**Project Title: Online Restaurant Web Application**

**Submitted by:**

**Arunima Suresh**

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Title** | **Pg.No** |
| **1** | **Introduction** | **3** |
| **2** | **Problem Statement** | **4** |
| **3** | **Proposed Solution** | **5** |
| **4** | **User Stories** | **7** |
| **5** | **Technology Stack** | **9** |
| **6** | **Development Environment** | **10** |
| **7** | **Project Timeline** | **11** |
| **8** | **Conclusion** | **12** |
| **9** | **Appendix** | **13** |

# 1. Introduction

The " Online Restaurant Web Application" is a full-stack project developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The primary goal of this application is to digitalize and simplify restaurant operations including online food ordering, table reservations, order delivery, and admin-level food and order management. This project is significant as it demonstrates how modern web technologies can improve customer experience and streamline business processes in the food industry.

In addition to customer and admin functionalities, the application also includes dedicated features for **Delivery Partners**. Once an admin assigns an order, delivery personnel can log in to their dashboard, view assigned orders, update delivery status, and ensure timely food delivery. This role-based functionality ensures smooth coordination between customers, restaurant staff, and delivery partners.

This project is significant as it demonstrates how modern web technologies can improve customer experience, optimize restaurant workflows, and streamline end-to-end food delivery processes in the food industry.

**Objectives:**

● To develop a responsive and user-friendly restaurant application.

● To allow customers to browse menus and place food orders online.

● To enable table reservations through the platform.

● To help restaurant admins manage food items, reservations, and customer orders.

● To assign delivery tasks and track order statuses through delivery dashboards.

● To provide delivery partners with access to their assigned deliveries and allow them to update order status.

● To practice and apply core full-stack development concepts including authentication, routing, and database integration.

**Significance:**

● Practical implementation of frontend and backend development using the MERN stack.

● Useful real-world application that enhances digital restaurant operations.

● Demonstrates full-stack developer skills in JWT-based authentication, role-based routing, CRUD operations, and responsive UI design.

● Includes role-based access for customers, admins, and delivery personnel—simulating real restaurant workflows.

● Enhances the ability to manage an end-to-end project lifecycle from design and development to testing and deployment.

## 2. Problem Statement:

In many traditional restaurant setups, operations such as food ordering, table reservations, and delivery coordination are handled manually or through disconnected systems. This can lead to various challenges such as:

* **Manual errors** in taking and fulfilling orders, which may result in incorrect food deliveries or missing items.
* **Long wait times** due to the lack of an efficient table reservation system, especially during peak hours.
* **Inefficient order tracking**, which creates confusion among kitchen staff, delivery personnel, and customers.
* **Poor communication** between different roles—such as servers, managers, and delivery partners—leading to delays and customer dissatisfaction.
* **Limited visibility** for restaurant administrators over ongoing orders, inventory, and delivery status, making it difficult to manage and optimize resources.

Furthermore, customers today expect **fast, seamless, and contactless service**, including the ability to order food online, reserve tables ahead of time, and receive real-time updates on their deliveries. In the absence of a digital system, restaurants struggle to meet these modern expectations.

This project addresses these challenges by providing a **centralized and responsive web application** where:

* Customers can place orders, reserve tables, and pay online.
* Admins can manage the menu, oversee orders/reservations, and assign delivery tasks.
* Delivery partners can view and update their assigned orders .

By offering dedicated dashboards and features tailored to each role, the application enhances coordination, improves customer satisfaction, and brings operational efficiency to restaurant management.

## 3. Proposed Solution:

To address the challenges faced in traditional restaurant operations, we propose a **role-based Restaurant Web Application** built using the **MERN (MongoDB, Express.js, React.js, Node.js)** stack. The application includes distinct functionalities for each type of user—**Customer**, **Admin**, and **Delivery Partner**—ensuring an organized and efficient restaurant workflow.

**Key Features by Role:**

* **Customer:**
  + Browse menu items with categorized listings and images.
  + Place food orders and view cart items.
  + Reserve tables in advance for dine-in.
  + Choose between **online (UPI/card)** or **offline (cash on delivery)** payment options.
  + Track order and reservation status through their dashboard.
* **Admin:**
  + Add, update, or delete food items in the restaurant menu.
  + View all orders and reservation requests.
  + Update the status of orders (e.g., "Preparing", "Ready", "Assigned", etc.) until they are handed over to delivery partners.
  + View available delivery partners and assign them to pending orders.
  + Monitor deliveries and ensure timely service.
* **Delivery Partner:**
  + Log in to their dedicated dashboard.
  + **Set their availability status** (e.g., Available / Not Available) to receive orders.
  + View the list of assigned orders with customer delivery details.
  + Update order status (e.g., "Out for Delivery", "Delivered") in real-time.

