PROJECT REPORT

(Project Term January- May 2023)

(Online Food Odering System)

Submitted by

NAME =Abhishek Kumar **Registration No =12216725**

Course Title = Advanced Web Development

Course Code = INT222

Submitted To

Faculty Name – Mr.Akash Pundir

School of Computer Science and Engineering

DECLARATION

I hereby declare that the project work entitled ("Order Food") is an authentic record of my own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara, under the guidance of Akash Pundir during January to May 2024. All the information furnished in this project report is based on my own intensive work and is genuine.

Name of Student:Abhishek Kumar

Registration
Number:12216725

(Signature of Student)
Date:30-04-24

CERTIFICATE

This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief. He /She have completed this Project under my guidance and supervision. The present work is the result of his/her original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara.

Signature of the Staff

School of Computer Science and Engineering,

Lovely Professional University, Phagwara,
Punjab.

Date : 30-04-24

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this project is to create a comprehensive Software Requirements Specification (SRS) document for an online food ordering System. The SRS will outline the functional and non-functional requirements, system architecture, user interfaces, and system behaviour of the website, providing a clear roadmap for the development team to design, implement, and maintain a robust and user-friendly e-commerce platform tailored specifically for selling clothes. By defining the project scope, objectives, and specifications in detail, the SRS aims to ensure alignment between stakeholders, guide the development process, and ultimately deliver a high-quality, feature-rich online food experience for customers while meeting the business goals of the food store.

1.2 Intended Audience

The document describes the scope, functionality and features of the online food store website which has a large audience. This document finds relevance to people from various technical and non-technical backgrounds. The document outlines various corporate goals, business strategies and design features that are important from a management point of view and can be used by project managers. It analyses performance, visibility and brand awareness which is important for marketing and advertising. With detailed analysis of the system design, features, implementation and performance, the document proves highly valuable to developers and testers.

Through the rest of the document, one becomes familiarized with the scope of these online shopping systems from their purpose, benefits, and businesses strategies. The context and origin of the product as well as basic functionality are then explained in relevant detail along with analysis of its different classes, design and implementation. We then detail the interface requirements build analysis models and examine system features and nonfunctional requirements.

1.3 Scope

The scope of a food ordering system encompasses the boundaries and objectives of the project, including the features, functionalities, and deliverables that will be included in the final product. Here's an overview of the scope of a typical food ordering system:

1. **User Interface (UI)**:
   - User-friendly interfaces accessible via web browsers and/or mobile applications.
   - Features for browsing restaurants, viewing menus, and placing orders.
   - Search functionality, filtering options, and visual displays of menu items.
2. **Menu Management** :
   - Tools for restaurants to manage their menus, including adding, updating, and removing items.
   - Options for setting prices, descriptions, and customization choices for menu items.
   - Ability to categorize menu items by cuisine type, dietary restrictions, or other attributes.
3. **Order Placement and Processing**:
   - Capability for users to add items to their cart, specify quantities, and make customizations.
   - Checkout process with options for payment methods (credit/debit cards, mobile wallets, cash on delivery).
   - Order confirmation and processing, including notifications to users and restaurants.
4. **Order Management** :
   - Interface for restaurants to view and manage incoming orders.
   - Options for accepting, rejecting, or updating order statuses (e.g., preparing, out for delivery).
   - Communication features to interact with customers regarding order details or issues.
5. **Delivery Coordination**:
   - Integration with delivery services or options for in-house delivery.
   - Assignment of delivery drivers to orders, including route optimization and real-time tracking.
   - Updates for users on delivery status and estimated arrival times.
6. **Customer Accounts and Profiles**:
   - User registration and login functionality.
   - Profiles for users to store personal information, delivery addresses, and

payment methods.

- Order history, reorder options, and subscription management features.

7. **Feedback and Reviews**:
   - Ability for users to provide feedback and ratings for restaurants, food items, and delivery service.
   - Display of reviews and ratings to help users make informed decisions.
   - Mechanisms for restaurants to respond to feedback and address customer concerns.
8. **Admin Dashboard** :
   - Centralized dashboard for administrators to manage system settings, user accounts, and restaurant profiles.
   - Analytics and reporting tools for monitoring order volumes, sales data, and customer feedback.
   - Options for configuring delivery zones, fees, promotions, and other system parameters.
9. **Integration with Third-Party Services**:
   - Integration with mapping/navigation APIs for accurate delivery estimates and tracking.
   - Integration with payment gateways and POS systems for secure transaction processing.
   - Collaboration with delivery logistics platforms for efficient delivery operations.
10. **Security and Compliance** :
    - Implementation of security measures to protect user data, payment information, and system integrity.
    - Compliance with relevant regulations and standards for data privacy, food safety, and online transactions.
    - Measures for preventing unauthorized access, data breaches, and fraudulent activities.

