RESTAURANT TABLE BOOKING WEBSITE

A PROJECT REPORT

Submitted by

GROUP ID : BCA22J007

Comprising of

Roll Number	Registration Number	Student Code	Student Name
22010301475	22013002926 of 2022-2023	BWU/BCA/22/564	Sneha Singh
22010301471	22013002922 of 2022-2023	BWU/BCA/22/558	Purnojit Kayal
22010301469	22013002920 of 2022-2023	BWU/BCA/22/556	Priya Mahapatra
22010301459	22013002909 of 2022-2023	BWU/BCA/22/545	Souvik Dutta
22010301474	22013002925 of 2022-2023	BWU/BCA/22/561	Subhradip Acharya
22010301461	22013002912 of 2022-2023	BWU/BCA/22/548	Subarna Modak

in partial fulfillment for the award of the degree

of

BACHELOR OF COMPUTER APPLICATIONS

Department of Computational Sciences

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125
January 2025

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

Department of Computational Sciences

<u>BONAFIDE CERTIFICATE</u>

Certified that this project report **"ONLINE RESTAURANT TABLE BOOKING"** is the bonafide work of "**Sneha Singh, Purnojit Kayal, Souvik Dutta, Priya Mahapatra, Subarna Modak, Subhradip Acharya**" who carried out the project work under my supervision.

--	--
SIGNATURE	**SIGNATURE**
Dr. Jayanta Aich	Mr. Souvik Bera
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Associate Professor Department of computational sciences University Building 6, Room. 306, Brainware University, Barasat, West Bengal, India	Asst. Professor Department of computational sciences University Building 6, Room. 305, Brainware University, Barasat, West Bengal, India

--	---
External Examiner	**Internal Examiner**

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who supported and guided us throughout the successful completion of our group project, “Online Restaurant Table Booking.”

First and foremost, we are deeply thankful to our respected project supervisor, Dr. Souvik Bera, for his continuous encouragement, expert guidance, and insightful feedback. His mentorship was instrumental in refining our approach and bringing this project to its final form.

We extend our sincere appreciation to the Department of Computational Sciences and BRAINWARE UNIVERSITY for providing the resources, technical support, and a collaborative environment essential for the execution of this project.

We are also grateful to our Head of the Department, Dr. Jayanta Aich, for his leadership and constant encouragement throughout our academic journey. Our sincere thanks to all the faculty members whose teachings and guidance shaped our knowledge and skills.

We would like to thank our peers and friends for their support, feedback, and positive motivation that helped us remain focused and committed throughout the development phase.

Last but not least, we express our deepest gratitude to our families for their unwavering love, patience, and belief in us, which gave us the strength to accomplish this project.

Thank you all for being an integral part of our journey and for contributing to the successful completion of Online Restaurant Table Booking.

SNEHA SINGH

PURNOJIT KAYAL

SOUVIK DUTTA

PRIYA MAHAPATRA

SUBARNA MODAK

SUBHRADIP ACHARYA

ABSTRACT

Online Restaurant Table Booking is a user-centric and efficient online food ordering platform designed to simplify the process of browsing, ordering, and tracking food delivery in real time. The system serves both customers and restaurant administrators through dedicated interfaces that ensure smooth interaction and transparent status updates.

Developed using modern web technologies including HTML, CSS, JavaScript(React framework) for the front end, Node.js and Express.js for the back end, and MongoDB for data management, Online Restaurant Table Booking supports a full-stack architecture optimized for responsiveness and scalability. The platform allows users to log in or sign up, explore the menu, place orders with delivery details, and monitor the delivery status all of which are reflected instantly on the admin panel.

Key features include real-time order processing, dynamic delivery tracking, and an intuitive user experience for both customers and admins. Future enhancements aim to introduce features such as live delivery tracking via maps, loyalty programs, and AI-powered menu recommendations to enrich user engagement and operational efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

In today's fast-paced digital era, online food ordering systems have revolutionized the way people interact with restaurants and food services. The demand for convenient, efficient, and user-friendly platforms for ordering food has grown significantly, especially in urban areas where time-saving solutions are highly valued.

Recognizing this need, our team has developed *Online Restaurant Table Booking,* a web-based food ordering system designed to streamline the process of browsing menus, placing orders, and tracking delivery status — all from a single platform.

Online Restaurant Table Booking aims to bridge the gap between customers and restaurant management by offering a seamless experience for both ends. Customers can easily register or log in, explore the menu, place their desired orders, and track delivery status. Meanwhile, the admin panel provides restaurant staff with real-time updates on customer orders and allows them to manage and update delivery statuses accordingly.

The project combines an intuitive user interface with efficient backend processes to ensure smooth operation and accurate order handling. By automating order management and enabling real-time communication between users and administrators, *Online Restaurant Table Booking* serves as a modern solution to the traditional dine-in or call-in ordering experience.

CHAPTER-2

OBJECTIVE

1. **To develop a user-friendly online platform**
   Enable customers to seamlessly browse menus, register/login, and place food orders with minimal effort and maximum clarity.

2. **To streamline order processing for administrators**
   Provide a real-time admin dashboard where staff can view incoming orders, update delivery status, and manage operations efficiently.

3. **To implement a responsive and accessible web design**
   Ensure that the website works smoothly across various devices, offering a consistent and intuitive user experience for all users.

4. **To facilitate real-time delivery tracking**
   Allow customers to track the status of their orders and receive timely updates, increasing transparency and trust in the service.

5. **To create a scalable and maintainable system**
   Design the website's backend and database architecture in a way that supports future feature upgrades and increased user traffic.

CHAPTER-3

PLANNING

To ensure the successful execution of the *Online Restaurant Table Booking* web application, our team followed a clear and systematic planning approach divided into four key phases. Each phase was designed to address different aspects of the project, from understanding the market to launching the final product.

1. Market Research and Analysis

Before initiating development, we conducted thorough research on existing food ordering platforms to identify user expectations, popular features, and market gaps. This helped us shape our own platform to meet user needs while offering a competitive edge. We also analyzed user behavior trends to design a more intuitive and appealing user experience.

2. Design and Development

In this phase, we created UI/UX designs for both customers and administrators. Wireframes and prototypes were developed to visualize the layout and flow. The development process was divided into modules such as user authentication, menu browsing, order placement, admin panel, and delivery tracking. Frontend and backend development progressed in parallel to ensure efficiency.

3. Testing and Feedback

After completing the initial development, we performed rigorous testing to identify and fix bugs, ensure data accuracy, and validate all functionalities. We gathered feedback from peers and potential users through demo sessions, which helped us improve usability and fix overlooked issues before final deployment.

4. Launch and Marketing

Once the system was stable and thoroughly tested, we prepared it for deployment on a live hosting environment. A basic marketing plan was created to introduce the platform to potential users through social media posts, word-of-mouth, and demos. This phase also included user onboarding and monitoring initial usage to gather further insights.

CHAPTER-4

REQUIREMENT ANALYSIS

Requirement analysis is a crucial step in the software development lifecycle, as it helps define the functionalities and expectations of the system from both the user and administrator perspectives. For *Online Restaurant Table Booking*, we carefully analyzed functional and non-functional requirements to ensure the platform meets the needs of all stakeholders.

1. Functional Requirements

a) User Side:

- **User Registration and Login:** Users should be able to sign up and log in securely.
- **Menu Browsing:** Users can explore a categorized food menu with item names, prices, and images.
- **Order Placement:** Users can select items, fill in personal and delivery details, and place orders.
- **Order Status Tracking:** Users can view the current status of their placed orders (e.g., Preparing, Out for Delivery).
- **User Profile Management:** Users should be able to view and update their personal information.

b) Admin Side:

- **Admin Login:** A secure login system for administrators.
- **Order Management:** Admins can view all customer orders in real-time.
- **Delivery Status Update:** Admins can update each order’s delivery status to keep customers informed.
- **Menu Management (optional):** Admins can update or modify the menu items (if this feature is included).

2. Non-Functional Requirements

- **Responsiveness:** The website should be accessible and functional across different devices and screen sizes.
- **Performance:** The system should load quickly and handle multiple simultaneous users without lag.
- **Security:** User data, especially login credentials and order information, must be securely handled and stored.
- **Scalability:** The platform should be able to handle growth in users and menu items in the future.
- **Usability:** The interface should be intuitive and easy to navigate for both users and admins.

3. System Requirements

- **Frontend Technologies:** HTML, CSS, JavaScript (and optionally frameworks like React or Bootstrap).
- **Backend Technologies:** PHP, Node.js, or Python (based on your actual stack).
- **Database:** MySQL, MongoDB, or any other suitable database for storing user, menu, and order data.
- **Hosting:** A reliable web hosting platform or local server setup for deployment and testing.

CHAPTER-5

SYSTEM FLOW (FLOWCHART)

The system flow of *Online Restaurant Table Booking* outlines the step-by-step interaction between the user, the system, and the admin. It represents how data moves through the platform, from the moment a user accesses the website to the final delivery of an order. This logical flow ensures smooth coordination between all modules of the system.

1. User Interaction Flow:

- **Start** → The user opens the *Food Mania* website.
- **Login/Sign Up** → If the user is new, they sign up; existing users log in using credentials.
- **Browse Menu** → The user views the available food items categorized for easy access.
- **Add to Cart** → Selected food items are added to the cart for review.
- **Enter Delivery Details** → The user fills out necessary delivery information (name, address, phone).
- **Place Order** → The user confirms and places the order.
- **Order Confirmation** → A success message is shown, and order data is sent to the admin panel.

