PROJECT TITLE :

FOOD ORDERING APPLICATION

TEAM MEMBERS AND THEIR ROLES :

1. BALAJI R   – [Managed Client-Side]
2. BHARANIDHARAN E   – [Managed Server-Side]
3. ERAIYANBU J   – [Managed Integration]
4. RAGHUL S   – [Managed Full Stack Execution]

INTRODUCTION :

In today's fast-paced world, food ordering websites have revolutionized the way we enjoy meals. My food ordering website aims to bring convenience and variety to users' dining experiences by offering a seamless platform to explore, order, and enjoy delicious food from their favorite restaurants. With a user-friendly interface and robust functionality, this website caters to both food enthusiasts seeking culinary adventures and busy individuals needing quick meal solutions.

The website is designed to provide a hassle-free experience, allowing users to browse an extensive menu of cuisines, filter by preferences, and place orders with ease. A secure login system ensures user data privacy, while advanced search and filtering options make finding the perfect meal straightforward. From appetizers to desserts, users can explore a variety of dishes from local eateries or global chains, ensuring there is something for everyone.

Our platform stands out with features such as real-time order tracking, secure payment gateways, and personalized recommendations based on user preferences. By leveraging responsive design and intuitive navigation, the website is accessible on all devices, ensuring users can order food anytime, anywhere. Additionally, the integration of reviews and ratings enables users to make informed choices, fostering trust and satisfaction.

PURPOSE AND FEATURES :

The primary purpose of my food ordering website is to simplify and enhance the process of ordering food online, catering to the needs of both users and restaurant partners. It aims to provide a platform where users can explore diverse cuisines, place orders effortlessly, and enjoy their favorite meals from the comfort of their homes. By connecting customers with a wide range of restaurants, the website ensures convenience, accessibility, and an improved dining experience. Additionally, the platform empowers restaurant owners to reach a larger audience, expand their customer base, and manage orders efficiently through a digital interface.

KEY FEATURES :

1.User-Friendly Interface :

- Easy-to-navigate design for browsing menus and placing orders.
- Responsive layout for seamless access on desktops, tablets, and smartphones.

2. Restaurant Listings and Menu Display :

- Comprehensive listings of restaurants with detailed menus and pricing.
- Cuisines categorized for quick and easy access.

3. Secure Login and Payment :

- User authentication for secure access to accounts and order history.
- Multiple payment options, including credit/debit cards, digital wallets, and UPI, ensuring safe and convenient transactions.

4. Order Tracking and Updates :

- Real-time tracking of orders from preparation to delivery.
- Notifications to keep users informed about order status.

5. Search and Filter Options :

- Advanced search features to find dishes, cuisines, or restaurants.
- Filters for sorting by ratings, price, distance, or special offers.

6. Personalized Experience :

- Recommendations based on user preferences and past orders.
- Options to save favorite restaurants and frequent orders for quick access.

7. Reviews and Ratings :

- Integrated system for users to leave feedback and rate dishes or restaurants.
- Helps new users make informed choices.

8. Restaurant Partner Portal :

- Tools for restaurant owners to manage menus, orders, and customer interactions.

- Analytics to track performance and customer preferences.

TECHNICAL ARCHITECTURE:

Frontend **Backend** **Database**

React(UI) Express JS Node JS Mongoose MongoDB
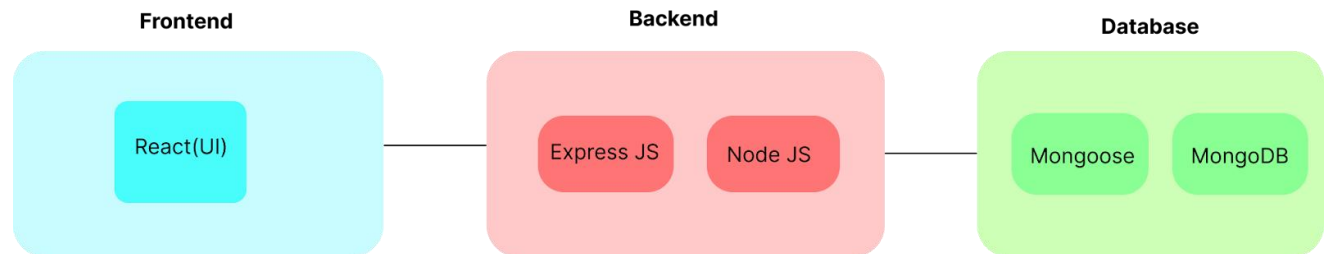
In This Architecture Diagram:

➢ The Frontend is represented by the "Frontend" section, including user interface components such as User Authentication, Cart, Products, Profile, Admin dashboard, etc.,

➢ The Backend is represented by the "Backend" section, consisting of API endpoints for Users, Orders, Products, etc., It also includes Admin Authentication and an Admin Dashboard.

➢ The Database section represents the database that stores collections for Users, Admin, Cart, Orders, and products.