1.4 Overview

1. **User Interface** : The system typically features a user-friendly interface accessible via web browsers or mobile applications. Users can browse through

a variety of restaurants, view menus, and place orders with ease. The interface often includes search functionality, filtering options, and visual displays of menu items to enhance the user experience.

2. **Menu Management**: Participating restaurants manage their menus within the system, including adding, updating, and removing items as needed. Menu management tools allow restaurants to showcase their offerings, set prices, describe dishes, and specify options for customization.
3. **Order Placement**: Users can add desired items to their virtual shopping cart, specify quantities, and make customizations or special requests (e.g., dietary restrictions, additional toppings). Once the order is complete, users proceed to checkout where they can review their order details and select payment and delivery options.
4. **Payment Processing**: The system supports secure payment processing, allowing users to pay for their orders using various payment methods such as credit/debit cards, mobile wallets, or cash on delivery. Payment gateways handle transaction processing and ensure the security of payment information

2. General Description

2.1 Product Perspective

1. **User-Centric Design** : The system should prioritize user experience by offering intuitive interfaces, easy navigation, and seamless ordering processes. It should cater to the diverse needs and preferences of users, including customization options, multiple payment methods, and order tracking features.
2. **Efficient Order Management**: From the perspective of restaurants and delivery partners, the system should streamline order management processes, including order processing, preparation, and delivery logistics. It should provide tools for managing menu items, updating availability, and tracking order statuses in real-time.
3. **Business Growth and Revenue Generation**: The system should support business growth and revenue generation for participating restaurants by

expanding their customer reach, increasing order volumes, and fostering customer loyalty through promotions and rewards programs. It should provide analytics and reporting tools to monitor performance and identify opportunities for improvement.

Scalability and Flexibility4. : The system should be designed to scale and adapt

to changing demands, including fluctuations in user traffic, menu offerings,

and delivery capacities. It should support the addition of new restaurants, integration with third-party services, and expansion into new geographic markets without compromising performance or usability.

5. **Data-Driven Insights**: The system should leverage data analytics to provide actionable insights for stakeholders, including user behavior analysis, order trends, and customer feedback. It should enable data-driven decision-making to optimize menu offerings, marketing strategies, and operational efficiency.

2.2 Product Functions

1. **User Registration and Authentication** :
   - Allow users to create accounts securely.
   - Enable users to log in using email, social media accounts, or mobile numbers.
   - Implement password reset and account recovery mechanisms.
2. **Browse Menu and Search** :
   - Provide a categorized menu of available food items.
   - Enable users to search for specific items using keywords or filters (e.g., cuisine type, dietary restrictions).
3. **Place Orders** :
   - Allow users to add food items to their virtual shopping cart.
   - Enable users to specify quantities, modifications (e.g., toppings, cooking preferences), and special instructions for each item.
   - Provide a checkout process for reviewing and confirming orders before finalizing the purchase.
4. **Payment Processing** :
   - Support multiple payment options (e.g., credit/debit cards, mobile

wallets, cash on delivery).
- Securely process payments and provide confirmation to users upon successful transactions.
- Automatically calculate taxes, fees, and discounts.

5. **Order Tracking and Status Updates** :

- Enable users to track the status of their orders in real-time (e.g., order received, preparation in progress, out for delivery).
- Send notifications or updates via email, SMS, or push notifications at key stages of the order process.
-

2.3 User Characteristics

1. **Demographics** : Users of the food ordering system can span various demographics, including age, gender, income level, occupation, and geographic location. Different demographic groups may have distinct preferences for cuisines, dietary restrictions, and ordering habits.
2. **Technological Proficiency**: Users may vary in their level of technological proficiency, ranging from tech-savvy individuals who are comfortable navigating digital platforms to those who may require assistance or guidance to use the system effectively. Design considerations should accommodate users with varying levels of digital literacy.
3. **Device Preferences**: Users may access the food ordering system through different devices, including smartphones, tablets, desktop computers, and smart speakers. Understanding device preferences can inform the design of responsive interfaces and cross-platform compatibility.
4. **Ordering Frequency**: Users' ordering frequency can vary, with some placing orders regularly for meals and others ordering less frequently for special occasions or convenience. The system should cater to both frequent and occasional users, providing features such as order history and quick reordering options

2.4 General Constraints

1. **Technology Limitations** : The system may be constrained by the available technology stack, including programming languages, frameworks, and

infrastructure. Compatibility with legacy systems or adherence to specific technology standards may also impose constraints.

2. **Budgetary Constraints**: The system's development and maintenance are constrained by budget limitations. This includes funding for software development, hardware infrastructure, third-party services, and ongoing operational costs.
3. **Time Constraints**: The system may have deadlines for development, testing, and deployment, which constrain the available time for implementing features, resolving issues, and meeting project milestones.
4. **Resource Constraints**: Constraints related to human resources, such as the availability of skilled developers, designers, testers, and other team members, can impact the project's timeline and deliverables.

