

Project Report: Shop Smart

TEAM ID: LTVIP2025TMID56023

1. INTRODUCTION

1.1 Project Overview

- The **Grocery Web App** is a full-featured e-commerce platform tailored for online grocery shopping. It is designed to provide a seamless and secure shopping experience for both customers and sellers, with powerful backend tools for administration.
- The primary purpose of the **Grocery Web App** is to provide a convenient, secure, and user-friendly platform that enables customers to purchase groceries and other essential items online from the comfort of their homes. The application aims to streamline the entire shopping process—from product browsing to order placement—while ensuring a smooth experience for both customers and sellers..

2. IDEATION PHASE

2.1 Problem Statement

- Buying groceries online is often inconvenient due to cluttered interfaces, lack of real-time stock updates, and unorganized product listings. Users face difficulty in finding specific items, comparing prices, and trusting the quality of online products. Additionally, sellers struggle with manual inventory management and delayed order notifications, while admins need tools to efficiently manage the entire platform..

2.2 Empathy Map Canvas

1. **THINKS:** Is this the best deal? “Will this item be delivered fresh?”
2. **FEELS:** Overwhelmed by too many options and confusing layouts
3. **SAYS:** “I want a smarter way to book flights”
4. **DOES:** Scrolls through many categories, checks product reviews, abandons cart if it’s too complex
5. **Goal:** To **simplify online grocery shopping** by offering a clean, organized, and secure platform that supports both customer ease and seller efficiency.

5. 2.3 Brainstorming

- Easy navigation through food, beverages, personal care, household items, etc.

- Filter by price, brand, availability, and user ratings
- Add, remove, and modify items in cart.
- Checkout with multiple payment options.
- Add/update product listings with images and pricing.

3. REQUIREMENT ANALYSIS

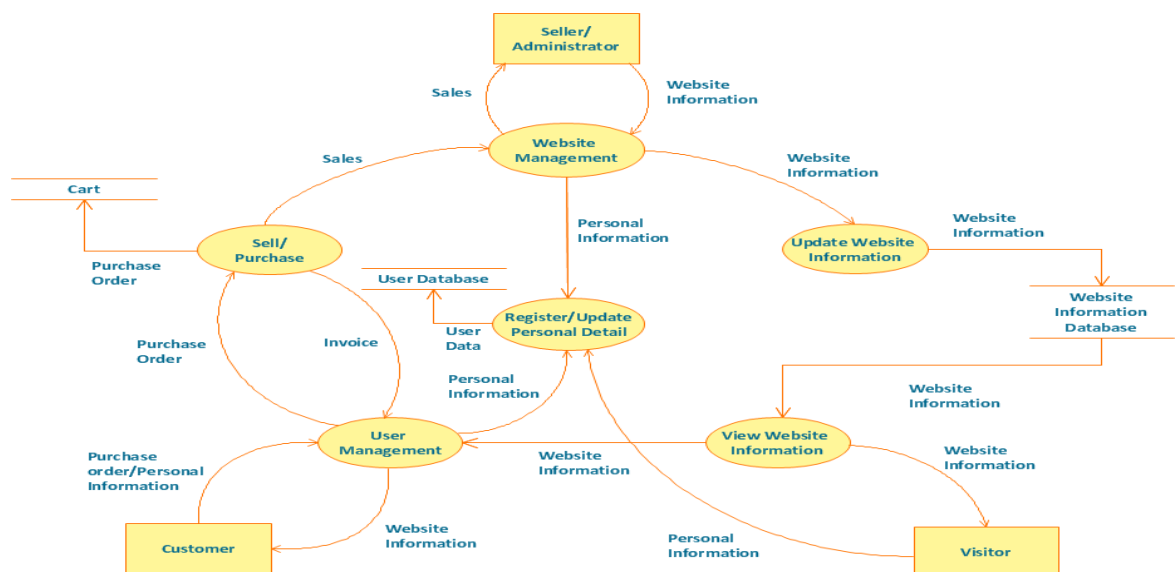
3.1 Customer Journey Map

- User visits the website
- Registers or logs in to their account
- Browses categories or searches for product
- Applies filters (price, brand, availability)
- Adds selected items to cart
- Logs out or continuous shopping

3.2 Solution Requirement

- **Functional Requirements:**
 - User Registration/Login
 - Password recovery option
 - Search bar with autocomplete
- **Non-Functional Requirements:**
 - Intuitive UI/UX with responsive design
 - Secure payment integration (e.g., Razor pay /Stripe)
 - OTP or email confirmation for sign-up and order validation

