



FRESHHIVE: ONLINE GROCERY E-COMMERCE SYSTEM USING MERN STACK

¹Aditya Rajendra Balghare, ²Soham Tatyabhau Adhav, ³Gaurav Rameshwar Bhandare,
⁴Yash Anil Bharane, ⁵Pranav Khanderao Bhosale, ⁶Prof.Sanjay Jagtap

¹UG Scholar, Computer, Dept. Bhivrabai Sawant Polytechnic, Wagholi

²UG Scholar, Computer, Dept. Bhivrabai Sawant Polytechnic, Wagholi

³UG Scholar, Computer, Dept. Bhivrabai Sawant Polytechnic, Wagholi

⁴UG Scholar, Computer, Dept. Bhivrabai Sawant Polytechnic, Wagholi

⁵UG Scholar, Computer, Dept. Bhivrabai Sawant Polytechnic, Wagholi

⁶Asst. Prof. Computer Dept. Bhivrabai Sawant Polytechnic, Wagholi

ABSTRACT- Long lines, store visits, short operation hours, and a lack of real-time inventory visibility are common features of traditional grocery shopping. For small retail stores, these difficulties raise operational inefficiencies and decrease client convenience. Using the MERN stack (MongoDB, Express.js, React.js, Node.js), Fresh Hive is an online supermarket e-commerce system that offers a quick, practical, and easy-to-use platform for food ordering. Customers may browse products, add things to a shopping cart, and submit orders with real-time product availability thanks to the system. An integrated dashboard allows administrators to manage orders, categories, and goods.

Tests of the platform's usability, accuracy, and performance revealed excellent success rates for product browsing, order placement, and login. The findings show that Fresh Hive makes a substantial contribution to upgrading regional retail establishments, cutting down on manual labor, and raising consumer happiness.

I. INTRODUCTION

The desire for quicker shopping options, shifting lifestyles, and digital revolution have all contributed to the sharp rise in demand for online grocery shopping. Conventional supermarket shopping techniques necessitate in-person visits, which result in lengthy lines and laborious billing procedures. Online grocery platforms provide convenience, accessibility, and time savings for consumers with hectic schedules, senior citizens, or urban dwellers.. By offering a fully digital platform where consumers can browse, search, and buy groceries online, FreshHive seeks to overcome the drawbacks of conventional methods. The system provides administrators and consumers with a contemporary, secure, and scalable shopping experience by utilizing the MERN stack.

1.1 Problem Statement

Traditional grocery systems face several challenges:

1. Time-consuming physical visits
2. Manual billing errors

3. Lack of real-time product availability
4. Limited store hours
5. Difficulty for small shops to digitize operations

1.2 Objectives

The system aims to:

1. Provide an online platform for browsing and purchasing grocery products
2. Ensure secure user registration and login
3. Implement a shopping cart and order placement module
4. Offer an admin dashboard for managing products and orders
5. Improve shopping convenience and operational efficiency

II. LITERATURE REVIEW

Current supermarket management systems include web-based markets, RFID-based inventory systems, barcode-based invoicing systems, and offline physical stores. Nevertheless, a lot of these solutions are expensive, not scalable, or inappropriate for small merchants..

Recent studies show:

- Barcode-based systems reduce errors but still require physical presence
 - Mobile apps offer convenience but often lack real-time inventory
 - Large e-commerce platforms are costly for small stores
- The research gap highlights the need for:
- Low-cost, scalable online grocery solutions
 - Real-time inventory management
 - Simple UI for customers and admins

FreshHive addresses these gaps by using the MERN stack to build a lightweight but powerful e-commerce platform.

III. METHODOLOGY

To guarantee seamless user-system interaction, the technique comprises of many phases.

3.1 Step 1 – User Registration & Login

Users register using email, password, and basic details. Login authentication uses secure token-based verification.

3.2 Step 2 – Product Browsing

Users browse categorized grocery items with real-time price and availability details.

3.3 Step 3 – Cart & Order Management

The cart allows users to add/remove/update items and proceed to checkout.

3.4 Step 4 – Admin Operations

Admins manage:

- 3.4.1 Product inventory

3.5 Step 5 – Output & Order Confirmation

Users receive confirmation for placed orders along with order summary and delivery details.

IV. SYSTEM ARCHITECTURE

It includes the following architecture:

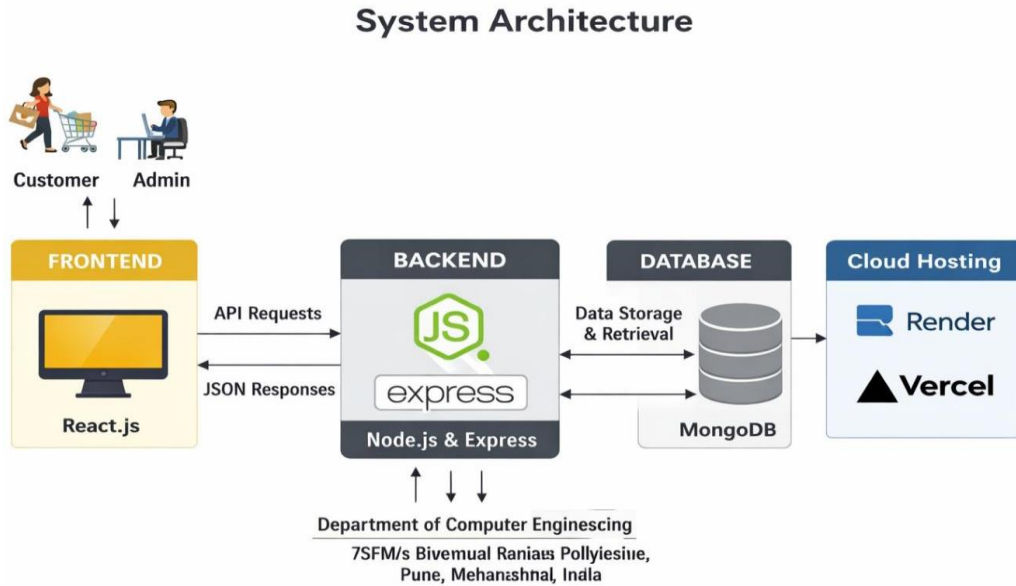


Fig.1- System Architecture

4.1 Frontend

- React.js
- Responsive UI
- Product browsing and cart pages

4.2 Backend

4.2.1 Node.js + Express.js

4.2.2 API handling

4.2.3 Authentication logic

4.3 Database

4.3.1 MongoDB Atlas

4.3.2 Stores users, products, categories, and orders

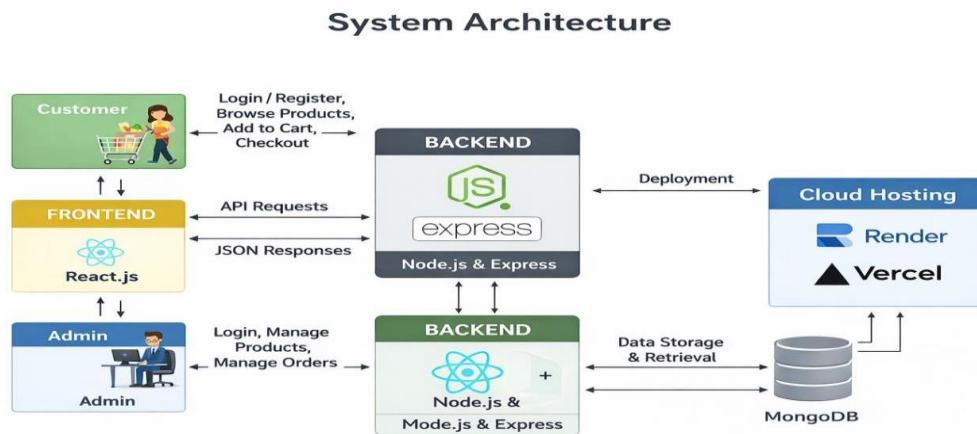


Fig.2-Block Diagram of System Architecture

V. MODELING AND ALGORITHMS

5.1 Product Filtering Algorithm

Uses category-based filtering to display relevant products.

5.2 Order Status Algorithm

Handles order states: *Pending* → *Confirmed* → *Delivered*

5.3 Authentication Algorithm (JWT)

User login → token generation → secure route authorization.



VI. IMPLEMENTATION

The implementation includes several modules:

5.1 User Module

- Registration / Login
- Product browsing
- Add to Cart
- Checkout

5.2 Admin Module

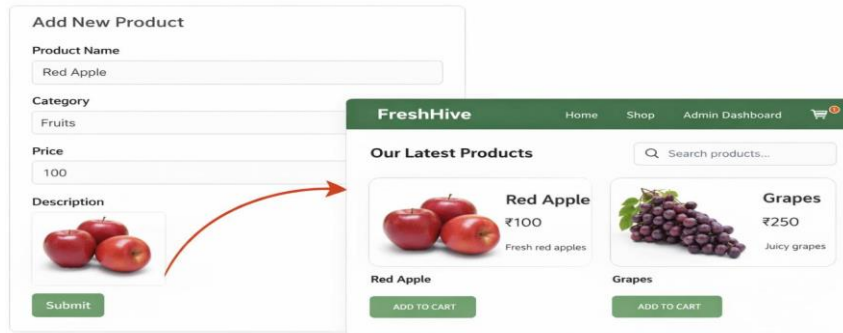
- Manage product list
- Update inventory
- View orders
- Modify order status

5.3 Database Module

Stores all user and admin data with real-time updates

VII. RESULTS & DISCUSSION

- The admin panel successfully adds a new product.
- Product details like **name, category, price, and image** are saved correctly.
- The added product is **immediately displayed on the main website.**
- The product appears in the **“Latest Products” section.**
- The **Add to Cart** button works for newly added products.
- This shows proper **connection between admin panel and user interface.** **Discussion**
- The system provides **real-time update** of products.
- Admin can easily manage products without technical difficulty.
- This reduces **manual work** and saves time.
- The design is **user-friendly** for both admin and customers.
- Such functionality is useful for **e-commerce websites.**
- It improves **efficiency and accuracy** in product management.



Admin Panel in admin panel on nw website

VIII. CONCLUSION

By automating conventional procedures, FreshHive effectively offers a contemporary grocery shopping experience. The MERN-based platform guarantees user ease, quick performance, and seamless operations. The technology increases consumer happiness and supports small merchants' digital growth.

REFERENCES

- Kumar A., Sharma R. "Online Grocery Management System," IJCA, 2021.
- React.js Documentation
- MongoDB Atlas Documentation
- Node.js Official Documentation
- Mehta S., Joshi P. "Secure E-commerce using MERN," 2022