.

2.5 Assumptions and Dependencies

Internet Connectivity: The system assumes that users, restaurants, and

1.

access to stable internet connectivity to interact with the
on internet infrastructure reliability is inherent.
The system assumes that users have compatible

Device Compatibility 2. :

devices (e.g., smartphones, tablets, computers) to access the platform.

Compatibility with various operating systems and browsers is also assumed. 3. **User Skills**: The system assumes that users possess basic digital literacy skills to navigate the platform, place orders, and interact with interface elements. Dependence on user familiarity with technology is implicit.

4. **Menu Availability**: The system depends on participating restaurants to maintain up-to-date menus with accurate item descriptions, prices, and availability. Any discrepancies may affect order accuracy and customer satisfaction.

Payment Gateways 5. : The system relies on third-party payment gateways for processing online payments securely. Dependence on the availability and reliability of payment gateway services is significant.

Delivery Services 6. : The system assumes access to reliable delivery services for fulfilling orders placed through the platform. Dependence on delivery partners' availability, capacity, and efficiency is crucial for timely order delivery.

3. Specific Requirements

Specific requirements for a food ordering system delve into the detailed functionalities and features necessary for its successful implementation. Here's a breakdown of specific requirements:

1. **User Registration and Authentication** :
   - Users should be able to register accounts with valid email addresses or phone numbers.
   - Account authentication should be secure, requiring password verification or multi-factor authentication.
   - Provide options for social media login for convenience.
2. **Menu Browsing and Search** :
   - Users should be able to browse menus by categories such as cuisine type, dietary restrictions, or meal type.
   - Implement search functionality allowing users to find specific items by name or keywords.
   - Filter options should include sorting by popularity, price, and ratings.
3. **Order Placement** :
   - Users should be able to add items to their cart with options for customization (e.g., toppings, cooking preferences).
   - Quantity selection and special instructions should be supported for each item.
   - Ensure a smooth checkout process with options for delivery or pickup.
4. **Payment Processing** :
   - Support multiple payment methods, including credit/debit cards, digital wallets, and cash on delivery.
   - Implement secure payment processing with encryption and compliance with PCI DSS standards.
   - Provide users with payment confirmation and receipts.
5. **Order Tracking and Status Updates** :

- Users should receive real-time updates on the status of their orders, including confirmation of order received, preparation, and delivery.
- Offer delivery tracking features with estimated arrival times and live updates on the delivery driver's location.

6. **Account Management** :
   - Allow users to manage their profiles, including updating contact information, delivery addresses, and payment methods.
   - Provide access to order history, allowing users to view past orders and reorder items easily.
   - Offer options for managing subscription preferences, notifications, and loyalty programs.
7. **Feedback and Reviews** :
   - Enable users to leave feedback and ratings for restaurants, food items, and delivery service.
   - Display reviews prominently on the platform to help users make informed decisions.
   - Allow restaurants to respond to feedback and address customer concerns.
8. **Admin Dashboard** :
   - Administrators should have access to a comprehensive dashboard for managing users, restaurants, and system settings.
   - Provide analytics and reporting tools for monitoring order volumes, sales data, and customer feedback.
   - Options for configuring delivery zones, fees, promotions, and other system parameters.
9. **Integration with Third-Party Services**:
   - Integrate with mapping/navigation APIs for accurate delivery estimates and tracking.
   - Integration with payment gateways and POS systems for secure transaction processing.
   - Collaborate with delivery logistics platforms for efficient delivery operations.
10. **Security and Compliance** :

- Implement measures to protect user data, payment information, and system integrity.
- Ensure compliance with relevant regulations and standards for data privacy, food safety, and online transactions.
- Conduct regular security audits and penetration testing to identify and mitigate vulnerabilities.

3.1.1 Hardware Interfaces

1. **Point-of-Sale (POS) Systems**:

- Many restaurants use POS systems to manage orders, process payments, and track inventory.
- The food ordering system should integrate with POS hardware such as terminals, cash registers, card readers, barcode scanners, and receipt printers.
- Integration with POS systems ensures seamless order processing and payment handling, allowing orders placed online to be synchronized with in-house operations.

2. **Mobile Devices**:

- Customers use mobile devices (smartphones, tablets) to access the food ordering system through mobile apps or web browsers.
- The food ordering system should be compatible with a variety of mobile devices and screen sizes, ensuring a consistent user experience across different platforms.
- Mobile devices may also be used by delivery drivers to receive order details, navigation instructions, and customer contact information.

3. **Kitchen Display Systems (KDS)** :

- In restaurant kitchens, KDS devices display incoming orders and provide instructions to kitchen staff for preparing and assembling dishes.