PREREQUISITES :

To develop a full-stack food ordering app using React JS, Node.js, and MongoDB, there are several prerequisites you should consider. Here are the key prerequisites for developing such an application:

Node.js and npm: Install Node.js, which includes npm (Node Package Manager), on your development machine. Node.js is required to run JavaScript on the server side. · Download: https://nodejs.org/en/download/

 Installation Instructions :
https://nodejs.org/en/download/package-manager/

MongoDB: Set up a MongoDB database to store hotel and booking information. Install MongoDB locally or use a cloud-based MongoDB service.

- Download: https://www.mongodb.com/try/download/community
- Installation instructions: https://docs.mongodb.com/manual/installation/

Express.js: Express.js is a web application framework for Node.js. Install Express.js to handle server-side routing,middleware, and API development.

- Installation: Open your command prompt or terminal and run the following command: **npm install express**

React.js: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications. To install React.js, a JavaScript library for building user interfaces, follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling,and JavaScript for client-side interactivity is essential.

Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

Front-end Framework: Utilize Angular to build the user-facing part of the application, including product listings, booking forms, and user interfaces for the admin dashboard.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at: https://git scm.com/downloads

Development Environment : Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

To Connect the Database with Node JS go through the below provided link:

Link: https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

•Visual Studio Code: Download from https://code.visualstudio.com/download

• Sublime Text: Download from https://www.sublimetext.com/download

• WebStorm: Download from https://www.jetbrains.com/webstorm/download

USER & ADMIN FLOW :

1. User Flow :

➢ Users start by registering for an account.

➢ After registration, they can log in with their credentials.

➢ Once logged in, they can check for the available products in the platform.

➢ Users can add the products they wish to their carts and orders.

➢ They can then proceed by entering address and payment details.

➢ After ordering, they can check them in the profile section.

2. Restaurant Flow :

➢ Restaurants start by authenticating with their credentials.

➢ They need to get approval from the admin to start listing the products.

➢ They can add/edit the food items.

3. Admin Flow :

➢ Admins start by logging in with their credentials.

➢ Once logged in, they are directed to the Admin Dashboard.

➢ Admins can access the users list, products, orders, etc.