**Other Technical Highlights:**

* **JWT-based authentication** to ensure secure access for all roles.
* **Role-based authorization** and protected routes with dynamic redirection based on user role.
* **Toast notifications** for real-time feedback on user actions (e.g., order placed, login success, etc.).
* **Clean, responsive UI** developed using React.js and Material UI/Tailwind CSS for a mobile-first user experience.
* **RESTful API** integration between frontend and backend for smooth communication.
* **MongoDB** for efficient data management and scalable performance.

This solution ensures smooth coordination between customers, admins, and delivery partners while offering flexibility, automation, and convenience in restaurant operations.

# 4.User Stories:

**User Story 1**

* **As a** customer,
* **I want to** order food online,
* **So that** I can receive it conveniently at home.

**User Story 2**

* **As a** customer,
* **I want to** reserve a table,
* **So that** I avoid waiting at the restaurant.

**User Story 3**

* **As an** admin,
* **I want to** manage food items and customer orders,
* **So that** operations are smooth and organized.

**User Story 4**

* **As an** admin,
* **I want to** manage table reservations,
* **So that** I can ensure better planning and avoid overbooking.

**User Story 5**

* **As a** delivery boy,
* **I want to** view and update my assigned deliveries,
* **So that** I can ensure timely and accurate delivery.

**User Story 6**

* **As a** delivery boy,
* **I want to** update my availability status,
* **So that** I receive delivery assignments only when I am ready.

## 5. Technology Stack:

**Frontend:**  
React.js, React Router, Material UI / Tailwind CSS.HTML,Bootstrap

**Backend:**  
Node.js, Express.js, JWT for authentication, Bcrypt for hashing

**Database:**  
MongoDB with Mongoose ODM

## 6. Development Environment:

* **IDE:** Visual Studio Code
* **Version Control:** Git + GitHub
* **Database Tools:** MongoDB Atlas & Compass
* **Testing:** Postman
* **Deployment:** Aws

# 7. Project Timeline: Milestones and Reporting

**Milestones and Reporting**

| **Milestone** | **Task** | **Date** |
| --- | --- | --- |
| **1 - Project Initialization and Planning** | **Finalized project idea, created repo, initialized folders** | **May 15, 2025** |
|  | **Prepared feature list and wireframe sketches** | **May 16, 2025** |
| **2 - Design Phase** | **Frontend UI design using , React component planning** | **May 17–18, 2025** |
| **3 - Backend Development** | **Setup Express server, MongoDB connection, basic APIs** | **May 19–21, 2025** |
|  | **Created models, routes, controllers for users, orders, food** | **May 22–23, 2025** |
| **4 - Frontend Development** | **Built user pages – Home, Menu, Cart, Order** | **May 24–26, 2025** |
|  | **Developed admin pages – Dashboard, Food List, Orders, Reservations** | **May 27–28, 2025** |
| **5 - Frontend-Backend Integration** | **Integrated API with React frontend (Axios, fetch data, display UI)** | **May 29–30, 2025** |
| **6 - Authentication & Authorization** | **JWT-based login, signup, protected routes, role-based dashboard routing** | **May 31–June 3, 2025** |
| **7 - Testing and Validation** | **Manual testing of all features, error handling, input validation** | **June 4–5, 2025** |
| **8 - Deployment and Hosting** | **Hosted frontend on** | **June 5, 2025** |
| **9 - Project Documentation** | **Finalized report, added screenshots, created walkthrough video** | **June 6–9, 2025** |