- The food ordering system should integrate with KDS hardware to transmit order information in real-time and update order statuses as they progress through the kitchen workflow.
- KDS devices may include touch-screen displays, monitors, or dedicated terminals installed in the kitchen.

4. **Digital Signage Displays** :

- Some restaurants use digital signage displays for advertising menu items, promotions, and special offers.
- The food ordering system may interface with digital signage hardware to update content dynamically based on menu changes, seasonal specials, or real-time order data.
- Digital signage displays may be controlled remotely or locally using specialized hardware and software solutions.

5. **Barcode/QR Code Scanners** :

- Restaurants may use barcode or QR code scanners to scan order receipts or loyalty cards for processing orders or applying discounts.
- The food ordering system should support integration with barcode/QR code scanner hardware to capture and interpret codes printed on physical media.
- Barcode/QR code scanners may be handheld devices or built into POS terminals or mobile devices.

3.1.2 Software Interfaces

1. **Database Interface** :

- The system interacts with a database management system (DBMS) to store and retrieve data related to users, restaurants, menus, orders, and transactions.

- Database interfaces include SQL (Structured Query Language) queries and commands used to perform CRUD (Create, Read, Update, Delete) operations on database tables.
- Object-relational mapping (ORM) frameworks may be used to abstract database interactions and simplify data access code.

2. **Payment Gateway Interface** :

- The system communicates with payment gateway APIs to process online payments securely.
- Payment gateway interfaces handle requests for authorizing transactions, capturing funds, and generating payment receipts.
- Integration with payment gateways typically involves sending payment requests with transaction details and receiving response codes indicating the status of the transaction (e.g., success, failure, pending).

3. **Third-Party Service Interfaces** :

- Integration with third-party services such as mapping/navigation APIs, SMS gateways, and delivery logistics platforms requires well-defined interfaces.
- APIs provided by third-party services define methods for sending requests and receiving responses to perform specific actions, such as geocoding addresses, calculating delivery routes, and tracking deliveries.
- Integration interfaces may include RESTful APIs, SOAP (Simple Object Access Protocol) APIs, or SDKs (Software Development Kits) provided by third-party service providers.

4. **User Interface Components**:

- The system's user interface components interact with front-end frameworks and libraries to render UI elements and handle user interactions.
- User interface components may include web pages, mobile app screens, form elements, buttons, input fields, and navigation menus.

- Interaction with UI components involves event handling, state management, and data binding to update UI elements based on user input and system state changes.

5. **Application Logic Interfaces**:
    - The system's application logic consists of business rules, algorithms, and workflows that govern order processing, menu management, payment handling, and other core functionalities.
    - Application logic interfaces define methods and protocols for invoking business logic components and processing requests from external clients.
    - Service-oriented architectures (SOA) or microservices architectures may use APIs (e.g., RESTful APIs, gRPC) to expose application logic as reusable services that can be invoked by other system components

3.1.3 Communications Interface

1. **User Interface (UI)**:
    - The user interface allows users to interact with the system through various channels such as web browsers, mobile apps, or voice assistants.
    - It should provide intuitive navigation, visual feedback, and input controls for browsing menus, placing orders, and managing account settings.
    - The UI should be responsive and adaptable to different screen sizes and devices.
2. **API (Application Programming Interface)** :
    - The system should expose APIs that enable communication between different modules and external services.
    - RESTful APIs are commonly used for retrieving menu information, processing orders, handling payments, and managing user accounts.
    - APIs should be well-documented with clear specifications for endpoints, request/response formats, authentication mechanisms, and error handling.
3. **Admin Interface** :

- Administrators and restaurant owners/managers require an interface for managing restaurant information, menu items, orders, and system settings.
- The admin interface should provide features for adding/editing restaurant details, updating menus, viewing order history, and generating reports.
- Access to the admin interface should be restricted to authorized users with appropriate permissions.

4. **Notification System** :
   - The system should include a notification system for sending alerts and updates to users, restaurants, and delivery partners.
   - Notifications may include order confirmation, order status updates (e.g., order received, order in preparation, order out for delivery), delivery ETA, and promotional messages.
   - Notifications can be sent via email, SMS, push notifications, or in-app messages, depending on user preferences and system capabilities.
5. **Payment Gateway Integration** :
   - Integration with payment gateways allows the system to securely process payments for orders placed by users.
   - Communication with payment gateways should adhere to industry standards for encryption and data security to protect sensitive payment information.
   - The system should support multiple payment methods (e.g., credit/debit cards, digital wallets, cash on delivery) to accommodate user preferences

3.2 Functional Requirements

1. **User Registration and Authentication** :
   - Users should be able to create accounts or profiles.
   - Users should be able to log in securely using email, social media accounts, or mobile numbers.
2. **Menu Browsing and Search** :

- Users should be able to browse through a categorized menu of available food items.
- Users should be able to search for specific items using keywords or filters (e.g., cuisine type, dietary restrictions).