FOLDER STRUCTURE :
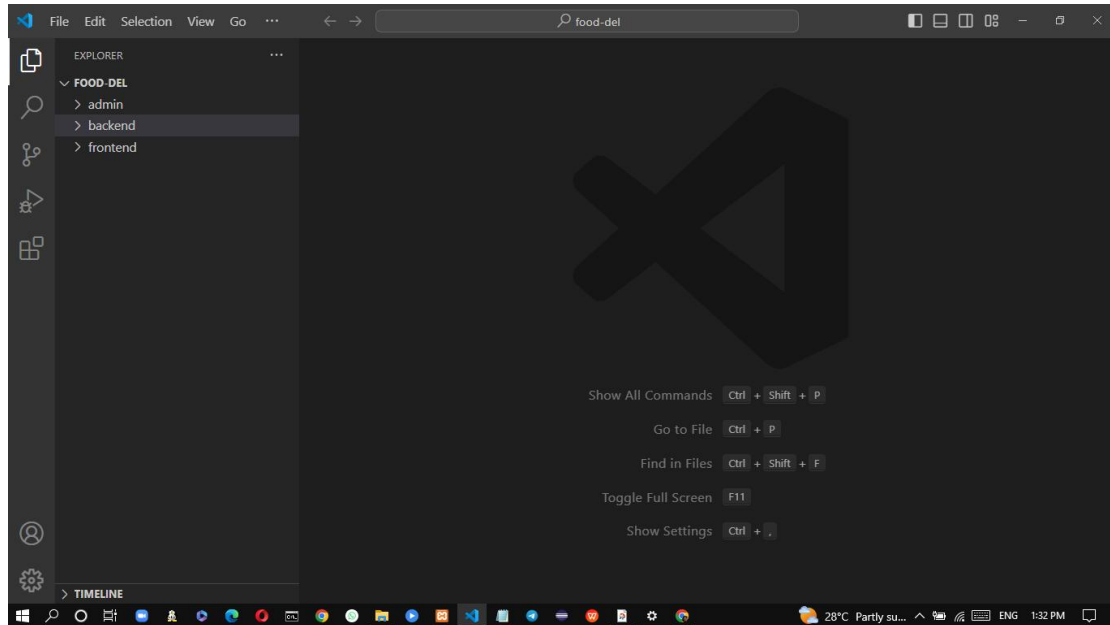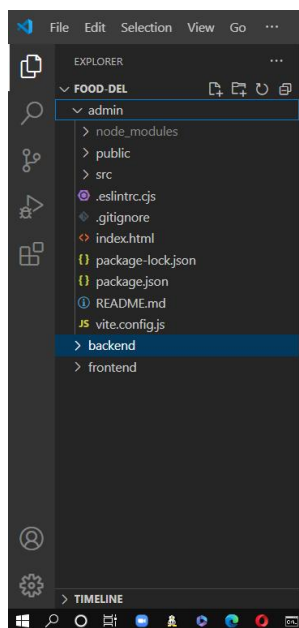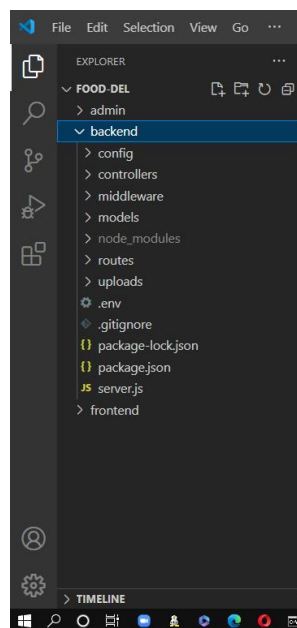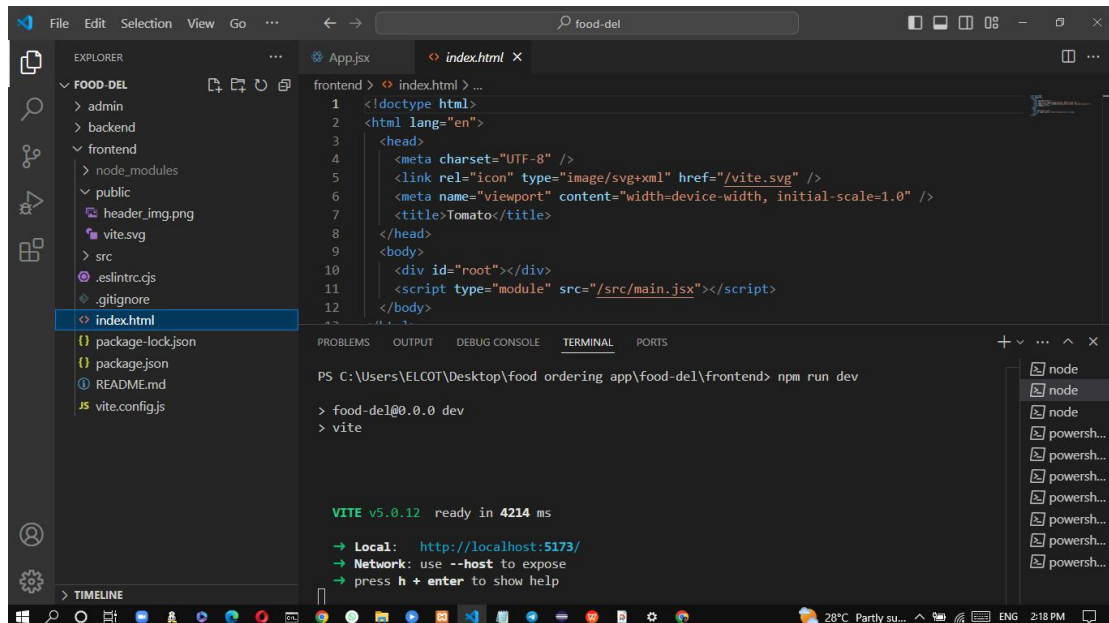
Admin :	Backend :	Frontend :
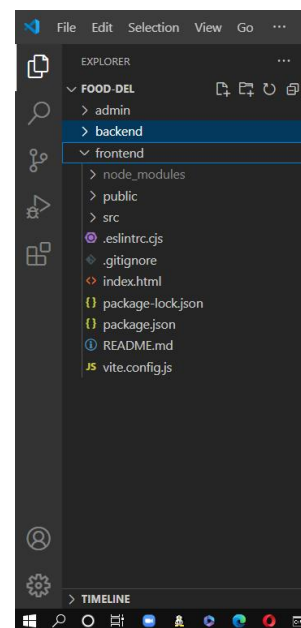
RUNNING THE APPLICATION :

FRONTEND :

BACKEND :

API DOCUMENTATION :

1. User Registration:

- User sends a request to register with their details (name, email, password, etc.).
- The system validates the input data.
- If validation passes, a new user account is created.
- The system responds with a success message and user ID.

---

2. User Login:

- User sends login credentials (email and password).
- The system verifies the credentials.
- If valid, a token (JWT or similar) is generated and returned.
- User uses the token for authenticated requests.

---

3. Get User Profile:

- User sends a request with the authentication token.
- The system verifies the token and retrieves user details.
- The system returns the user's profile information.

Admin API Steps :

1. Admin Login:

- Admin provides credentials to log in.
- The system verifies the credentials.
- If valid, a token is issued for admin-level access.

---

2. View All Users:

- Admin sends a request with their authentication token.
- The system verifies the token and fetches all user records.
- The system returns the list of users.

