****

**Online Food Delivery System- “Food Sewa”**

Manyata Roka, Kritan Shrestha, Sandesh Shahi, Subhayu Jung Thapa

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST5009CEM Software Development

Sandis Prajapati

May 9, 2025

**Table of Contents**

**Table of Figures**

# Introduction

In the digital age, where accessibility, speed, and ease of use are crucial, the meal delivery sector has grown significantly. The modern consumer seeks effective ways to satisfy their appetites without sacrificing comfort or time. We are thrilled to present Food Sewa, a cutting-edge, responsive, and user-friendly web application that aims to transform the online meal delivery experience after realizing this need.

By providing a smooth platform where users can browse menus, place orders, make safe payments, and follow the progress of their food deliveries in real time, Food Sewa seeks to close the gap between customers and nearby dining establishments. The word "Sewa," which means "service," embodies our goal of offering a dependable and fulfilling meal delivery experience via a skillfully designed digital service.

For this project, our team has decided to use React as the primary front-end technology. We can design a highly dynamic and interactive user interface with React's component-based architecture, which guarantees quick page loads, real-time changes, and seamless cross-device navigation. Additionally, React allows us to create reusable components, which improves code maintainability and scalability as the platform expands.

Modern online standards, such as accessibility, performance optimization, and mobile responsiveness, are being prioritized in the platform's development. Whether a consumer uses a smartphone or a laptop to place an order, Food Sewa guarantees a consistent and enjoyable experience. We intend to incorporate safe APIs and effective data processing technologies on the backend to facilitate user profiles, order histories, restaurant listings, and payment processing.

In addition to being useful, Food Sewa prioritizes helping neighborhood eateries and food vendors by providing them with a platform to increase their visibility. In addition to helping the businesses, this allows customers to choose from a wider range of cuisines and eating options with just a few clicks.

Through teamwork, effective project management, and cutting-edge web technologies, our team is dedicated to creating a platform that is dependable, user-focused, and prepared for the future. Food Sewa is a step toward wiser digital service in the food industry, not merely a website for food delivery.

# Aim

In addition to giving local restaurants an effective way to handle their orders and expand the customers they serve, the Food Sewa project aims to design and develop a cutting-edge, responsive, and user-friendly online food delivery web platform using React that allows users to conveniently browse menus, place food orders, make secure payments, and track deliveries in real time.

# Objective

The objectives of the Online Food Delivery System (Food Sewa) are:

1. Use React to Create and Develop a Responsive Web Platform  
   Using React, create a cutting-edge, responsive online application that guarantees a fluid, dynamic, and mobile-friendly user experience on a range of devices.
2. Put in place safe user and restaurant management  
   Provide restaurants the opportunity to create and maintain their listings, update menus, and track orders through a dedicated dashboard, in addition to facilitating user registration, login, and profile management.
3. Create a Simplified System for Ordering and Checkout  
   To improve customer satisfaction, incorporate an easy-to-use food ordering system with features like a shopping cart, customizable menu options, safe payment processing, and real-time order tracking.
4. Assure privacy, data security, and feedback systems  
   Use online security best practices to protect user information and transactions while letting clients provide evaluations and ratings to preserve openness and high standards of service.
5. Build an Architecture That Is Scalable and Maintainable  
   Scalability should be considered while designing the platform to accommodate upcoming additions like mobile app development, delivery partner integration, and loyalty programs.

# Problem Statement

Even though the market for online meal delivery services is expanding quickly, there are still a number of enduring issues. Poor user interfaces, sluggish loading speeds, and challenging navigation plague many of the platforms currently in use, particularly on mobile devices. Because of overloaded menus and insufficient filtering options, users frequently find it tedious to search for their favorite meals or restaurants. Furthermore, local or smaller eateries find it difficult to get noticed on large platforms, which are usually dominated by well-known food chains.

The absence of transparency and real-time updates in order tracking is another major problem. Uncertainty over delivery schedules is a common problem for customers, which can cause discontent and recurrent customer support inquiries. Additionally, some platforms charge unreasonable commission costs, which lower partner restaurants' earnings and hinder future cooperation.