3. **Order Placement** :

- Users should be able to add food items to their virtual shopping cart.
- Users should be able to specify quantities, modifications (e.g., toppings, cooking preferences), and special instructions for each item.
- Users should be able to review and confirm their orders before finalizing the purchase.

4. **Payment Processing** :

- Users should be able to choose from multiple payment options (e.g., credit/debit card, mobile wallets, cash on delivery).
- The system should securely process payments and provide confirmation to users upon successful transactions.
- The system should support automatic calculation of taxes, fees, and discounts.

5. **Order Tracking and Status Updates** :

- Users should be able to track the status of their orders in real-time (e.g., order received, preparation in progress, out for delivery).
- Users should receive notifications or updates via email, SMS, or push notifications at key stages of the order process.

6. **Account Management** :

- Users should be able to view and update their profile information (e.g., contact details, delivery addresses).
- Users should be able to view their order history and reorder previous orders with ease.
- Users should be able to manage subscription preferences, if applicable (e.g., newsletter subscriptions, loyalty programs).

7. **Feedback and Reviews** :

- Users should be able to provide feedback and ratings for food items, restaurants, and overall service quality.

- Users should be able to view reviews and ratings submitted by other customers to inform their purchasing decisions.

8. **Restaurant Management** :

- Restaurants should be able to manage their menus, including adding, updating, and removing items.
- Restaurants should be able to set availability times for ordering and specify delivery areas.
- Restaurants should have access to order management tools for processing incoming orders and updating order statuses

3.3 Non-Functional Requirements

1. **Performance** : The system should be able to handle a high volume of concurrent users without significant performance degradation, ensuring quick response times for order placement, updates, and queries.

2. **Scalability**: The system should be designed to scale horizontally or vertically to accommodate increased load during peak times or future growth without sacrificing performance.

3. **Reliability**: The system should be highly reliable, with minimal downtime or service interruptions. It should be resilient to failures and able to recover quickly from any unexpected issues.

Security 4. : The system should ensure the security and privacy of user data, including personal information, payment details, and order history. It should implement strong authentication, encryption, and access control mechanisms to prevent unauthorized access or data breaches.

5. **Usability** : The system should have a user-friendly interface that is intuitive and easy to navigate, catering to users of varying technical abilities. It should provide clear instructions and feedback to users throughout the ordering process.

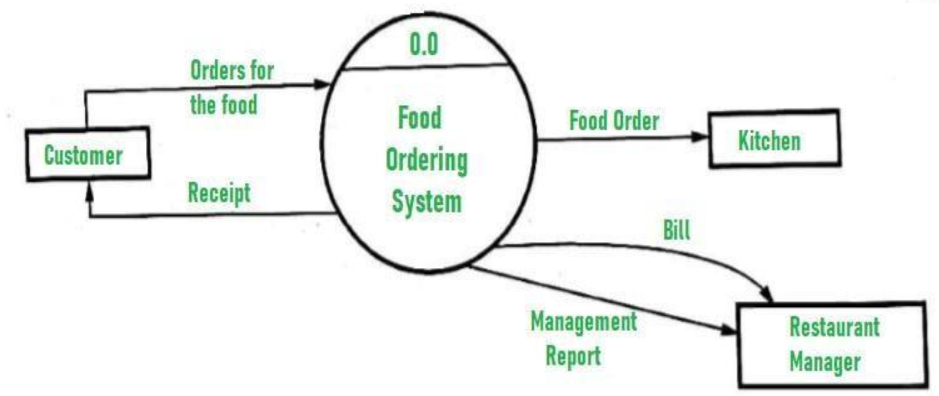
6. **Accessibility**: The system should be accessible to users with disabilities, complying with relevant accessibility standards such as WCAG (Web Content

Accessibility Guidelines) to ensure that all users can interact with the system effectively.

7. **Compatibility**: The system should be compatible with a variety of devices and platforms, including web browsers, mobile devices, and operating systems, to reach a wide range of users.
8. **Maintainability**: The system should be easy to maintain and update, with well-documented code and modular architecture that allows for efficient troubleshooting, bug fixes, and enhancements