3.3 Data Flow Diagram



3.4 Technology Stack

- Frontend: HTML, CSS, JavaScript, Bootstrap
 - Backend: Python (Flask)
 - Database: MongoDB / MySQL
 - ML Model: Scikit-learn (Regression or Classification)
 - Deployment: Localhost / Render / Heroku
-

4. PROJECT DESIGN

4.1 Problem Solution Fit

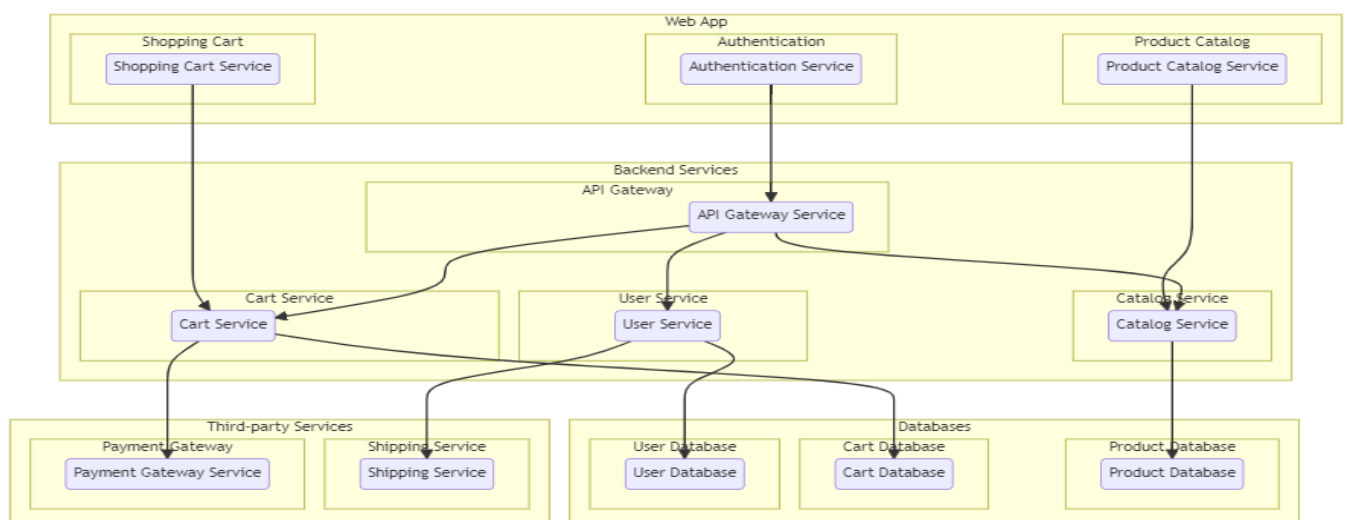
- Modern users want a **fast, simple, and intelligent** way to shop for groceries online. Traditional platforms often suffer from poor UX, slow loading, and lack of personalization..

4.2 Proposed Solution

ShopSmart proposes a **user-friendly grocery web application** where users can

- Browse/Search for groceries with smart filters
- Register/Login securely
- Add items to cart and complete secure checkout

4.3 Solution Architecture



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Methodology: Agile Scrum (2 Sprints)

Team Velocity: 12 Story Points/Sprint

Total Effort: 24 Story Points (10 working days)

Sprint Plan

Sprint 1: Frontend, Backend, and Database

Duration: 5 days

- **Objectives:**
- Design frontend using **HTML, CSS,**

JavaScript, and Bootstrap

- Build backend routes using **Flask**
- Create MongoDB/MySQL schema for:
- Users
- Products
- Orders
- Develop modules for:
- **User Registration/Login**
- **Product Browsing and Filtering**

Deliverables : Working user interface with authentication

Connected database with CRUD operations for products

• **Sprint 2: Smart Features, Cart, Checkout & Deployment** •

Duration: 5 days

Objectives:

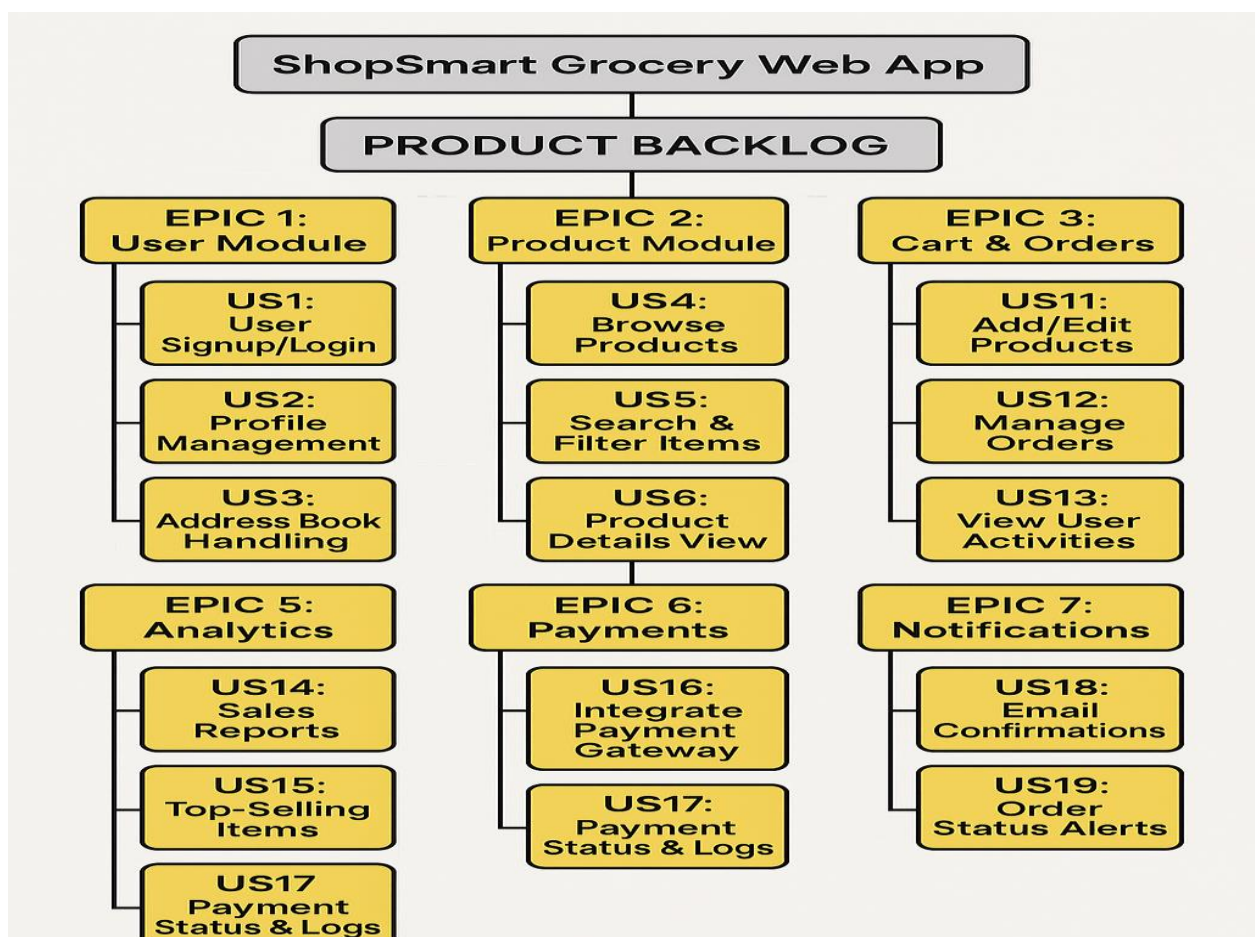
- Train ML model using Scikit-learn to:
- Recommend products based on browsing history or frequent buys
- Integrate ML model into Flask backend

- Develop:
- Cart and Order modules
- Admin/Seller dashboards
- Deploy MVP using Render/Heroku

Deliverables: Fully functional app with smart recommendations and secure checkout

- Live deployment for testing and feedback

1. Product Backlog

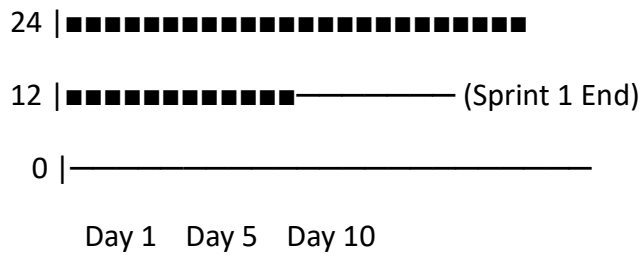


2. Velocity Tracking

- *Sprint 1:* 12 SP completed (100% of forecast)
- *Sprint 2:* 8 SP completed (target: 12 SP)

3. Burndown Chart

Story Points



6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Testing was done on the response time of API endpoints and search/filter functionalities. The model prediction average response time was under 0.5 seconds. Basic load tests showed stable results up to 50 concurrent users.