In order to overcome these obstacles, Food Sewa provides a cutting-edge, user-friendly, and performance-enhancing online platform constructed with React. Users will benefit from quicker load times, easier navigation, and an all-around better ordering experience across devices with a responsive and clean design. Throughout their transactions, users will benefit from transparency and confidence thanks to real-time tracking capabilities and order progress updates.

Food Sewa provides a low-barrier-to-entry platform that makes it simple for neighborhood eateries to handle their menus, take orders, and expand their clientele. Secure payment integration and data protection features will boost user trust, while the admin dashboard and simplified order management system will aid in increasing operational efficiency. Food Sewa wants to provide a more intelligent, inclusive, and fulfilling food delivery experience for patrons and establishments alike by resolving these fundamental issues.

# **Functional Requirement**

1. User Authentication
   * Users must be able to register and log in securely using email and password.
   * Passwords should be encrypted and stored securely.
   * Users should be able to recover their password via email or security question.
2. User Dashboard
   * Users can view their profile, order history, saved addresses, and current orders.
   * Users can update personal information and change passwords.
3. Restaurant Management
   * Restaurants can register/login and manage their profile.
   * Restaurants can add, edit, or remove food items from their menu.
   * Restaurants can manage orders (accept, prepare, complete, cancel).
4. Food Browsing and Ordering
   * Users can browse restaurants and filter by cuisine, rating, or availability.
   * Users can view menu items, add them to a cart, and customize selections (e.g., quantity, toppings).
   * Users can place an order and receive confirmation.
5. Cart and Checkout System
   * Users can review and update items in the cart before checkout.
   * Users can choose a delivery address and payment method.
   * The system calculates total cost including taxes and delivery fees.
6. Payment Integration
   * Secure online payment processing using a third-party API (e.g., Esewa, Khalti.).
   * Payment confirmation and receipt generation.
7. Order Tracking
   * Users can view real-time status updates (e.g., preparing, out for delivery, delivered).
   * Restaurants can update order statuses accordingly.
8. Review and Rating System
   * Users can rate restaurants and leave feedback after order completion.
   * Reviews are visible to other users.
9. Admin Dashboard
   * Admin can manage users, restaurants, and orders.
   * Admin can approve or deactivate restaurant listings and monitor platform activity.
10. Notification System

* Send order confirmations, status updates, and promotional messages via email or in-app alerts.

# Non-Functional Requirement

1. Performance

* The application should load in under 3 seconds on a standard internet connection.
* The system should handle at least 100 simultaneous users without noticeable performance degradation.

2. Scalability

* The system architecture should support easy scaling to accommodate growing users, restaurants, and data.
* React’s component-based structure and backend API design should support horizontal scaling when needed.

3. Reliability and Availability

* The platform should maintain 99.9% uptime to ensure continuous access to users and restaurant partners.
* In case of server failure, the system should recover quickly using backup and failover mechanisms.

4. Security

* All user data, including login credentials and payment information, must be securely encrypted.
* The system must implement protection against common attacks like SQL injection, XSS, and CSRF.
* Role-based access control should be enforced (user, restaurant, admin).

5. Usability

* The platform should offer a clean, intuitive, and responsive interface for both customers and restaurant partners.
* It must be easy to navigate, with clear call-to-actions and accessible design for users of all technical levels.

6. Maintainability

* The codebase should follow best practices, be modular, and well-documented to simplify future updates and debugging.
* Technologies like React and version control tools (e.g., Git) will ensure easier collaboration and ongoing improvements.

7. Portability

* The web app should be accessible on all modern browsers (Chrome, Edge, Firefox, Safari) and devices (desktop, tablet, mobile).
* It should be easy to migrate to cloud hosting environments or different platforms if required.

8. Localization Support (Optional/Future Scope)

* The system should be flexible enough to support multiple languages and regional settings in future releases.