## 8. Project Structure:

**Frontend (/client)**

/client

│

├── /public

│ └── images/

│

├── /src

│ ├── /assets

│ │ └── Home.css

│

│ ├── /components

│ │ ├── Login.jsx

│ │ ├── Signup.jsx

│ │ ├── Navbar.jsx

│ │ ├── Menu.jsx

│ │ ├── Order.jsx

│ │ ├── Reservation.jsx

│ │ ├── MyReservation.jsx

│ │ ├── MapSelector.jsx

│ │ ├── MapModal.jsx

│ │ ├── Payment.jsx

│ │ ├── ResetPassword.jsx

│ │ ├── PrivateRoutes.jsx

│ │ └── Main.jsx

│

│ ├── /components/AdminDashboard

│ │ ├── AdminDashboard.jsx

│ │ ├── FoodForm.jsx

│ │ ├── FoodItems.jsx

│ │ ├── OrderList.jsx

│ │ ├── ReservationList.jsx

│ │ └── ReservationListJ.jsx

│

│ ├── /context

│ │ ├── CartContext.jsx

│ │ └── DeliveryDash.jsx

│

│ ├── App.jsx

│ ├── App.css

│ ├── main.jsx

│ ├── index.css

│ ├── axiosInterceptor.js

│ └── index.html

**Backend (/server)**

/server

├── /controllers

│ └── authController.js

│

├── /db

│ └── connection.js

│

├── /model

│ ├── foodData.js

│ ├── orderData.js

│ ├── reservationData.js

│ ├── slotData.js

│ └── userData.js

│

├── /routes

│ ├── authRoutes.js

│ ├── foodRoutes.js

│ ├── orderRoutes.js

│ ├── reservationRoutes.js

│ ├── userRoutes.js

│ ├── paymentRoutes.js

│ ├── jwt.js

│

├── app.js

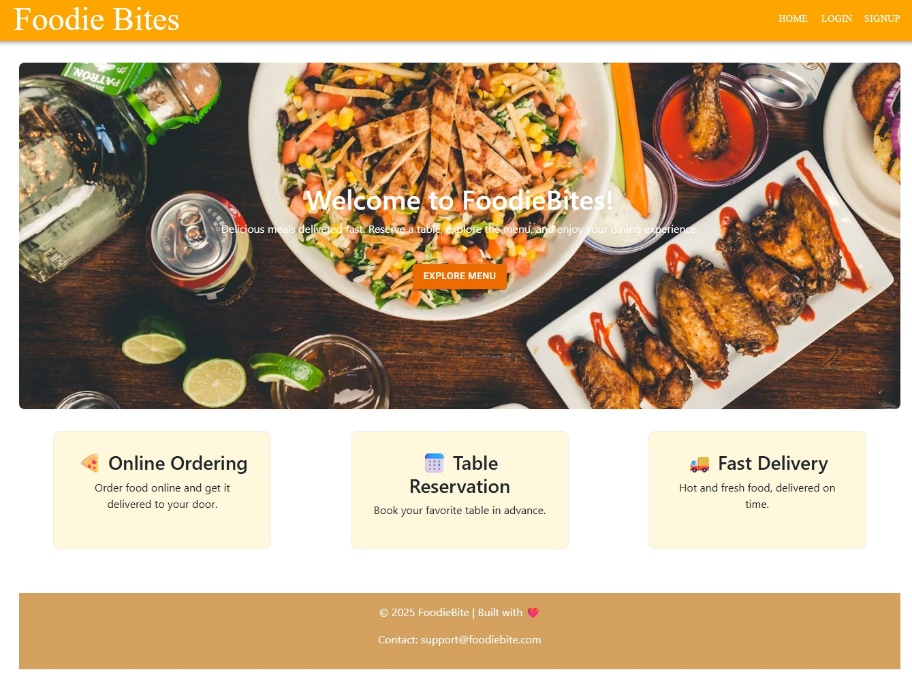
├── hashPasswords.js

# 9. Conclusion:

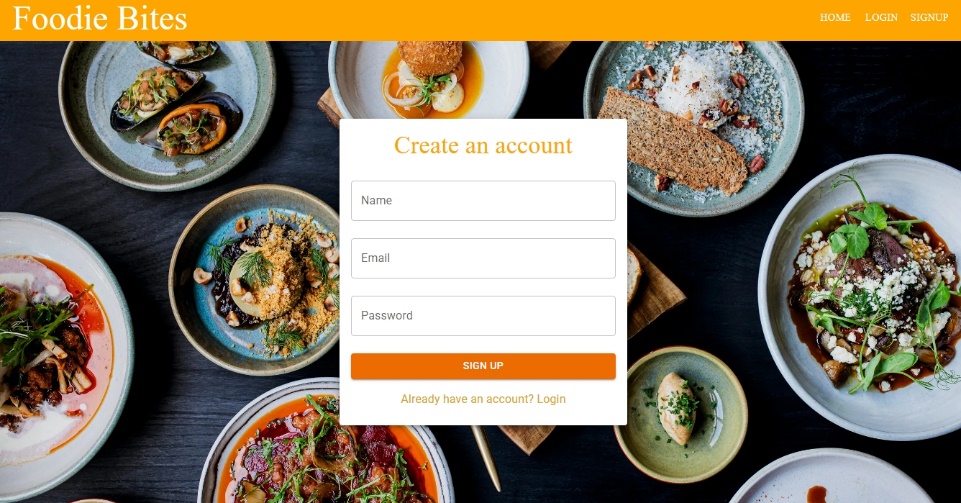
* ✅ The **Online Restaurant Web Application** offers a complete digital solution for restaurant management, enhancing efficiency and customer satisfaction.
* ✅ It simplifies core operations such as **food ordering, table reservations, and order delivery** with role-based access for customers, admins, and delivery partners.
* ✅ The project provided hands-on experience with the **MERN stack**, allowing the implementation of both frontend and backend technologies.
* ✅ It helped build a solid understanding of **RESTful API design**, **JWT-based authentication**, and **user role-based routing**.
* ✅ I practiced and applied important concepts like **state management ,** **form validation**, **payment integration**, and **Google Maps API**.
* ✅ The project improved my confidence in deploying full-stack applications and managing the **entire development lifecycle** from planning to deployment.
* ✅ Overall, this project is a strong representation of my skills as a full-stack developer and demonstrates my ability to create practical, real-world web applications.