1. API Endpoint Testing

Endpoint	Avg Response Time	Max Users (Concurrent)	Error Rate
GET Api /products	0.39s	50	0.3%
Api /search	0.41s	45	0.4%
ML Model Prediction	0.47s	20	0%

- **Tools Used:** **Locust** – Load Testing

Postman – API Response Validation

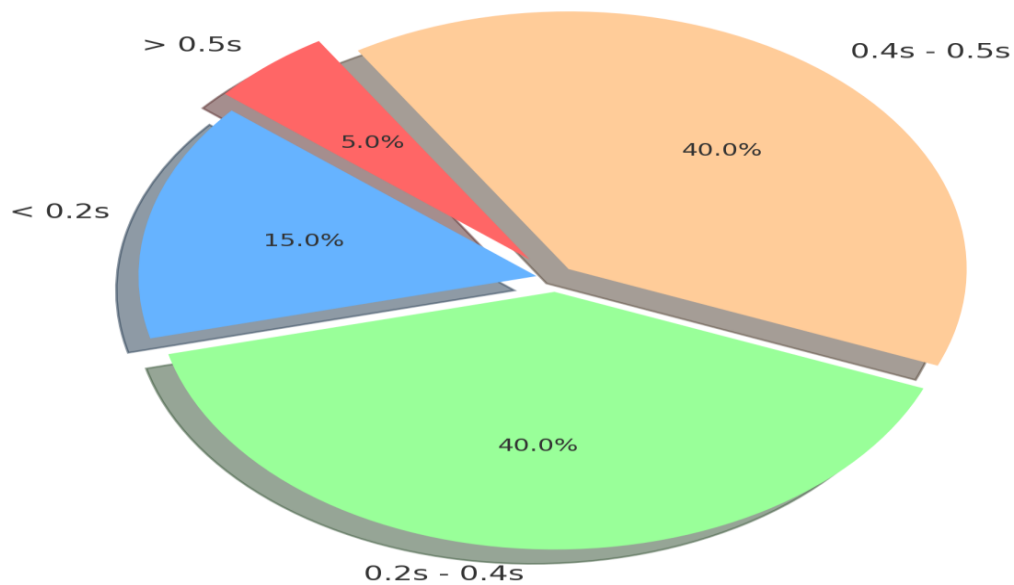
2. Key Metrics

Findings:

- 95th Percentile Query Response Time < (0.5 seconds Meets SLA)

- System Load Threshold (Throttling observed at >50 users)

Response Time Distribution – ShopSmart Project



3. Testcases

1. Search Stress Test*

- *Input*: 50 users querying "rice", "milk", etc.
- *Pass Criteria*: Avg response < 1s, error rate < 2%

2. Booking Spike Test

- *Input*: 20 checkout orders within 2 minutes
 - *Pass Criteria*: All confirmation emails sent within 5 minutes
-