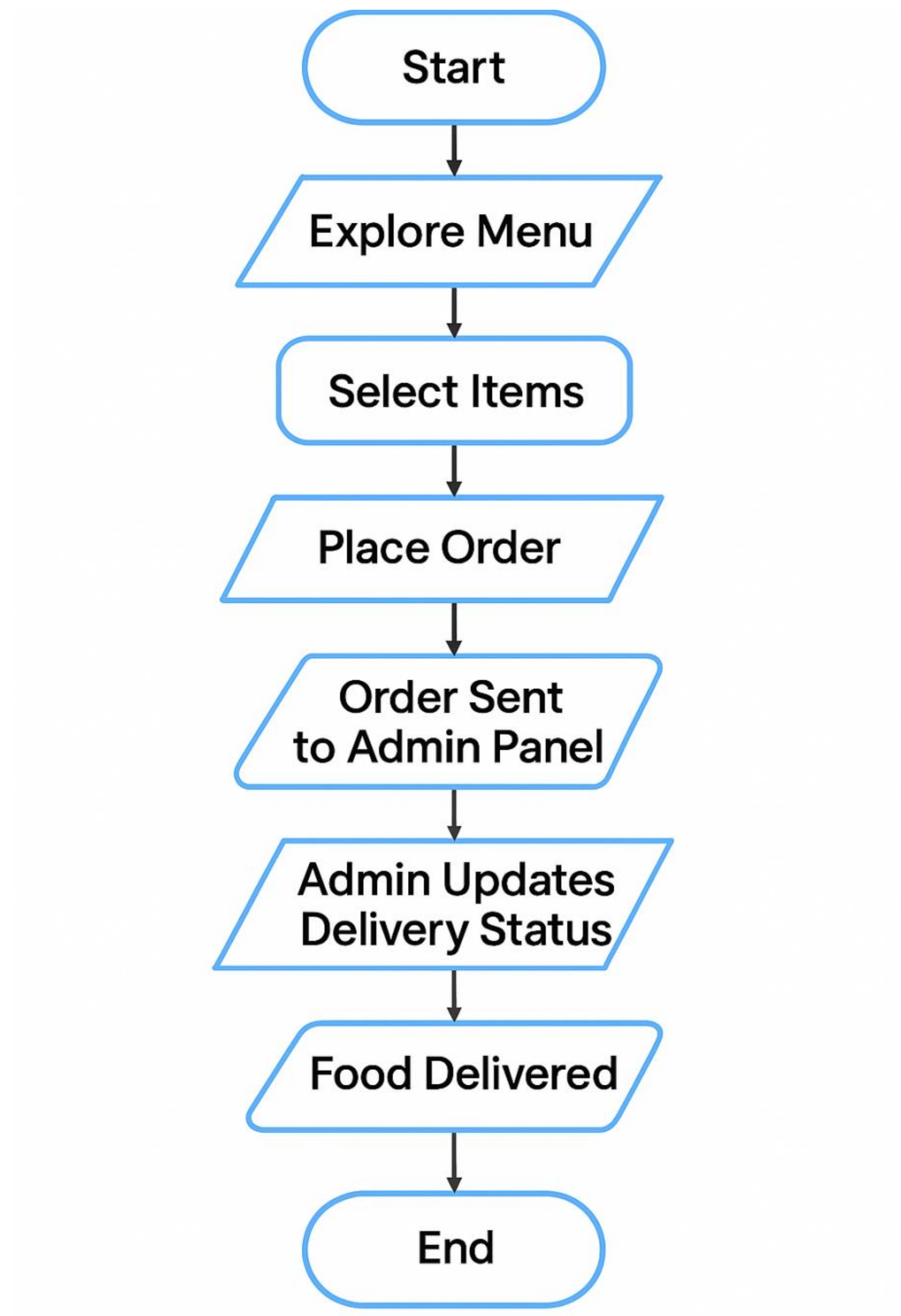
2. Admin Interaction Flow:

- **Admin Login** → Admin securely logs into the admin panel.
- **View Orders** → The admin receives and views all new customer orders in real-time.
- **Process Order** → The admin begins preparing the food based on the order details.
- **Update Delivery Status** → The admin updates the status (e.g., Preparing, Out for Delivery, Delivered).
- **Status Sync** → The updated status is reflected in the user's interface for tracking.

3. Order Completion:

- Once the food is delivered and the status is marked as **Delivered**, the order is archived.
- **End of Process** → Both user and admin may view past order history for reference.

This flow ensures that all users experience a logical and seamless process, while the admin can efficiently manage orders and keep users informed in real-time.



Flow of Online Food Booking Process

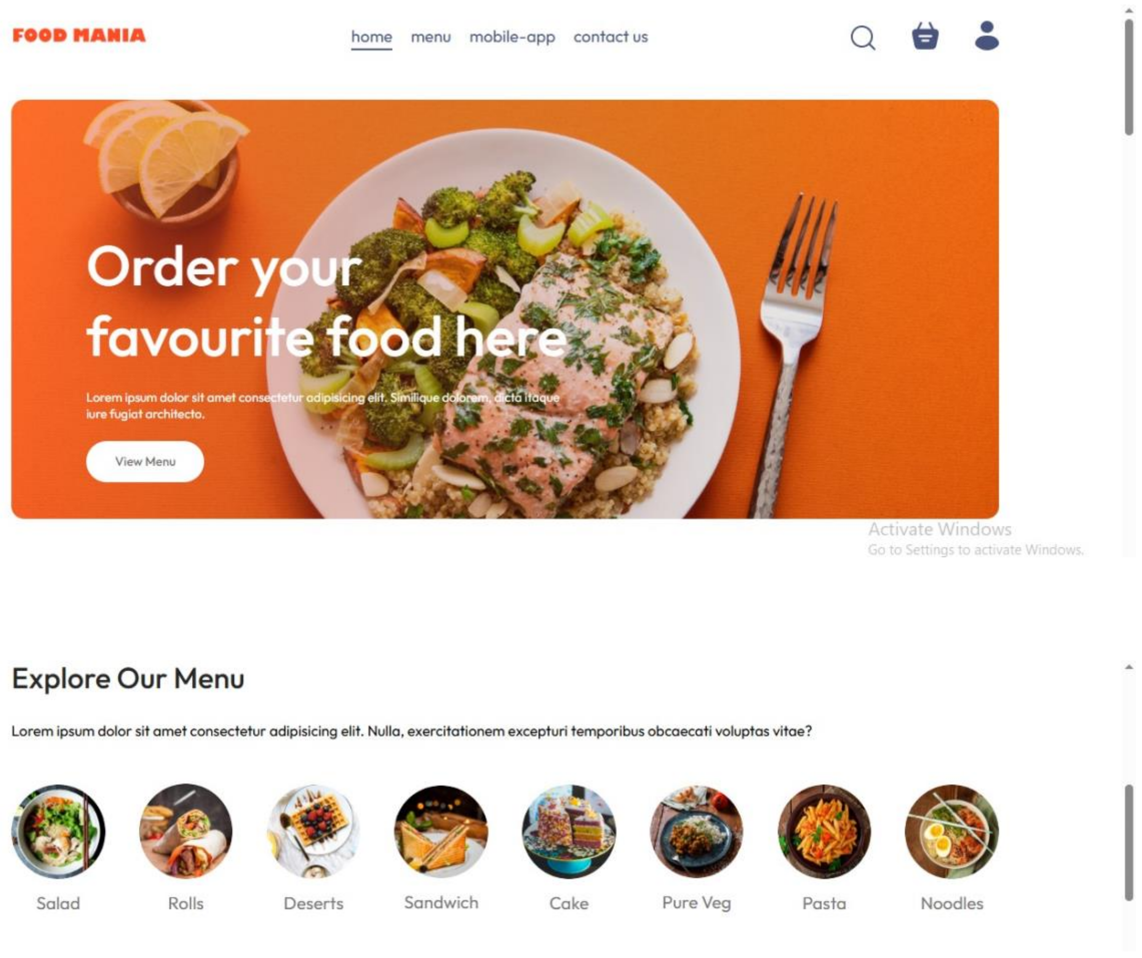
The flowchart represents the typical structure of the **online food ordering process** for customers and admin. Below is a detailed breakdown of each component:

1. **Start**
   The process begins when a user accesses the website, intending to order food.
2. **Login / Sign Up**
   The user either logs into an existing account or creates a new one to access the menu and place orders.
3. **Explore Menu**
   After successful login, the user is directed to the menu page, where they can browse through available food items and offerings.
4. **Select Items**
   The user adds one or more food items to their cart based on their preferences.
5. **Enter Delivery Details**
   Before proceeding, the user fills in necessary delivery information such as address, contact number, and any special instructions.
6. **Place Order**
   Once details are complete, the user confirms the order. The system processes the data and finalizes the request.
7. **Order Sent to Admin Panel**
   The order information is reflected on the admin panel in real-time, where it can be tracked and managed by the restaurant staff.

8. **Admin Updates Delivery Status**
   As the order progresses, the admin updates its delivery status — such as “Confirmed,” “Preparing,” “Out for Delivery,” and “Delivered.”
9. **Food Delivered**
   The food is delivered to the customer's provided address. This marks the fulfillment of the order.
10. **End**
    The process concludes here. The user may optionally leave feedback or place another order.

CHAPTER-6

PROPOSED DESIGN



Lorem ipsum dolor sit amet consectetur adipisicing elit. Similique dolorem, dicta itaque iure fugiat architecto.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nulla, exercitationem excepturi temporibus obcaecati voluptas vitae?

Landing Page

The design focuses on providing a seamless user experience using modern technologies:

- **User Interface (UI):**
    - Clean, minimal layout with interactive elements, built using **HTML** and **CSS**.
    - Dynamic content and smooth transitions powered by **React**.
    - Responsive design ensuring compatibility across devices.

- **User Experience (UX):**
    - Simple and intuitive booking flow with instant feedback using **React**.
    - A visual progress bar to guide users through the process.

- **Backend System:**
    - **MongoDB** for efficient data storage and retrieval.
    - Secure data handling with proper validation.

