SB FOODS

ONLINE FOOD ORDERIND APP

INTRODUTION

Food Order is an innovative online food ordering app designed to revolutionize the way you order food. Our app brings together your 9 restaurants, mouth-watering dishes, and hassle-free ordering experience, all under one roof. With Food Order, you can bid farewell to waiting in long queues or on phone calls to place orders, limited dining options, and uncertain delivery times.

DESCRIPTION

An online food ordering system is a digital platform that enables customers to browse menus, place orders, and make payments directly from their devices, connecting them seamlessly with restaurants and food establishments. This system streamlines the entire ordering process by providing a user-friendly interface for selecting dishes, customizing orders, and tracking delivery in real time. Integrated with various payment gateways and location services, the system ensures secure transactions and efficient delivery coordination. Online food ordering systems benefit restaurants by expanding their customer reach, optimizing order management, and improving customer satisfaction through convenience and quick service.

PROJECT OVERVIEW

The Food Order project aims to develop a user-friendly online food ordering app, connecting customers with their iuy6t5 `1 restaurants and delivering a seamless dining experience. Leveraging the MERN (MongoDB, Express.js, React.js, and Node.js) technology stack, this 12-week project will integrate features such as restaurant listings, menu browsing, secure payment processing, real-time order tracking, and personalized recommendations. The app will cater to a diverse user base, including foodies, busy professionals, students, and families, providing convenience, variety, and cost-effectiveness.

ARCHITECTURE OVERVIEW

1. User Interface (Web and Mobile)

- Handles user input and displays menu, order status, and other relevant information

- Built using React.js, Redux, and React Router

2. Order Service

- Handles order creation, updates, and cancellations

- Integrates with Restaurant Service and Payment Service

- Built using Node.js, Express.js, and MongoDB

3. Restaurant Service

- Handles restaurant registration, menu management, and order management

- Integrates with Order Service and Payment Service

- Built using Node.js, Express.js, and MongoDB

4. Payment Service

- Handles payment processing and transaction management

- Integrates with Stripe payment gateway

- Built using Node.js, Express.js, and MongoDB

5. Delivery Service

- Handles delivery personnel management and order assignment

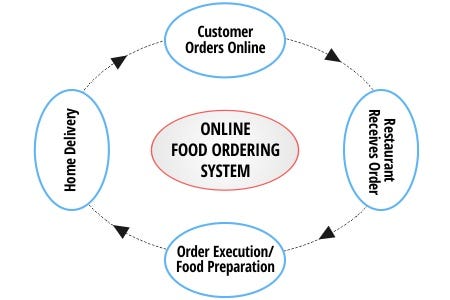
- Integrates with Order Service and Restaurant Service

- Built using Node.js, Express.js, and MongoDB

6. Notification Service

- Handles SMS and push notifications for order updates and delivery status

- Integrates with Twilio (SMS) and Firebase Cloud Messaging (FCM)



SCENARIO

Scenario: SB FOODS Online Food Delivery

1. Browsing and Selection:

- Alice logs into the SB FOODS app using her credentials.

- She browses through the list of available restaurants and filters for Italian cuisine.

- Alice selects Bella’s Pizzeria from the results and views its menu, checking customer reviews and item descriptions.

- She decides on a Margherita pizza and adds it to her cart, along with a side of garlic bread.

2. Customization and Checkout:

- Before checking out, Alice customizes her pizza with additional toppings (extra cheese and mushrooms).

- She reviews her order, ensures it meets her preferences, and proceeds to the checkout page.

- Alice selects her preferred payment method (credit card) and enters a discount code she received via email.

- The system calculates the total price with the discount and displays it for final confirmation.

3. Order Processing:

- Alice confirms her order, and the system sends a notification to Bella’s Pizzeria.

- The restaurant acknowledges the order and provides an estimated preparation time.

- Alice receives an in-app notification and a confirmation email with the estimated delivery time.

4. Preparation and Tracking:

- Bella’s Pizzeria begins preparing the order and updates its status on the SB FOODS system.

- Once the order is ready, Jake, the delivery person assigned to the route, picks it up.

- Alice receives a real-time tracking link to monitor Jake’s location as he heads to her address.