# 10. Appendix:

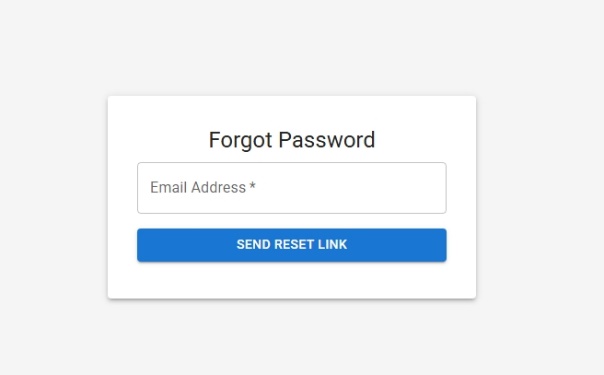
**Home Page:**

****

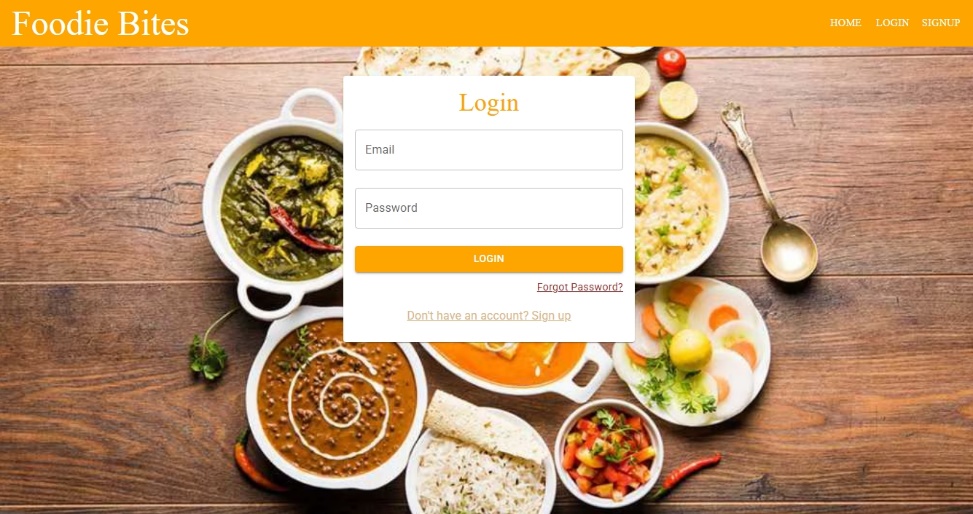
**Signup Page:**



