

Full Stack Development with MERN

Project Documentation

1. Introduction

- Project Title: ShopSmart: Grocery Store
- Team Members:
 - Dasari Krupa Jyothi – Full Stack Developer
 - Kunche Alekhya – Full Stack Developer
 - Kasindala Pardhasaradhi – Backend Developer
 - Puvvala Hruayeswar – Frontend Developer

2. Project Overview

- Purpose:

ShopSmart is a digital grocery store web application aimed at simplifying grocery shopping by providing a convenient and efficient online platform for users to browse, filter, and purchase products.
- Features:
 - User registration and login
 - Product browsing and category filtering
 - Add to cart and checkout
 - Admin panel to manage products
 - MongoDB-based data storage
 - Responsive UI using React

3. Architecture

- Frontend:

Developed using React.js, with components for product display, cart, navigation, and login/signup. It communicates with the backend using RESTful APIs via fetch or axios.
- Backend:

Built with Node.js and Express.js. It handles user authentication, product management, and order processing.
- Database:

MongoDB is used for data storage. Collections include Users, Products, and Orders. Mongoose is used for schema modeling and queries.

4. Setup Instructions

- Prerequisites:
 - Node.js
 - MongoDB
 - npm (Node Package Manager)

- Installation:
git clone <your-repo-url>
cd ShopSmart

```
# Backend  
cd server  
npm install  
npm start
```

```
# Frontend  
cd ../client  
npm install  
npm start
```

5. Folder Structure

- Client (Frontend - React):
client/
 - |─ src/
 - | |─ components/
 - | |─ pages/
 - | |─ App.js
 - | |─ index.js

- Server (Backend - Node.js/Express):
server/
 - |─ routes/
 - |─ models/
 - |─ controllers/
 - |─ server.js / index.js

6. Running the Application

- Frontend:
cd client
npm start

- Backend:

cd server

npm start

7. API Documentation

Endpoints:

- GET /api/products → Get all products
- GET /api/products/:id → Get product by ID
- POST /api/users/register → Register a new user
- POST /api/users/login → User login
- POST /api/cart → Add item to cart

Example response for GET /api/products:

```
[
  {
    "_id": "abc123",
    "name": "Tomato",
    "price": 20,
    "category": "Vegetables"
  }
]
```

8. Authentication

- Method: JSON Web Tokens (JWT)
- Storage: Tokens are stored in local storage
- Flow:
 - On login, the server generates a JWT token
 - Token is sent with each protected request via Authorization header

9. User Interface

UI includes:

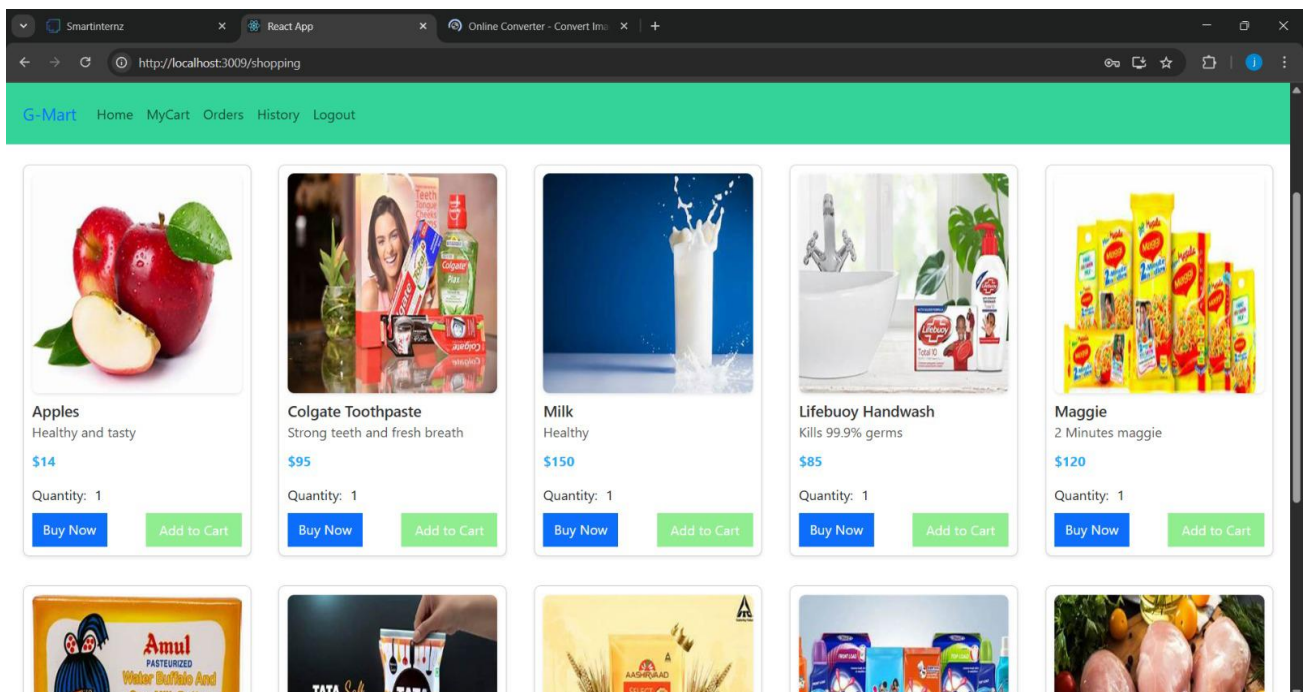
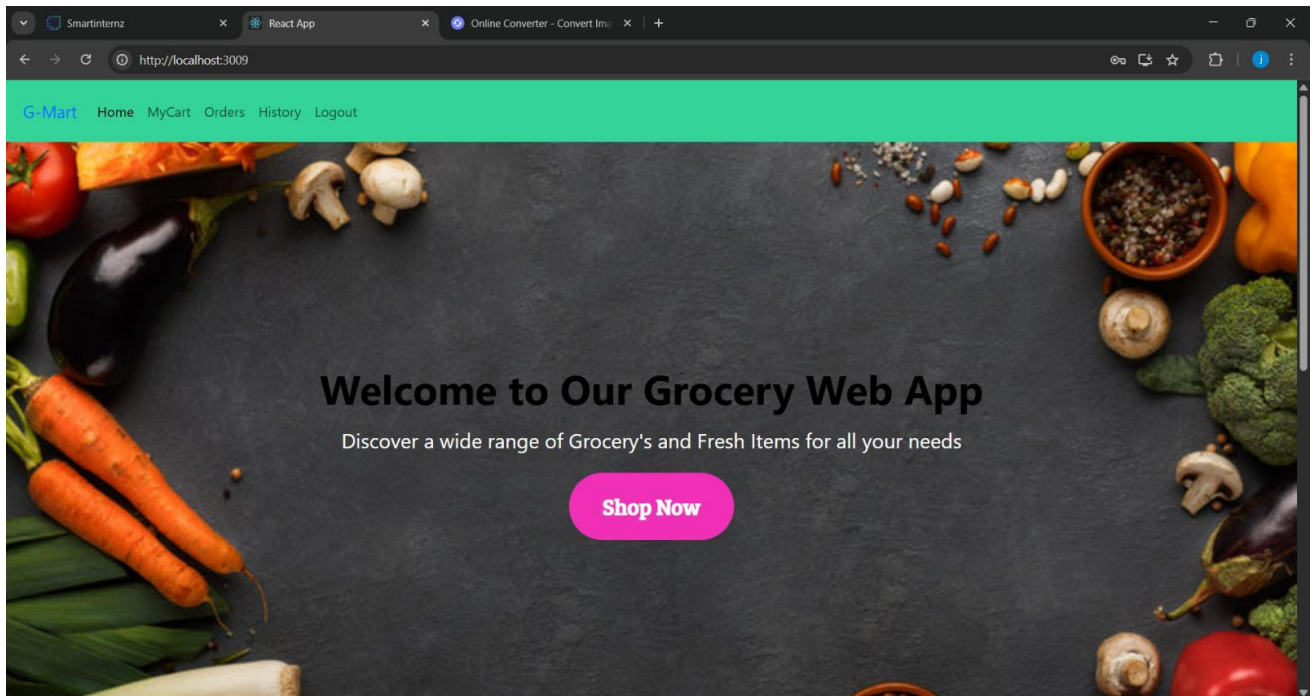
- Product listing
- Category filters (e.g., Fruits, Vegetables)
- Shopping cart
- Login and Signup forms

