SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category - I University by UGC
Accredited "A++" by NAAC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Fast and Reliable

ONLINE GROCERY SYSTEM

PROJECT STUDENT:

Dasari Nithish
(43110196)

Online Grocery System

- Full Stack Implementation using ReactiveJS (Front-end), Spring Boot WebFlux (Back-end), and MongoDB (Database
- The Online Grocery System aims to revolutionize grocery shopping by providing a fast, reliable, and modern web-based platform that connects retailers and customers in real-time.

Agenda

- 1. Problem Statement
- 2. Abstract
- 3. Objectives
- 4. System Architecture
- 5. Process / Methods Adopted
- 6. Module Descriptions & Workflow
- 7. System Requirements
- 8. Results & Screenshots
- 9. Security & Performance
- 10. Conclusion & Future Scope

Problem Statement

Traditional grocery shopping involves long queues, limited product visibility, and dependency on physical stores. Customers struggle with stock issues and price inconsistencies. Retailers face challenges in inventory tracking, data management, and real-time updates.To overcome these limitations, this project proposes an Online Grocery System that enables customers to browse, purchase, and track groceries efficiently through a digital interface, powered by a reactive and scalable technology stack.

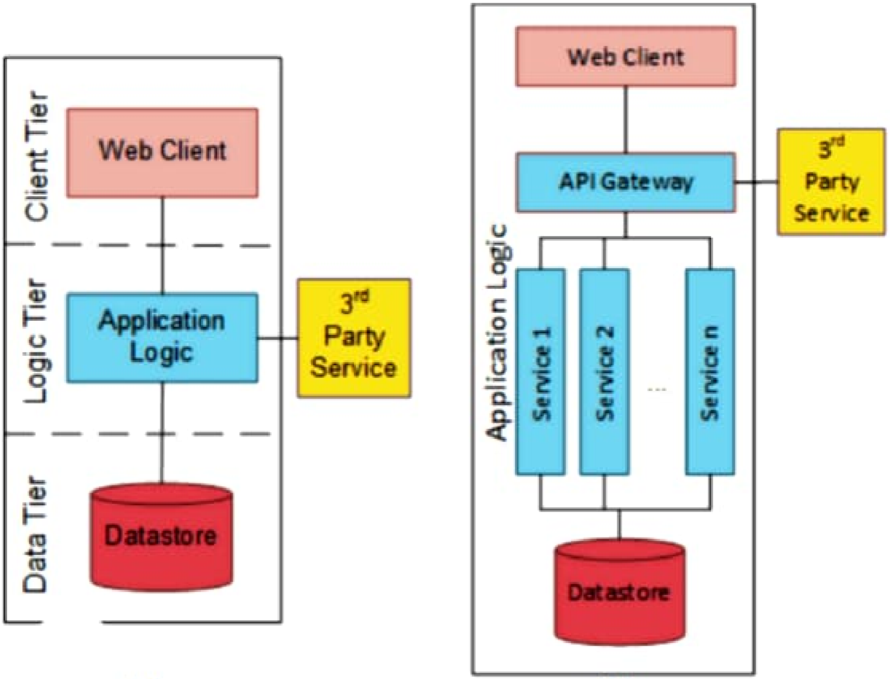
Abstract

- The Online Grocery System is a full-stack web application developed using ReactiveJS for the client-side, Spring Boot WebFlux for the back-end, and MongoDB for the database. The system provides real-time, non-blocking communication for better scalability and responsiveness. Users can browse products, add them to cart, make secure payments, and track orders in real-time. Admin users can manage inventory, track orders, and view analytics. The combination of a reactive front-end and non-blocking back-end ensures efficient performance even under heavy user load.

Objectives

- Design a dynamic and user-friendly front-end using ReactiveJS for real-time updates and smooth navigation.
- Develop a non-blocking API using Spring Boot WebFlux for high concurrency and efficient performance.
- Integrate MongoDB to handle flexible, schema-less product and user data.
- Implement key modules: Authentication, Product Catalog, Cart, Checkout, Order Management, and Notifications.
- Ensure scalability, fault tolerance, and secure transactions using JWT and HTTPS.
- Support multi-device accessibility with responsive UI.

Architecture of the System



(a)

(b)

Process / Methods Adopted

1. Requirement Analysis: Defined functional and non-functional requirements for users and admins.
2. System Design: Created data models, UI wireframes, and API blueprints.
3. Implementation: Developed front-end using ReactiveJS, back-end using Spring Boot WebFlux, and database with MongoDB.
4. Integration & Testing: Integrated modules and performed unit, integration, and load testing.
5. Deployment: Deployed the application using Docker and configured for cloud scalability.
6. Maintenance: Set up monitoring, logging, and periodic backups for data reliability.

Modules & Workflow

1. User Module: Registration, Login (JWT Authentication), Profile Management.
2. Product Module: Search, Filter, and View Product Details.
3. Cart Module: Add, Update, and Remove Items with Real-time Price Calculation.
4. Checkout & Payment: Payment Gateway Integration, Address Selection, and Order Confirmation.
5. Order Management: Live order tracking with WebSocket updates.
6. Admin Module: Product CRUD, Inventory Control, and Offers Management.
7. Notification Module: Automated emails/SMS for order updates.

Workflow: User browses → Adds to cart → Confirms order → Makes payment → Tracks delivery.

System Requirements

Software Requirements:

- - Front-end: Node.js, ReactiveJS (RxJS + Framework), Webpack/Vite
- - Back-end: Java 17, Spring Boot WebFlux, Spring Security
- - Database: MongoDB 4.4+
- - Tools: Docker, Postman, Kubernetes (for deployment)

Hardware Requirements:

- - Development: 8GB RAM, Quad-core CPU, 20GB storage
- - Production: Scalable cloud setup with autoscaling containers.

Results

- - Login & Signup Pages
- - Product List with Search and Filters
- - Cart and Checkout Process
- - Order Tracking Dashboard
- - Admin Inventory Management

Security & Performance

- HTTPS encryption and JWT-based authentication.
- Secure password hashing using BCrypt.
- MongoDB indexes and pagination for optimized queries.
- Load testing to ensure smooth performance under concurrent requests.
- Optimistic locking for inventory control to avoid overselling.
- Rate limiting and input validation to prevent abuse.

Conclusion & Future Scope

- This project demonstrates a full-stack, reactive, and scalable e-commerce solution for groceries. The integration of ReactiveJS, Spring Boot WebFlux, and MongoDB enables seamless real-time data flow and efficient performance.
- In the future, the system can be enhanced with AI-based product recommendations, voice ordering, predictive restocking, and multi-store management.
- Acknowledgements: We extend our gratitude to our guide, mentors, and institution for continuous support throughout the development.