# 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.

### 3.3 Technology Stack

- **Frontend:** React
- **Backend:** Node.js / Express
- **Database:** MongoDB

## 4. Project Design

**4.1 Problem Solution Fit** Provides a digital, web-based solution to the physical limitations and inconveniences of traditional grocery shopping.

**4.2 Proposed Solution** A role-based web application for customers and an administrator. Customers can browse and buy products, while the admin manages the store's inventory.

**4.3 Solution Architecture** Frontend (Client Side) -> Backend API -> MongoDB (Database)

## 5. Project Planning & Scheduling

| Week | Task |
|---|---|
| 1 | Requirements & Planning |
| 2 | UI/UX Design & Prototyping |
| 3 | Frontend Development |
| 4 | Backend Development & API |
| 5 | Integration & Testing |
| 6 | Final Submission & Review |

Export to Sheets

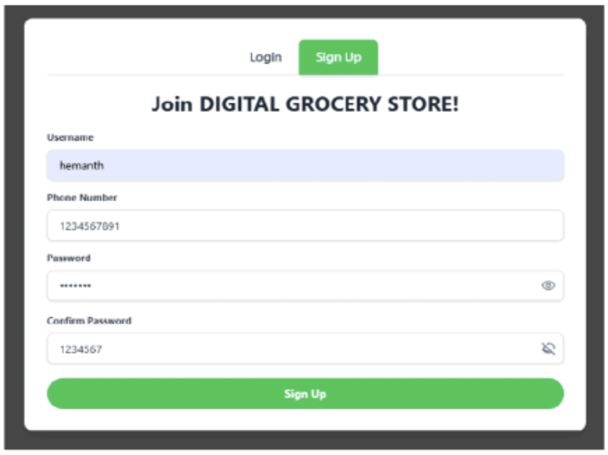
## 6. Functional and Performance Testing

- **Performance Testing Tools:** Postman (for API response time), Browser Dev Tools (for frontend load time).
- **API Response Time:** Optimized to be less than 500ms for most queries.
- **Database Query Time**: Optimized through proper indexing on the MongoDB collections.

## 7. Results

The project resulted in a fully functional e-commerce web application with the following key screens:

- **User Sign Up & Login**



Join DIGITAL GROCERY STORE!

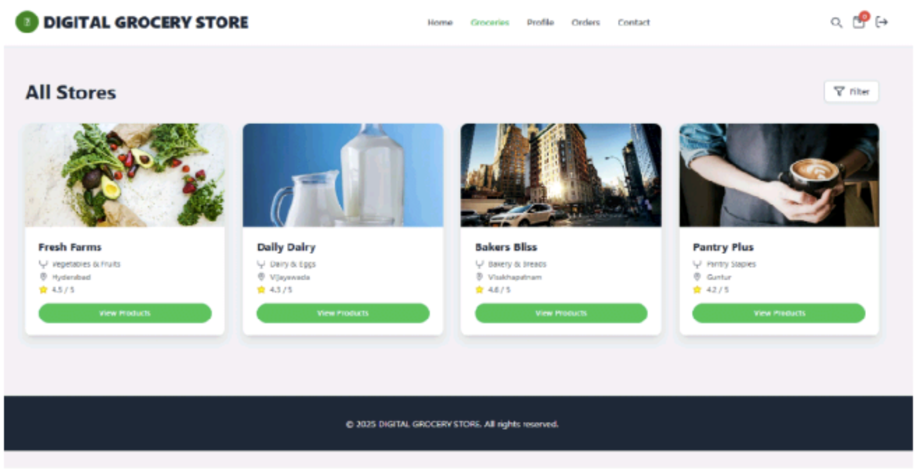
Username: hemanth

Phone Number: 1234567891

Password: •••••••

Confirm Password: 1234567

- **Home Page & Grocery Stores**



All Stores

Fresh Farms
Vegetables & Fruits
Hyderabad
4.5 / 5

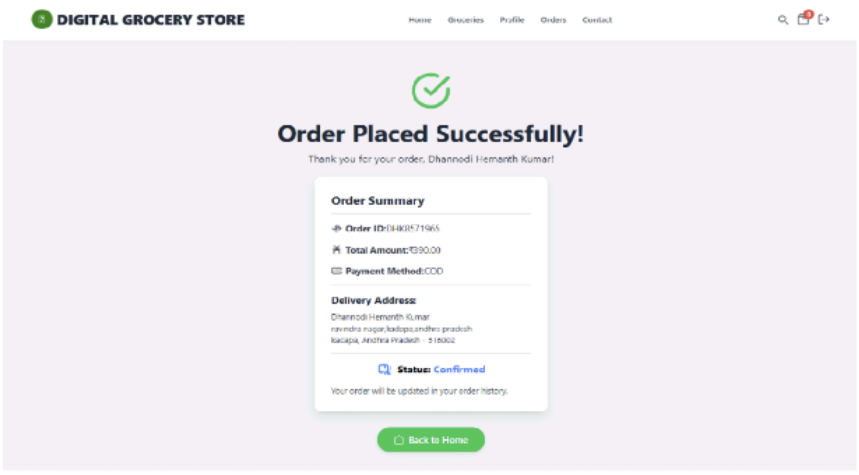
Daily Dairy
Dairy & Eggs
Vijayawada
4.3 / 5