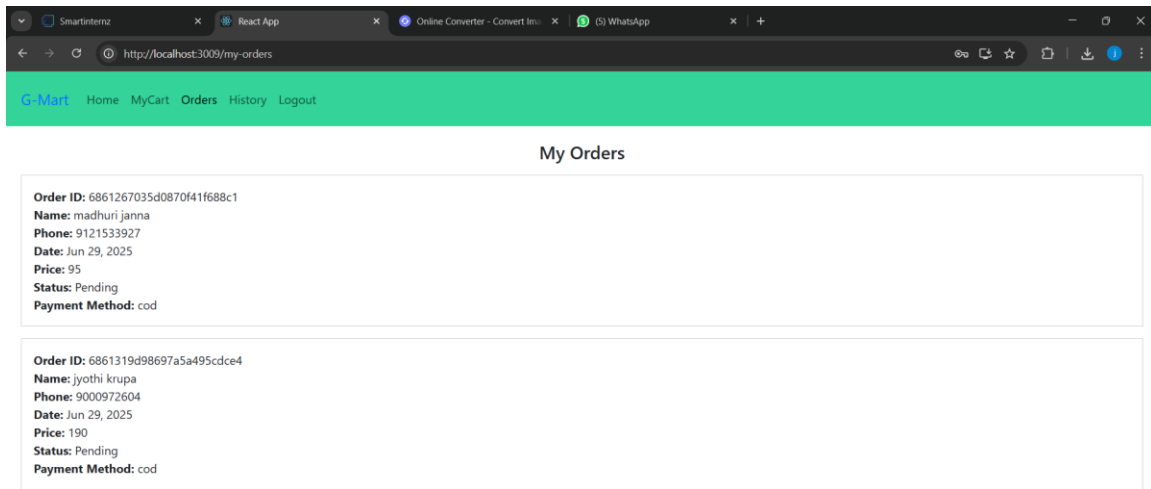
10. Testing

- Tools Used: Manual testing in browser and Postman
- Strategy:
 - Tested all routes via Postman
 - UI tested on different screen sizes
 - Cross-browser testing for Chrome and Firefox

11. Screenshots or Demo

Below are the screenshots showcasing different features of the ShopSmart Grocery Store application.





12. Known Issues

- Product images may not load properly on slow connections
- No payment gateway integration yet
- Admin role management is basic

13. Future Enhancements

- Add Razorpay/Stripe for payment gateway
- Role-based admin dashboard
- Sorting/filtering products by price, rating
- Wishlist functionality
- PWA support for mobile devices