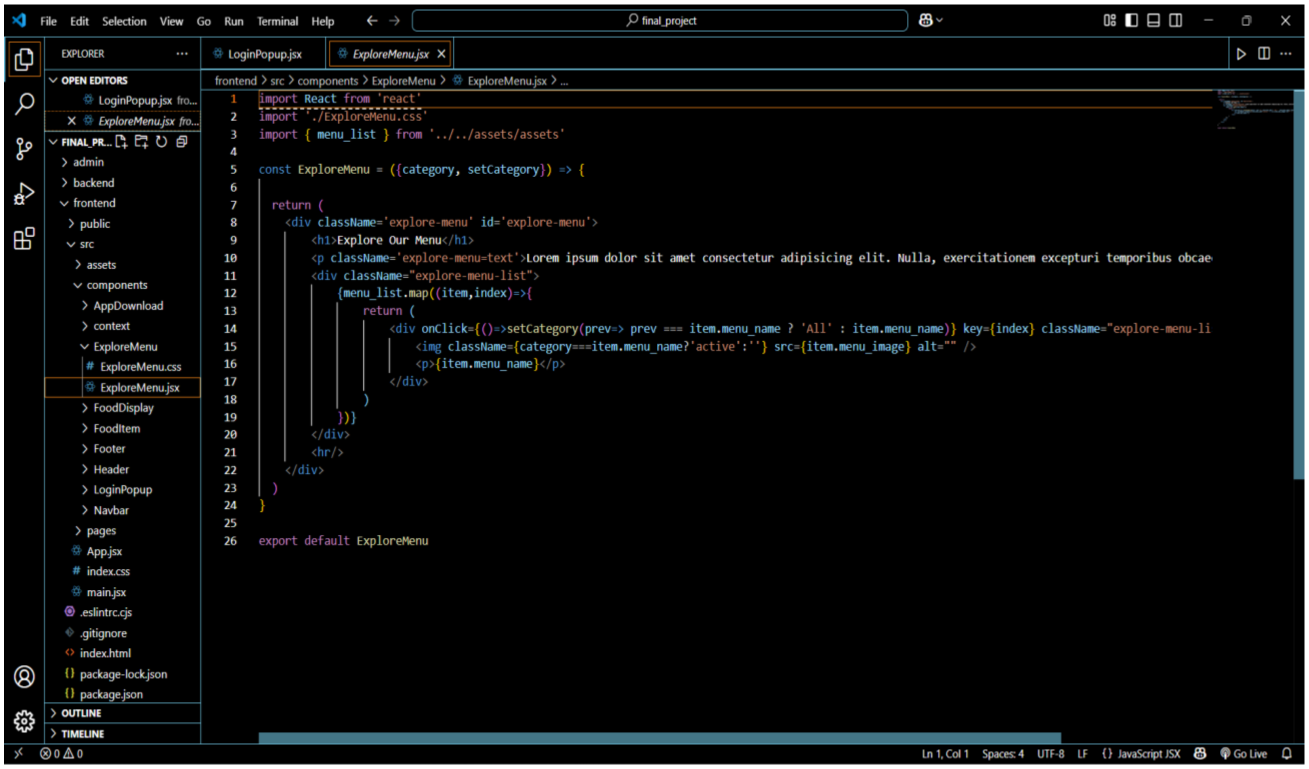
This design ensures an intuitive, responsive, and secure platform for users to easily book food and them delivered to their location.

CHAPTER-7

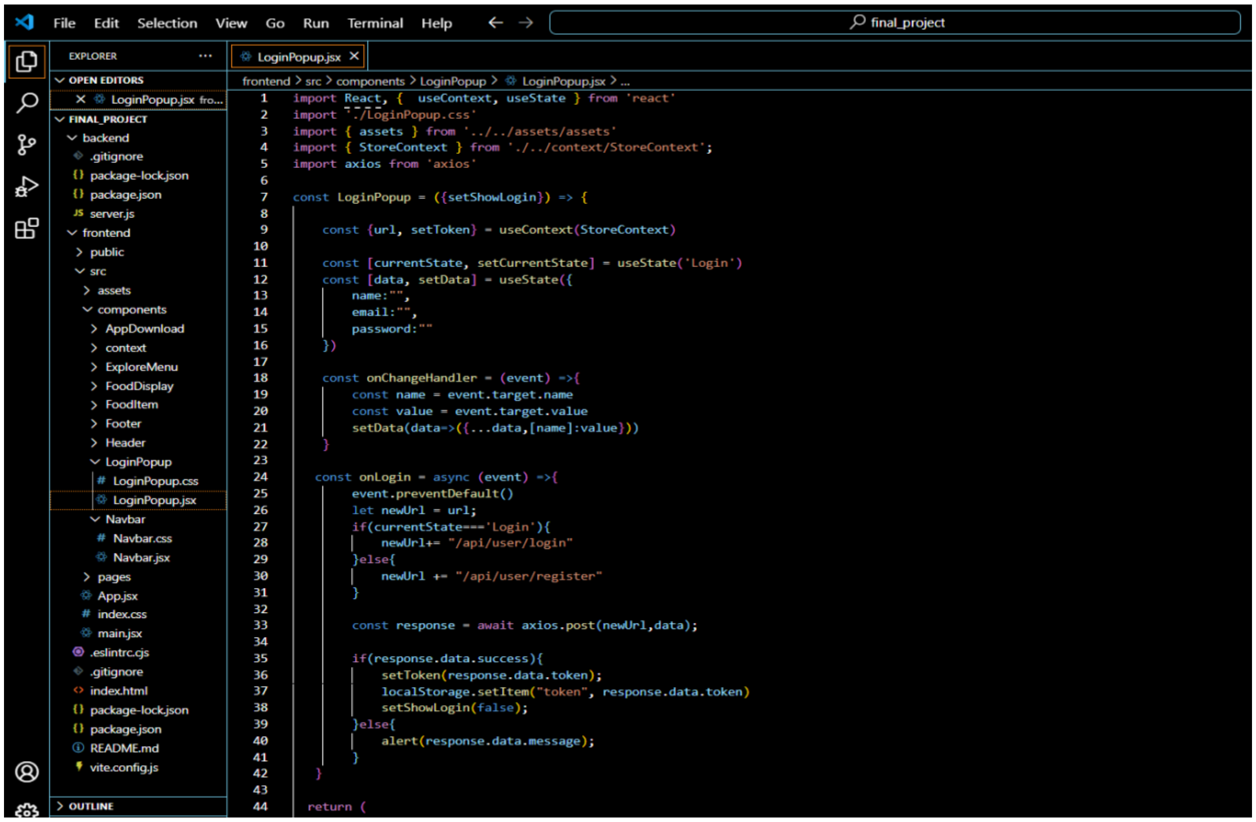
SIMPLE CORE CODE

Below is a snippet of the core code illustrating the decision-making mechanism and progress saving