---

3. Delete a User:

- Admin sends a request to delete a specific user, providing the user ID.
- The system verifies the admin token and checks if the user exists.
- If valid, the user is deleted, and the system responds with a confirmation.

---

4. Update User Role:

- Admin sends a request to update a user's role, specifying the user ID and new role.
- The system verifies the admin token and checks the validity of the new role.
- If valid, the user's role is updated, and the system responds with a success message.

AUTHENTICATION MECHANISM :

1.User/Admin Registration:

- Collect user/admin details (e.g., email, password) and store the password securely after hashing.

2.Login:

- User/admin provides email and password for authentication.

3.Credential Verification:

- Validate the email and password by comparing the provided password (hashed) with the stored hash.

4.Token Generation:

- On successful login, generate a secure token (e.g., JWT) containing user/admin details and expiry.

5.Token Usage:

- Client includes the token in the **Authorization header** for subsequent requests.

6.Token Verification:

- Server verifies the token, checks user/admin permissions, and processes the request if authorized.

USER INTERFACE :

SCREENSHOTS :

LANDING PAGE :
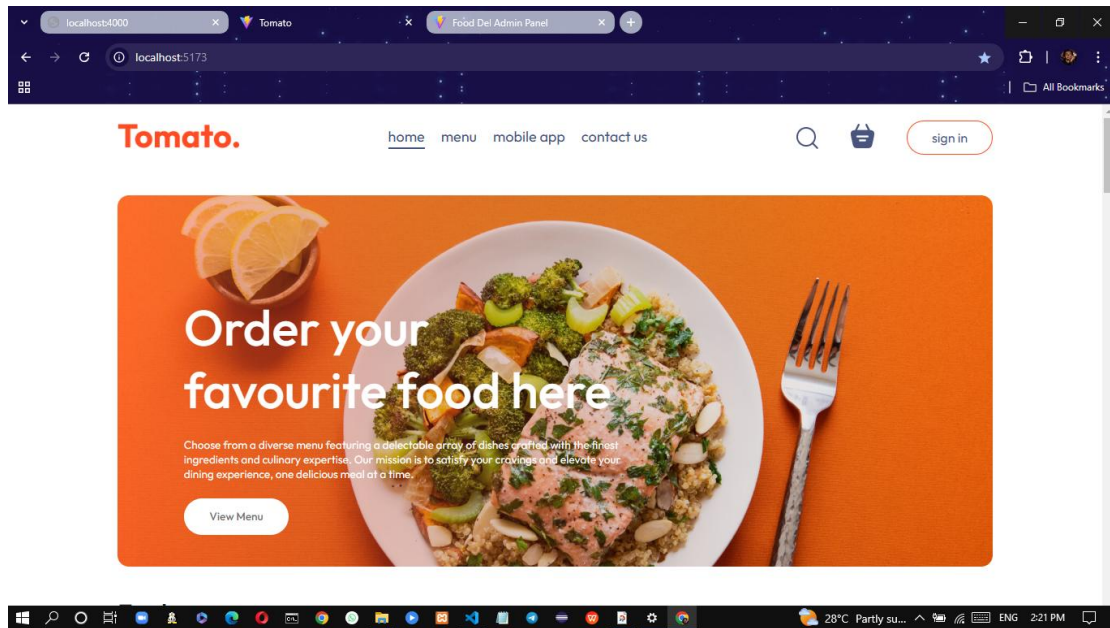
RESTAURANT MENU PAGE:
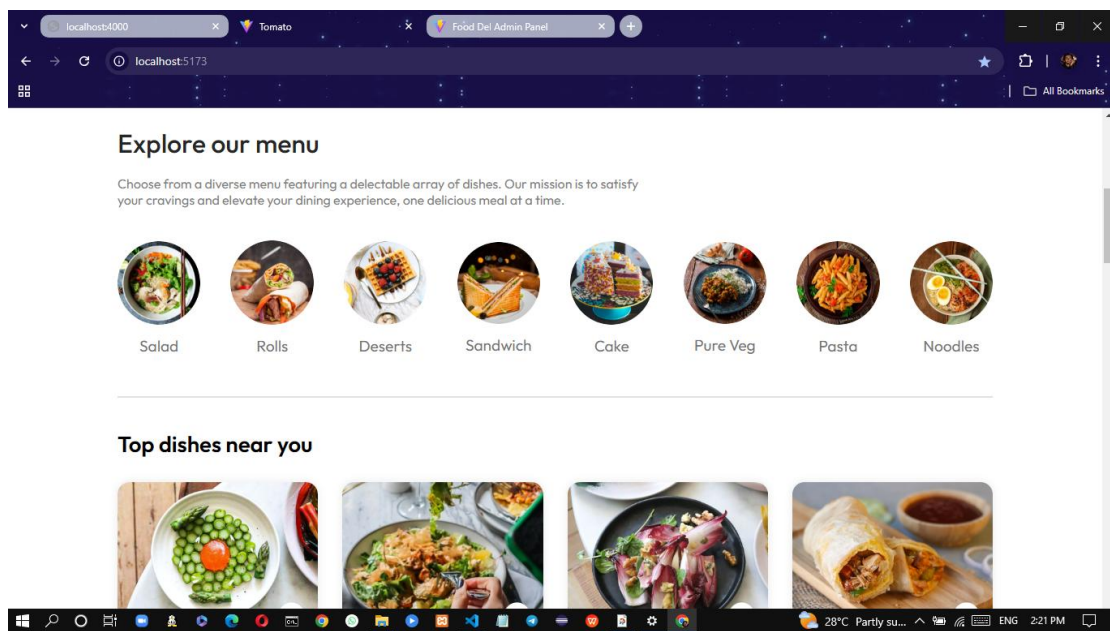
CONTACT DETAILS PAGE:
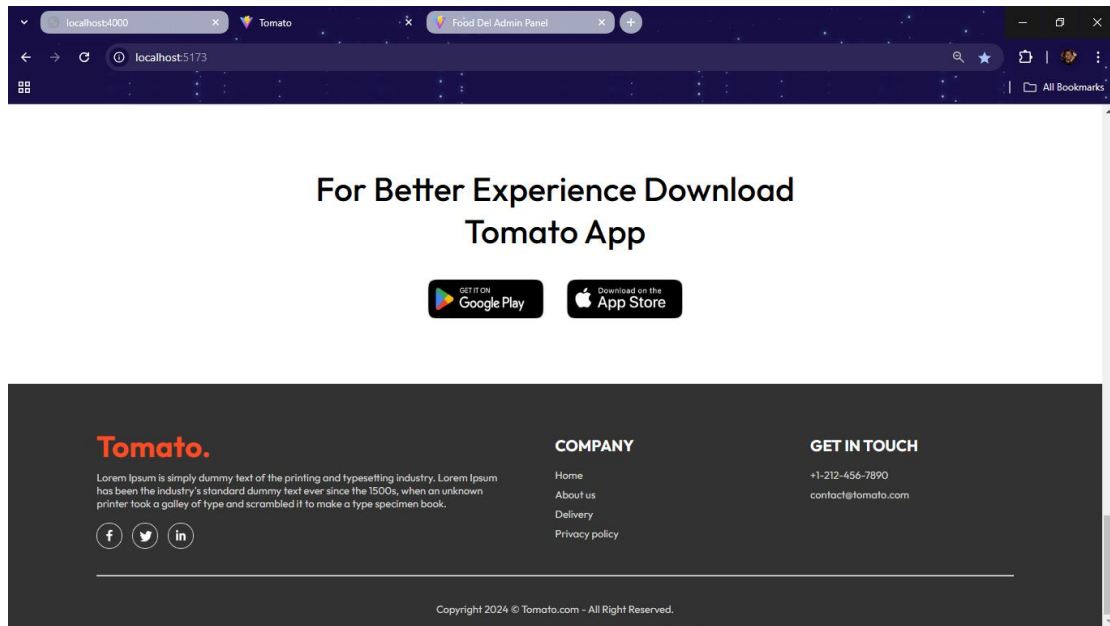
LOGIN AND SIGNUP PAGES :

