

Table of Contents

[Introduction](#)

[System Architecture](#)

[Features](#)

- [User Authentication](#)
- [Ordering System](#)
- [Delivery Tracking](#)
- [Payment Integration](#)

[Technologies Used](#)

[Development Environment Setup](#)

[API Documentation](#)

- [User Authentication API](#)
- [Ordering API](#)
- [Delivery Tracking API](#)
- [Payment API](#)

[Database Schema](#)

[Security Considerations](#)

[Testing](#)

- [Unit Testing](#)
- [Integration Testing](#)
- [User Acceptance Testing](#)

[Deployment](#)

[Monitoring and Logging](#)

[Maintenance and Updates](#)

Conclusion

1. Introduction

The Food Delivery App is a web and mobile application designed to streamline the process of ordering food and having it delivered to the user's location. This technical documentation provides an in-depth overview of the system architecture, features, technologies used, development environment setup, API documentation, database schema, security considerations, testing procedures, deployment strategies, and ongoing maintenance.

2. System Architecture

The Food Delivery App follows a client-server architecture. The client-side consists of web and mobile applications, while the server-side includes a set of APIs for handling user authentication, order processing, delivery tracking, and payment integration. The system leverages a cloud-based infrastructure for scalability and reliability.



3. Features

User Authentication

The app includes a robust user authentication system that supports secure registration, login, and password recovery functionalities. Users can create accounts using email or social media credentials.

Ordering System

Users can browse through a list of restaurants, view menus, place orders, and customize their selections. The ordering system includes features like adding items to the cart, specifying delivery preferences, and viewing order summaries.

Delivery Tracking

Real-time order tracking is available for users and delivery drivers. The system integrates with GPS services to provide accurate location updates. Users receive notifications about the status of their orders.

Payment Integration

Secure payment processing is implemented using industry-standard encryption protocols. The app supports various payment methods, including credit cards, digital wallets, and cash on delivery.

4. Technologies Used

- Frontend: React.js (web), React Native (mobile)
- Backend: Node.js with Express.js
- Database: MongoDB
- Authentication: JWT (JSON Web Tokens)
- Payment: Stripe API
- Real-time Communication: WebSocket
- Cloud Hosting: AWS (Amazon Web Services)

5. Development Environment Setup

Detailed instructions for setting up the development environment for both frontend and backend development can be found in the [Development Environment Setup Guide](#).

6. API Documentation

Detailed API documentation for each major feature is available in the [API Documentation](#) section.

User Authentication API

- `/api/auth/register`: Create a new user account.
- `/api/auth/login`: Authenticate a user.
- `/api/auth/forgot-password`: Initiate the password recovery process.

Ordering API

- `/api/orders/create`: Place a new order.
- `/api/orders/track/:orderId`: Get real-time tracking updates for an order.

Delivery Tracking API

- `/api/delivery/track/:driverID`: Get real-time location updates for a delivery driver.
- `/api/delivery/update/:driverID`: Update the delivery status.

Payment API

- `/api/payment/process`: Process payment for an order.

7. Database Schema

The database schema is outlined in the [Database Schema](#) document.

8. Security Considerations

The [Security Considerations](#) document provides insights into security measures implemented in the Food Delivery App.

9. Testing

Various testing methodologies are employed to ensure the reliability and functionality of the app.

Unit Testing

Unit tests cover individual components, functions, and modules.

Integration Testing

Integration tests verify the interaction between different parts of the system.

User Acceptance Testing

User acceptance tests involve real users to assess the app's usability, functionality, and overall satisfaction.

10. Deployment

The [Deployment Guide](#) outlines the steps to deploy the Food Delivery App to a production environment.

11. Monitoring and Logging

Monitoring tools are implemented to track system performance, identify bottlenecks, and log potential issues. The [Monitoring and Logging](#) document provides details on these practices.

12. Maintenance and Updates

Ongoing maintenance involves addressing bug reports, security patches, and feature updates. The [Maintenance and Updates](#) document provides guidance on these activities.

13. Conclusion

This technical documentation serves as a comprehensive guide for developers, administrators, and stakeholders involved in the Food Delivery App. Regular updates will be made to ensure accuracy and relevance as the application evolves. If you have any questions or feedback, please contact the development team.