5. Delivery:

- Jake arrives at Alice’s location and delivers the food. Alice receives a notification to confirm delivery.

- She checks the order, finds everything in good condition, and marks the delivery as complete in the app.

6. Post-Delivery:

- Alice rates the food and leaves a review for Bella’s Pizzeria, mentioning the delicious taste and prompt service.

- The system stores her review and updates the restaurant’s rating accordingly.

- Alice receives a follow-up message thanking her for using Local Bites and encouraging her to order again with a future discount.

SETUP INSTRUCTION

1. Clone repository from GitHub

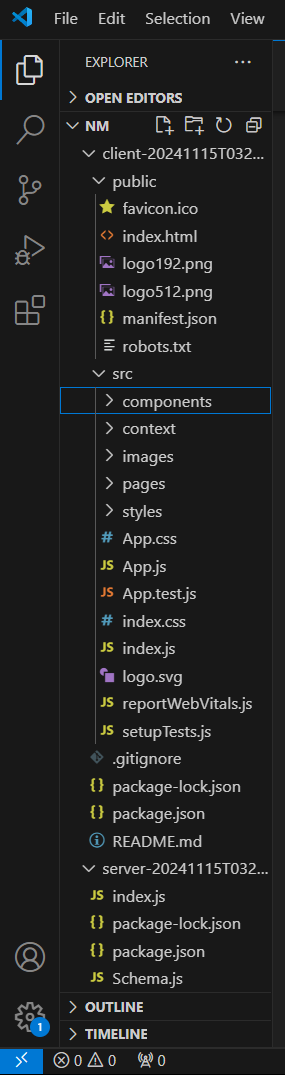
2. Install dependencies (npm install)

3. Configure environment variables (MongoDB, Stripe)

4. Run backend server (node server.js)

5. Run frontend server (npm start)

FOLDER STRUCTURE



API DOCUMENTATION

Restaurant API

* GET /restaurants - Retrieve all restaurants
* GET /restaurants/:id - Retrieve restaurant by ID
* POST /restaurants - Create new restaurant
* PUT /restaurants/:id - Update restaurant
* DELETE /restaurants/:id - Delete restaurant

Order API

* POST /orders - Place new order
* GET /orders - Retrieve all orders
* GET /orders/:id - Retrieve order by ID
* PUT /orders/:id - Update order status

User API

* POST /users - Register new user
* GET /users - Retrieve all users
* GET /users/:id - Retrieve user by ID
* PUT /users/:id - Update user profile

