1. Introduction

Project Title: Food Order on the Go
Team ID: LTVIP2025TMID53004
Team Members:
- B Samuel Joe Paul - Team Leader
- Arimilli Mokshznanivas - Frontend Developer
- Aravapalli Hema Sri Pragna - Backend Developer
- Arepalli Madhu Sri - Database & Integration

2. Project Overview

Purpose:
To develop a smart and user-friendly food ordering platform that allows users to quickly browse nearby restaurar

Features:
- User authentication (Login/Register)
- Restaurant listing and detail pages
- Menu browsing and item selection
- Add to cart and order placement
- Real-time order status tracking
- Admin and restaurant dashboards

3. Architecture

Frontend:
Built using React.js with context API for state management and React Router for navigation. Tailwind CSS is use

Backend:
Developed using Node.js and Express.js. The API handles authentication, user and order management, and con

Database:
MongoDB Atlas stores user data, restaurant data, orders, and menu items. Mongoose is used for schema model

4. Setup Instructions

Prerequisites:
- Node.js (v18+)
- MongoDB Atlas account or local MongoDB
- Git

Installation:
git clone https://github.com/your-repo/food-order-go
cd food-order-go

For frontend
cd client-app
npm install

For backend
cd ../server
npm install

Environment Variables:
Create a .env file in the server folder:
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret

5. Folder Structure

Client (React):
client-app/
■■■ src/
■ ■■■ components/
■ ■■■ pages/
■ ■■■ context/
■ ■■■ App.js

Server (Node/Express):
server/
■■■ models/
■■■ routes/
■■■ controllers/
■■■ server.js

6. Running the Application

Frontend:
cd client-app
npm start

Backend:
cd server
npm start

7. API Documentation

Method	Endpoint	Description
POST	/api/register	Register a new user
POST	/api/login	Login existing user
GET	/api/restaurants	Get list of restaurants
GET	/api/menu/:id	Get menu of a restaurant
POST	/api/order	Place a new order
GET	/api/orders	Fetch user orders

8. Authentication

JWT (JSON Web Tokens) is used for secure login and route protection.
Token is stored in local storage and passed in headers for protected routes.

9. User Interface

Include your UI screenshots here - Login Page, Home Page, Restaurant List, Order Summary, etc.

10. Testing

Manual testing for all endpoints using Postman.
Component-level testing using React Testing Library (in progress).

11. Screenshots or Demo

Add screenshots of major pages or attach a demo link (YouTube, Netlify, Vercel, etc.).

12. Known Issues

- Delay in live order updates without real-time WebSocket integration

- Admin dashboard is under development

13. Future Enhancements

- Add real-time order tracking with Socket.IO
- Implement payment gateway integration
- Push notifications for order status
- Add customer reviews and ratings