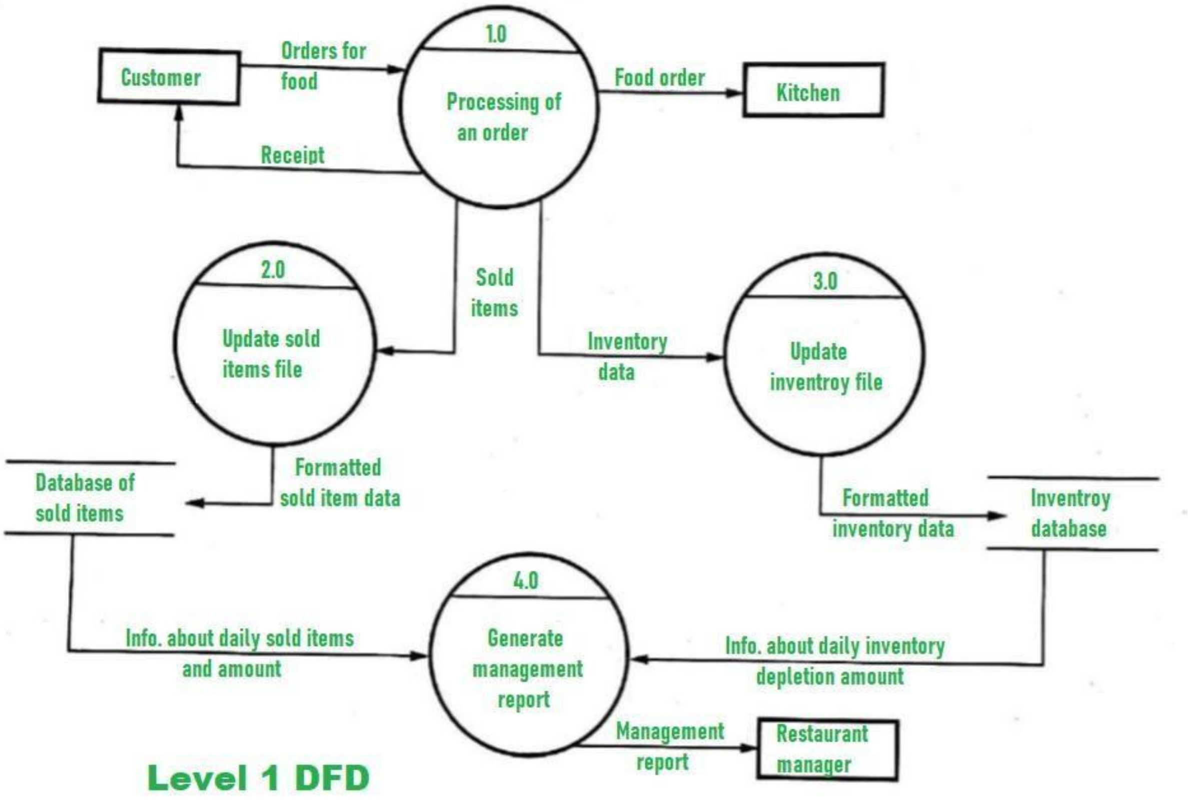
4. DFD Diagrams

4.1 Level 0 DFD

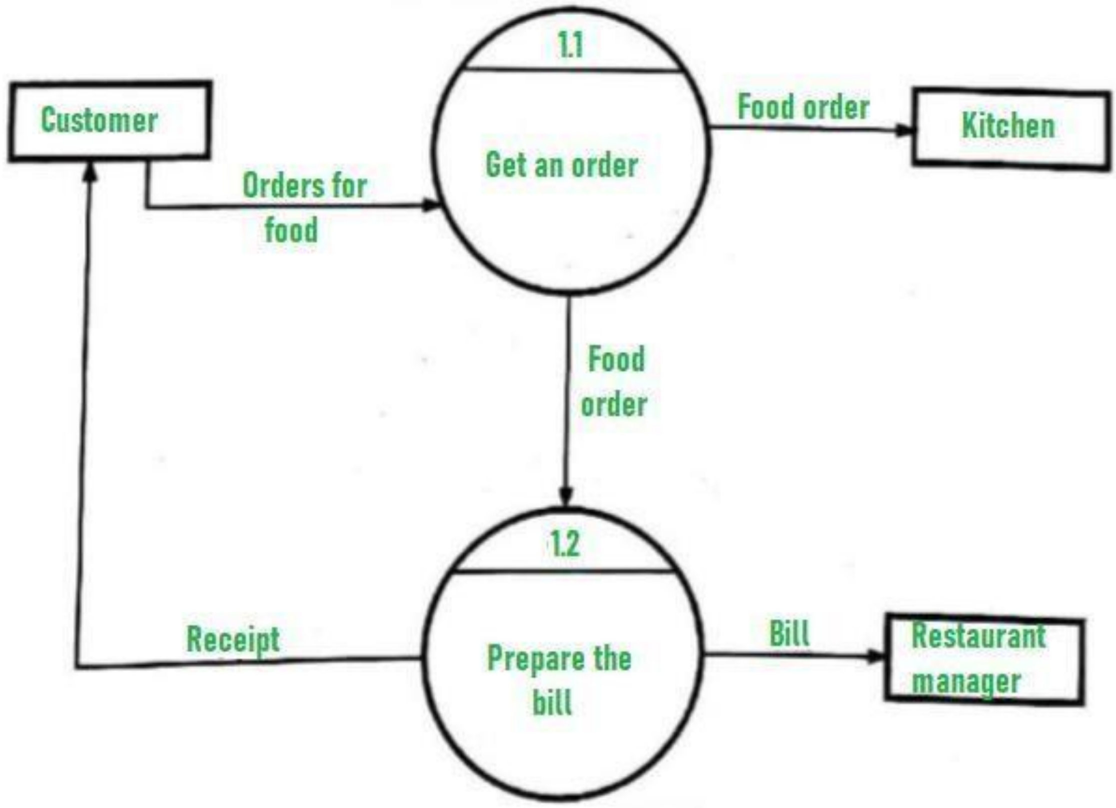


Level 0 DFD (Context Level

4.2 Level 1 DFD



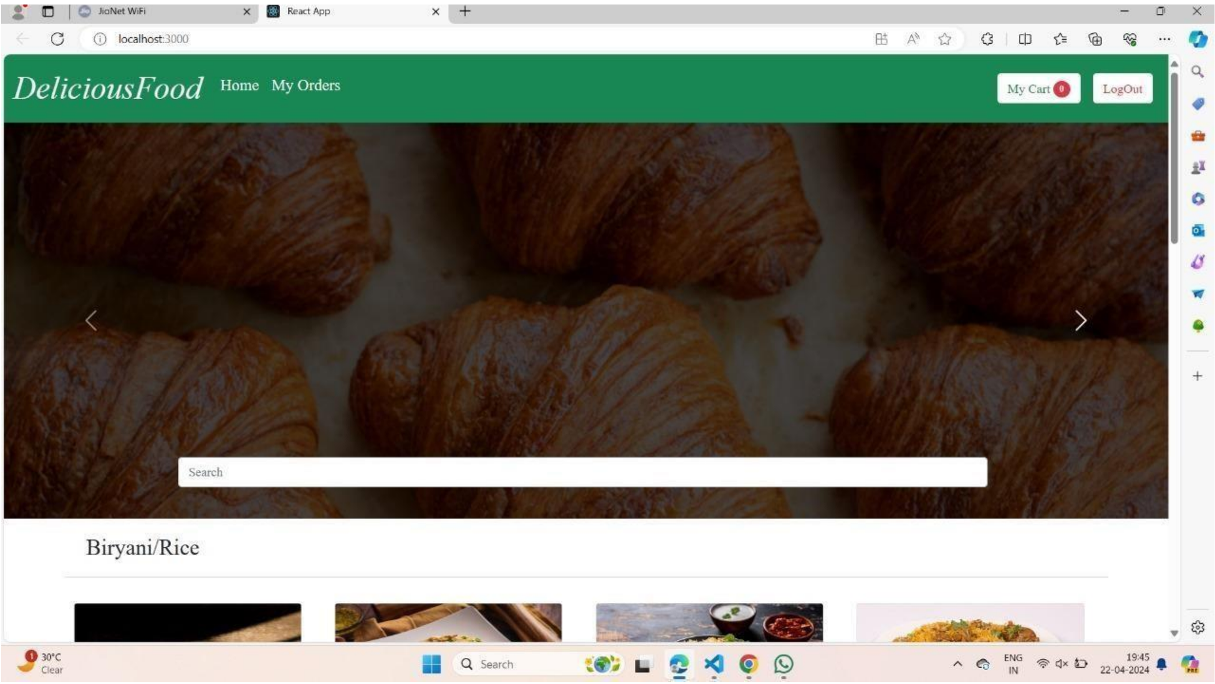
4.3 Level 2 DFD



Level 2 DFD

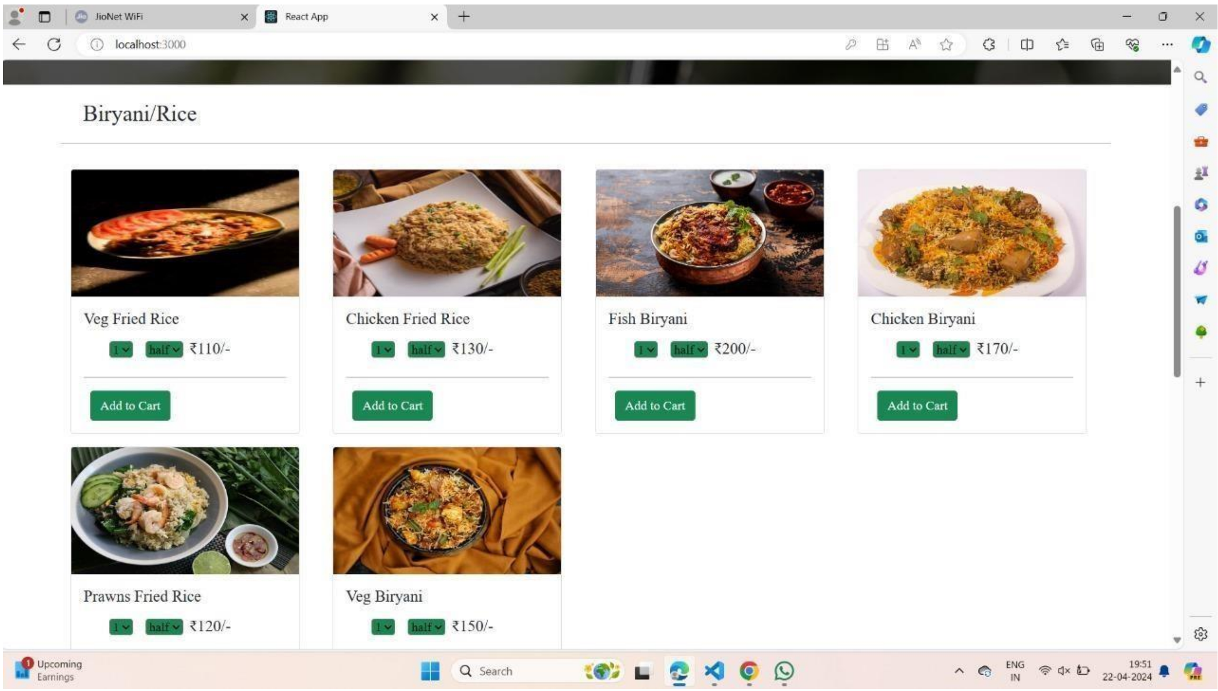
