

Full Stack Development with MERN

Project Documentation

1.Introduction

- Project Title: OrderOnTheGo-Your On-Demand Food Ordering Solution
- Team members: Katam Siva Naga Sravanthi – Team Leader
Bandla Krishna Priya – Team member
Bonigala Nandhini – Team member
Rangisetty Deepika – Team member

2.Project Overview

- Purpose: The Food Ordering App is a web application designed to facilitate online food ordering and delivery services. It connects customers with restaurants, allowing users to browse menus, place orders, make payments, and track deliveries.
- Features: User registration and authentication system
Restaurant browsing with search and filter capabilities
Dynamic menu display with categories and item details
Shopping cart functionality with quantity management
Secure payment processing integration
Order tracking and status updates
User profile management and order history
Admin dashboard for restaurant and order management
Responsive design for mobile and desktop platforms

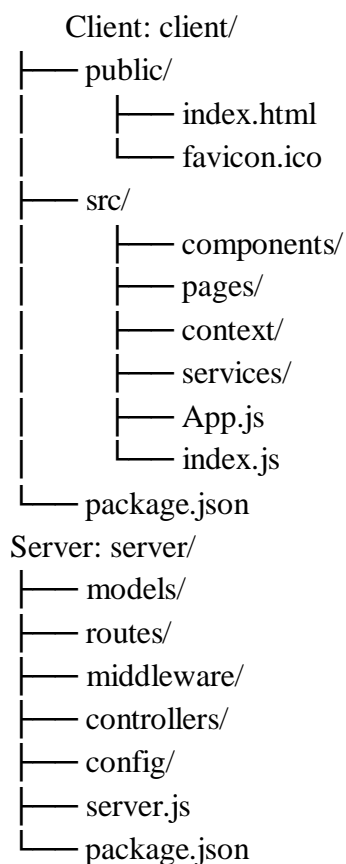
3.Architecture

- Frontend: Built with React.js using functional components and hooks
 - React Router for client-side navigation
 - Context API for global state management
 - Axios for HTTP requests to backend APIs
 - CSS modules for component styling
- Backend: Node.js runtime environment
 - Express.js framework for RESTful API development
 - JWT (JSON Web Tokens) for authentication
 - Multer middleware for file uploads
 - CORS middleware for cross-origin requests
- Database: MongoDB as the primary database
 - Mongoose ODM for object modeling and schema validation
 - Collections include Users, Restaurants, MenuItems, Orders, and Reviews
 - Indexes on frequently queried fields for performance optimization

4.Setup Instructions

- Prerequisites: Node.js (version 14.0 or higher)
 - MongoDB (version 4.4 or higher)
 - npm package manager
 - Git for version control
- Installation: Clone the repository:
 - git clone: [SRAVANTHI-KATAM/Food-Ordering-App-MERN-main](https://github.com/SRAVANTHI-KATAM/Food-Ordering-App-MERN-main)
 - cd Food-Ordering-App-MERN-main
- Install backend dependencies:
 - cd server
 - npm install
- Install frontend dependencies:
 - cd ./client
 - npm install
- Create .env file in server directory:
 - Port= 3000
 - MongoDB_Url=mongodb://localhost:3000/foodorderingapp
 - JWT_SECRET=your_jwt_secret_key NODE_ENV=development
 - Start MongoDB servive locally

5.Folder Structure



6. Running the Application

- Frontend: cd client
 - npm start (runs at http://localhost:3000)
- Backend: cd server
 - npm start (runs at http://localhost:5000)
- Concurrent Dev:
 - npm run dev (runs both frontend and backend)

7.API Documentation

- Authentication:
 - POST /api/auth/register
 - POST /api/auth/login
 - GET /api/auth/profile (Protected)
- Restaurants:
 - GET /api/restaurants
 - GET /api/restaurants/:id
- Menu:
 - GET /api/menu/:restaurantId
- Orders:
 - POST /api/orders (Protected)
 - GET /api/orders/user/:userId (Protected)
- Example Response:

```
{  
  
  "success": true,  
  "data": { "restaurant": {  
    "_id": "60f1b2b3c45a1b2c3d4e5f67", "name": "Pizza Palace",  
    "cuisine": "Italian", "rating": 4.5,  
    "deliveryTime": "30-45 mins"  
  }  
}
```

8.Authentication

- The application uses JWT tokens:
 - User credentials checked in MongoDB
 - Token generated on successful login
 - Stored in localStorage on frontend
 - Token required for protected routes
 - Token expires in 24 hours
- JWT Structure:
 - Header: Type and algorithm

- Payload: User ID and expiration
- Signature: Encrypted with secret

9. User Interface

- Pages:
 - Homepage: Search, featured restaurants, categories
 - Restaurant List: Filters, search, ratings
 - Menu: Item categories, add to cart, images
 - Cart: Order summary, checkout
 - Profile: Order history, address, settings

10. Testing

- Testing Approach:
 - Unit: Jest
 - Component: React Testing Library
- API: Postman
- Integration: Frontend-backend interaction
- Coverage:
 - Auth flows
 - Menu/order management
 - Form validation

11. Screenshots or Demo

Video Demo: https://drive.google.com/file/d/1y9anr_7fVWWuz-6J5QEm47KBegifBoNP/view?usp=sharing

12. Known Issues

- Payment testing needed in production
 - Image upload is slow for large files
 - WebSocket tracking not done
 - Mobile layout needs improvement
 - Search lacks fuzzy matching
- Browser Compatibility:
 - Chrome, Firefox, Safari (tested)
 - IE support limited

13. Future Enhancements

- Push notifications
- Fuzzy search
- Multiple payment gateways

- React Native app
- GPS delivery tracking
- Customer reviews
- Loyalty rewards
- Multi-language support
- Restaurant analytics dashboard
- Voice ordering
- Migrate to TypeScript
- GraphQL APIs
- PWA support
- Input sanitization, rate limiting
- MongoDB performance tuning
- Microservices architecture