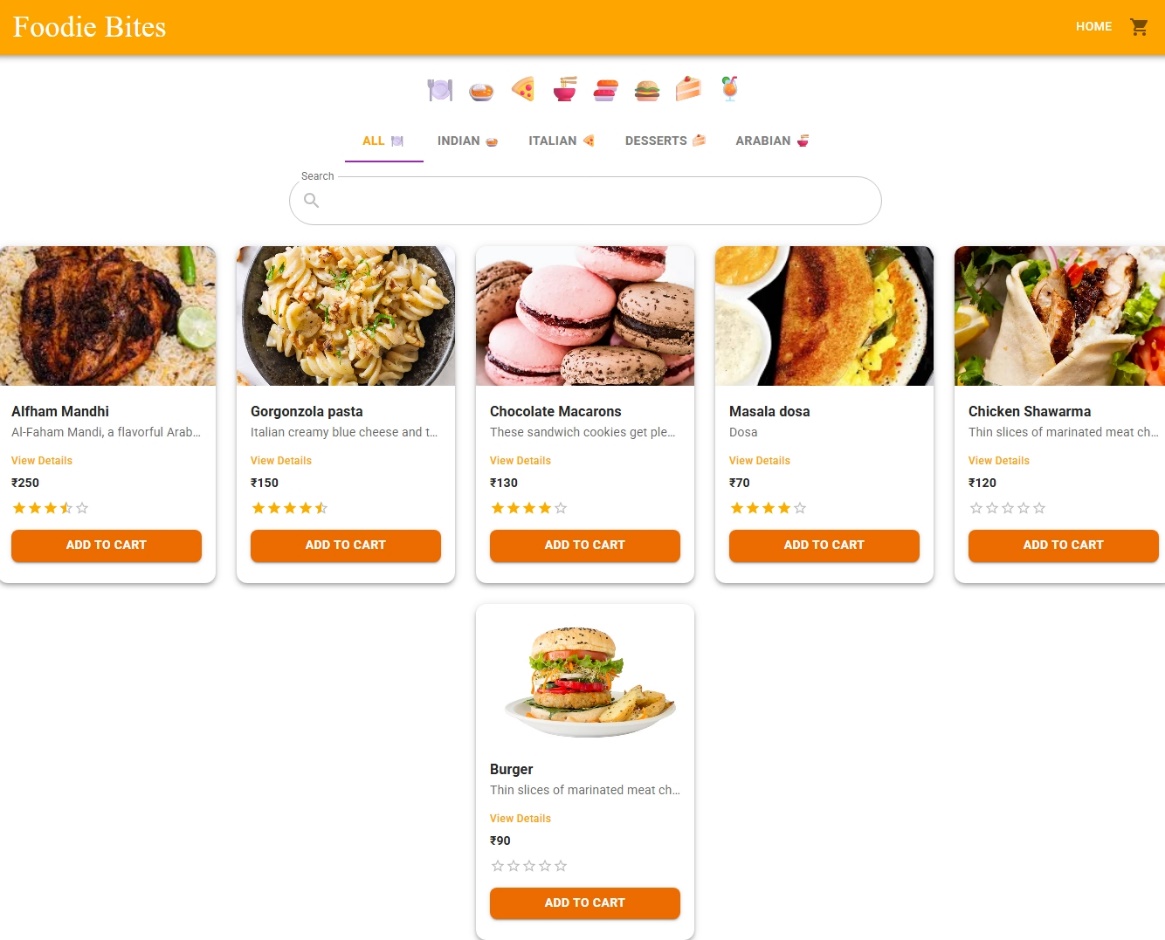
**Forgot password Page:**

****

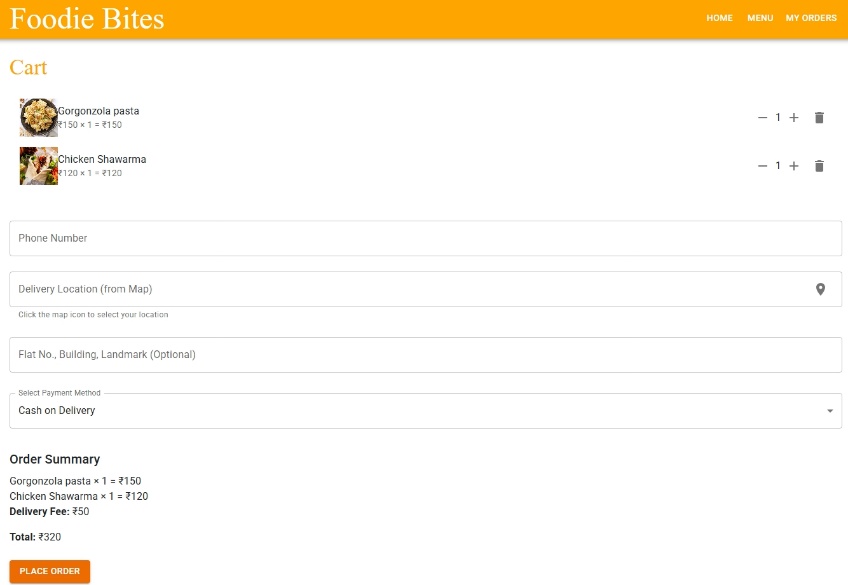
**Login Page:**



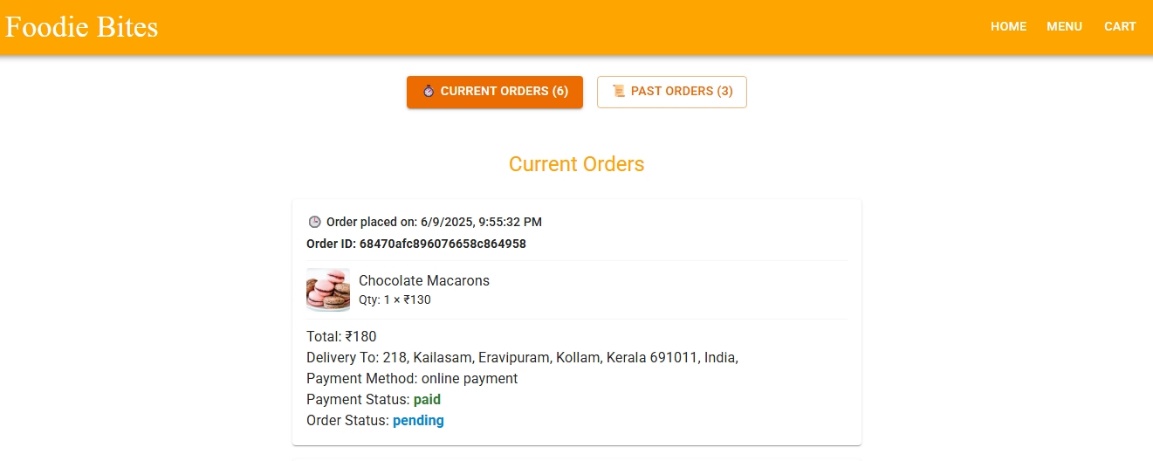