# Methodology

## Agile

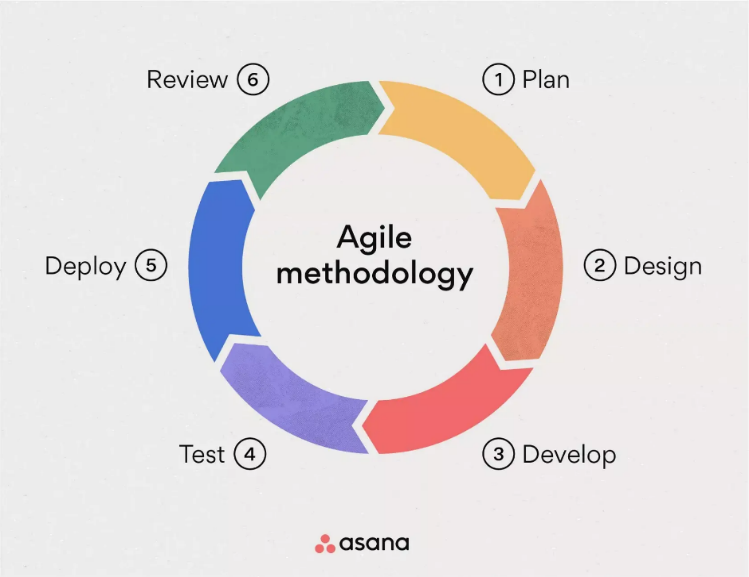
The incremental and iterative nature of these agile methodologies in software development can be adjusted within the specifications so that they are able to meet the demands of the client. It provides iterative development together with adaptive planning.

Boxing and time. The Agile approach also holds the analysis of requirements, design, coding, testing, partial implementation of software but it only waits for customers' feedback at each process, Maximum customer satisfaction is the prime concentration throughout the whole process. The time used in development gets sped up accordingly. Sarkar, Gupta and Sharma 2012.

[(Laoyan, 2024).](#first)

**Figure 2:**

Agile infographics



## Reasons for using Agile Methodology

* Iterative Development: Agile breaks down projects into small pieces of known as sprint rapid delivery to make the functional component deliverable as soon as possible.
* Flexibility: Teams easily adapt to changes due to requirements and priority based on user feedback and market conditions
* Stronger collaboration: Agile encourages close collaboration within cross-functional teams; hence communications and transparency via regular standup meetings and sprint reviews.
* Improved Quality: Continuous testing and integration coupled with development contribute a lot to high quality with minimal defects.
* Customer Involvement: Since the stakeholders have already been a part of the development process right from the beginning, it comes out as expected by the users.
* Risk Reduction: Early detection through feedback and subsequent adjustment minimize the risks of project failure.
* Improved Satisfaction: Emphasize the delivery value to the customer to increase overall satisfaction in the end product.
* Team Morale: Teams make sure morale is brought out, increasing ownership of projects, hence improving performance in the long run[. Rudder (n.d.)](#h)

# Team Roles and Responsibilities (Based on Scrum Methodology)

🔹 Subhayu Jung Thapa – Backend Developer & Scrum Master

* Role in Scrum: Scrum Master & Development Team Member
* Responsibilities:
* Facilitate Scrum ceremonies such as sprint planning, daily stand-ups, sprint reviews, and retrospectives.
* Remove obstacles that may hinder the team’s progress and promote an Agile mindset within the group.
* Coordinate backend development tasks including API development, user authentication, and data management.
* Contribution to Success:
* Acts as the team’s guide in applying Scrum principles, ensuring smooth collaboration and continuous progress. Also plays a key role in building the backend foundation for the project.

🔹 Sandesh Shahi – Backend Developer

* Role in Scrum: Development Team Member
* Responsibilities:
* Share responsibility for backend development tasks including database design and integration.
* Optimize backend performance, security, and API responsiveness.
* Assist in setting up server environments and deployment processes.
* Contribution to Success:
* Ensures the reliability and efficiency of the application’s backend operations, working closely with other team members to meet backend-related sprint goals.