AUTHENTICATION

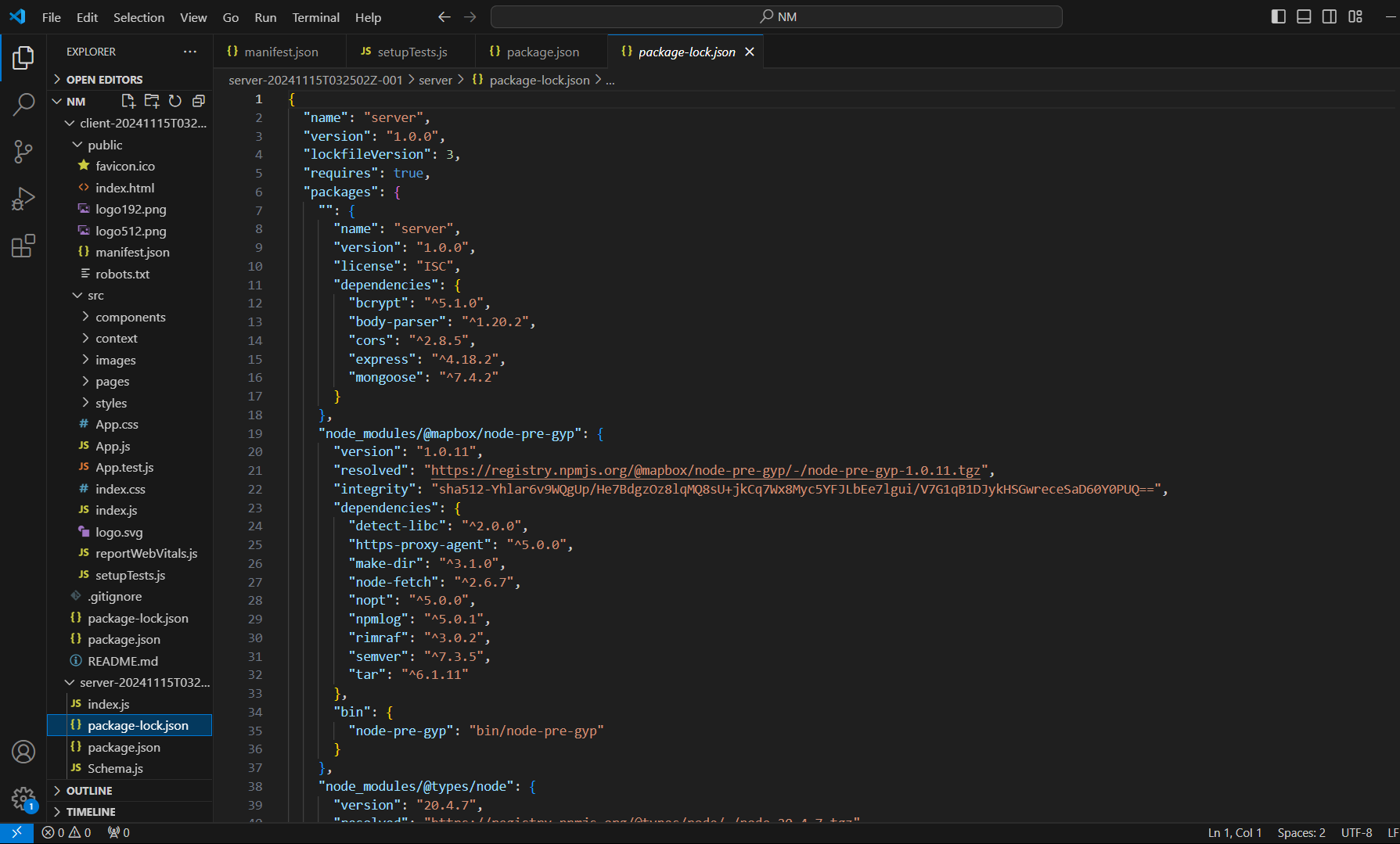
The online food ordering app utilizes secure authentication methods to verify user identity, ensuring authorized access and protecting sensitive data. Techniques include JSON Web Tokens (JWT), OAuth 2.0, and session-based authentication. Users register/login via email/password, social media, or phone number (OTP), generating a token validated on each request. Best practices include password hashing (bcrypt), secure password reset, two-factor authentication (2FA), and regular security audits. Authentication components comprise login/registration forms, authentication APIs, token storage, and validation middleware, leveraging tools like Passport.js, Auth0, Firebase Authentication, and Okta.