**Menu Page:**



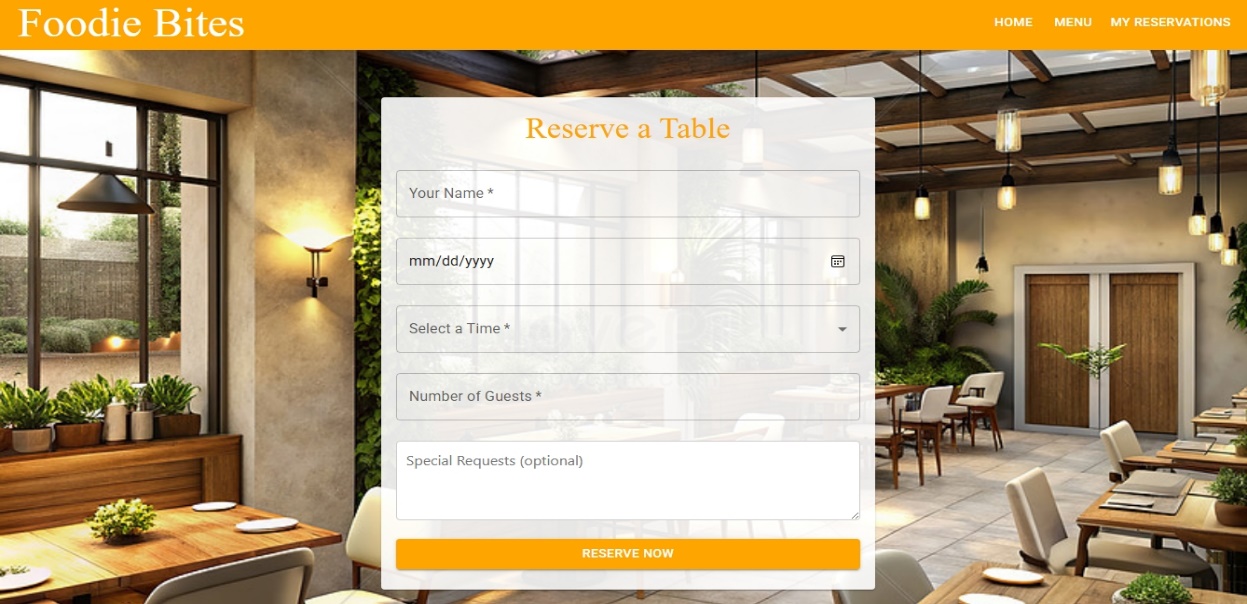
**Cart Page:**



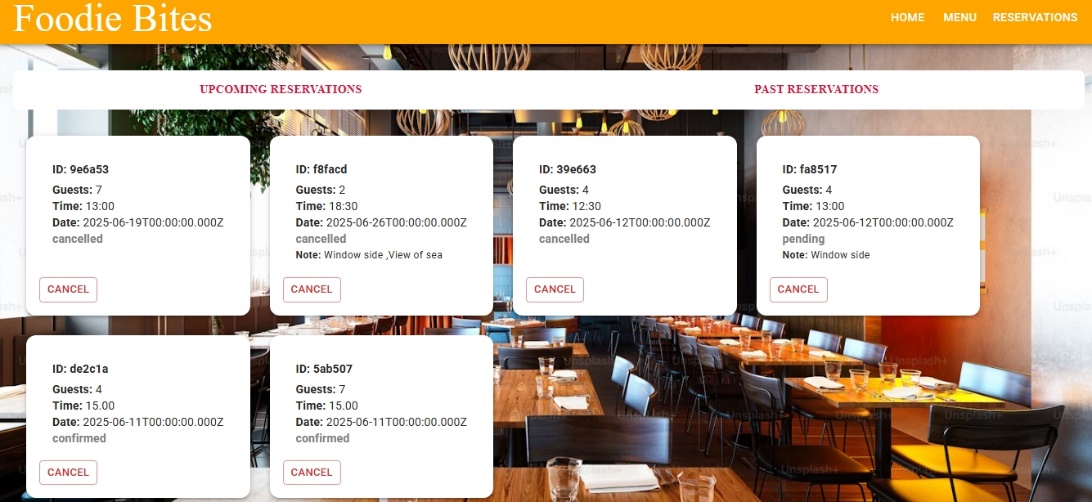
