

Full Stack Development with MERN

ShopSmart: Your Digital Grocery Store Experience

1. Introduction

- **Project Title:** ShopSmart: Your Digital Grocery Store Experience

- **Team Members:**

- o U. Kavitha
- o L. Padma Priya
- o J. Sony Bai
- o P. Sowmya
- o S. Afroz

2. Project Overview

- **Purpose:**

The purpose of ShopSmart is to create a user-friendly online grocery shopping platform that allows customers to browse products, add items to cart, and complete purchases securely. The system also enables admins to manage products, users, and orders efficiently.

- **Key Features:**

- o User roles: Customer, Admin
- o Browse products by category
- o Search and filter products
- o Add to cart and checkout
- o Order tracking
- o Admin dashboard for product and order management

3. Architecture

- **Frontend:**

Built using Angular / React (based on your project structure).

- o HTML
- o CSS
- o Bootstrap
- o TypeScript / JavaScript
- o Axios / HTTP Client

- **Backend:**

Developed using Node.js and Express.js.

- o RESTful API design
- o JWT Authentication
- o Middleware for route protection

- **Database (MongoDB):**

Main Collections:

- o Users: { _id, name, email, password, role }
- o Products: { _id, title, category, price, stock, description }
- o Orders: { _id, userID, products, totalAmount, orderStatus }

4. Setup Instructions

- **Prerequisites:**

- o Node.js
- o MongoDB
- o Angular / React
- o Git

- **Installation Steps:**

```
git clone [your-repository-link]  
cd your-project-folder
```

Frontend

```
cd frontend  
npm install
```

Backend

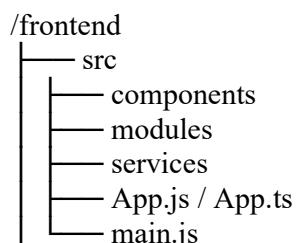
```
cd backend  
npm install
```

- **Environment Variables (.env file in backend):**

```
PORT=3000  
MONGO_URI=your_mongo_connection_string  
JWT_SECRET=your_secret_key
```

5. Folder Structure

- **Frontend:**



- **Backend:**

```
/backend
└── controllers
└── models
└── routes
└── middleware
└── config
└── server.js
└── .env
```

6. Running the Application

- Frontend:

```
cd frontend
npm start
```

- Backend:

```
cd backend
npm start
```

- Access:

<http://localhost:3000>

7. API Documentation

- POST /api/auth/register – Register User
- POST /api/auth/login – Login User
- GET /api/products – Get All Products
- POST /api/cart – Add to Cart
- POST /api/orders – Place Order
- DELETE /api/product/:id – Delete Product (Admin)

8. Authentication

- JWT-based authentication
- Token generated on login
- Middleware verifies token before accessing protected routes

9. User Interface Screens

- Landing Page
- Login / Register Page

- Product Listing Page
- Cart Page
- My Orders Page
- Admin Dashboard

10. Testing

- Manual testing for customer and admin roles
- API testing using Postman
- UI responsiveness testing

11. Screenshots / Demo

Demo Link: (Add your Drive link here)

12. Known Issues

- Payment gateway integration may require live credentials
- Basic error handling on some routes

13. Future Enhancements

- Online payment integration
- Discount coupon system
- Wishlist feature
- Mobile application version
- Real-time order tracking