Project Title:  
SHOPMART – DIGITAL GROCERY STORE EXPERIENCE

# Team ID:

LTVIP2025TMID48243

# Team Name:

Team ShopSmart

# Team Members:

Raahitya Illuri  
 Register Number: SBAP0048723  
 Hall Ticket No.: 22AT1A0544  
 College: G. Pullaiah College of Engineering and Technology

# Table of Contents

|  |  |
| --- | --- |
| S. No. | Section Title |
| 1. | Introduction |
| 2. | Features |
| 3. | Technology Stack |
| 4. | Folder Structure |
| 5. | Backend Implementation |
| 6. | Frontend Implementation |
| 7. | Challenges Faced |
| 8. | Screenshots & Results |
| 9. | Advantages & Disadvantages |
| 10. | Conclusion |
| 11. | Future Scope |
| 12. | Appendix |

# 1. INTRODUCTION

## 1.1 Project Overview

Shopmart – Digital Grocery Store is a full-stack web application designed for seamless online grocery shopping. Customers can browse products, register, log in, and manage shopping carts. Admins can manage products and user data. The platform is built on a modern MERN stack and prioritizes user experience with attractive styling and intuitive navigation.

## 1.2 Purpose

The purpose of Shopmart is to:  
- Create a modern online grocery platform.  
- Enable secure user registration and authentication.  
- Allow product browsing and management.  
- Provide admin features for product and user control.  
- Showcase full-stack application development skills.

# 2. FEATURES

## User Features

- User Registration  
- User Login  
- View Product Listings  
- Add Products to Cart  
- Interactive Home Page with background images and UI

## Admin Features

- Add, Edit, Delete Products  
- View All Products  
- Manage User Accounts

# 3. TECHNOLOGY STACK

|  |  |
| --- | --- |
| Layer | Tools/Technologies |
| Frontend | React, Vite, Bootstrap, Axios |
| Backend | Node.js, Express.js, MongoDB, Mongoose |
| Others | JWT Authentication, Thunder Client, GitHub |

# 4. FOLDER STRUCTURE

shopmart/  
│  
├── shopmart-backend/  
│ ├── models/  
│ ├── routes/  
│ ├── controllers/  
│ └── server.js  
│  
└── shopmart-frontend/  
 ├── public/  
 ├── src/  
 ├── api/  
 ├── assets/  
 ├── components/  
 ├── context/  
 ├── pages/  
 ├── Home.jsx  
 ├── Products.jsx  
 ├── Login.jsx  
 ├── Register.jsx  
 ├── Cart.jsx  
 ├── App.jsx  
 ├── index.css  
 └── main.jsx

# 5. BACKEND IMPLEMENTATION

- Set up Express.js server in server.js.  
- Connected MongoDB via Mongoose.  
- Created Mongoose models for:  
 - User  
 - Product  
- Implemented routes for:  
 - User registration and login  
 - Product CRUD operations  
- Handled JWT token-based authentication.  
- Tested APIs using Thunder Client and Postman.

# 6. FRONTEND IMPLEMENTATION

- Created React app with Vite for faster build.  
- Integrated Bootstrap for layout and styling.  
- Developed:  
 - Home Page with background image.  
 - Navbar with routing.  
 - Products Page fetching products from API.  
 - Login and Registration forms.  
 - Cart page with local storage persistence.  
- Managed routing using React Router.  
- Used Axios for backend API calls.  
- Implemented conditional UI based on user roles.

# 7. CHALLENGES FACED

- CORS issues between frontend and backend.  
- Vite asset imports for images caused path errors.  
- Styling navbar and sticky footer.  
- Handling dynamic product images vs placeholders.  
- Properly linking routes for React Router.

# 8. SCREENSHOTS & RESULTS

(Add your screenshots here when preparing the final report. Examples include Home Page, Products listing page, Registration/Login screens, and Cart page view.)

# 9. ADVANTAGES & DISADVANTAGES

## Advantages

- Full-stack architecture demonstrates real-world development skills.  
- Attractive UI using Bootstrap and Vite.  
- Secure authentication with JWT.  
- Dynamic product listing connected to MongoDB.  
- Clear folder structure and modular codebase.

## Disadvantages

- No payment gateway implemented yet.  
- Static placeholder images used instead of per-product images.  
- No cloud deployment yet.  
- Role-based UI permissions are basic.

# 10. CONCLUSION

The Shopmart – Digital Grocery Store project successfully demonstrates a full-stack web application. From backend APIs to frontend user experience, the app provides core e-commerce features while offering a strong base for future enhancements like payments, cloud deployment, and admin dashboards.

# 11. FUTURE SCOPE

- Payment gateway integration (Stripe, Razorpay, etc.)  
- Product image uploads.  
- Admin dashboard for managing products via UI.  
- User profile and order history pages.  
- Deployment on cloud platforms (e.g., Vercel, AWS).  
- Role-based UI restrictions.  
- Analytics dashboard for admin insights.

# 12. APPENDIX

Source Code Repository:  
https://github.com/illuriraahitya/shopmart

# Author

Raahitya Illuri  
Register No.: SBAP0048723  
Hall Ticket No.: 22AT1A0544  
G. Pullaiah College of Engineering and Technology