LOGIN PAGE :
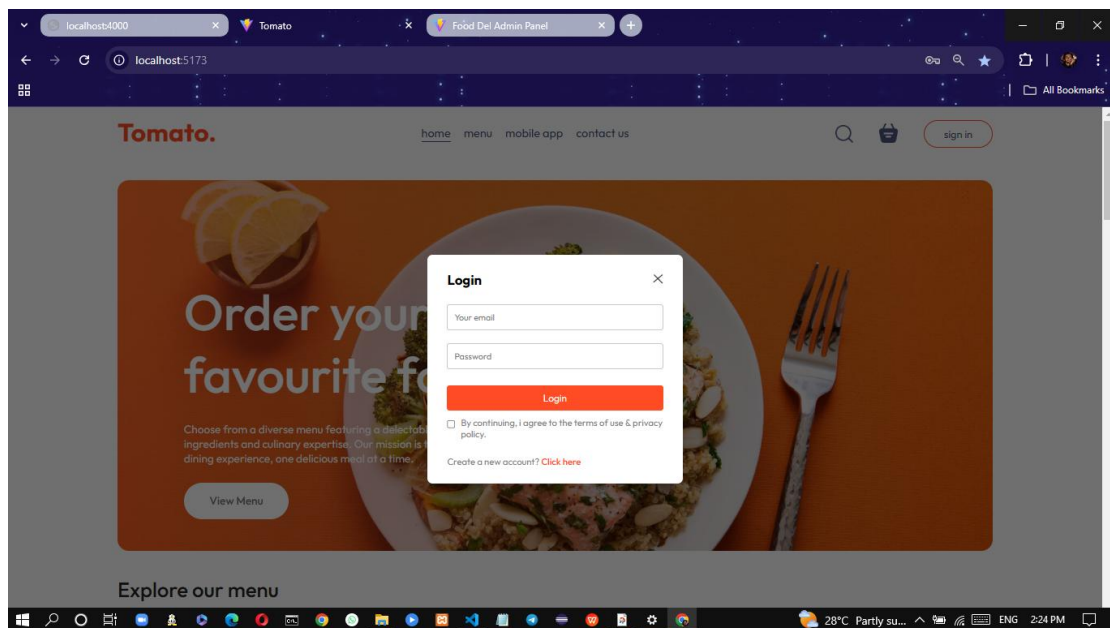
SIGNUP PAGE :
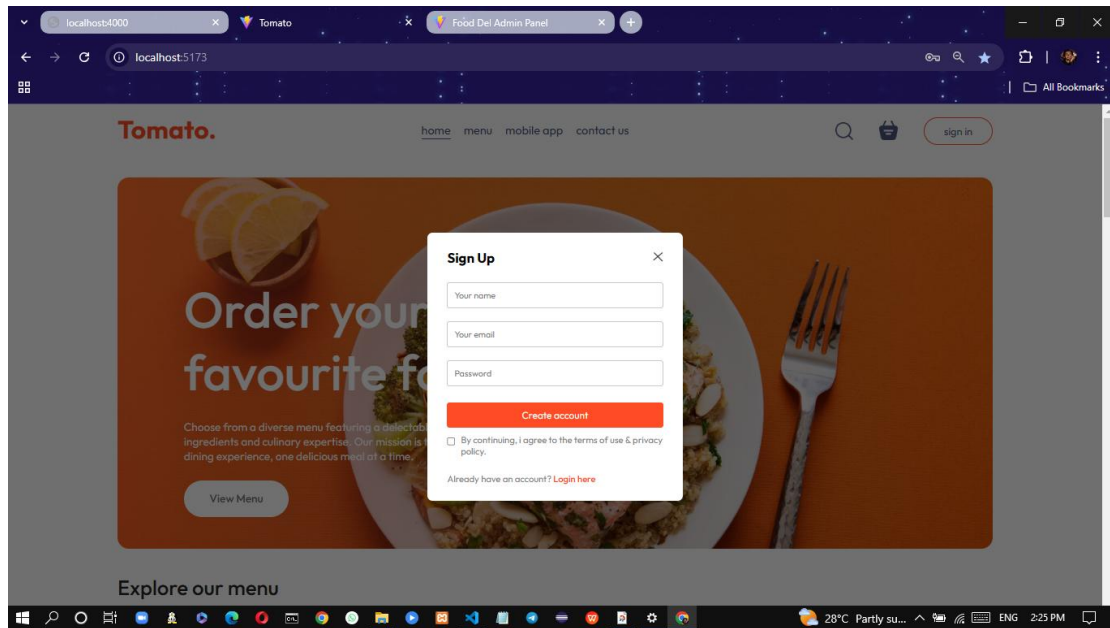
USER PROFILE PAGE:

CART ITEMS PAGE :

ADDRESS AND PAYMENT METHOD PAGE:
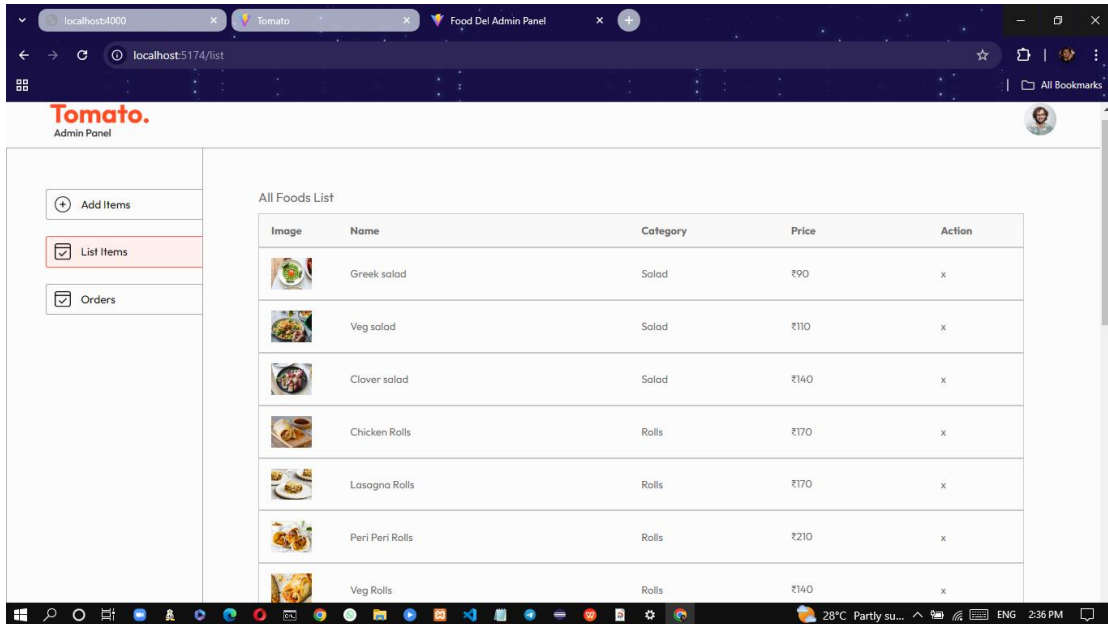
ADMIN DASHBOARD PAGE:
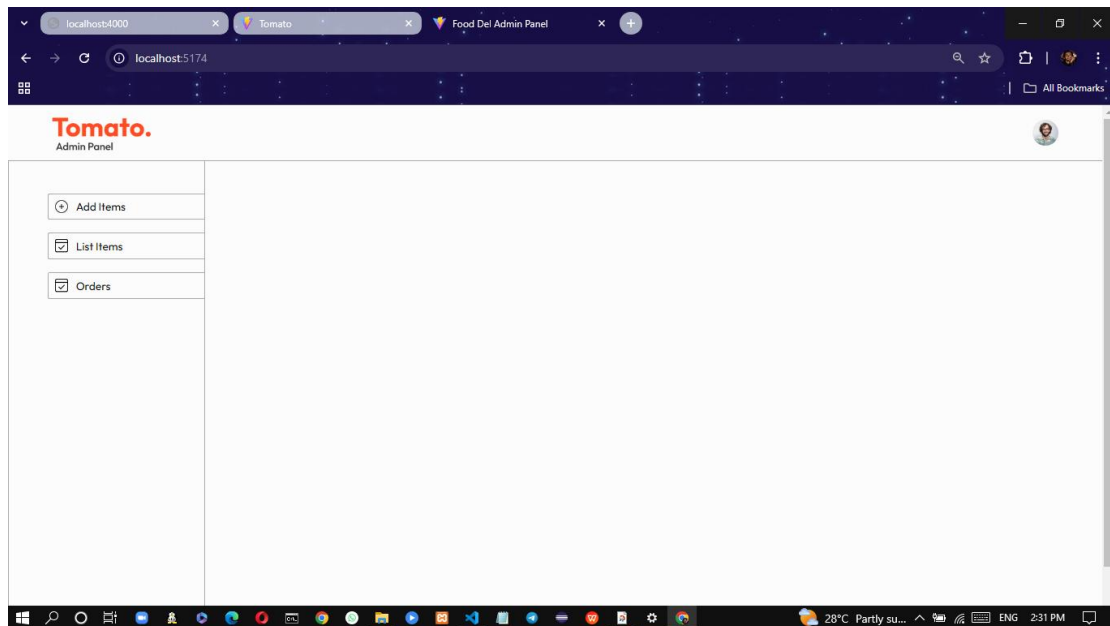
ADDING NEW ITEMS PAGE:

LIST ITEMS PAGE:

ORDERS PAGE :

USERS CAN ALSO TRACK THE ORDERS PAGE :

LOGOUT PAGE :