🔹 Kritan Shrestha – Frontend Designer and Tester

* Role in Scrum: Development Team Member
* Responsibilities:
* Design and implement interactive UI components using React.
* Ensure frontend responsiveness and functionality across devices.
* Write test cases and perform functional testing on developed features.
* Collaborate with UI/UX designers to maintain consistency in design.
* Contribution to Success:
* Bridges development and testing to ensure the app is visually appealing and functionally correct, delivering a reliable user experience.

🔹 Manyata Roka – Frontend Designer and UI/UX Designer

* Role in Scrum: Development Team Member
* Responsibilities:
* Design user-friendly and attractive interfaces following UI/UX best practices.
* Create wireframes, prototypes, and final visual designs for the platform.
* Collaborate with frontend developers to translate design into code.
* Conduct usability testing and iterate based on feedback.
* Contribution to Success:
* Ensures the platform is intuitive, visually consistent, and engaging, directly influencing user satisfaction and platform usability.

# Tools and Technologies

The Online Food Delivery System (Food Sewa) uses following tools and technologies for the project:

## Technologies

***Figure 3:***

Technologies used

# 

## Tools

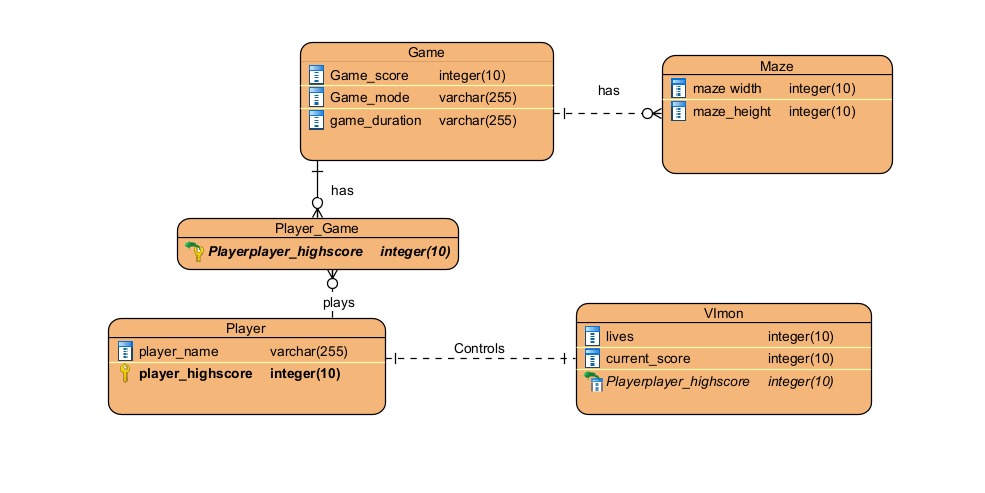
**Figure 4:**

Tools used

# 

# Conceptual Diagram

## ER Diagram



**Prototype**

The prototype showcases the core gameplay mechanics, including maze navigation, item collection, and enemy avoidance. [Kirvan (2023)](#f)

Presented below are both high-fidelity and low fidelity prototypes that shows the various aspects of game in different stages.