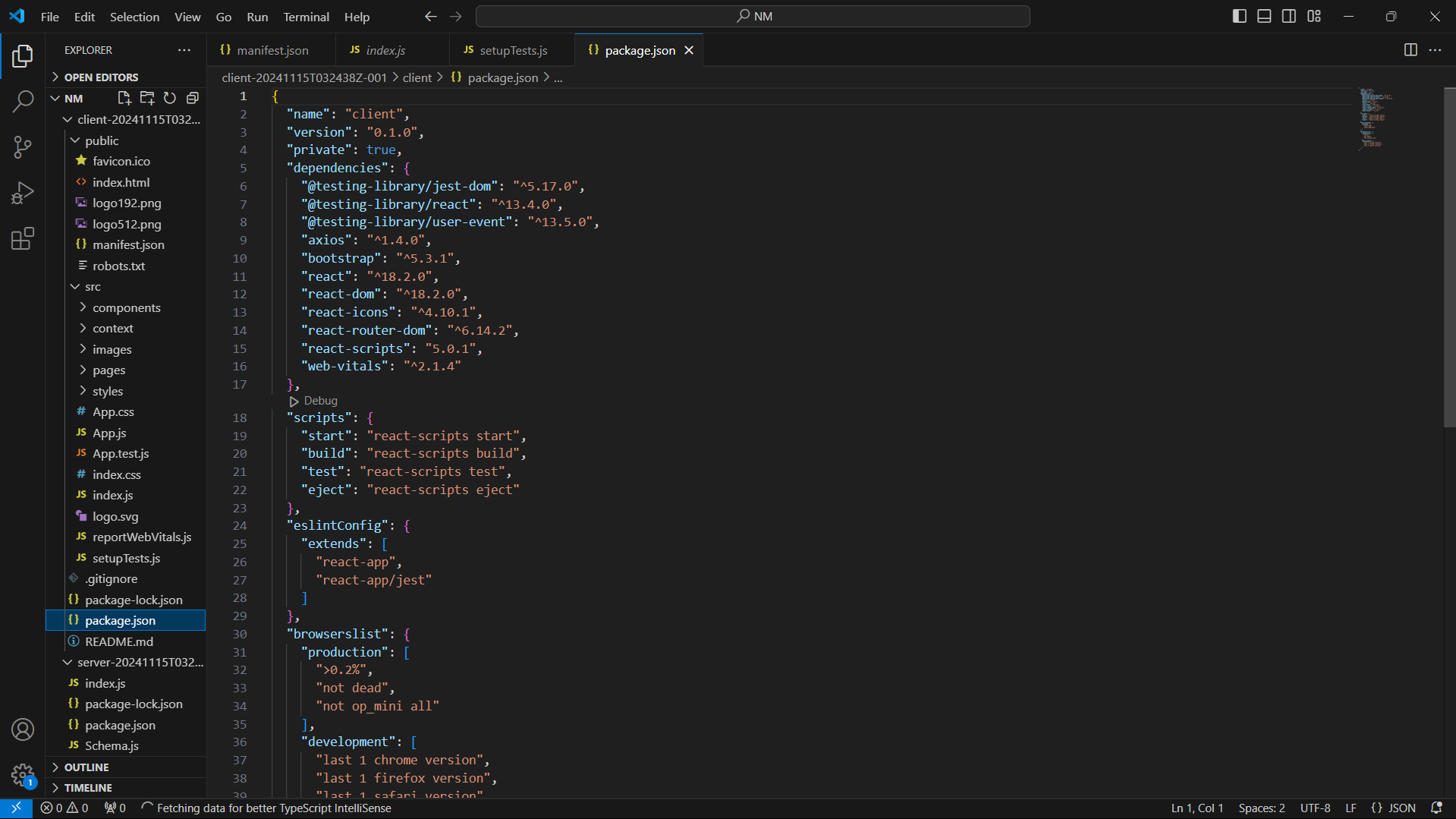
USER INTERFACE

The Food Order app features an intuitive interface with vibrant food images, easy search, clear menu displays, and secure payment processing. Its responsive design adapts to mobile, tablet, and web devices, ensuring seamless navigation and ordering. Key elements include streamlined ordering, real-time tracking, and personalized recommendations.

