**Full Stack Development with MERN**

**Project Documentation**

**OrderOnTheGo – On-Demand Food Ordering Solution (SB Foods)**

**1. Introduction**

Project Title: OrderOnTheGo – On-Demand Food Ordering Solution (SB Foods)

Team ID : LTVIP2026TMIDS71750

Team Size : 4

Team Leader : Om Abhiram Beella

Team Member : Velugula Hemanth

Team Member : Sriya Badampudi

Team Member : Teki Vijayaratnam

**2. Project Overview**

Purpose: OrderOnTheGo (SB Foods) is a full-stack MERN application developed to provide a seamless and efficient online food ordering experience.

Features include User Registration & Login, Restaurant Registration & Approval, Browse Food Items, Add to Cart, Secure Payment Integration, Order Tracking, Admin Dashboard, and Role-Based Authentication.

**3. Architecture**

Frontend: Built using React.js with reusable components, routing and API integration.

Backend: Developed using Node.js and Express.js providing REST APIs.

Database: MongoDB using Mongoose ODM with collections for Users, Restaurants, Food Items, Cart and Orders.



**4. Setup Instructions**

Prerequisites: Node.js, MongoDB, Express.js, React.js, Git, VS Code.

Installation: Clone repository, install server dependencies (npm install), install client dependencies (npm install), configure .env file.

**5. Folder Structure**

Client Structure:

client/ public/ src/api src/assets src/components src/pages/admin src/pages/customer src/pages/restaurant App.js main.js index.css package.json README.md

client/

├── public/

├── src/

│ ├── api/

│ ├── assets/

│ ├── components/

│ ├── pages/admin/

│ ├── pages/customer/

│ ├── pages/restaurant/

│ ├── App.js

│ ├── main.js

│ └── index.css

├── package.json

└── README.md

Server Structure:

server/ config/db.js controllers/ middleware/ models/ routes/ .env index.js package.json RAZORPAY\_SETUP.md README.md

server/

├── config/

│ └── db.js

├── controllers/

│ ├── adminController.js

│ ├── authController.js

│ ├── cartController.js

│ ├── foodController.js

│ ├── orderController.js

│ ├── paymentController.js

│ ├── restaurantController.js

│ └── userController.js

├── middleware/

│ ├── authMiddleware.js

│ └── roleMiddleware.js

├── models/

│ ├── Admin.js

│ ├── Cart.js

│ ├── Food.js

│ ├── Order.js

│ ├── Restaurant.js

│ └── User.js

├── routes/

│ ├── adminRoutes.js

│ ├── authRoutes.js

│ ├── cartRoutes.js

│ ├── foodRoutes.js

│ ├── orderRoutes.js

│ ├── paymentRoutes.js

│ ├── restaurantRoutes.js

│ └── userRoutes.js

├── .env

├── index.js

├── package.json

├── RAZORPAY\_SETUP.md

└── README.md

**6. Running the Application**

Backend: cd server -> npm start (http://localhost:5000)

Frontend: cd client -> npm run dev (http://localhost:5173)

**7. API Documentation**

POST /api/auth/register – Register User

POST /api/auth/login – Login User

GET /api/foods – Get Food Items

POST /api/cart – Add to Cart

POST /api/orders – Place Order

GET /api/admin/users – Get All Users

**8. Authentication**

JWT-based authentication. Tokens generated on login and validated using middleware.

Role-based access control for User, Restaurant and Admin.

**9. Testing**

Manual testing of all user flows.

API testing using Postman.

Form validation and protected route testing.

**10. Known Issues**

Minor UI responsiveness adjustments may be required for smaller screens.

Real-time tracking not implemented.

**11. Future Enhancements**

Real-time tracking using WebSockets.

Push notifications.

AI-based recommendations.

Ratings and Reviews system.

Mobile Application version.

**12. Screenshots**

