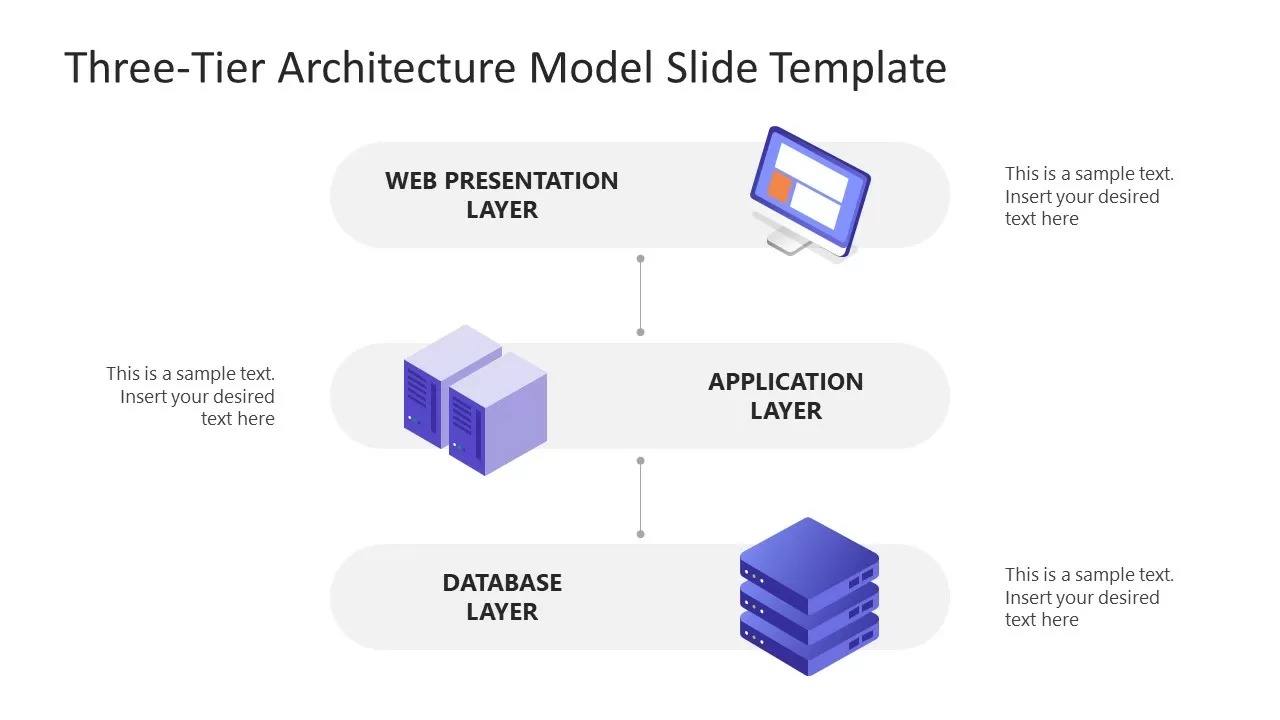
# System Architecture

Our Online Food Delivery System follows a structured system architecture to ensure scalability, maintainability, and efficiency. The system is divided into three main tiers:

* 1. Presentation Layer (Frontend)
* Built using React
* Handles UI, routing, state management (e.g., using Redux, Context API)
* Interacts with the backend through REST APIs
* Responsive design for desktop/mobile access
  1. Application Layer (Backend/API Layer)
* Handles business logic, authentication, and request routing
* Built using:
* Node.js + Express.js (if staying in JS ecosystem)
* Connects with the database
* Implements APIs for order management, user sessions, restaurant data, etc.
  1. Data Layer (Database)
* Stores user profiles, restaurant data, menu items, orders, and transactions
* Use:
* MongoDB for flexibility with unstructured data
* or MySQL/PostgreSQL for relational, structured data
* Add a caching layer (e.g., Redis) for performance if needed

[(*Search Results for “Three Tier Architecture” – MUNICIPALITY OF LUPON*, n.d.)](#d)

The modular approach gives flexibility for easy updating and expanding Food Sewa in the future



# MVC Pattern

## MVC Architecture

Food Sewa is designed based on the MVC architectural pattern that separates the concerns within the application for better clarity and scalability. The three components of MVC are:

### Model:

Responsible for retrieving, storing, and managing data related to:

* Users
* Restaurants
* Menus
* Orders
* Payments

### View:

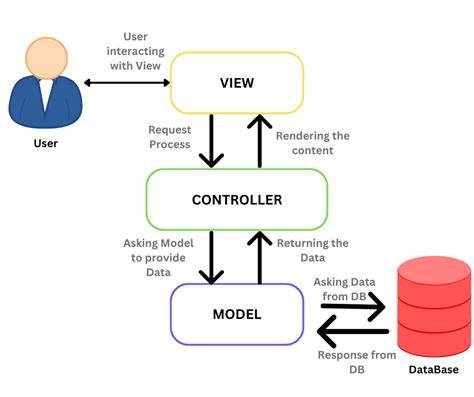
 Represents the frontendor UI.

