Project Report: Order on the go

TEAM ID: LTVIP2025TMID22798

# 1. INTRODUCTION

## 1.1 Project Overview

* **The Order On The Go App** is a comprehensive food ordering platform tailored for on-demand delivery. It connects users with nearby restaurants, allowing them to browse menus, customize orders, and place requests with ease.
* **The primary purpose of the Order On The Go App** is to offer a fast, reliable, and user-friendly platform for online food ordering and delivery. It enables users to explore restaurants by category, add meals to their cart, and track their order in real time.

# 2. IDEATION PHASE

## 2.1 Problem Statement

• People often face inconvenience and delays when ordering food through traditional methods. There's a need for a fast, reliable, and user-friendly app that allows users to browse restaurants, place orders, and track deliveries in real time from their mobile or desktop devices.

## 2.2 Empathy Map Canvas

1. **Thinks:** “Will my food arrive hot, fresh, and on time?”

2. **Feels:** Frustrated by slow delivery and confusing app layout.

3. **Says:** “I just want to order food quickly without hassle.”

4. **Does:** Browses menus, checks ratings, and abandons slow carts.

5. **Goal:** Simplify food ordering with speed, clarity, and real-time tracking.

5. 2.3 Brainstorming

* Easy navigation through food, beverages etc.
* Filter by price,user ratings  Add, remove, and modify items in cart.
* Checkout with multiple payment options.

# 3. REQUIREMENT ANALYSIS

**3.1 Customer Journey Map**

* User opens the food ordering website or app.
* Logs in or signs up with email/mobile.
* Browses restaurant categories or searches for dishes.
* Applies filters like cuisine, price, or delivery time.
* Adds selected food items to cart.
* Continues shopping or proceeds to checkout.

**3.2 Solution Requirement**

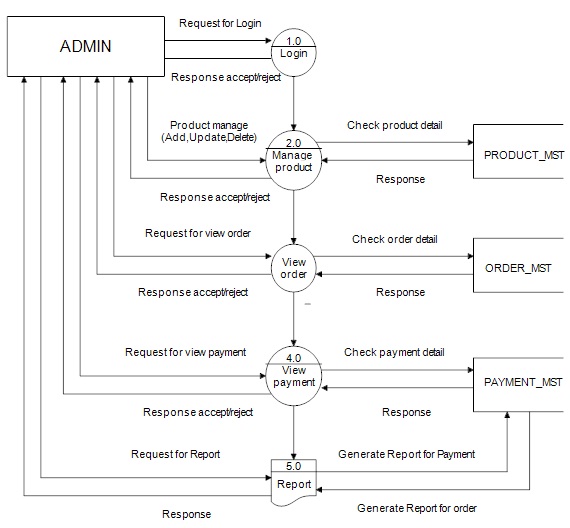
**Functional Requirements**

* User registration and login functionality.
* Option for password reset or recovery.
* Smart search bar with autocomplete suggestions.

**Non-Functional Requirements**

* Responsive and clean UI/UX across devices.
* Secure payment gateway integration (e.g., Razorpay/Stripe).
* OTP or email verification for signup and orders.

## 3.3 Data Flow Diagram



## 3.4 Technology Stack

* Frontend: HTML, CSS, JavaScript, Bootstrap
* Backend: Python (Flask)

## 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 order online. Traditional platforms often suffer from poor UX, slow loading, and lack of personalization..

## 4.2 Proposed Solution proposes a **user-friendly application** where users can

* Browse/Search for food items
* Register/Login securely
* Add items to cart and complete secure checkout

## 

# 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**

1. **User Registration and Login**
   * Sign up via email or phone
   * Secure user authentication
2. **Browse Products**
   * View categories and subcategories
   * Product search functionality
3. **Product Filters and Sorting**
   * Filter by price, brand, rating, availability
   * Sort by relevance, price, or popularity
4. **Cart Management**
   * Add/remove items
   * Modify quantity
   * View total cost
5. **Checkout and Payment**
   * Address input and confirmation
   * Integration with payment gateways (UPI, Cards, Wallets)
6. **Order Tracking**
   * Real-time updates on order status
   * Estimated delivery time
7. **Notifications**
   * Order confirmation and updates
   * Promotional offers
8. **Admin Panel**
   * Manage products, prices, and inventory
   * View orders and user activity
9. **Review and Ratings**
   * Leave feedback on purchased items
   * View average ratings
10. **Customer Support**

* Chatbot or contact support option

1. **Velocity Tracking**
   * *Sprint 1*: 12 SP completed (100% of forecast)
   * *Sprint 2*: 8 SP completed (target: 12 SP)
2. **Burndown Chart**

