FOOD DELIVERY SYSTEM

Project Report

Project Title: Food Delivery System
Student Name: Seyha
Project Type: Individual Project
Technology Stack: MERN Stack (MongoDB, Express.js, React, Node.js)
Academic Purpose: University / College Submission & Portfolio

---

1. Introduction

With the rapid growth of internet usage and mobile technologies, online food delivery systems have become an essential part of modern life. Customers increasingly prefer ordering food online due to convenience, time efficiency, and easy access to various restaurants and menus.

The **Food Delivery System** is a web-based application designed to allow users to browse food items, add them to a cart, place orders, and manage deliveries through an admin dashboard. This project simulates a real-world food delivery platform by integrating frontend, backend, authentication, database management, and deployment.

This project was developed as an **individual university project** and also serves as a **professional portfolio project** demonstrating full-stack development skills.

---

2. Problem Statement

Traditional food ordering methods such as phone calls or physical visits are inefficient and prone to errors. Customers may face issues such as: - Long waiting times - Incorrect orders - Lack of order tracking - No centralized management system for admins

There is a need for a centralized, user-friendly, and secure food delivery platform that simplifies food ordering and management for both customers and administrators.

---

3. Project Objectives

The main objectives of this project are: - To develop a responsive food delivery web application - To allow users to browse food items and place orders online - To implement secure user authentication and authorization - To provide an admin dashboard for managing foods, users, and orders - To store and manage data efficiently using a database - To deploy the application for real-world access

---

4. Scope of the Project

4.1 User Scope

- User registration and login
- Browse food items
- Add food items to cart
- Place and track orders

4.2 Admin Scope

- Secure admin login
- Manage food items (add, update, delete)
- Manage users
- Manage and update order status
- View overall system data

---

5. Technology Stack

5.1 Frontend

- **React.js** – Component-based UI development
- **Vite** – Fast build tool
- **Tailwind CSS** – Modern utility-first styling
- **React Router** – Client-side routing
- **Axios** – API communication

5.2 Backend

- **Node.js** – Server-side runtime
- **Express.js** – Backend framework

5.3 Database

- **MongoDB** – NoSQL database for storing users, foods, and orders

5.4 Tools & Deployment

- **Git & GitHub** – Version control
- **Vercel** – Frontend deployment
- **REST API** – Backend communication

---

6. System Architecture

The Food Delivery System follows a **client-server architecture**:

1. The **frontend (React)** sends requests to the backend using REST APIs
2. The **backend (Express & Node.js)** processes requests
3. The **database (MongoDB)** stores and retrieves data
4. Responses are sent back to the frontend and displayed to users

This architecture ensures scalability, maintainability, and separation of concerns.

---

7. Database Design

7.1 User Collection

- User ID
- Name
- Email
- Password (encrypted)
- Role (User / Admin)

7.2 Food Collection

- Food ID
- Name
- Price
- Category
- Image
- Availability

7.3 Order Collection

- Order ID
- User ID
- Ordered Items
- Total Price
- Order Status
- Date

---

8. Key Features Implementation

8.1 Authentication

- Secure login and registration system
- JWT-based authentication

- Role-based access control (Admin / User)

8.2 Cart System

- Add or remove food items
- Quantity management
- Real-time total price calculation

8.3 Order Management

- Place orders
- Admin can update order status
- Users can view order history

8.4 Admin Dashboard

- Food management
- Order management
- User management

---

9. User Interface Design

The application is fully responsive and optimized for: - Desktop - Tablet - Mobile devices

Tailwind CSS is used to ensure clean, modern, and consistent UI design.

---

10. Challenges and Solutions

Challenge 1: API Integration

Solution: Axios was used to handle API requests efficiently with proper error handling.

Challenge 2: Authentication Security

Solution: Implemented JWT tokens and protected routes.

Challenge 3: Deployment Issues

Solution: Proper routing configuration and environment variables were set for Vercel deployment.

---

11. Testing

The system was tested using: - Manual testing - API testing via Postman - Browser testing across different screen sizes

All major functionalities were tested to ensure reliability and performance.

---

12. Future Enhancements

Possible future improvements include: - Online payment integration - Real-time order tracking - Restaurant management module - Push notifications - Mobile application version

---

13. Conclusion

The Food Delivery System successfully meets its objectives by providing a functional, secure, and scalable web application. This project demonstrates practical knowledge of full-stack web development, REST APIs, authentication, database design, and deployment.

This project is suitable for **university submission** and also serves as a **professional portfolio project**, reflecting real-world software development practices.

---

14. References

- React Documentation
- Node.js Documentation
- MongoDB Documentation
- Tailwind CSS Documentation