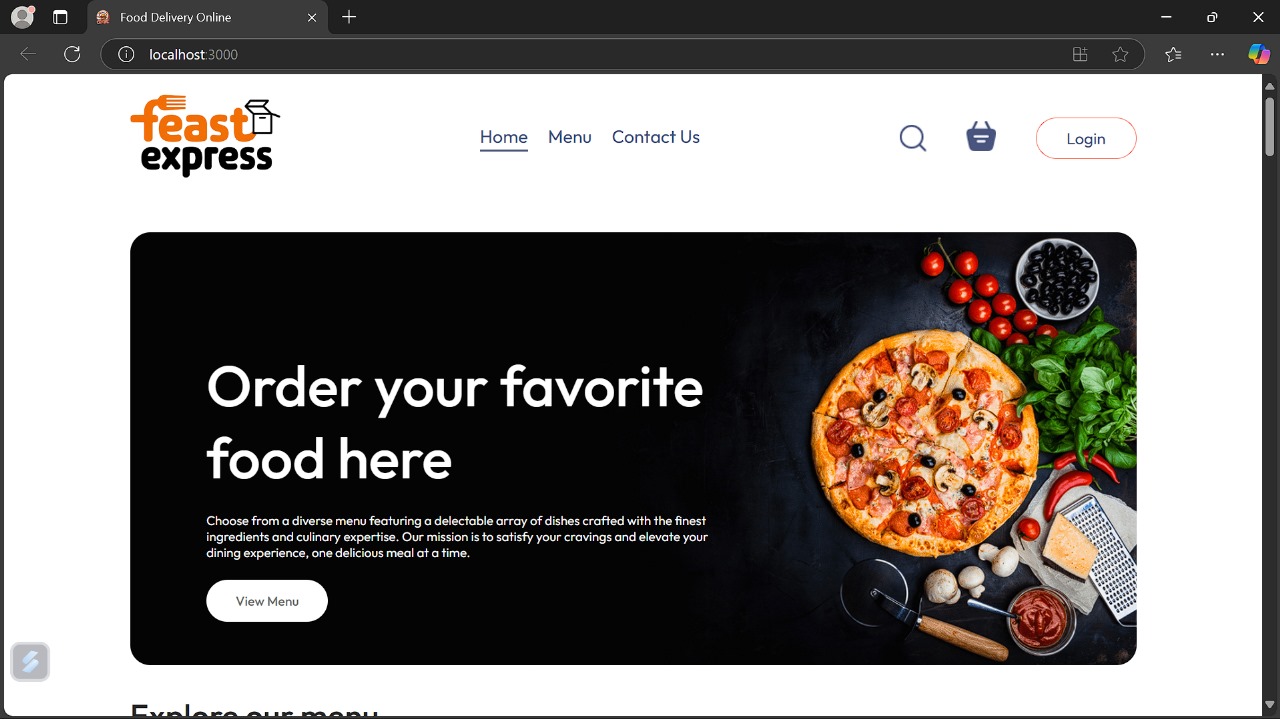
TESTING

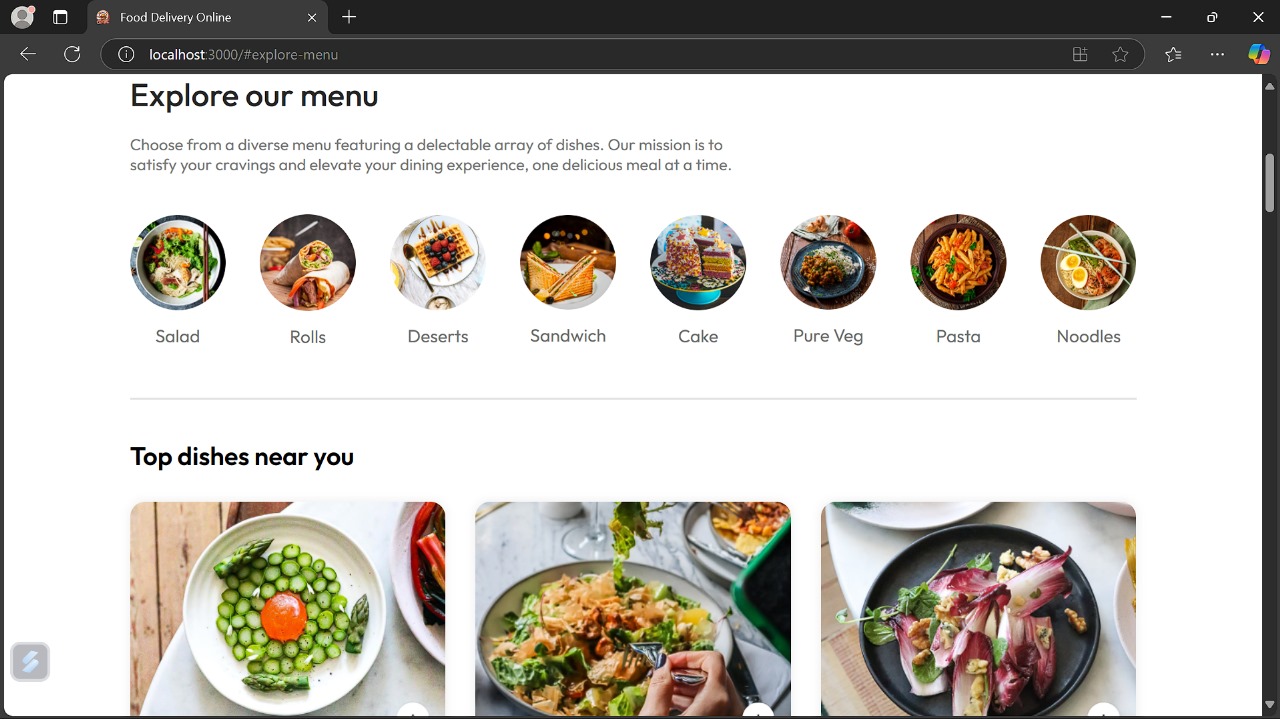
Testing the Food Order app using the MERN (MongoDB, Express.js, React.js, Node.js) stack involves leveraging frameworks like Jest, Mocha, Enzyme, and Cypress. Unit testing, integration testing, and end-to-end testing ensure individual components and interconnected systems function correctly. Testing scenarios include user registration, food ordering, payment processing, order tracking, and restaurant management. Developers utilize tools like Supertest for API testing and Mongoose for database testing. Example test code using Jest illustrates the simplicity of testing, such as creating a new order with a 201 status code. By adopting MERN testing frameworks and tools, developers guarantee the app's reliability, stability, and performance.