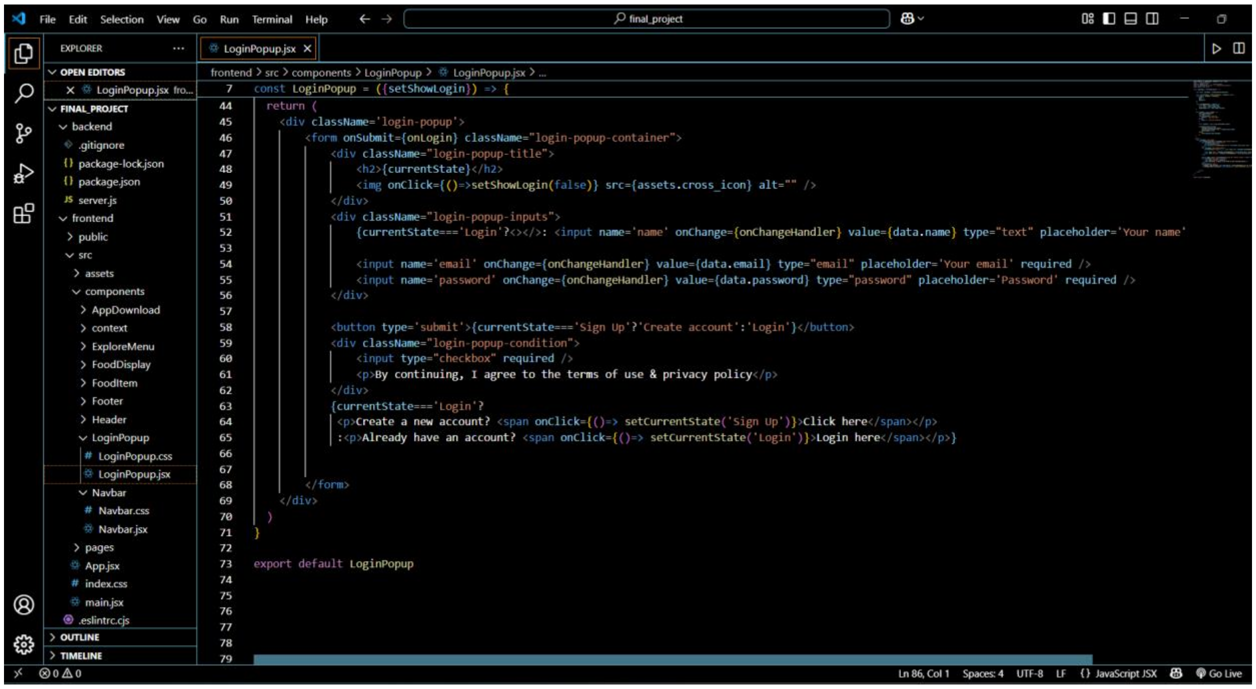
FRONTEND CODES



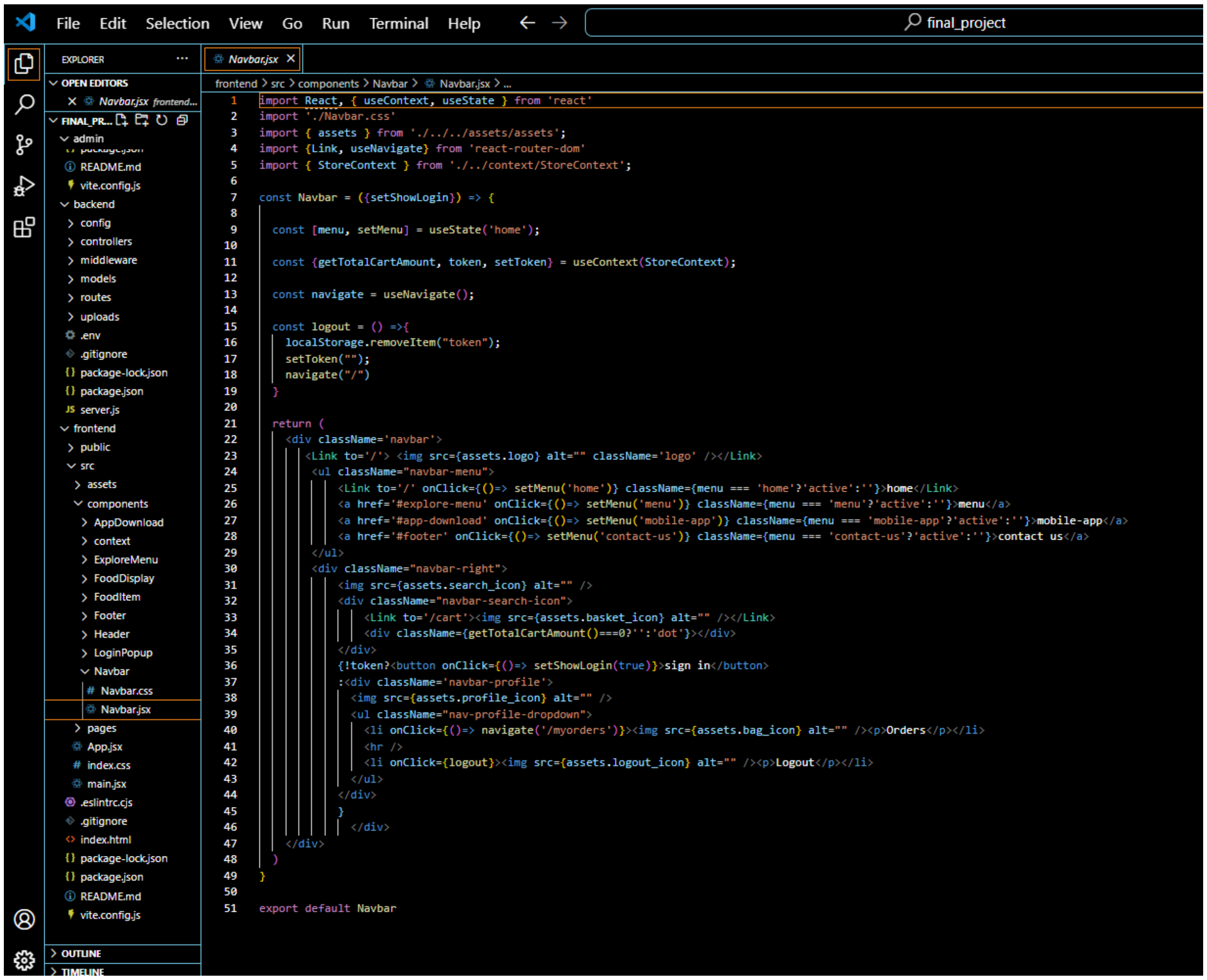
```jsx
import React from 'react'
import './ExploreMenu.css'
import { menu_list } from '../../assets/assets'

const ExploreMenu = ({category, setCategory}) => {

  return (
    <div className='explore-menu' id='explore-menu'>
        <h1>Explore Our Menu</h1>
        <p className='explore-menu=text'>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nulla, exercitationem excepturi temporibus obcae
        <div className="explore-menu-list">
            {menu_list.map((item,index)=>{
                return (
                    <div onClick={()=>setCategory(prev=> prev === item.menu_name ? 'All' : item.menu_name)} key={index} className="explore-menu-li
                        <img className={category===item.menu_name?'active':''} src={item.menu_image} alt="" />
                        <p>{item.menu_name}</p>
                    </div>
                )
            })}
        </div>
        <hr/>
    </div>
  )
}

export default ExploreMenu
```



```
import React, {  useContext, useState } from 'react'
import './LoginPopup.css'
import { assets } from '../../assets/assets'
import { StoreContext } from '../../context/StoreContext';
import axios from 'axios'

const LoginPopup = ({setShowLogin}) => {

    const {url, setToken} = useContext(StoreContext)

    const [currentState, setCurrentState] = useState('Login')
    const [data, setData] = useState({
        name:"",
        email:"",
        password:""
    })

    const onChangeHandler = (event) =>{
        const name = event.target.name
        const value = event.target.value
        setData(data=>({...data,[name]:value}))
    }

   const onLogin = async (event) =>{
        event.preventDefault()
        let newUrl = url;
        if(currentState==='Login'){
            newUrl+= "/api/user/login"
        }else{
            newUrl += "/api/user/register"
        }

        const response = await axios.post(newUrl,data);

        if(response.data.success){
            setToken(response.data.token);
            localStorage.setItem("token", response.data.token)
            setShowLogin(false);
        }else{
            alert(response.data.message);
        }
   }

  return (
```

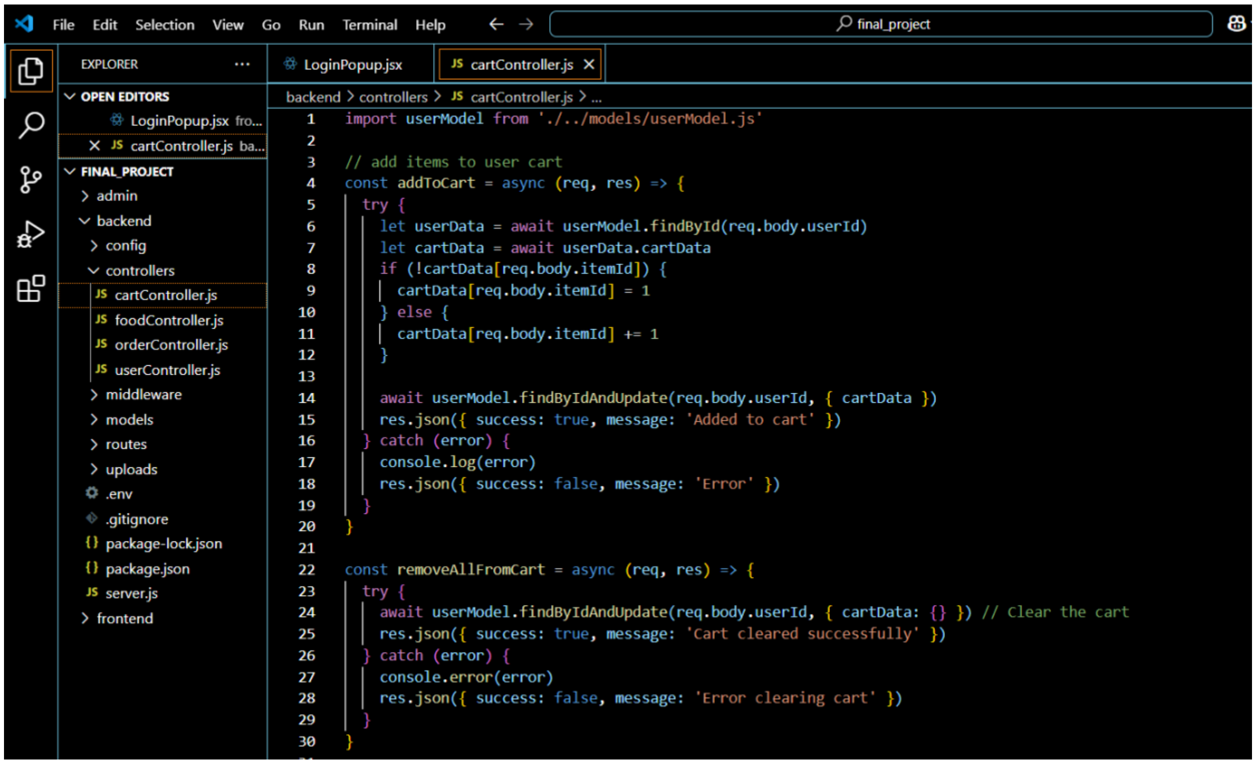


```
const LoginPopup = ({setShowLogin}) => {
  return (
    <div className='login-popup'>
        <form onSubmit={onLogin} className="login-popup-container">
            <div className="login-popup-title">
                <h2>{currentState}</h2>
                <img onClick={()=>setShowLogin(false)} src={assets.cross_icon} alt="" />
            </div>
            <div className="login-popup-inputs">
                {currentState==='Login'?<></>: <input name='name' onChange={onChangeHandler} value={data.name} type="text" placeholder='Your name'

                <input name='email' onChange={onChangeHandler} value={data.email} type="email" placeholder='Your email' required />
                <input name='password' onChange={onChangeHandler} value={data.password} type="password" placeholder='Password' required />
            </div>

            <button type='submit'>{currentState==='Sign Up'?'Create account':'Login'}</button>
            <div className="login-popup-condition">
                <input type="checkbox" required />
                <p>By continuing, I agree to the terms of use & privacy policy</p>
            </div>
            {currentState==='Login'?
             <p>Create a new account? <span onClick={()=> setCurrentState('Sign Up')}>Click here</span></p>
             :<p>Already have an account? <span onClick={()=> setCurrentState('Login')}>Login here</span></p>}


        </form>
    </div>
  )
}

export default LoginPopup
```