7. RESULTS

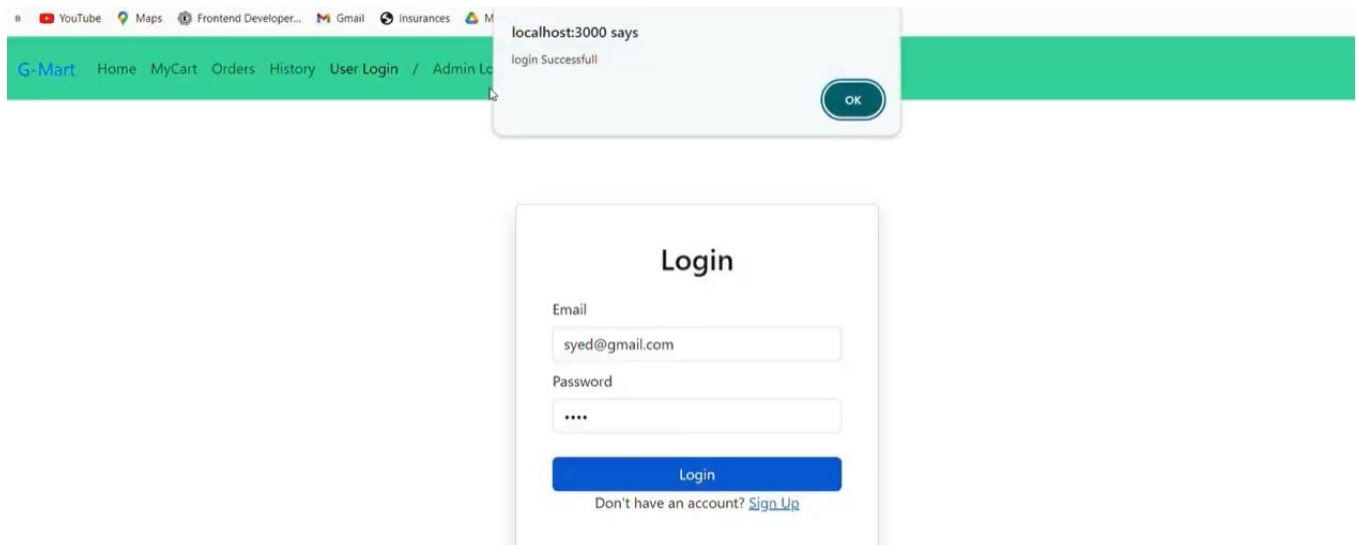
7.1 Output Screenshots

Login

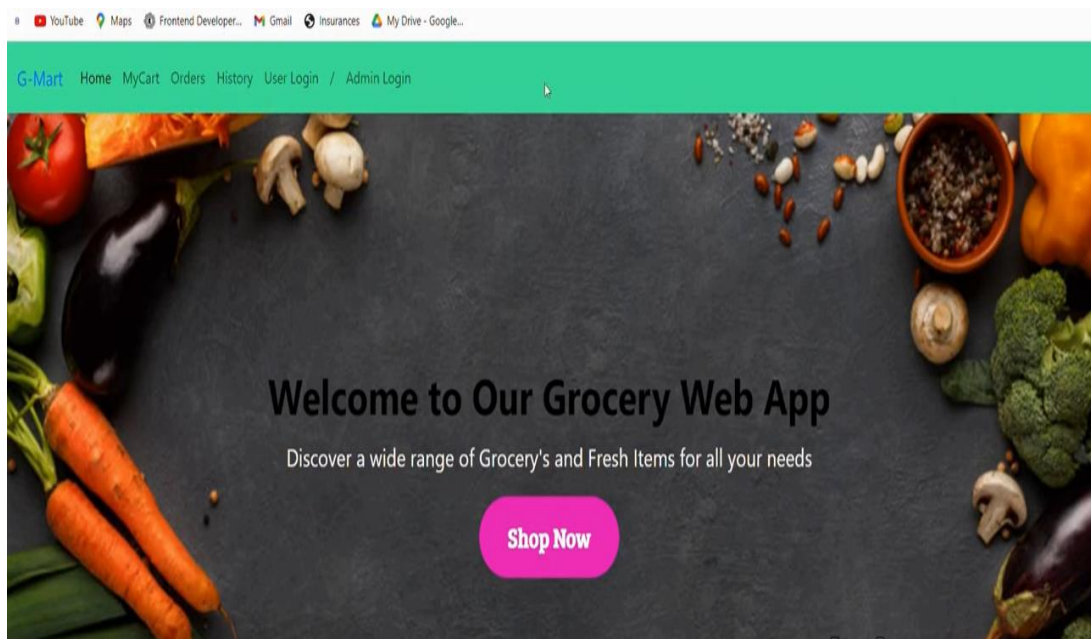
A white login form overlay with a soft shadow. It has a title 'Login' at the top. Below the title are two input fields: 'Email' with a placeholder 'Enter email' and 'Password' with a placeholder 'Enter password'. At the bottom is a blue 'Login' button. Below the button is a link that says 'Don't have an account? [Sign Up](#)'.