OutLook

1.



Biryani/Rice

2.



Biryani/Rice

Veg Fried Rice
1 half ₹110/-
Add to Cart

Chicken Fried Rice
1 half ₹130/-
Add to Cart

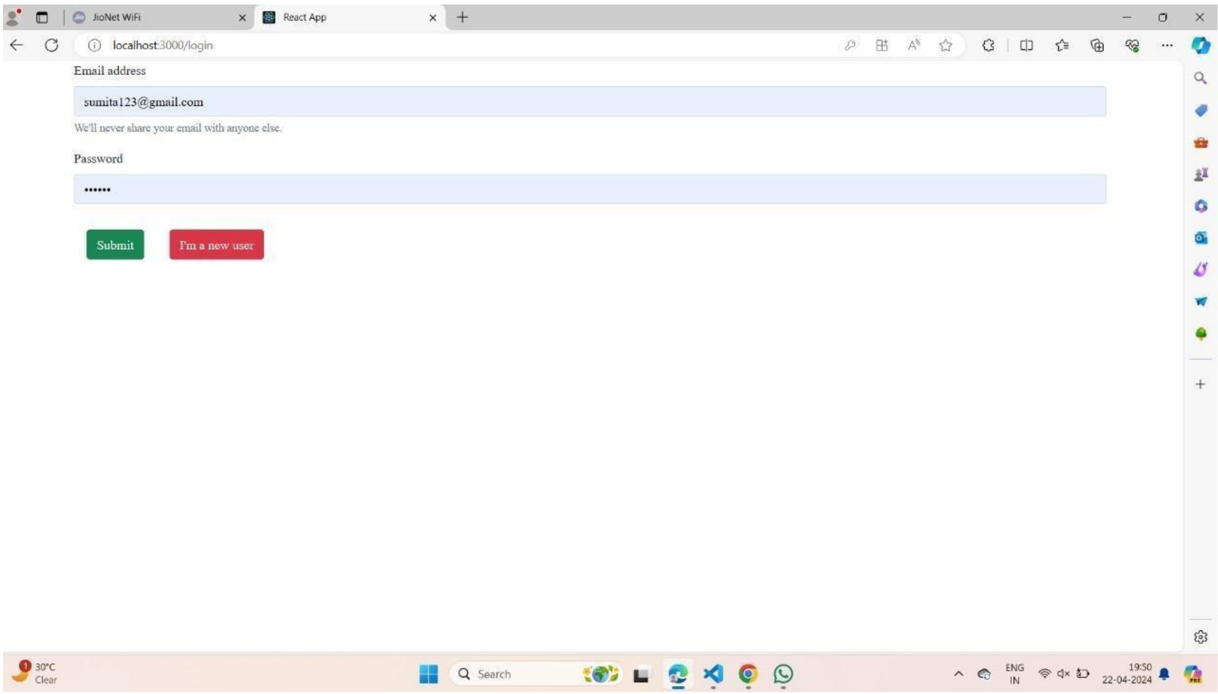
Fish Biryani
1 half ₹200/-
Add to Cart

Chicken Biryani
1 half ₹170/-
Add to Cart

Prawns Fried Rice
1 half ₹120/-

Veg Biryani
1 half ₹150/-

3.



Email address

sumita123@gmail.com

We'll never share your email with anyone else.

Password

••••••

Submit

I'm a new user

8.GITHUB LINK

Link:

https://github.com/Abhishek6725/OrderFood-Backend-Project