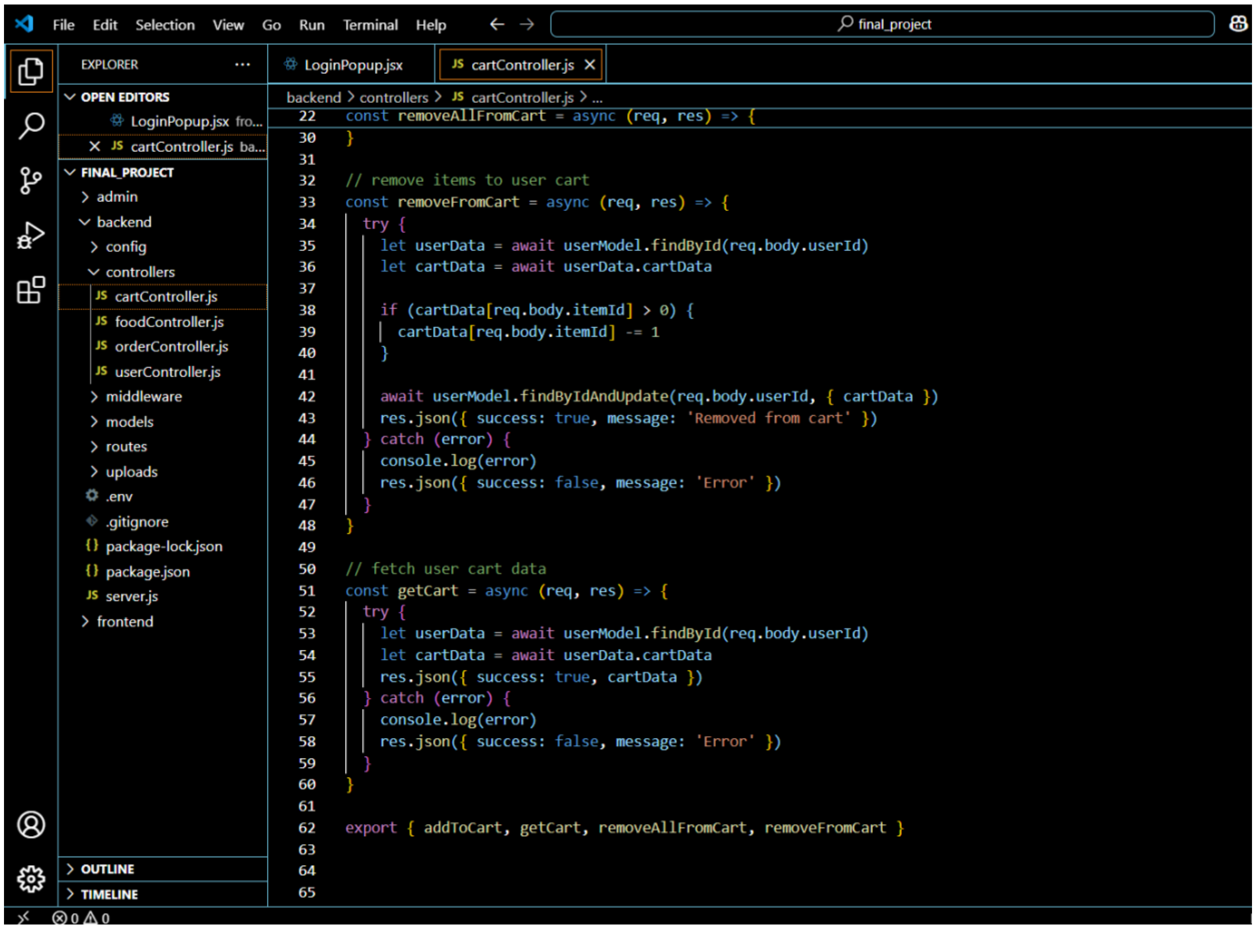


```jsx
import React, { useContext, useState } from 'react'
import './Navbar.css'
import { assets } from './../../assets/assets';
import {Link, useNavigate} from 'react-router-dom'
import { StoreContext } from './../context/StoreContext';

const Navbar = ({setShowLogin}) => {

  const [menu, setMenu] = useState('home');

  const {getTotalCartAmount, token, setToken} = useContext(StoreContext);

  const navigate = useNavigate();

  const logout = () =>{
    localStorage.removeItem("token");
    setToken("");
    navigate("/")
  }

  return (
    <div className='navbar'>
       <Link to='/'> <img src={assets.logo} alt="" className='logo' /></Link>
        <ul className="navbar-menu">
          <Link to='/' onClick={()=> setMenu('home')} className={menu === 'home'?'active':''}>home</Link>
          <a href='#explore-menu' onClick={()=> setMenu('menu')} className={menu === 'menu'?'active':''}>menu</a>
          <a href='#app-download' onClick={()=> setMenu('mobile-app')} className={menu === 'mobile-app'?'active':''}>mobile-app</a>
          <a href='#footer' onClick={()=> setMenu('contact-us')} className={menu === 'contact-us'?'active':''}>contact us</a>
        </ul>
        <div className="navbar-right">
          <img src={assets.search_icon} alt="" />
          <div className="navbar-search-icon">
            <Link to='/cart'><img src={assets.basket_icon} alt="" /></Link>
            <div className={getTotalCartAmount()===0?'':'dot'}></div>
          </div>
          {!token?<button onClick={()=> setShowLogin(true)}>sign in</button>
          :<div className='navbar-profile'>
            <img src={assets.profile_icon} alt="" />
            <ul className="nav-profile-dropdown">
              <li onClick={()=> navigate('/myorders')}><img src={assets.bag_icon} alt="" /><p>Orders</p></li>
              <hr />
              <li onClick={logout}><img src={assets.logout_icon} alt="" /><p>Logout</p></li>
            </ul>
          </div>
          }
            </div>
    </div>
  )
}

export default Navbar
```

BACKEND CODES

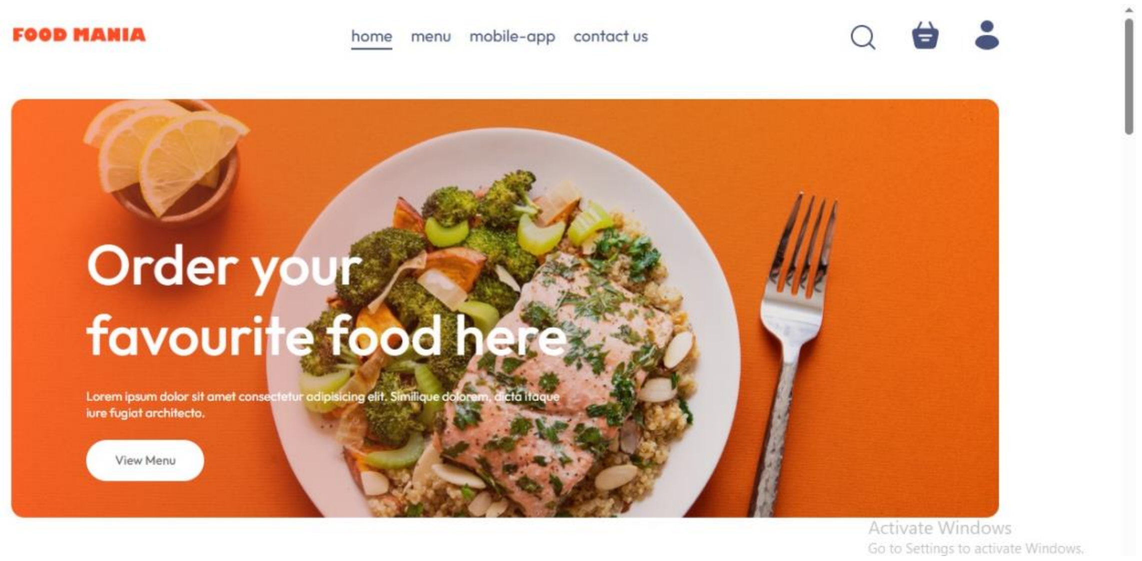


```
import userModel from './../models/userModel.js'

// add items to user cart
const addToCart = async (req, res) => {
  try {
    let userData = await userModel.findById(req.body.userId)
    let cartData = await userData.cartData
    if (!cartData[req.body.itemId]) {
      cartData[req.body.itemId] = 1
    } else {
      cartData[req.body.itemId] += 1
    }

    await userModel.findByIdAndUpdate(req.body.userId, { cartData })
    res.json({ success: true, message: 'Added to cart' })
  } catch (error) {
    console.log(error)
    res.json({ success: false, message: 'Error' })
  }
}

const removeAllFromCart = async (req, res) => {
  try {
    await userModel.findByIdAndUpdate(req.body.userId, { cartData: {} }) // Clear the cart
    res.json({ success: true, message: 'Cart cleared successfully' })
  } catch (error) {
    console.error(error)
    res.json({ success: false, message: 'Error clearing cart' })
  }
}
```

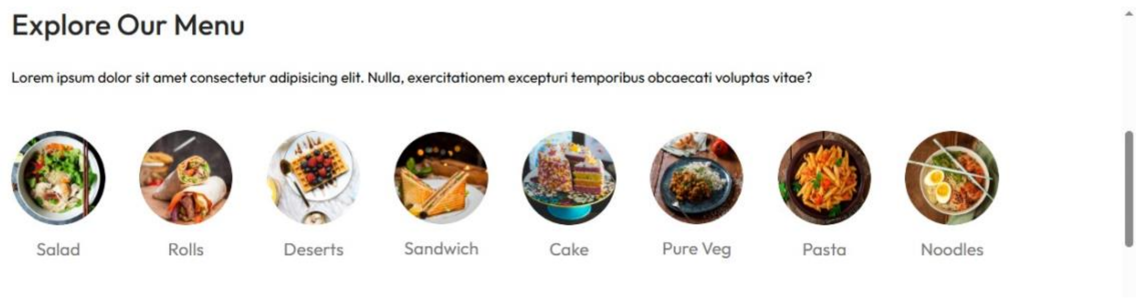


```javascript
const removeAllFromCart = async (req, res) => {
}

// remove items to user cart
const removeFromCart = async (req, res) => {
  try {
    let userData = await userModel.findById(req.body.userId)
    let cartData = await userData.cartData

    if (cartData[req.body.itemId] > 0) {
      cartData[req.body.itemId] -= 1
    }

    await userModel.findByIdAndUpdate(req.body.userId, { cartData })
    res.json({ success: true, message: 'Removed from cart' })
  } catch (error) {
    console.log(error)
    res.json({ success: false, message: 'Error' })
  }
}

// fetch user cart data
const getCart = async (req, res) => {
  try {
    let userData = await userModel.findById(req.body.userId)
    let cartData = await userData.cartData
    res.json({ success: true, cartData })
  } catch (error) {
    console.log(error)
    res.json({ success: false, message: 'Error' })
  }
}

export { addToCart, getCart, removeAllFromCart, removeFromCart }
```

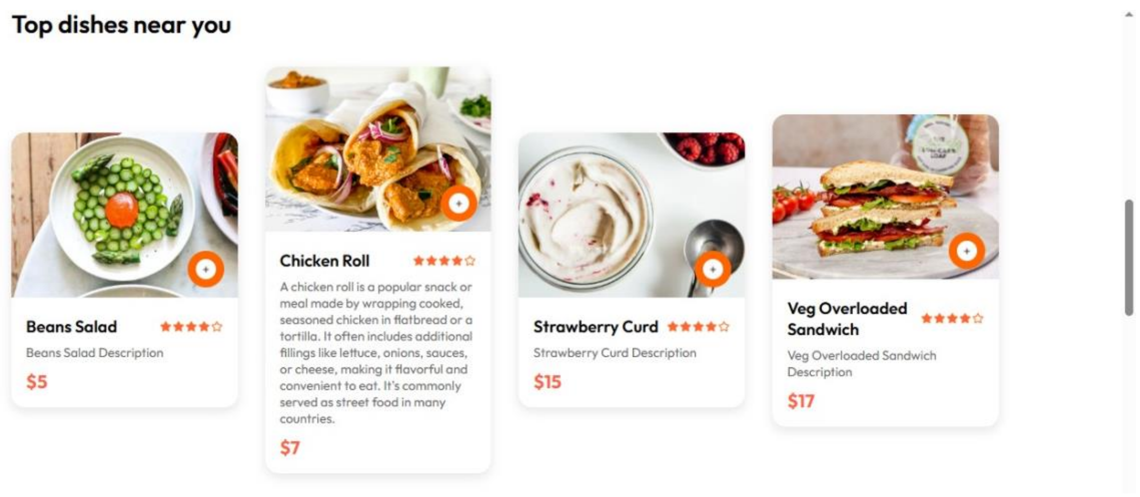
EXPERIMENTAL RESULT


FOOD MANIA
home
menu
mobile-app
contact us
Order your favourite food here
View Menu
Activate Windows
Go to Settings to activate Windows.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Similique dolorem, dicta itaque iure fugiat architecto.