 Since you are using React, this is your view layer.

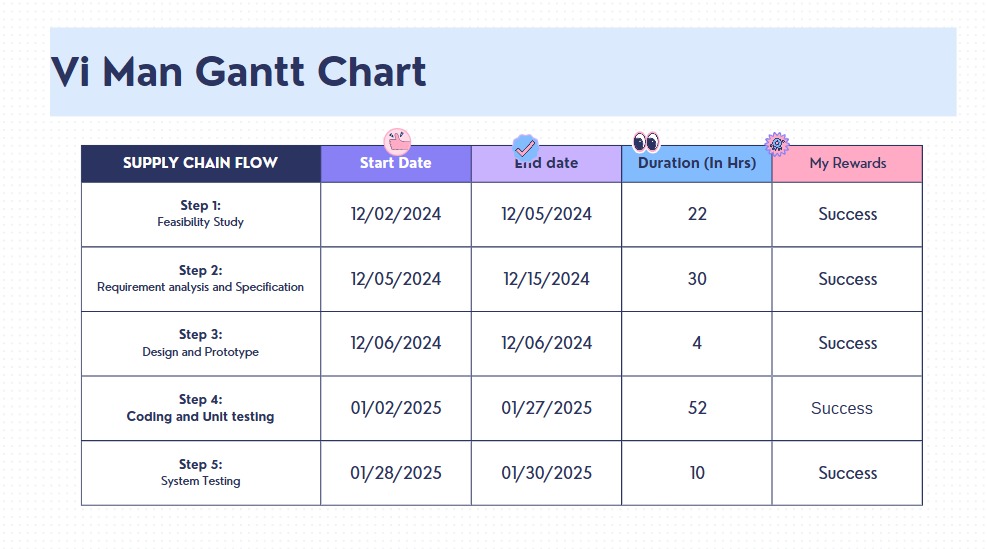
 It displays the data to users and captures user interactions.

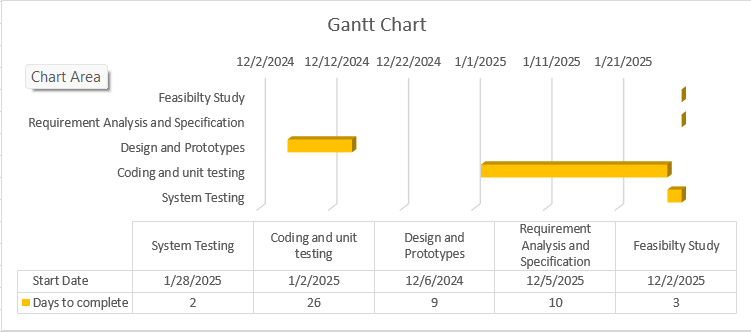
 React fetches data from controllers (via API endpoints) and renders dynamic components accordingly.

### Controller:

Between View and Model, the Controller acts as a mediator. It interprets player input, including movement commands, modifies the Model appropriately, and alters the display as a result. [Sadika (2023)](#e) 

# Project Plan

Project planning refers to the phase in project management in which you determine the actual steps to complete a project. This includes laying out timelines, establishing the budget, setting milestones, assessing risks, and solidifying tasks and assigning them to team members [Staff (2024)](#b). 



****

# Risk And Challenges

* Technical Complexity

Risk: Bugs or performance problems could arise from integrating several systems (frontend, backend, database, payment gateway, and real-time tracking).

Mitigation: Clear documentation, frequent testing, and modular development will aid in complexity management.

* Deadlines and Time Management

Risk: Not finishing all scheduled features by the deadline because of unforeseen delays or underestimating workloads.