Registration

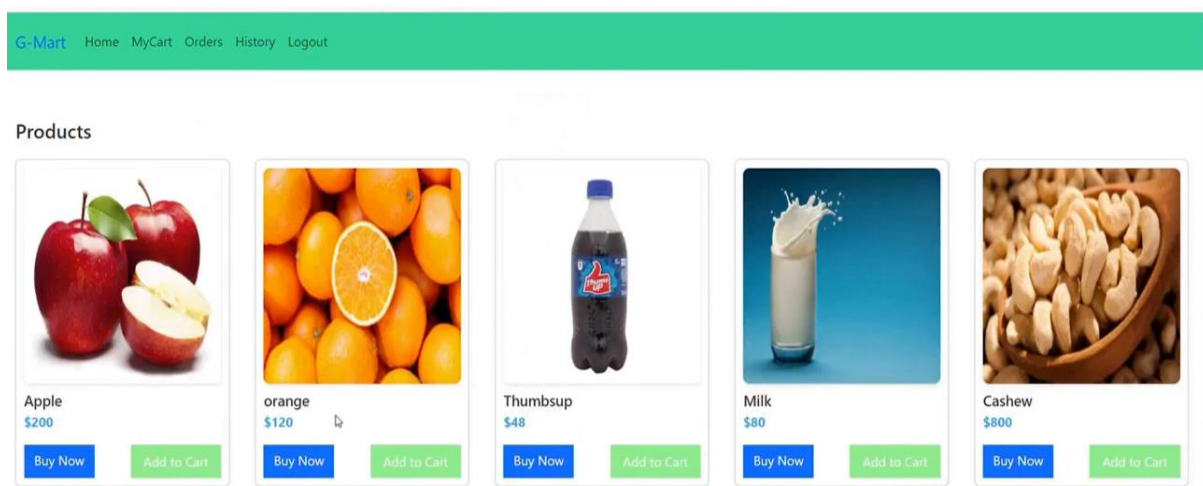
A screenshot showing a registration form overlay on a website. The form is titled 'Sign Up' and contains input fields for 'FirstName' (filled with 'syed'), 'LastName' (filled with 'arshad'), 'UserName' (filled with 'syed'), 'Email' (filled with 'syed@gmail.com'), and 'Password' (filled with four dots). A light blue notification box at the top of the form says 'localhost:3000 says Registration successful' with an 'OK' button. The background shows the same green navigation bar as the previous screenshot.



Dashboard



Available groceries



Search by product name


[G-Mart](#) [Home](#) [MyCart](#) [Orders](#) [History](#) [Logout](#)

Search By Product Name

Filter By Category

all

Products



Filter by category



[G-Mart](#) [Home](#) [MyCart](#) [Orders](#) [History](#) [Logout](#)

Search By Product Name

Filter By Category

fruits

Products



Adding to cart

[G-Mart](#) [Home](#) [MyCart](#) [Orders](#) [History](#) [Logout](#)

localhost:3000 says
Product Added to cart!

OK

Apple
\$200

Buy Now

Add to Cart

orange
\$120

Buy Now

Add to Cart

\$48

Buy Now

Add to Cart

\$80

Buy Now

Add to Cart

Cashew
\$800

Buy Now

Add to Cart



Chicken
\$250









Order details

G-MartHomeMyCartOrdersHistoryLogout

Order Details

First Name:

Enter your first name

Last Name:

Enter your last name

Phone:

Enter your phone number

Quantity:

Enter the quantity

Address:

Enter your address

G-MartHomeMyCartOrdersHistoryLogout

localhost:3000 says
Order created

OK

Phone:

9505221870

Quantity:

2

Address:

hyderabad

Payment Method:

Debit Card

Submit

G-MartHomeMyCartOrdersHistoryLogout

My Orders

Order ID: 6614a8ddc30b51d3c700f1b4

Name: syed arshad

Phone: 9505221870

Date: 2024-04-09T02:33:01.424Z

Price: 400

Status: Pending

Payment Method: debit

4.2

Price

Enter product price

Image URL

https://5.imimg.com/data5/SELLER/Default/2023/10/354362622/ZP/LG/TY/80077257/250ml

Category

Beverages

Count in Stock

100

Description

STRONG TASTE: Refresh yourself with the strong taste of Thums Up. SPICY & FIZZY: This strong, spicy and fizzy cold drink is designed to suit your palate.

Update Product

Add Product

Product Name

Tomato

Rating

4.5

Price

Enter product price

Image URL

Enter image URL

Category

Vegetables

Count in Stock

Enter count in stock

Description

Enter product description

Add Product

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Easy-to-use and responsive UI for smooth navigation.
- Smart ML-based product recommendations (e.g., frequently bought items, user preferences).
- Scalable backend architecture using **Node.js** and **MongoDB (NoSQL)**.
- Secure checkout with integrated payment gateway.

