# Full Stack Development with MERN

# Team Id : LTVIP2025TMID56240 Project Documentation

## 1. Introduction

• Project Title: ShopSmart: Your Digital Grocery Store Experience

• Team Members:

- Bhavagna – Full Stack Developer: Manages end-to-end development, authentication systems, and deployment.

- Ampa Chaitanya Naga Durga Ganesh – Frontend Developer: Responsible for React.js UI components, user experience design, and client-side functionality.

- Boda Prasanna Venkata Siva Kumar – Backend Developer: Handles Node.js/Express.js server development, API creation, and database integration.

- Bolla Geethika Naga Sai Sridevi – Database Administrator: Designs MongoDB schemas, optimizes queries, and manages data relationships.

## 2. Project Overview

• Purpose: ShopSmart aims to simplify grocery shopping with a seamless digital experience by offering product search, cart management, and secure checkout.

• Features: User authentication, product browsing, add to cart, payment integration, responsive UI.

## 3. Architecture

• Frontend: Built using React.js, components manage state with hooks, and React Router for page navigation.

• Backend: Developed with Node.js and Express.js, exposing RESTful APIs and handling business logic.

• Database: MongoDB is used to store user data, product information, and order history.

## 4. Setup Instructions

• Prerequisites: Node.js, MongoDB, Git

• Installation:

- Clone the repository: git clone [repo link]

- Navigate to the root directory and install dependencies for both client and server

- Set environment variables for MongoDB URI, JWT secret, and payment gateway keys

## 5. Folder Structure

• Client: Contains all React components, pages, styles, and context for state management.

• Server: Contains API routes, models, controllers, and middleware for backend logic.

## 6. Running the Application

• Frontend: Run `npm start` inside the `client` directory.

• Backend: Run `npm start` inside the `server` directory.

## 7. API Documentation

• GET /products – Fetch all products  
• POST /login – User login  
• POST /register – New user registration  
• POST /cart – Add product to cart  
• POST /checkout – Process payment

## 8. Authentication

• JWT-based authentication for secure login.  
• Tokens are stored in localStorage and verified on protected routes.

## 9. User Interface

• Clean and responsive UI built using React and CSS. Includes homepage, product listing, cart page, and login/signup pages.

## 10. Testing

• Manual testing for UI and Postman for API endpoint testing. Jest can be used for future test automation.

## 11. Screenshots or Demo

• Demo video link: https://youtu.be/Kdjh3AL\_OAQ

## 12. Known Issues

• Payment gateway sandbox only, no real transactions.  
• UI might break on smaller screens without responsive adjustments.

## 13. Future Enhancements

• Admin dashboard for managing inventory  
• Real-time order tracking  
• Progressive Web App (PWA) support