FoodExpress – Online Food Delivery System

1. Abstract

FoodExpress is a MERN-stack-based web application designed to streamline online food ordering and delivery. Users can browse menus, add items to a cart, and place orders, while restaurant admins can manage menu items, track orders, and update delivery status. The platform aims to bridge the gap between customers and restaurants with a user-friendly interface, secure transactions, and real-time order management.

2. Introduction

2.1 Project Overview

FoodExpress provides a centralized platform for restaurants and customers. It offers separate interfaces for customers (ordering) and admins (restaurant management). The system is built with React (frontend), Node.js & Express (backend), and MongoDB (database), ensuring scalability and performance.

2.2 Purpose

- Enable customers to browse menus and place orders online.
- Provide restaurants with an easy-to-use admin panel to manage orders.
- Support secure payment and real-time order tracking.
- Improve operational efficiency for restaurants.

3. Problem Statement

With busy lifestyles, customers prefer online ordering over dining in. However, small and medium restaurants often lack affordable, easy-to-manage systems. FoodExpress addresses this by providing an accessible, feature-rich, and cost-effective platform for food ordering and delivery.

4. Requirement Analysis

4.1 Functional Requirements

Requirement	Description
User Registration/Login	Customers and admins must authenticate to access features.
Menu Browsing	Customers can view and filter menu items.
Cart Management	Users can add, remove, and update items in their cart.
Order Placement	Orders can be placed with payment integration.
Order Tracking	Customers can see order status in real-time.
Menu Management	Admins can add, update, or delete menu items.
Order Management	Admins can view and update order statuses.

4.2 Non-Functional Requirements

Requirement	Description
Usability	Responsive UI for all devices.
Performance	Orders processed within 2 seconds.
Security	Passwords hashed; secure payment processing.
Availability	Accessible via web browsers 24/7.
Scalability	Supports multiple restaurants and large user bases.

5. System Design

Frontend: React (Vite) for admin and customer UI.
Backend: Node.js with Express for API handling.
Database: MongoDB for storing menu, orders, and user data.
Storage: Cloud/Local for images of food items.
Hosting: Render (backend), GitHub Pages (frontend).

6. Technology Stack

Frontend: React.js, Vite, HTML, CSS, JavaScript
Backend: Node.js, Express.js
Database: MongoDB
Tools: GitHub, Render, Postman
Other: Cloudinary/Local uploads for images

7. Implementation

Workflow:
1. Customer registers/login.
2. Browse menu and add items to cart.
3. Place order with payment.
4. Backend processes order and updates status.
5. Admin receives order in dashboard, updates progress.
6. Customer receives real-time status updates.

Modules:
- Customer Module: Registration, browsing, ordering, tracking.
- Admin Module: Menu CRUD operations, order management.
- Authentication Module: Secure login for both users and admins.

8. Testing & Results

Test Case	Input	Expected Output	Actual Output	Status
TC1	Customer adds item to cart	Item visible in cart	Item visible in cart	Pass
TC2	Admin adds menu item	Item shown in customer menu	Item shown	Pass
TC3	Customer places order	Order appears in admin dashboard	Order appears	Pass

9. Advantages & Limitations

Advantages:
- Easy to use for customers and restaurants.
- Real-time updates on orders.
- Scalable and customizable.
- Secure payment integration.

Limitations:
- Requires internet connection.
- Backend hosting cost increases with traffic.
- Payments depend on third-party gateways.

10. Conclusion

FoodExpress successfully integrates restaurant management and customer ordering into a single platform. It delivers a smooth user experience for both sides while ensuring scalability and security.

11. Future Scope

- Mobile app development (Android/iOS).
- AI-based food recommendations.
- Integration with delivery tracking APIs.
- Multi-restaurant support with location-based filtering.
- Wallet system and loyalty points.