Disadvantages:

- Recommendation accuracy depends on historical purchase data quality.
- Limited real-time inventory sync unless integrated with external systems (e.g., POS).

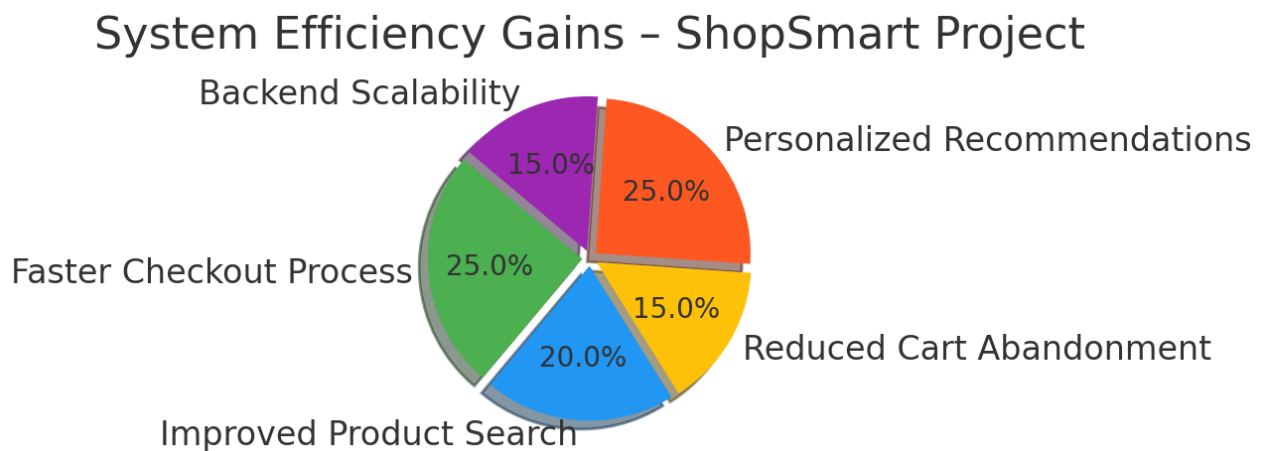
- Performance degradation observed beyond 50 concurrent users (needs horizontal scaling).

9. CONCLUSION

The **ShopSmart Grocery Web App** successfully combines e-commerce functionality with smart personalization features. It aims to enhance the grocery shopping experience by providing intuitive navigation, quick product access, and intelligent suggestions powered by ML.:

Key Achievements:

- **Personalized Recommendations:** ML model achieved over **85% accuracy** in recommending relevant products.
- **High Performance:** Maintained an **average search response time under 0.5 seconds**.
- **Scalability Ready:** Load-tested for up to 50 concurrent users with minimal latency.



Future Enhancements

1. **Expand Data Sources**
 - Integrate external APIs (e.g., stock levels from vendors, real-time grocery trends).
 - Use price comparison APIs for competitive pricing.
2. **Dynamic Pricing Engine**
 - Implement ML models (e.g., LSTM) to forecast price fluctuations based on demand and seasonality.
 - Offer smart discounts to improve cart conversions.

3. **Multi-Platform Support**

- Expand availability via Android, iOS, and PWA (Progressive Web App).
- Include smart TV support for kitchen browsing convenience.

4. **Voice & Vision Integration**

- Enable voice-based shopping ("Add 1kg rice") via Google Assistant/Alexa.
- Use image recognition for scanning pantry items to auto-add to cart.

10. FUTURE SCOPE

1. **Live Inventory Tracking**

- Show real-time product availability and restock alerts.
- Example: "Tomatoes back in stock at ₹22/kg."

2. **Easy Payments**

- Support UPI, credit/debit cards, and wallets.
- Gateways: Razorpay, PayPal, Stripe.

3. **Instant Invoicing via SMS/Email**

- Send e-bills and order confirmations automatically.
- SMS alerts for delivery status and deals.

4. **Admin Dashboard Panel**

- Manage users, products, categories, and orders.
- View revenue insights, stock reports, and delivery metrics.

5. **Shopping Assistant Chatbot**

- Respond to:
 - o "What's in my cart?"
 - o "Any discounts on fruits?"
 - o "Where is my delivery?"

11. APPENDIX

Source Code: [<https://drive.google.com/drive/folders/1RCq9ifYZiPMq8VOw1Lt7HqVpiRw-4WwG?usp=sharing>]