**My orders Page:**



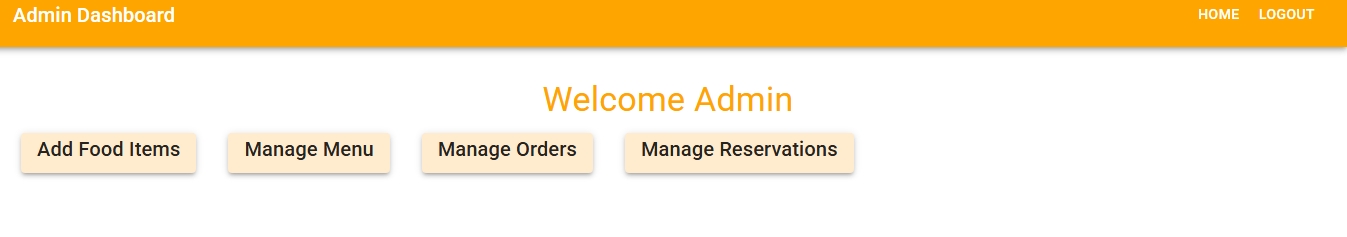
**Reservation Page:**

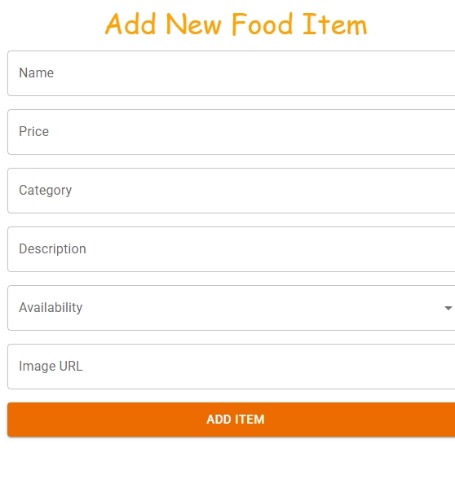
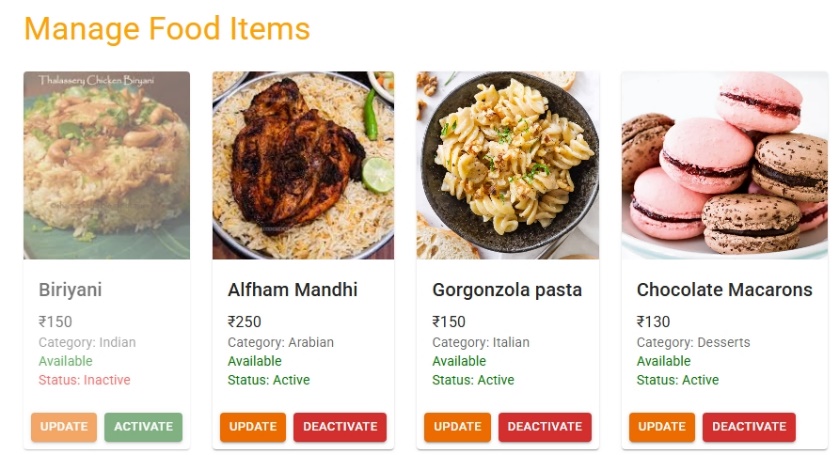