Lorem ipsum dolor sit amet consectetur adipisicing elit. Nulla, exercitationem excepturi temporibus obcaecati voluptas vitae?



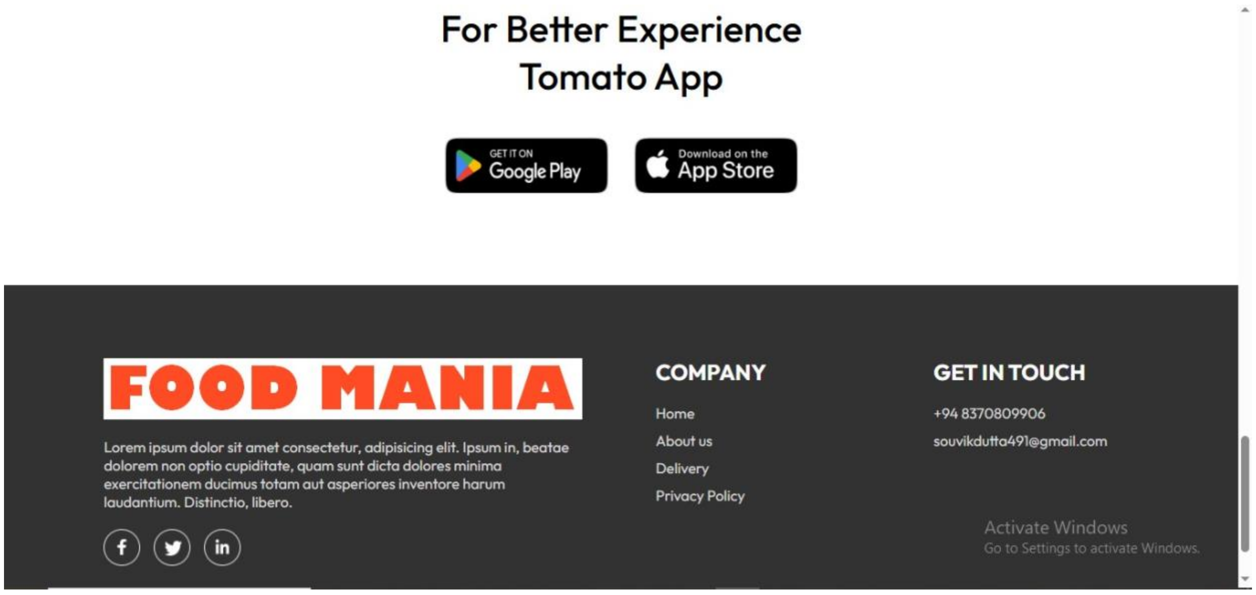
Top dishes near you

Beans Salad ★★★★☆
Beans Salad Description
$5

Chicken Roll ★★★★☆
A chicken roll is a popular snack or meal made by wrapping cooked, seasoned chicken in flatbread or a tortilla. It often includes additional fillings like lettuce, onions, sauces, or cheese, making it flavorful and convenient to eat. It's commonly served as street food in many countries.
$7

Strawberry Curd ★★★★☆
Strawberry Curd Description
$15

Veg Overloaded Sandwich ★★★★☆
Veg Overloaded Sandwich Description
$17



For Better Experience
Tomato App

FOOD MANIA

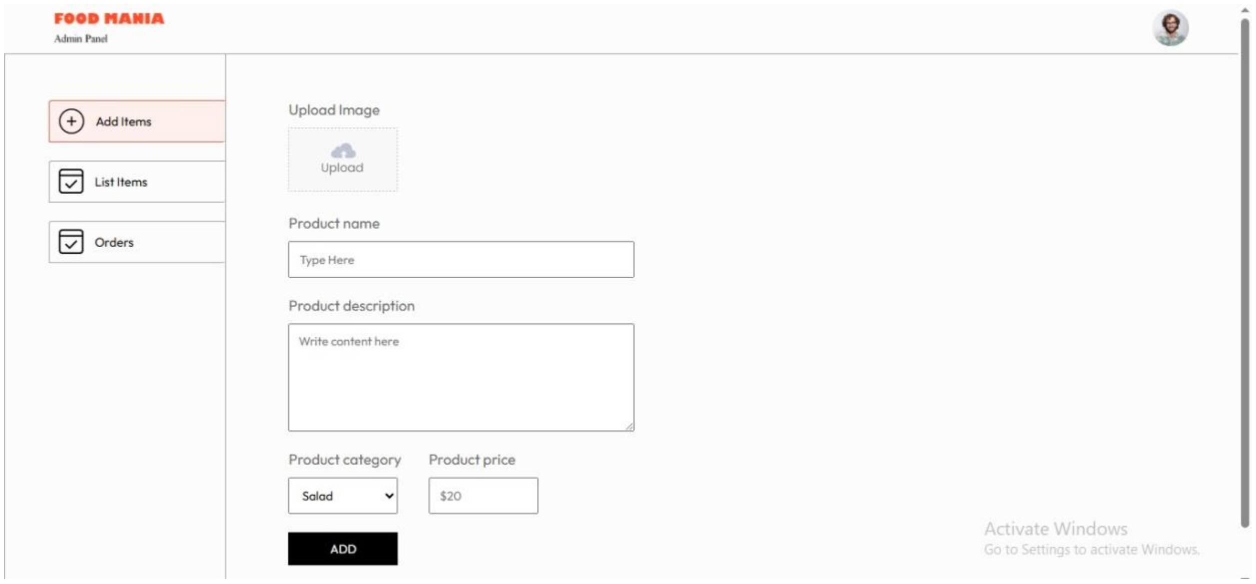
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ipsum in, beatae dolorem non optio cupiditate, quam sunt dicta dolores minima exercitationem ducimus totam aut asperiores inventore harum laudantium. Distinctio, libero.

COMPANY
Home
About us
Delivery
Privacy Policy

GET IN TOUCH
+94 8370809906
souvikdutta491@gmail.com

Activate Windows
Go to Settings to activate Windows.