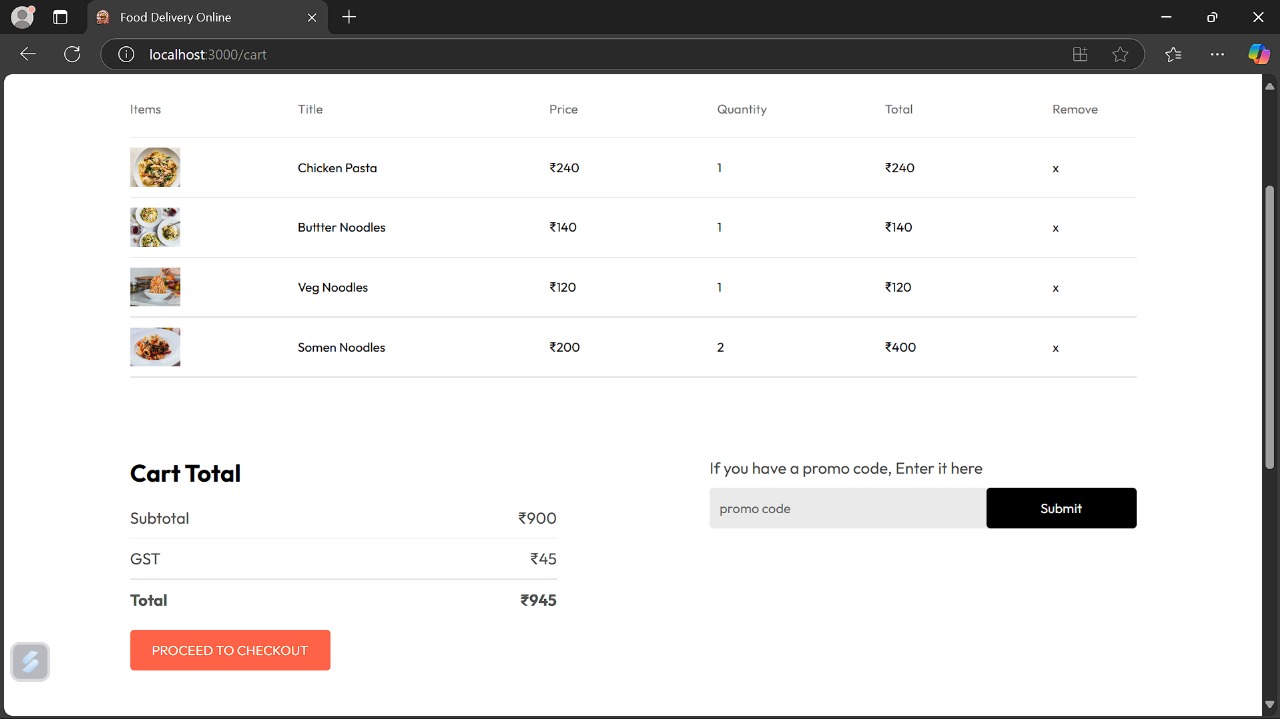


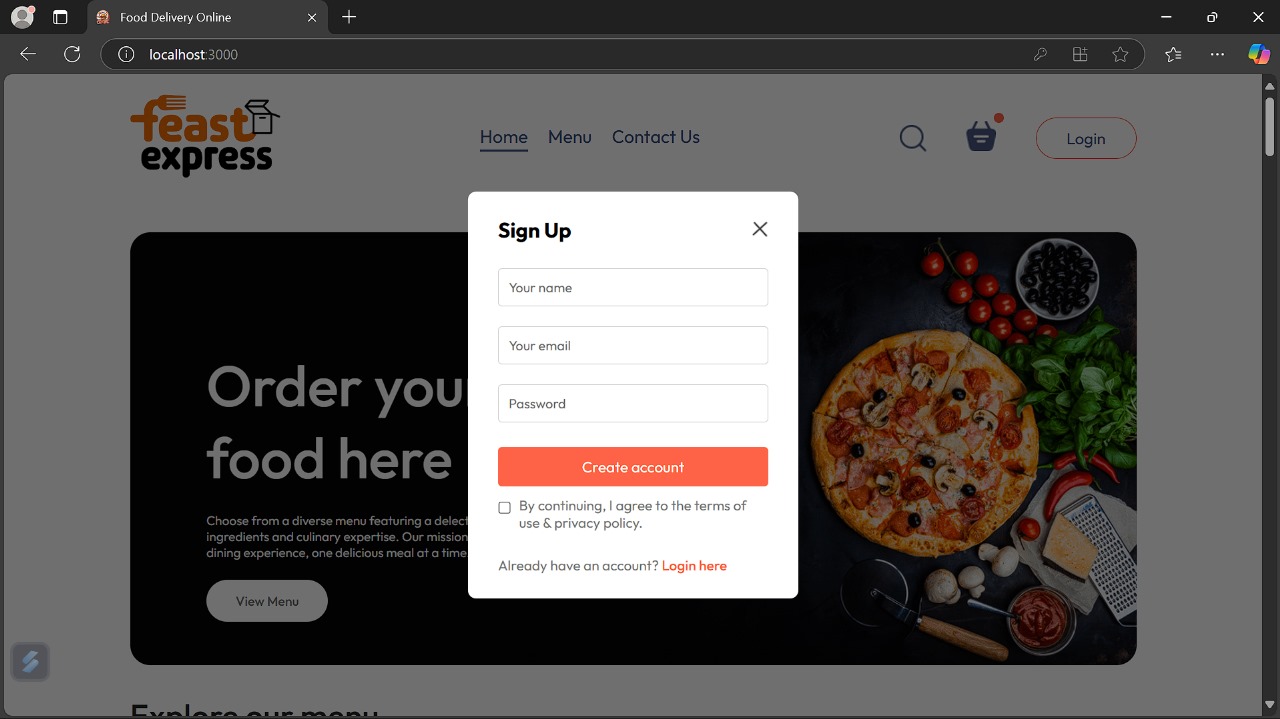


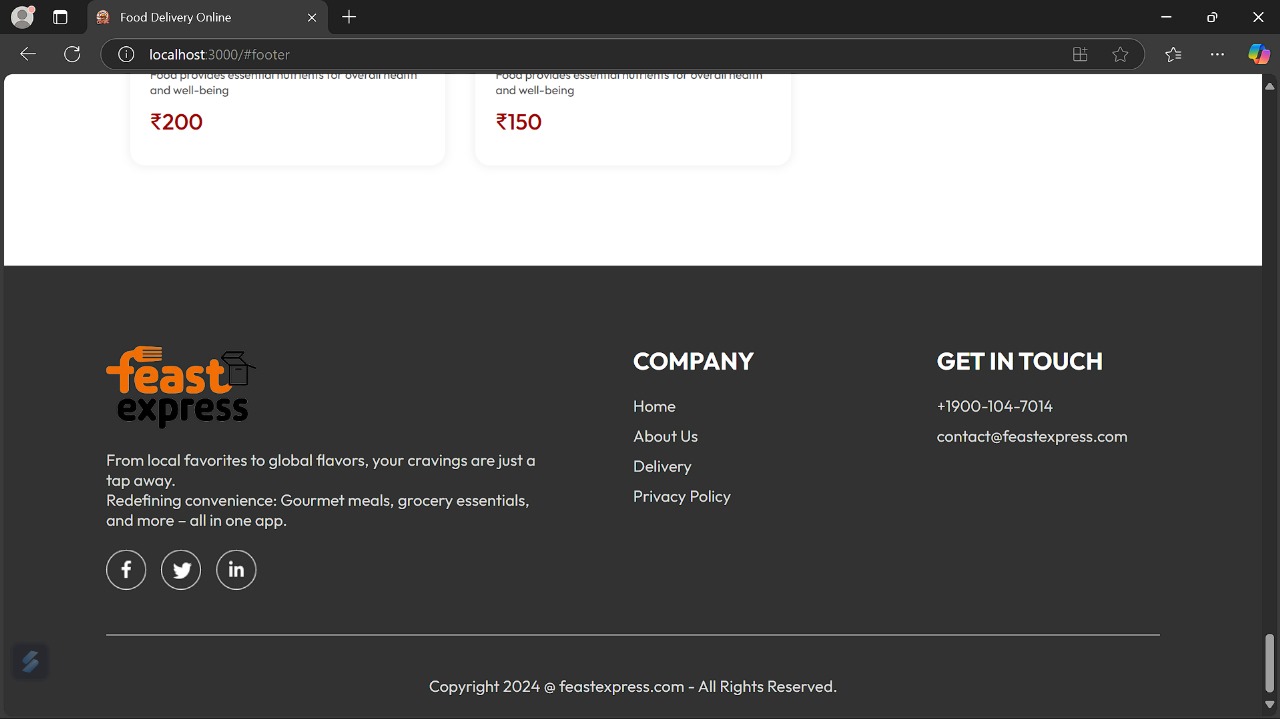
Screenshot or Demo











KNOWN ISSUE

Online food ordering apps have revolutionized the way people access meals, but they are not without issues. One of the most common problems is delivery delays, often caused by logistical inefficiencies, traffic, or a lack of available delivery personnel, which can lead to cold or stale food. Users frequently face incorrect or incomplete orders, where items are missing or differ from what was ordered. High service fees, including hidden charges and surge pricing during peak hours, often frustrate customers. App performance issues, such as slow loading times, crashes, or payment processing errors, add to the dissatisfaction. Furthermore, limited restaurant availability in certain areas and discrepancies in pricing between the app and the restaurant can deter users. Concerns