17
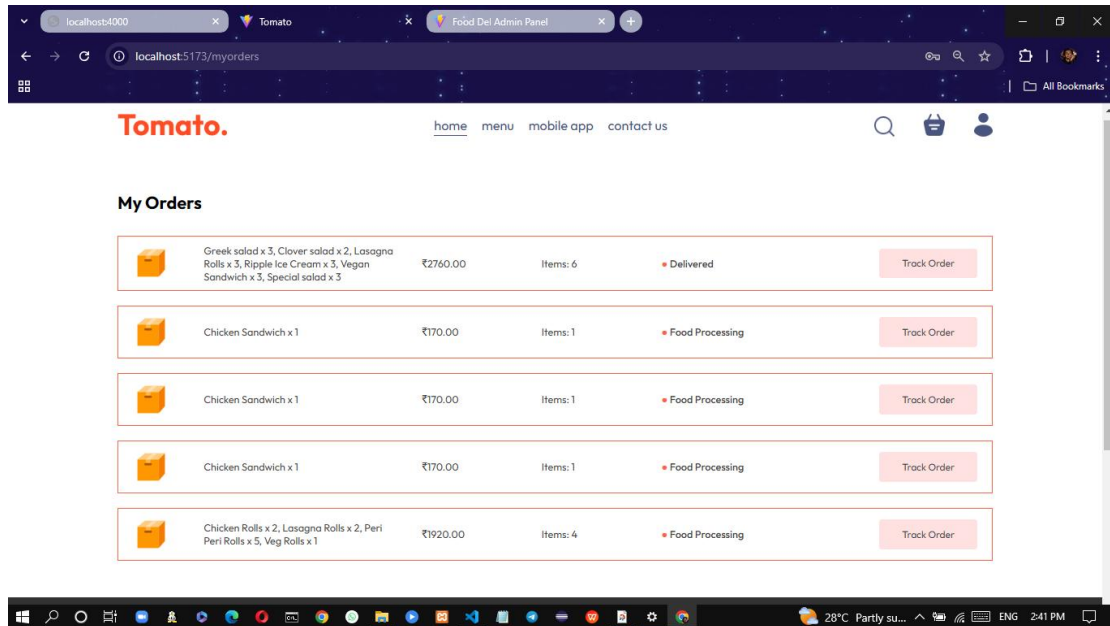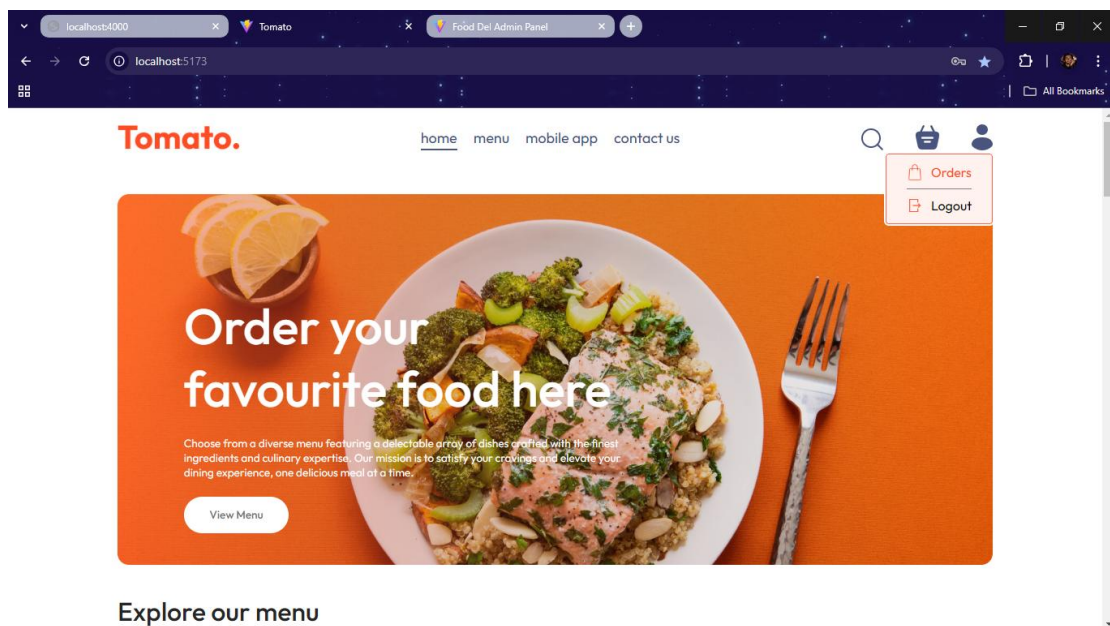
TESTING STRATEGY :

1. Functional Testing:
- Purpose: Verify that all core features, like order placement, payment, and item selection, work as expected.
- Tools: Selenium, Postman (for API testing), Cypress

2. Security Testing:
- Purpose: Check for vulnerabilities like SQL injection, XSS, and ensure secure handling of sensitive data (e.g., passwords and payment details).
- Tools: OWASP ZAP, Burp Suite, Acunetix

3. Performance Testing:
- Purpose: Test how the system performs under heavy traffic, checking for slowdowns or crashes.
- Tools: Apache JMeter, LoadRunner, Gatling

DEMO VEDIO LINK :

https://drive.google.com/file/d/1yeZ85Jq7kGxlGPz9tFNGbHd8K_P9G_fC/view?usp=drivesdk

KNOWN ISSUES :

1. Order Processing Errors: Incorrect status updates or failed order processing due to transaction issues.

2. Payment Gateway Integration: Payment failures or lack of transaction confirmation after successful payment.

3. Database & Inventory Management: Incorrect stock levels or delayed queries affecting item availability.

FUTURE ENHANCEMENTS :

1. Mobile App Integration:

- ➢ **Enhancement:** Develop a mobile application for iOS and Android to reach a broader audience and improve user experience.
- ➢ **Tools**: React Native, Flutter, Xamarin

2. AI and Personalization:

- ➢ **Enhancement:** Implement AI-driven features like personalized recommendations, predictive text for faster ordering, and user behavior analysis for targeted promotions.
- ➢ **Tools:** TensorFlow, Python, Google Cloud AI, Firebase ML

3. Real-time Order Tracking:

- ➢ **Enhancement:** Add real-time order tracking so customers can monitor their order status and delivery progress.
- ➢ **Tools:** Firebase Realtime Database, WebSockets, Google Maps API.

REFERENCES :

GITHUB REPO :

https://github.com/RaghulExtremze/Food_ordering_web.git

DEMO VEDIO :

https://drive.google.com/file/d/1yeZ85Jq7kGxlGPz9tFNGbHd8K_P9G_fC/view?usp=drivesdk