Full Stack Development with MERN

Project Documentation

1. Introduction

- **Project Title:** OrderOnTheGo: Your On-Demand Food Ordering Solution (DHK FODDIES DEVILERY APP)

- **Team Members:**

  o   D. Hemanth kumar (Team Leader)

  o   D. Keerthi

  o   Y. Keerthi

  o   S. Topi vali

  o   K. Leela Siddu

2. Project Overview

- **Goal:** To develop a full-featured, functional prototype of a modern food ordering and delivery system using the MERN stack.

- **Key Features:**

  o   **User Authentication:** Secure user registration and login system with email and password.

  o   **Restaurant & Product Listing:** Dynamically fetches and displays food items from a MongoDB database.

  o   **Shopping Cart:** Fully functional cart where users can add, view, and manage items before checkout.

  o   **Data Persistence:** Uses a real MongoDB database, not mock data, ensuring data is saved and retrieved.

  o   **Backend API:** A robust backend built with Node.js and Express to handle all business logic.

o **Responsive UI:** A user-friendly interface built with React that works on different screen sizes.

3. Architecture

- **Frontend:** React.js for building the user interface and managing component state.
- **Backend:** Node.js and Express.js to create a RESTful API for handling data, users, and orders.
- **Database:** MongoDB with Mongoose for data modeling and persistent storage of users, products, and orders.

4. Setup Instructions

- **Requirements:** Node.js, NPM, and a running MongoDB instance.
- **Run:**
  1. Clone the repository and install dependencies for both client and server using npm install.
  2. **Run the custom seeder script** to populate the database with initial data: cd server then node seeder.js.
  3. Start the backend server: cd server then npm start.
  4. Start the frontend server in a separate terminal: cd client then npm start.

5. Folder Structure

- client/ (Contains the React Frontend application)
- server/ (Contains the Node.js/Express Backend API)
- seeder.js (The custom script created to import data into MongoDB)

6. API Endpoints

- POST /api/users/register (Handles new user registration)
- POST /api/users/login (Authenticates and logs in existing users)

- GET /api/products (Fetches all restaurant/food items to display on the homepage)

7. Authentication

- Uses Email & Password for user registration and login.

- Backend handles user creation, data validation, and verification.

- Frontend client communicates with the backend via a proxy configured in client/package.json.

8. User Interface Screens

- Login & Register Pages

- Home Page (Product Listing)

- Shopping Cart

- User Profile (for viewing order history)

9. Testing

- **Frontend:** Manual testing of UI components and user flows (like adding to cart and logging in) was conducted across different browsers using Chrome DevTools.

- **Backend:** API endpoints were tested using **Postman** to ensure correct request handling, data validation, and response structure for all routes.

10. Screenshots or Demo

- Screenshots and a video demonstration of the working application can be provided separately in the project folder.