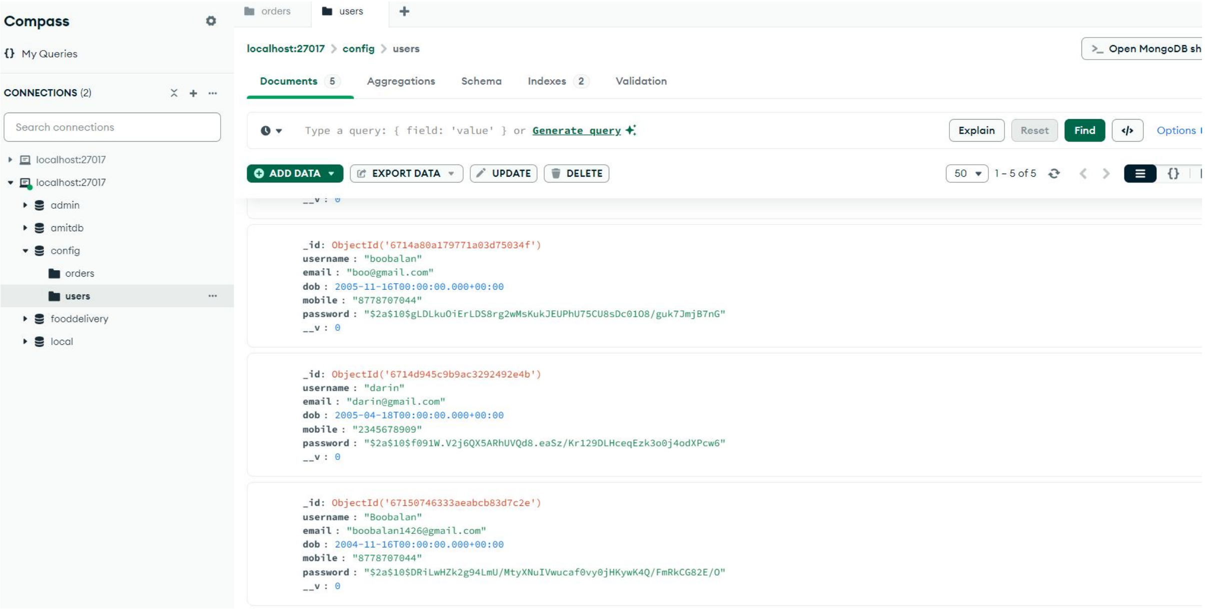
Order Details:

- Paneer Tikka - ₹250
- Dosa - ₹120
- Chole Poori - ₹200

Total: ₹570

Figure 3.4 Add To Cart

The add-to-cart page (**Figure 3.4**) allows users to review selected food items before finalizing their orders, displaying details such as item names, quantities, and prices. It provides options to modify quantities or remove items, ensuring an easy and efficient checkout process.



```
__v : 0

_id: ObjectId('6714a80a179771a03d75034f')
username : "boobalan"
email : "boo@gmail.com"
dob : 2005-11-16T00:00:00.000+00:00
mobile : "8778707044"
password : "$2a$10$gLDLkuOiErLDS8rg2wMsKukJEUPhU75CU8sDc0108/guk7JmjB7nG"
__v : 0

_id: ObjectId('6714d945c9b9ac3292492e4b')
username : "darin"
email : "darin@gmail.com"
dob : 2005-04-18T00:00:00.000+00:00
mobile : "2345678909"
password : "$2a$10$f091W.V2j6QX5ARhUVQd8.eaSz/Kr129DLHceqEzk3o0j4odXPcw6"
__v : 0

_id: ObjectId('67150746333aeabcb83d7c2e')
username : "Boobalan"
email : "boobalan1426@gmail.com"
dob : 2004-11-16T00:00:00.000+00:00
mobile : "8778707044"
password : "$2a$10$DRiLwHZk2g94LmU/MtyXNuIVwucaf0vy0jHKywK4Q/FmRkCG82E/O"
__v : 0
```

Figure 3.5 Backednd Connection

The backend connection page (**Figure 3.5**) manages the communication between the frontend and server, facilitating data exchange for menu items, user authentication, and order processing. It ensures secure API interactions, enabling the seamless functionality of the food ordering system.

3.2 Results

Thus, we were able to create and implement a storyteller Website with the MERN Stack structure

Frontend and Backend components:

1. **Frontend:**
    - **Components**: Register,Login,Home,Cart,Order confirmatiom (JS files).
2. **Backend:**
    - **Controllers**: Handle backend logic (items and User controllers).
    - **Models**: Define the database schemas (items and User models in MongoDB).
    - **Routes**: Define API endpoints for itmes and users (Express.js routes).

This setup supports user authentication, items display, and essential features like login, register, and food items management.

CHAPTER 4

CONCLUSION

In conclusion, the developed food ordering system is well-structured with a clear separation between the frontend and backend. The frontend, built with React, utilizes components to deliver a dynamic and interactive user experience for browsing menus and placing orders. The backend, managed through controllers, models, and routes using Node.js and Express.js, ensures efficient order processing, secure user authentication, and effective management of food items. This architecture, leveraging the MERN stack, provides the scalability, flexibility, and robust foundation necessary for creating a modern and feature-rich food ordering platform.

References

1. **MDN Web Docs -** Comprehensive guide for web technologies like HTML, CSS, and JavaScript.
   MDN Web Docs
2. **React Official Documentation -** Learn about React components and best practices for building frontend applications.
   React Docs
3. **MERN Stack Guide -** Learn how to create a full-stack project using MongoDB, Express.js, React, and Node.js.
   MERN Guide
4. **Tailwind CSS Documentation -** For utility-first CSS framework used in the frontend.
   Tailwind CSS Docs
5. **Node.js Official Docs -** Backend technology for building scalable applications.
   Node.js Documentation