Final Project Report

ShopMart: Your Digital Grocery Store Experience
Team ID: LTVIP2025TMID55453
Team Lead: Vanjarapu Suryavardhan
Team Members: V. Naga Venkata Pavan, V. Sai Viswanadh, Uppada Sagar
Date: 19-07-2025

1. Introduction

ShopMart is a full-stack, MERN-based (MongoDB, Express, React, Node.js) e-commerce application designed to deliver a seamless online grocery shopping experience. The platform empowers customers to easily browse, filter, and purchase products, while providing robust management tools for admins. ShopMart streamlines inventory, order management, and integrates best practices for security and scalability.

2. Ideation Phase

Empathy & Problem Discovery

Problem Statement

Modern e-commerce platforms often suffer from a lack of features for sellers, poor user navigation, and limited admin controls. Small vendors lack streamlined order management, while customers face complicated interfaces and checkout processes. Admins struggle with maintaining system integrity and overseeing platform health.

Empathy Map Canvas

Customer

Says: "I want to find what I need quickly," "Checkout should be easy."

Thinks: "Are these products reliable and affordable?", "Will my order be secure and on time?"

Does: Browses, compares, reads reviews, manages cart.

Feels: Frustrated by missing reviews; confident with clear information.

Needs: Simple navigation, order tracking, secure checkout.

Pain Points: Poor product info, complex checkout, payment security concerns.

Admin

Says: "Everything must run smoothly."

Thinks: "Are users getting a seamless, secure experience?"

Does: Manages users/products, reviews platform activity.

Feels: Pressured for uptime and data safety.

Needs: Comprehensive controls and analytics.

Pain Points: Outdated UI, limited insights, high responsibility.

3. Requirement Analysis

Customer Journey Map

Awareness: User discovers ShopMart via search or referral.

Browsing: Explores products with filters/sorting.

Selection: Adds products to cart.

Cart Management: Manages cart before checkout.

Checkout: Completes a secure transaction.

Order Confirmation: Receives summary and notification.

Post-Purchase: Tracks delivery/status.

Support: Leaves feedback or gets support.

Solution Requirements

Clean homepage with quick navigation.

Advanced filtering and product details.

Editable, transparent cart.

Fast, secure, multi-method checkout.

Real-time order confirmations (screen, email, SMS).

Live order tracking.

Responsive customer support.

Data Flow Overview

User logs in, system verifies credentials.

Product listings fetched/displayed.

Filters applied dynamically.

Cart linked to sessions.

Address/order data saved securely.

Secure payments via integrated gateways.

Confirmation and tracking provided post-checkout.

4. Technology Stack

Layer	Technology
Frontend	React.js, Tailwind CSS
Backend	Node.js, Express.js
Database	MongoDB Atlas
Storage	Cloudinary, Payment Gateway
Tools	Redux Toolkit, GitHub, PayPal

5. Project Planning & Scheduling

Agile, Phased Development

Requirement Gathering: User research, feature definition.

Design: Wireframes, UI mockups, architecture.

Development: Frontend, backend, APIs, database.

Integration & Testing: Module connections, bug-fixing.

Deployment: Cloud hosting.

Documentation & Review: Final report, technical docs.

6. Solution Architecture

Roles & Modules

Customer Portal: Browse, search, cart, purchase, tracking.

Admin Panel: Manage data, users, report generation.

(Future) Seller Module: List products, view sales.

Data Flow

Frontend: React fetches/pushes via REST APIs.

Backend: Express.js handles auth, logic, validation.

Database: MongoDB stores all transactional data.

Payment: Secure APIs for order processing.

7. Functional & Performance Testing

Page Speed: Optimized with React's virtual DOM.

API Response: Fast endpoints, Express middleware.

Database: Indexed queries, low-latency.

Concurrency: Simulated users to verify stability and performance.

Monitoring

Chrome DevTools for load metrics.

Backend logging and resource monitoring.

CPU, memory, and error tracking during stress tests.

8. Results

Advantages

Intuitive, responsive UI.

Secure, encrypted transactions.

Fast product access and filtering.

Real-time updates for orders and cart.

Efficient admin management.

Disadvantages

No seller panel in the current version.

Internet-dependent.

Scalability on free cloud hosting is limited.

9. Conclusion

ShopMart provides a robust online shopping experience, combining efficiency for customers and powerful admin functionality. The project exemplifies modern e-commerce using scalable web

architecture and user-centered design, setting the foundation for future innovations like seller panels and enhanced analytics.

10. Future Scope

Seller Dashboard: Direct inventory/order management for vendors.

Advanced Payments: Integration of services like Razorpay/Stripe.

Wishlist & Reviews: Save/favorite products, leave feedback.

Mobile App: Broader accessibility via native apps.

AI Recommendations: Smart product suggestions for users.