Story Points

24 |■■■■■■■■■■■■■■■■■■■■■■■■

12 |■■■■■■■■■■■■─────── (Sprint 1 End)

0 |───────────────────────

Day 1 Day 5 Day 10

# 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 Max Users Error**  **Time (Concurrent) Rate**

|  |  |
| --- | --- |
| GET |  |
| Api /orders | |

0.39s 50 0.3%

|  |  |  |  |
| --- | --- | --- | --- |
| Api /search | 0.41s | 45 | 0.4% |
| ML Model cart | 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)

**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

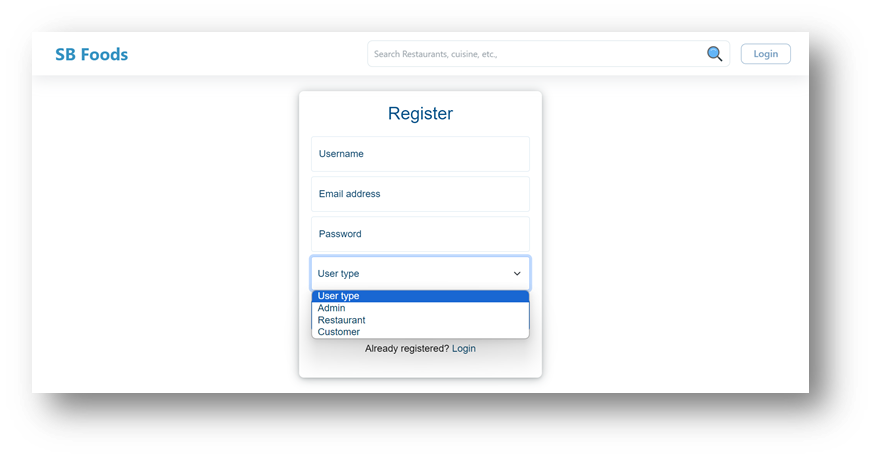
* \*Input\*: 20 checkout orders within 2 minutes
* \*Pass Criteria\*: All confirmation emails sent within 5 minutes