ADMIN PANEL



FOOD MANIA
Admin Panel

- Add Items
- List Items
- Orders

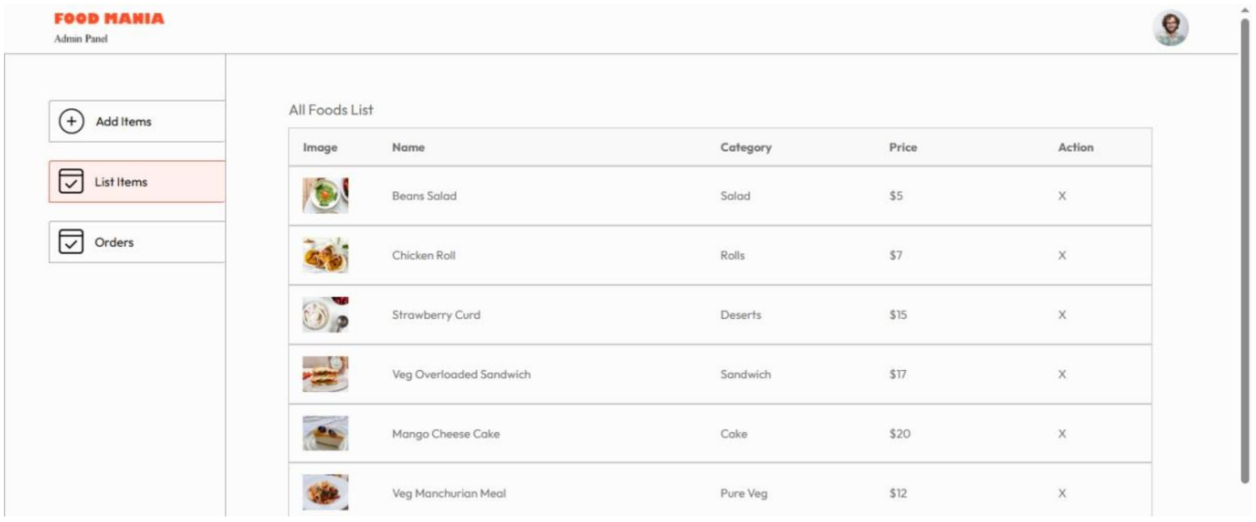
Upload Image
Upload

Product name
Type Here

Product description
Write content here

Product category: Salad
Product price: $20

ADD

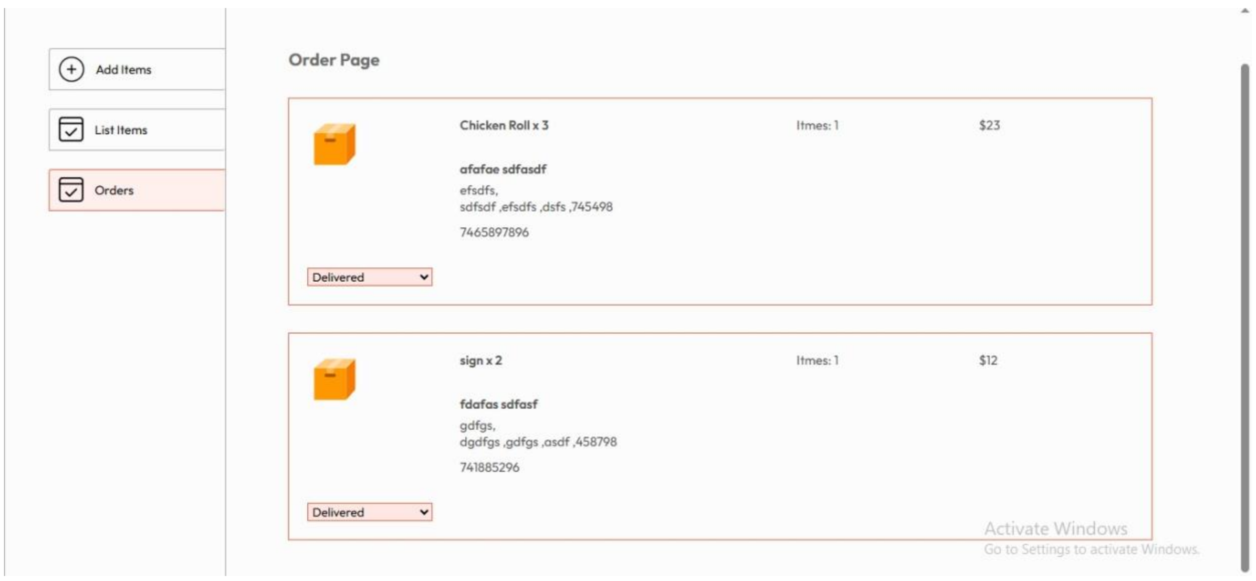


FOOD MANIA
Admin Panel

- Add Items
- List Items
- Orders

All Foods List

Image	Name	Category	Price	Action
	Beans Salad	Salad	$5	X
	Chicken Roll	Rolls	$7	X
	Strawberry Curd	Deserts	$15	X
	Veg Overloaded Sandwich	Sandwich	$17	X
	Mango Cheese Cake	Cake	$20	X
	Veg Manchurian Meal	Pure Veg	$12	X



Order Page

Chicken Roll x 3 Itmes: 1 $23

afafae sdfasdf

efsdfs,
sdfsdf ,efsdfs ,dsfs ,745498

7465897896

Delivered

sign x 2 Itmes: 1 $12

fdafas sdfasf

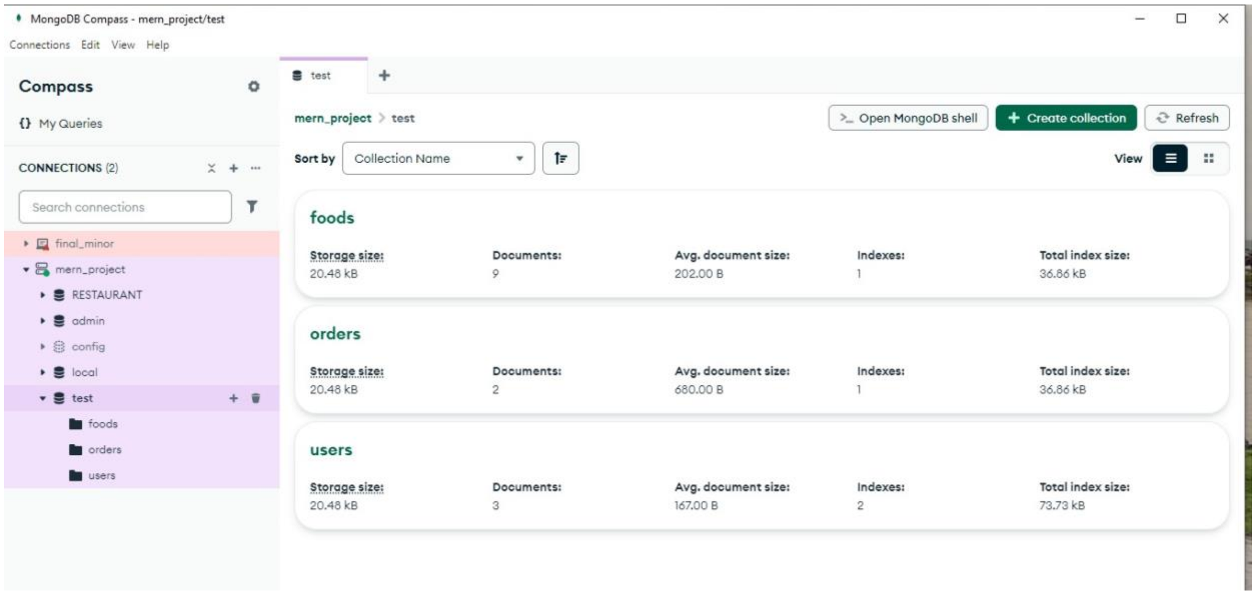
gdfgs,
dgdfgs ,gdfgs ,asdf ,458798

741885296

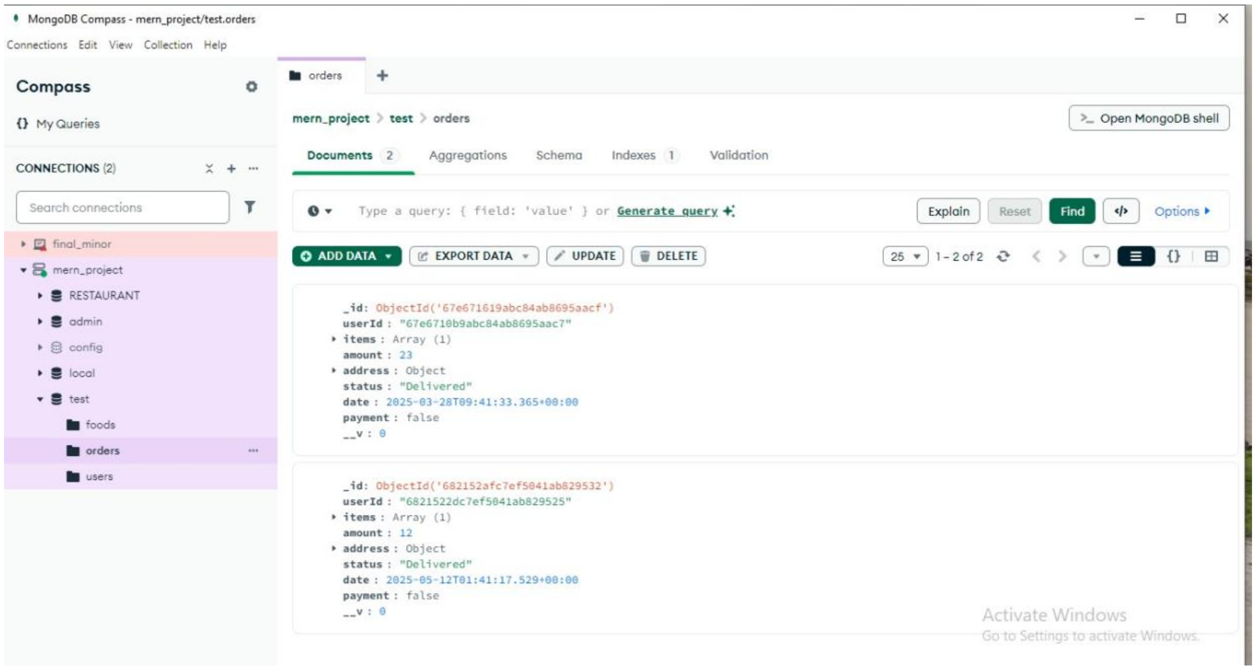
Delivered

Here, the users fill up their details in the process of booking their food online.

CUSTOMER'S INFORMATION STORED IN DATABASE

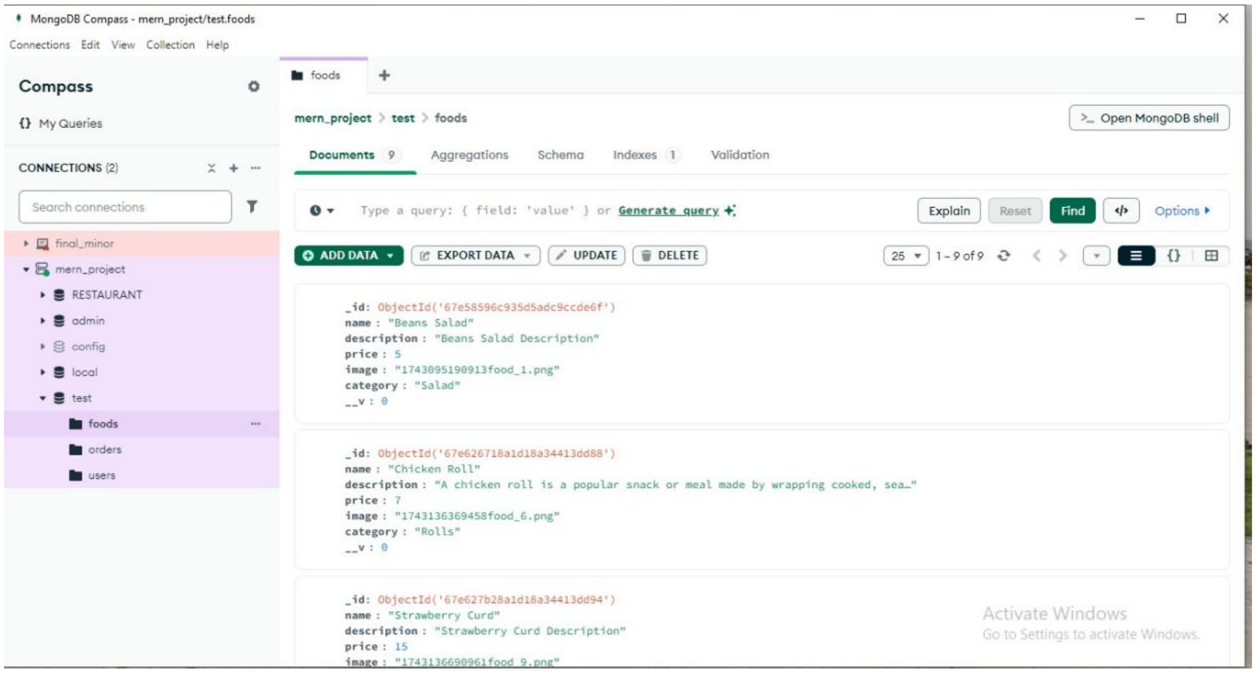


	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
foods	20.48 kB	9	202.00 B	1	36.86 kB
orders	20.48 kB	2	680.00 B	1	36.86 kB
users	20.48 kB	3	167.00 B	2	73.73 kB