Bakers Bliss
Bakery & Breads
Visakhapatnam
4.6 / 5

Pantry Plus
Pantry Staples
Guntur
4.2 / 5

© 2025 DIGITAL GROCERY STORE. All rights reserved.

- Order Details & Order Status



Order Placed Successfully!

Thank you for your order, Dhannodi Hemanth Kumar!

Order Summary

Order ID:DHK8571965

Total Amount:₹390.00

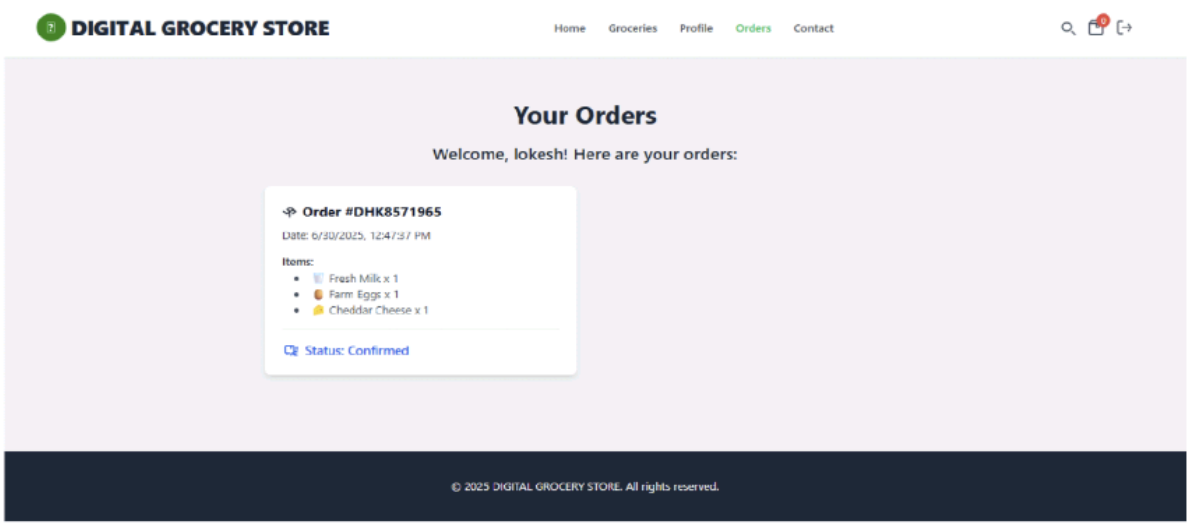
Payment Method:COD

Delivery Address:
Dhannodi Hemanth Kumar
ravindra nagar,kadapa,andhra pradesh
kadapa, Andhra Pradesh - 516002

Status: Confirmed

Your order will be updated in your order history.

- **Order History**



Your Orders

Welcome, lokesh! Here are your orders:

Order #DHK8571965

Date: 6/30/2025, 12:47:37 PM

Items:

- Fresh Milk x 1
- Farm Eggs x 1
- Cheddar Cheese x 1

Status: Confirmed

© 2025 DIGITAL GROCERY STORE. All rights reserved.

## 8. Advantages & Disadvantages

**Advantages:**

- **Convenience:** Customers can shop 24/7 from anywhere.
- **Scalable:** The application is built to handle a growing number of products and users.
- **Simple UI:** The user interface is clean and easy to navigate.
- **Centralized Management:** Easy for the admin to manage all products from one dashboard.

**Disadvantages:**

- **No Payment Gateway:** Does not include a real-world payment processing system.
- **Basic UI Design:** The UI is functional but could be enhanced with more advanced design elements.
- **No Physical Inspection:** Customers cannot physically see or touch products before buying.

## 9. Conclusion

The Digital Grocery Store project successfully provides an efficient and effective online shopping system. It connects customers with products in a seamless digital environment and provides a solid foundation for a real-world e-commerce business.

## 10. Future Scope

- **Payment Gateway Integration:** Integrate Stripe or PayPal for real online transactions.
- **JWT Authentication:** Enhance security with JSON Web Token (JWT) for users and admins.
- **Cloud Deployment:** Deploy the application on a cloud service like AWS, Heroku, or Vercel.
- **Mobile App:** Develop a native mobile application for Android and iOS.
- **SMS/Email Notifications:** Implement automatic notifications for order confirmation and shipping updates.