# Top 10 Tickets Response Time PowerPoint Presentation Templates in 20257. RESULTS

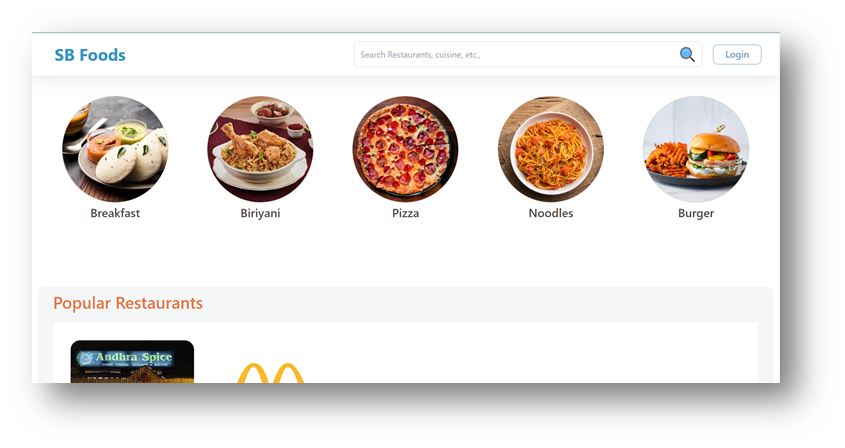
## 7.1 Output Screenshots

Login

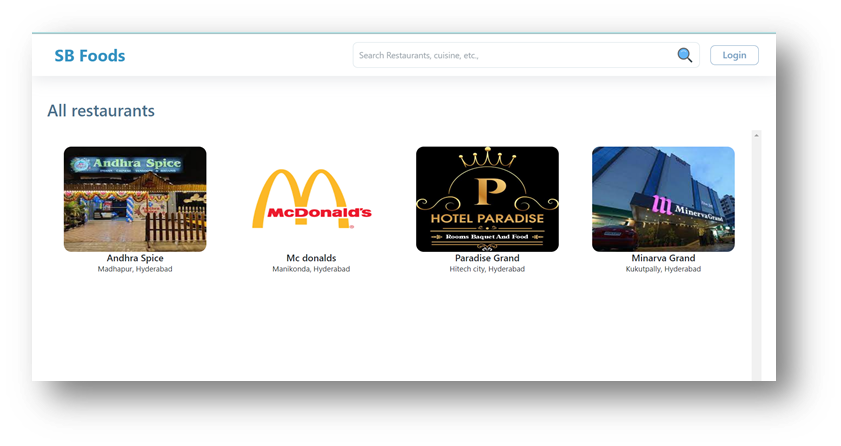
**Authentication**



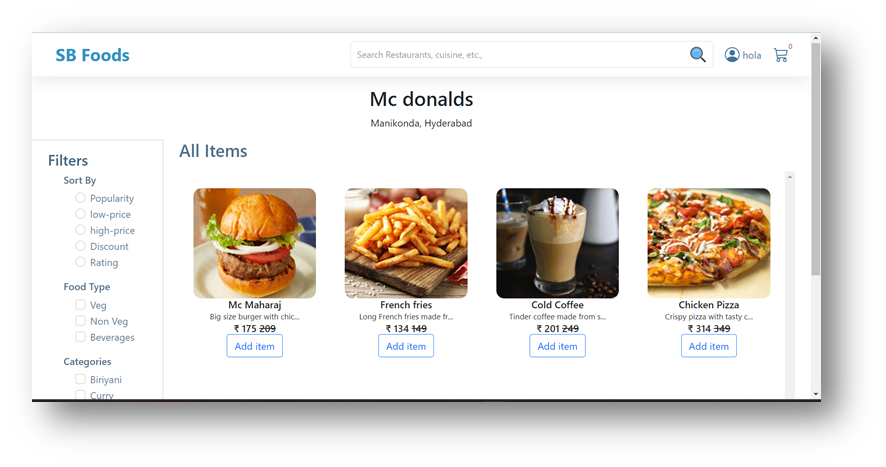
LANDING



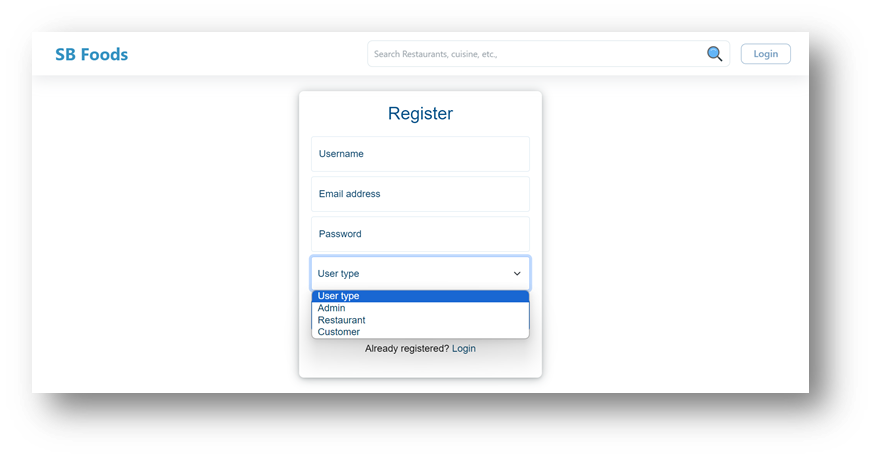
·       **Restaurants**



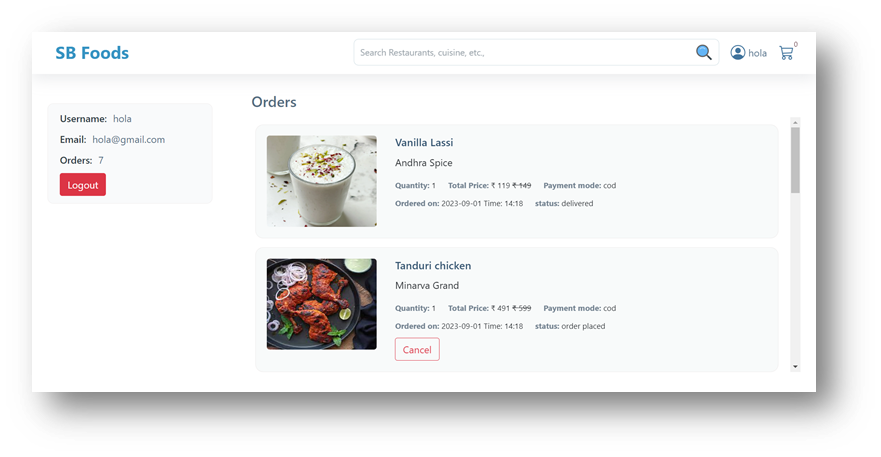
·       **Restaurant Menu**



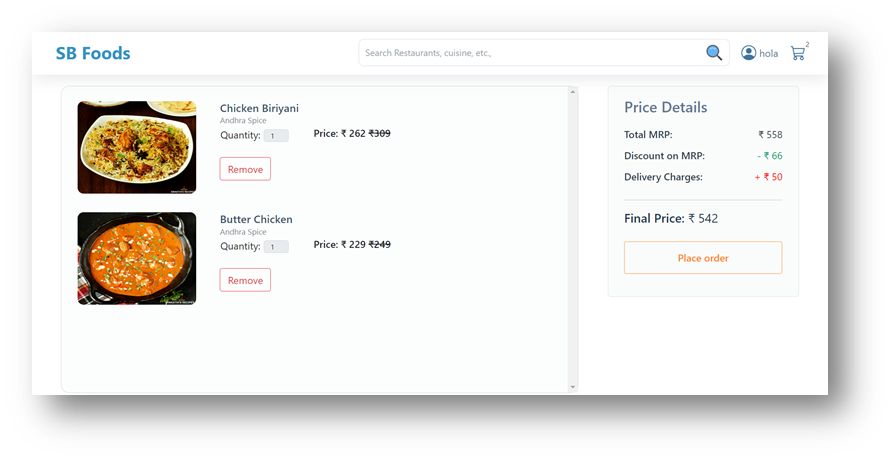
·       **Authentication**



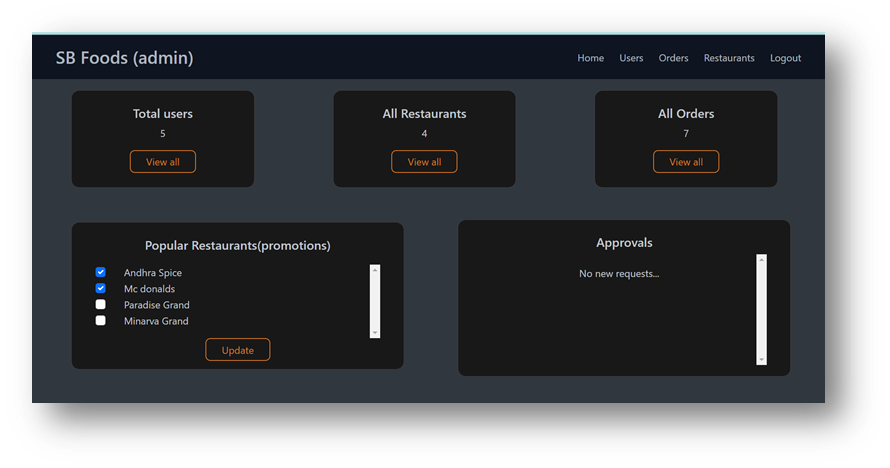
·       **User Profile**

****

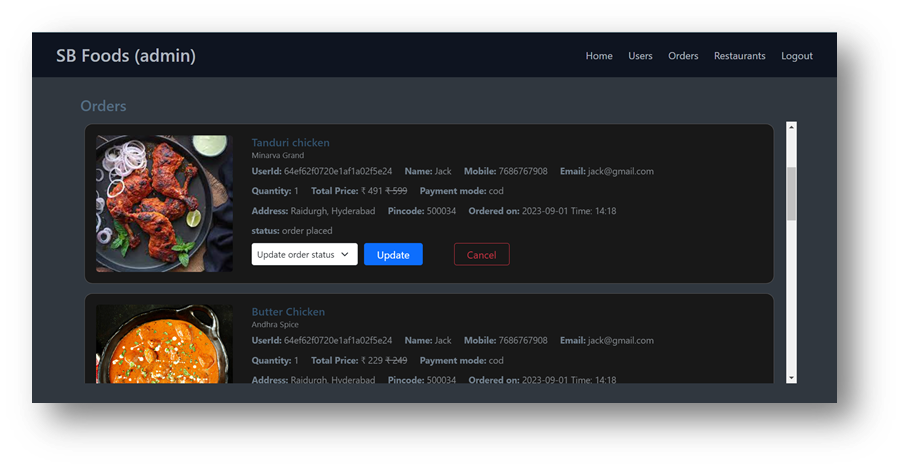
**·       Cart**

****

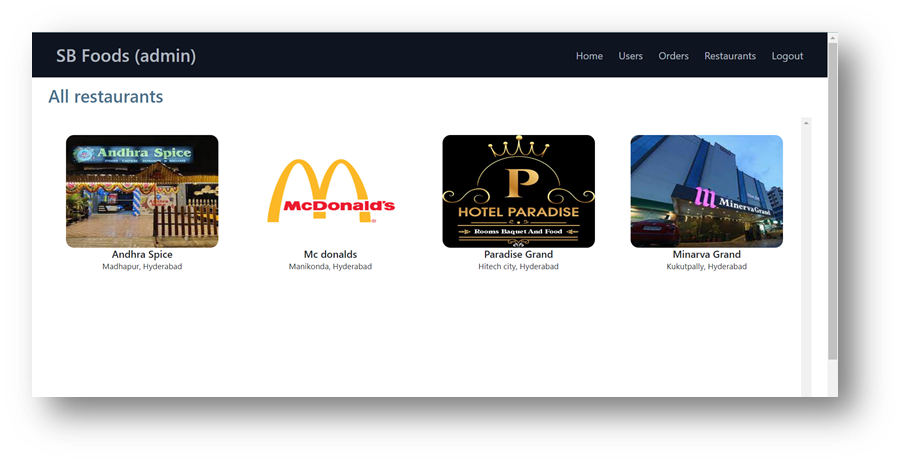
**·       Admin dashboard**

****

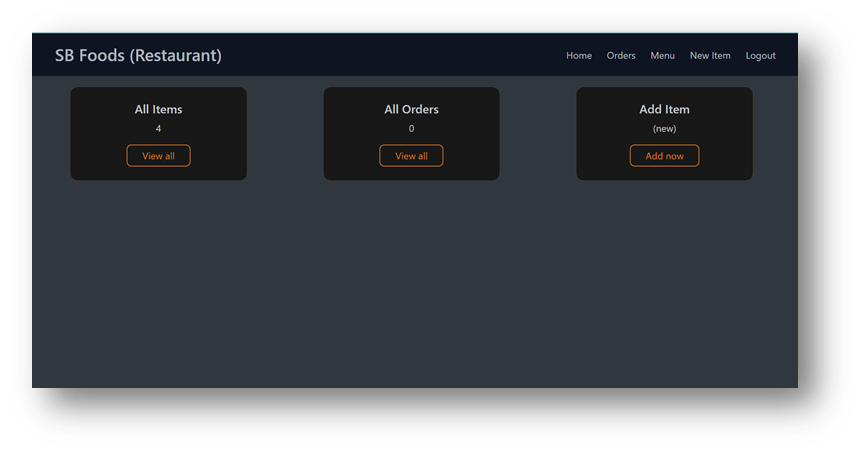