Mitigation: Prioritize MVP features first, use Scrum sprints with explicit deliverables, and monitor progress with daily stand-ups.

* Team Coordination

Risk: Confusion and a slowed pace of work might result from poor communication or an ambiguous responsibility allocation.

Mitigation strategies include using collaborative platforms like Trello, Git, or Slack, holding regular meetings, and clearly defining roles—as you have previously done.

* Data and Security Privacy

Risk: If sensitive data, such as payment details and user credentials, is handled carelessly, it may result in vulnerabilities.

Mitigation strategies include using HTTPS, encrypted passwords, secure third-party payment APIs, and secure authentication (like JWT).

* Third-Party Dependency Failures

Risk: Dependency on services that could fall down or have policies changed, such as hosting platforms, payment gateways, or geolocation APIs.

Mitigation strategies include using dependable and well-documented services, keeping an eye on uptime, and having contingency preparations (such as alternative APIs or caching).

* Problems with Scalability

Risk: The system may slow down or crash as the number of establishments and users increases.

Mitigation: Make use of scalable technologies (such as cloud-hosted databases, React, and Node.js) and plan for future growth.

* Cross-Device Compatibility

Risk: Not all devices and screen sizes will experience the UI/UX as well.

Mitigation: Throughout development, create responsive layouts and test them on a range of devices and browsers.

* User Adoption and Retention

Risk: Due to a subpar user interface, a dearth of functionality, or competition in the market, even a well-designed app may have trouble gaining or keeping users.

Mitigation: Pay attention to fundamental features, a seamless user experience, reliable performance, and user input for enhancements.

# Conclusion

The Food Sewa project seeks to provide an online meal delivery service that links consumers with a range of nearby eateries in a way that is user-friendly, effective, and aesthetically pleasing. We are creating a system that provides secure transactions, real-time ordering, and a smooth user experience by utilizing contemporary web development technologies, namely React for the frontend and a safe, scalable backend. This technology is intended to help businesses increase their online presence in addition to making food ordering easier.

Our team's roles and responsibilities are well-defined thanks to the Scrum approach, which encourages cooperation, accountability, and ongoing development. The goal of each sprint is to produce useful modules that lead to a finished, integrated application.

We have described mitigation measures to proactively handle each of the possible issues, including integration complexity, tight timeframes, and security maintenance. Food Sewa will distinguish itself as a dependable, scalable, and user-friendly solution in the expanding online meal delivery market by placing a high priority on both the user experience and technical robustness.

To sum up, Food Sewa is more than simply a project; it represents a step toward innovation in regional food services by fusing a thorough comprehension of customer demands with technical prowess. We have no doubt that this platform will be a useful tool in the digital food economy, benefiting both users and business partners.

# References

Laoyan, S. (2024, February 2). What is Agile Methodology? (A Beginner’s Guide) [2024] • Asana. *Asana*. <https://asana.com/resources/agile-methodology>

Staff, C. (2024, November 14). *What is project planning? benefits, tools, and more*. Coursera. <https://www.coursera.org/articles/project-planning>

Solutions, G. (2024, July 12). Functional and Non-Functional Requirements - with examples | Medium. *Medium*. <https://medium.com/@growsolutions/functional-and-non-functional-requirements-the-ultimate-checklist-with-examples-cde16aba33d7>

*Search Results for “Three tier architecture” – MUNICIPALITY OF LUPON*. (n.d.). <https://lupon.gov.ph/?s=Three+tier+architecture>

Sadika. (2023, September 19). The MVC Architecture - Sadika - Medium. *Medium*. <https://medium.com/@sadikarahmantanisha/the-mvc-architecture-97d47e071eb2>

Kirvan, P. (2023, April 13). *prototype*. Search ERP. <https://www.techtarget.com/searcherp/definition/prototype>

Rudder, C. (n.d.). *Agile project management: 10 reasons to use it*. The Enterprisers Project. <https://enterprisersproject.com/article/2019/9/agile-project-management-10-reasons-to-use>