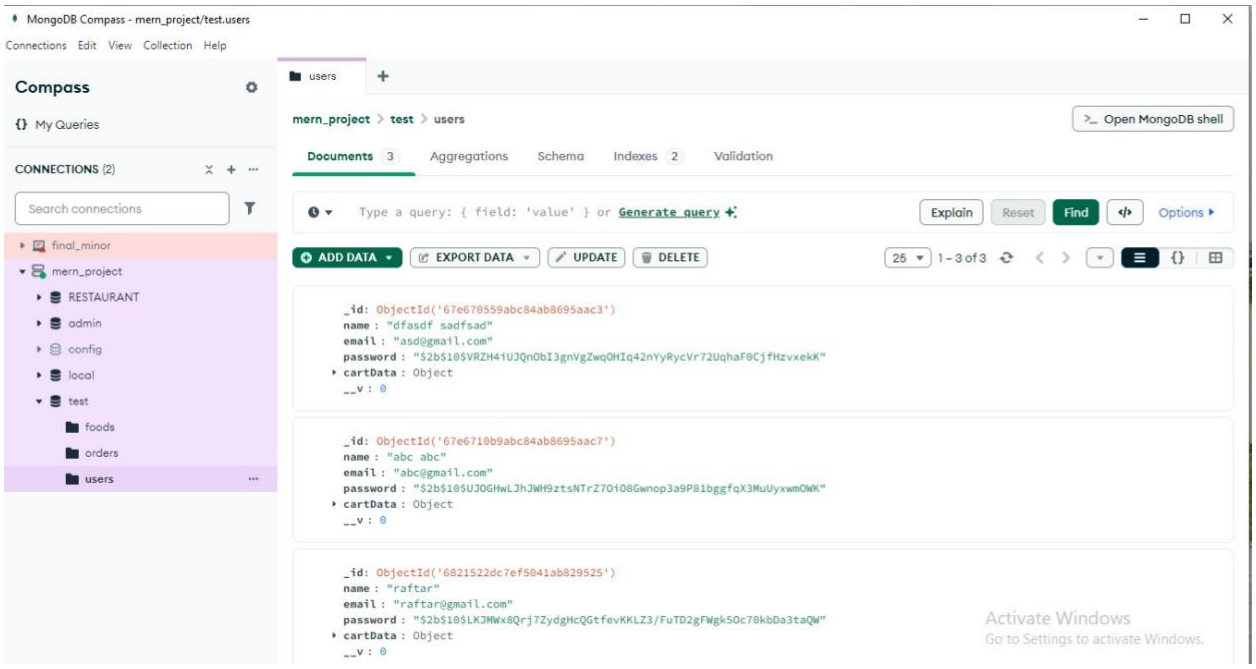


```
_id: ObjectId('67e671619abc84ab8695aacf')
userId : "67e6710b9abc84ab8695aac7"
items : Array (1)
amount : 23
address : Object
status : "Delivered"
date : 2025-03-28T09:41:33.365+00:00
payment : false
__v : 0
```

```
_id: ObjectId('682152afc7ef5041ab829532')
userId : "6821522dc7ef5041ab829525"
items : Array (1)
amount : 12
address : Object
status : "Delivered"
date : 2025-05-12T01:41:17.529+00:00
payment : false
__v : 0
```



```
_id: ObjectId('67e58596c935d5adc9ccde6f')
name : "Beans Salad"
description : "Beans Salad Description"
price : 5
image : "1743095190913food_1.png"
category : "Salad"
__v : 0

_id: ObjectId('67e626718a1d18a34413dd88')
name : "Chicken Roll"
description : "A chicken roll is a popular snack or meal made by wrapping cooked, sea…"
price : 7
image : "1743136369458food_6.png"
category : "Rolls"
__v : 0

_id: ObjectId('67e627b28a1d18a34413dd94')
name : "Strawberry Curd"
description : "Strawberry Curd Description"
price : 15
image : "1743136690961food_9.png"
```



```
_id: ObjectId('67e670559abc84ab8695aac3')
name : "dfasdf sadfsad"
email : "asd@gmail.com"
password : "$2b$10$VRZH4iUJQnObI3gnVgZwqOHIq42nYyRycVr72UqhaF0CjfHzvxekK"
cartData : Object
__v : 0

_id: ObjectId('67e6710b9abc84ab8695aac7')
name : "abc abc"
email : "abc@gmail.com"
password : "$2b$10$UJOGHwLJhJWH9ztsNTrZ7Oi08Gwnop3a9P81bggfqX3MuUyxwmOWK"
cartData : Object
__v : 0

_id: ObjectId('6821522dc7ef5041ab829525')
name : "raftar"
email : "raftar@gmail.com"
password : "$2b$10$LKJMWx8Qrj7ZydgHcQGtfevKKLZ3/FuTD2gFWgk5Oc70kbDa3taQW"
cartData : Object
__v : 0
```

Our Team Profile

INTRODUCTION OF

OUR TEAM

Souvik Dutta
Web Developer

Purnojit Kayal
Frontend Developer

Sneha Singh
Database Administrator

Priya Mahapatra
UI/UX Designer

Subarna Modak
Frontend Developer

Subhradip Acharya
Backend Developer

CHAPTER-8

FUTURE SCOPE

While Food Mania currently provides essential features for online food ordering and delivery tracking, there are several areas for future expansion and enhancement. These improvements can increase functionality, user satisfaction, and overall platform efficiency.

1. Online Payment Integration

Incorporating secure online payment options like UPI, credit/debit cards, and wallets will enable cashless transactions, providing a faster and more convenient checkout experience for users.

2. Real-Time Delivery Tracking with Maps

Adding live GPS-based tracking would allow customers to monitor the real-time location of their food delivery, increasing transparency and improving the overall user experience.

3. Mobile Application Development

Launching dedicated Android and iOS applications would make the platform more accessible and user-friendly, especially for mobile users. It would also allow push notifications for order updates.

4. Multiple Vendor Support

Expanding the system to accommodate multiple restaurants or food providers would transform Food Mania into a marketplace model, giving users more options and increasing the platform's reach.

5. Loyalty and Rewards Program

Introducing a loyalty system with points, discounts, and rewards for frequent users can boost customer retention and incentivize repeat orders.

6. Advanced Analytics Dashboard for Admin

An upgraded analytics dashboard could provide admins with detailed reports on sales trends, top-selling items, customer behavior, and more — helping in strategic decision-making.

7. AI-Based Food Recommendations

Using user data and preferences, AI algorithms could suggest popular or personalized dishes, making the experience more tailored and user-friendly.

These features will continue to enhance the user experience, making the platform even more efficient, interactive, and enjoyable for customers and cafe owners.

CHAPTER-9

CONCLUSION

Online Restaurant Table Booking presents a modern, practical solution for online food ordering and delivery management, designed to enhance the experience for both customers and administrators. By integrating essential features such as user registration, menu browsing, real-time order placement, and a responsive admin panel, the platform streamlines the entire food ordering process from start to finish.

Built using contemporary web technologies, the system ensures smooth navigation, clear communication between users and admins, and efficient delivery status updates. The dual-interface approach allows customers to conveniently place orders, while administrators can monitor and manage them in real time — all within a single, unified platform.

As the demand for digital food services continues to grow, *Online Restaurant Table Booking* is well-positioned for future enhancements such as online payments, real-time GPS tracking, mobile app development, and support for multiple vendors. These additions will further strengthen the platform's usability and impact.

In conclusion, this project showcases how technology can modernize everyday services, offering a faster, smarter, and more connected food ordering experience. *Online Restaurant Table Booking* not only simplifies operations for businesses but also raises the standard for convenience and satisfaction in the food service industry.

APPENDICES

Appendix A: Technical Specifications

- **Technologies Used:** Overview of the tools and frameworks used in development, such as HTML5, CSS3, JavaScript (or React), backend language (e.g., Node.js/PHP), and database (e.g., MongoDB/MySQL).
- **Development Environment:** Description of the development setup including code editors (e.g., VS Code), version control (e.g., Git/GitHub), and browser testing tools.
- **Project Architecture:** Explanation of client-server architecture and modular design principles used in building the platform.

Appendix B: System Design

- **Platform Overview:** Description of the *Food Mania* concept, objectives, and how it connects users and admins.
- **System Flow:** Logical flow of how a user interacts with the platform, from login to food delivery.
- **Architecture Diagram:** High-level system architecture showing frontend, backend, and database interactions.
- **Flowcharts:** Visual representation of the user and admin workflows, including ordering and delivery tracking processes.

Appendix C: Code Snippets

- **User Authentication:** Sample code for user login and registration functionality.
- **Order Management:** Key code showing how user orders are processed and stored.
- **Admin Panel Logic:** Snippets illustrating how admins view and update order statuses.
- **Comments and Annotations:** Code examples with clear comments for understanding logic and flow.

Appendix D: Testing and Evaluation

- **Testing Strategies:** Description of the types of testing performed, such as unit testing, functional testing, and user acceptance testing (UAT).
- **Test Cases:** Summary of test scenarios for core features (e.g., placing an order, updating delivery status).
- **Results and Feedback:** Evaluation of test outcomes and user feedback collected during trials or demo sessions.

Appendix E: User Guide

- **How to Use the Platform:** Step-by-step instructions for end users on signing up, browsing menus, placing orders, and checking delivery status.
- **Admin Instructions:** Guide for administrators to manage incoming orders and update delivery progress.
- **FAQs and Troubleshooting:** Answers to common user questions and basic troubleshooting tips.

Appendix F: Future Enhancements

- **Planned Features:** Outline of proposed upgrades such as online payment integration, mobile app support, GPS-based tracking, and loyalty programs.
- **Development Roadmap:** Timeline or roadmap for implementing future features and platform improvements.

.

REFERENCE

- **https://www.w3schools.com/REACT/DEFAULT.ASP**
- **https://www.tutorialspoint.com/mongodb/index.htm**