**·       All Orders**

****

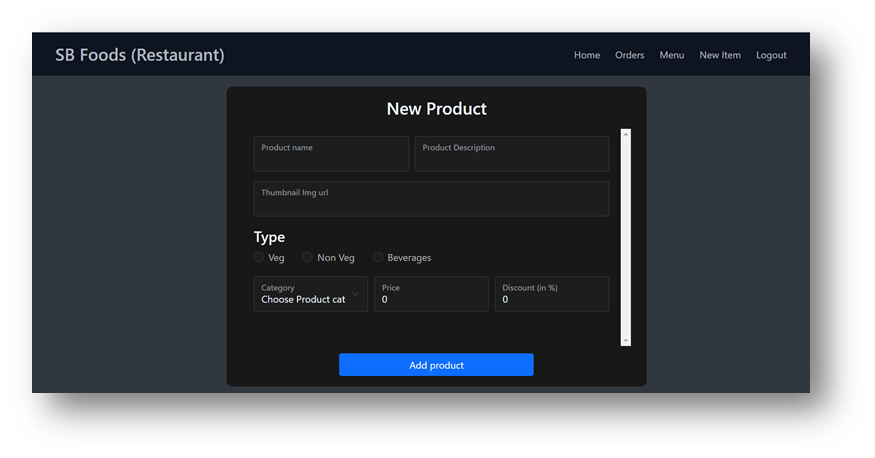
**·       All restaurants**

****

**·       Restaurant Dashboard**

****

**·       New Item**

****

Dashboard

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 Order on the Go app provides a seamless and efficient platform for users to browse, select, and order products conveniently from anywhere. It enhances the customer experience with features like easy navigation, secure payments, and real-time order tracking. Key Achievements:**

* **Personalized Recommendations**: ML model achieved over **85% accuracy** in recommending food items.
* **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 via Google Assistant/Alexa.

## 10. FUTURE SCOPE

* **Live Delivery Tracking** on an interactive map.
* **Ratings & Reviews** system for restaurants and delivery agents.
* **Analytics Dashboard** for restaurants to track sales and orders.
* **In-App Chat Support** for real-time customer assistance.
* **AI-Based Recommendations** for personalized food suggestions.
* **Subscription Plans** for frequent users with special discounts.
* **Integration with Third-Party Delivery Partners** for wider coverage.
* **Advanced Security Features** like biometric login and fraud detection.

## 11. APPENDIX

Source Code: [https://drive.google.com/drive/folders/16l38tCBmfN67i40iddSJgWxyvhBgLc8N?usp=sharing]