**Full Stack Development with MERN Project Documentation format**

# 1. Introduction

* **Project Title:** OrderOnTheGo: Your On-Demand Food Ordering Solution
* **Team Members:**.

1. Marem Rekha
2. Rushitha Konangi
3. K.Sahithi
4. V.Sarath Kumar

# 2. Project Overview

OrderOnTheGo is a web-based food delivery application designed to streamline the food ordering process for customers, restaurants, and administrators. The platform enables users to browse restaurant menus, add food items to a cart, and place orders online, while also providing dashboa

* **Purpose:**
* The purpose of this project is to build a responsive, intuitive, and fully functional food ordering solution using HTML, CSS, and JavaScript. The system ensures seamless interaction among all stakeholders, with features like session-based login, cart management, menu browsing, and administrative control.
* **Features:**

User authentication (Login/Register)

Restaurant and food item listings

Persistent cart with live updates

Admin dashboard for restaurant/item management

Restaurant dashboard for order tracking and menu updates

# 3. Architecture

* **Frontend**

Developed entirely in HTML, CSS, and JavaScript with modular JS files (orders.js, restaurant.js, etc.). Persistent navigation and cart bar across all pages..

* **Backend:**

Not implemented for this phase. Data is simulated using localStorage for all CRUD operations.

* **Database:**

Users, Restaurants, Food Items, Orders — all stored and manipulated via localStorage. Future plans include MongoDB schema with collections

**4. Setup Instructions**

* **Prerequisites:**

Any modern web browser. No installations required as project is frontend-only.

* **Installation:**

1. Clone or download the project folder.

2. Open index.html in browser.

3. Navigate between pages using the navigation bar.

**5. Folder Structure**

* **Client:**
* /index.html – Landing page
* /restaurants.html – Lists restaurants
* /menu.html – Restaurant-specific menu
* /cart.html – View and checkout cart
* /admin.html, /dashboard.html – Admin/restaurant views
* /css/ – Contains all styling files
* /js/ – Contains orders.js, restaurant.js, auth.js, etc.
* **Server:**
* Not applicable. All logic is in client-side JS using localStorage.

# 6. Running the Application

• Open index.html in a browser.

•No terminal commands required.

•All pages interlinked and functional offline through localStorage

**o Frontend**

Developed entirely in HTML, CSS, and JavaScript with modular JS files (orders.js, restaurant.js, etc.). Persistent navigation and cart bar across

**o Backend**:

Not implemented for this phase. Data is simulated using localStorage for all CRUD operations

# 7. API Documentation

* No backend APIs in current version
* Future plan: REST APIs with Node.js (Login, Register, Orders CRUD, Restaurant Management).

# 8. Authentication

* Custom login and registration system using JavaScript.
* Sessions managed using localStorage keys.
* Role-based redirection for users, restaurants, and admins.

1. **User Interface**

* Swiggy-like design using HTML & CSS.
* Home bar and cart bar persistent across all pages.
* Restaurant cards with image, name, cuisine.
* Food cards with pricing and add-to-cart buttons.

1. **Testing**

* Manual testing on Chrome, Firefox, and Edge.
* Tested across devices (desktop, tablet, mobile).
* Verified session handling, cart updates, and navigation logic.

1. **Screenshots or Demo**

* [Insert Screenshots of landing page, restaurant view, menu, cart, admin dashboard]
* [Insert GitHub repository/demo link here]

1. **Known Issues** 
   * Clearing browser storage deletes all data
   * No actual payment or backend validation
   * Limited scalability without backend integration

# 13. Future Enhancements

* Add backend using Node.js, Express, MongoDB
* Real-time order updates for restaurants
* User profile and order history page
* Payment gateway integration (Razorpay, Stripe)
* Mobile app version using React Native or Flutter.