about food hygiene, spillage due to poor packaging, and substandard quality are significant drawbacks. Poor customer support makes resolving issues like refunds or cancellations challenging, while security vulnerabilities raise concerns about fraudulent transactions. Additionally, the dependence on internet connectivity can disrupt the ordering experience. These issues highlight the need for better operational efficiency, customer service, and technological robustness in food ordering apps.

FUTURE ENHANCEMENT

The future of online food ordering apps lies in leveraging advanced technologies and user-centric innovations to enhance the overall experience. AI-powered personalization will allow apps to suggest meals tailored to individual preferences, dietary needs, and past orders. Autonomous delivery systems, such as drones and robots, can ensure faster and contactless deliveries, while real-time GPS tracking will provide precise delivery updates. Sustainability features, including eco-friendly packaging and carbon footprint tracking, will appeal to environmentally conscious users. Improved customer support through AI chatbots and streamlined refund processes will enhance reliability. Seamless payment options, including cryptocurrencies and Buy Now, Pay Later services, along with transparent pricing models, will build trust. Apps could integrate health and wellness features, such as calorie tracking and nutritional information, to cater to health-focused users. Augmented reality (AR) could allow customers to preview meals before ordering, and gamification with loyalty programs can boost user engagement. Expanding services to include groceries and everyday essentials, along with offline functionality for areas with poor connectivity, will make these platforms more versatile. These enhancements will ensure a more engaging, reliable, and innovative food ordering experience.