**My Reservation Page:**

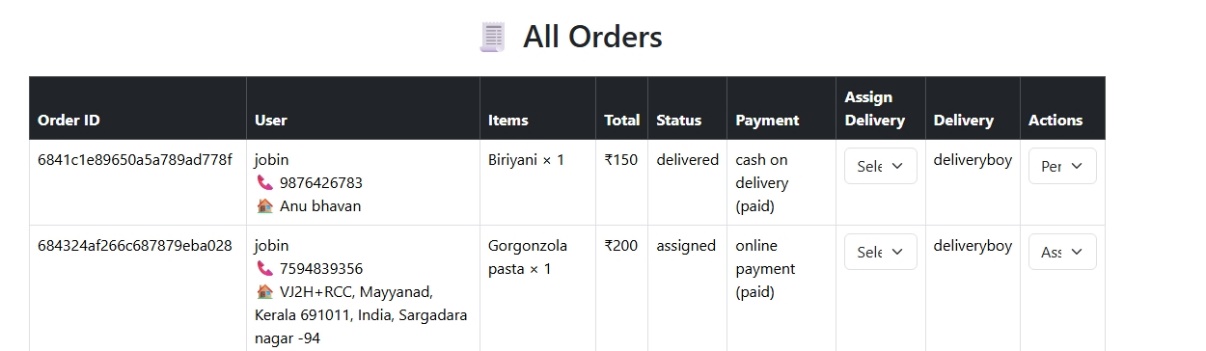


**Admin dashboard Page:**

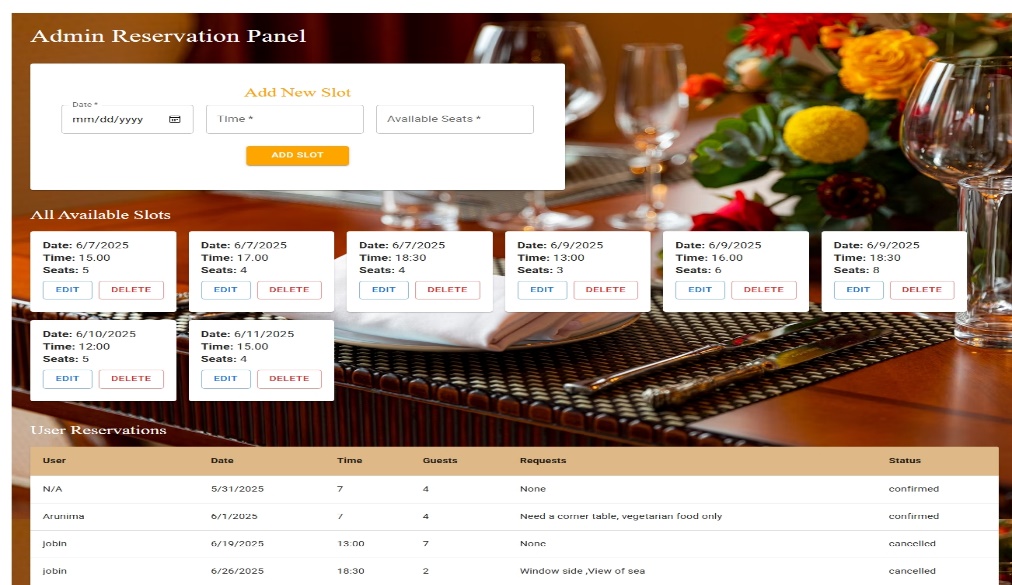


**Admin dashboard Page: Add item** **Manage food item**

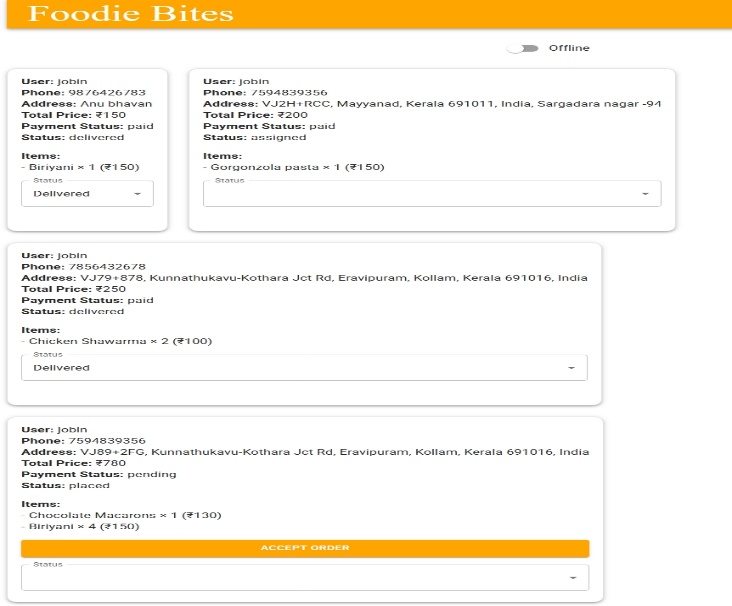
**Orders Page:**

****

**Reseravtion Page:**

****

**Delivery dashboard Page:**

****

**Login Credentials:**

** Admin**: [admin123@gmail.com](mailto:admin123@gmail.com)

Password:admin@123

 **User**: 1)[jobin@gmail.com](mailto:jobin@gmail.com)

Password:jobin@1234

2)[arunima@gmail.com](mailto:arunima@gmail.com)

Password:arun@123

** Delivery**: deliveryboy12@gmail.com

Password: deliveryboy@123

**Hosted Link:http://13.204.83.27/**

**Repo Link:** <https://github.com/Arunima-72/Restauarnt.git>

**Link of recorded video of the working of your application (with voice explanation):** **https://drive.google.com/file/d/1v7KEk7cgoqJ-XT691WMhmoWGUAdXs6fI/view?usp=sharing**