

# Full Stack Development with MERN

## Project Documentation

### 1. Introduction

- **Project Title:** ShopSmart – Your Digital Grocery Store (Grocery-Shop)
- **Team Members:**
  - Bhanu Gandluru (Team Leader)
  - Kopparapu Siva Sai Harish
  - Shaik Mohammed Khayum
  - Velpula Rama Thulasi
  - C Abdul

### 2. Project Overview

- **Purpose:** To provide a modern, real-time online platform for customers to browse and purchase groceries, and for administrators to manage products and users efficiently.
- **Features:** User authentication (register/login), product Browse and searching, shopping cart functionality, user profile management, and an admin dashboard for inventory control.

### 3. Architecture

- **Frontend:** Developed using **React** for a dynamic and responsive user interface. Handles all client-side logic and presentation.
- **Backend:** Built with **Node.js** and the **Express** framework. It handles API requests, business logic, and server-side operations.

- **Database: MongoDB** (connected via MongoDB Atlas) is used as the database to store all application data, including user credentials, product information, and orders.

#### 4. Setup Instructions

- **Prerequisites:** Node.js, npm (Node Package Manager), Git, and a MongoDB Atlas account are required.

#### 5. Folder Structure

- **frontend:** Contains the complete React frontend source code, including components, pages, and styles.
- **root (/):** Contains the backend server implemented with Express, including models, routes, controllers, and middleware.

#### 6. Running the Application To start the application:

- **Backend:** Run `npm run dev` from the project's **root** directory.
- **Frontend:** Run `npm start` from inside the **/frontend** directory.

#### 7. API Documentation The backend exposes REST APIs for essential operations. Key endpoints include:

- `POST /api/auth/register` - User registration
- `POST /api/auth/login` - User login
- `GET /api/products` - Retrieve a list of all groceries
- `GET /api/products/:id` - Retrieve a single grocery item by its ID
- `POST /api/cart` - Add an item to the user's shopping cart

#### 8. Authentication

- Standard email and password-based login for both Users and Admins.
- The system uses **JWT (JSON Web Tokens)** for securing API routes and maintaining user sessions for enhanced security.

**9. User Interface** Screens provided in the system include:

- Login & Register Page
- Home Page (Product Listing)
- Product Details Page
- Shopping Cart Page
- User Profile/Dashboard
- Admin Dashboard (for managing products, orders, and users)

## **10. Testing**

- **Backend:** API endpoints can be tested using tools like **Postman** for validation and debugging.
- **Frontend:** The user interface and functionality can be tested manually or by using the browser's built-in **Chrome DevTools**.

## **11. Screenshots or Demo**

- Screenshots and demo videos can be provided separately in the project folder to showcase the application's functionality.