The Recspicy Index

Design Document

Team #1

James McGuigan

Steven Forester

SUNY College at Old Westbury

[**1. Introduction 4**](#_unn1lxhfo074)

[**2. Problem Statement 5**](#_o4m10gw17e98)

[**3. Assumptions, Constraints, and Risks 6**](#_84p8wixbpgsc)

[Assumptions: 6](#_ufen4g6v24nf)

[Constraints: 6](#_ip9ozf5rmcmu)

[Risks: 7](#_8fg996l20bkv)

[**4. System Overview 8**](#_fnyn03a8dz95)

[Recspicy Tech. stack: 8](#_20t6qbypyffa)

[**5. Detailed Design - Database Models and Design 8**](#_901o0w9bsp9)

[Data Model: 8](#_8os5sqsbic1z)

[The User Model: 9](#_ts261hzha0c)

[Recipe Model: 11](#_nx236h2bxemn)

[Meal Plan Model: 12](#_ykv8kdcwj8cn)

[Meal Model: 13](#_mkhri9twruea)

[**6. User Interface - Structure and Navigation 14**](#_figse8kkxlp)

[Navigation Bar & Key Sections 14](#_lhwlsbbi8iej)

[Additional UI Features 14](#_lxylcplq0tlf)

[Wireframe 1: Home Page 15](#_sgsyr4vwn1ag)

[Wireframe 2: Recipe Detail Page 15](#_dnirf9w2q1t2)

[Wireframe 3: Meal Planning Interface 16](#_ot2l0phrx9fh)

[**7. API Development 16**](#_jgpxc76cwrps)

[**API Functions 16**](#_zdjz9cybzn01)

[External API Integration 17](#_3sx5r4i2qp5u)

[Performance & Security 17](#_cae58dvwvfsq)

[**8. Search Engine and Filtering 18**](#_569wq93aqpts)

[Search Parameters 18](#_ld9juhkkhmj5)

[Search Optimization 19](#_r9mv4q8722e4)

[**9. Meal Planning System 19**](#_qijeafmxle1q)

[Meal Plan Features 20](#_rwsmmvmtzaox)

[Enhancements 20](#_73zuoa365p7b)

[**10. Admin Controls and Moderation 20**](#_vbpxw2p3pyf2)

[Moderation 21](#_vz9sfqy9kbaz)

[Analytics & Monitoring 21](#_lolzyzr0xudd)

[**11. Non-Functional Requirements 22**](#_eu91ghtg2kpw)

[Scalability 22](#_ws4u5d7i4ujk)

[Security 22](#_d5peo92jdsr5)

[Performance 23](#_uknyfemh1b9y)

[**12. Functional Requirements 23**](#_kgffz7r9obkt)

[User Authentication & Profiles 23](#_o9iljm5woydb)

[Recipe Management 23](#_oz9gxdo1kjf8)

[Meal Planning System 24](#_yo724tjd9rm2)

[Admin & Moderation Controls 24](#_nqqfwez9shhx)

[**13. Testing Strategy 25**](#_7zb4ij1pna8f)

[Unit Testing 25](#_n9ce68fdc75y)

[Integration Testing 25](#_wk081kz6isqm)

[UI Testing 26](#_ffjuz8pml8ac)

[**14. Appendix & References 26**](#_hrv4c58oavjj)

[Appendix 27](#_luhbizz775c)

[References 27](#_er6k4g3tghxn)

## 1. Introduction

Recspicy is a web-based platform designed for recipe discovery, meal planning, and community-driven content sharing. Our backend is built from API-based recipe data and user-generated content that allows users to create, modify, and share meal plans. Our design is built to focus on simplicity and functionality for the user so we want to allow the person to experience ease in navigation and utility.

Who is this for? We've designed this for food enthusiasts, at home cooks, and individuals looking to easily build their meal planning process within our system. What is this? a platform combining recipe discovery, creation, and advanced meal planning features. Where is this available? This is accessible as a web application usable on any browser. Why have we built this? existing platforms lack unique filtering and meal planning tools making recipe creation, profile management, and meal discovery inefficient and difficult.

## 2. Problem Statement

Existing recipe management platforms struggle to provide essential features such as meal planning tools, advanced search capabilities, and easy-to-use interfaces for dietary needs. Users on other platforms will struggle with disorganized recipe searches, limited filtering options, and no meal planning integration. Recspicy’s aim is to resolve these issues by simplicity and design of our platform for efficient meal organization, creation, and discovery.

Who is affected by this problem? anyone hoping for an easy time looking at recipes, planing meals, or even just dietary tracking. What is our solution? A structured recipe and meal planning system designed to address the current limitations in competitor platforms. When is the solution applicative? Our solution is useful and anytime that meal planning or recipe discovery is desired, or whenever a user has questions about nutrition. Why not pick a competitor? Current competitors do not provide meal organizations alongside tailored filters which can create difficulty in use leading to user frustration.

## 3. Assumptions, Constraints, and Risks

### Assumptions:

Users want to easily find, store, and organize recipes. The platform will be scalable to accommodate growing user contributions. Users are actively looking for an efficient and structured way to discover, save, and organize their favorite recipes. The platform is designed with the expectation that users will appreciate the integration of meal planning tools alongside recipe discovery, helping them create well-balanced meal plans with ease. Users prefer a personalized experience, including the ability to filter recipes based on dietary restrictions, cooking time, and ingredients. From a technical standpoint, the platform assumes scalability will be a necessity, meaning the system must be able to accommodate an increasing number of users and a growing recipe database without performance issues.

### Constraints:

The platform must be responsive and work across various browsers. API rate limits and data availability may restrict functionality. Recspicy operates within several constraints that impact its functionality and development. The platform must maintain a responsive design, ensuring that it is intended to work seamlessly across various screen sizes, from desktops to mobile devices. Our Platform relies on external data sources, such as the Tasty API, which may impose rate limits and availability restrictions, affecting how often users can access new recipes. Security and authentication mechanisms are also a constraint, requiring implementations to protect user data while maintaining their accessibility. Development time and resources further limit the project's scope, requiring careful prioritization of features to deliver a functional and user-friendly system within a reasonable time frame.

### Risks:

Security risks related to user data privacy. System performance issues as database size increases. Risks must be managed to ensure Recspicy operates effectively, efficiently and securely. Another risk is data security, as user authentication, personal preferences, and saved meal plans must be protected from unauthorized access or breaches. The reliance on external APIs presents a type of risk, if an API goes offline or changes its structure, Recspicy’s recipe discovery functionality could be disrupted.

Another potential issue is system performance, the platform scales and more users contribute their own recipes, database efficiency and query optimization will be necessary to work out in preventing slow load times or crashes. User engagement is also a minor risk factor, a lack of compelling features or a difficult onboarding process could result in low uptimes and user usage rates as well as new users adapting to the platform. Moderation of user-generated content is another major challenge, recipes should meet quality standards and be free of inappropriate content.

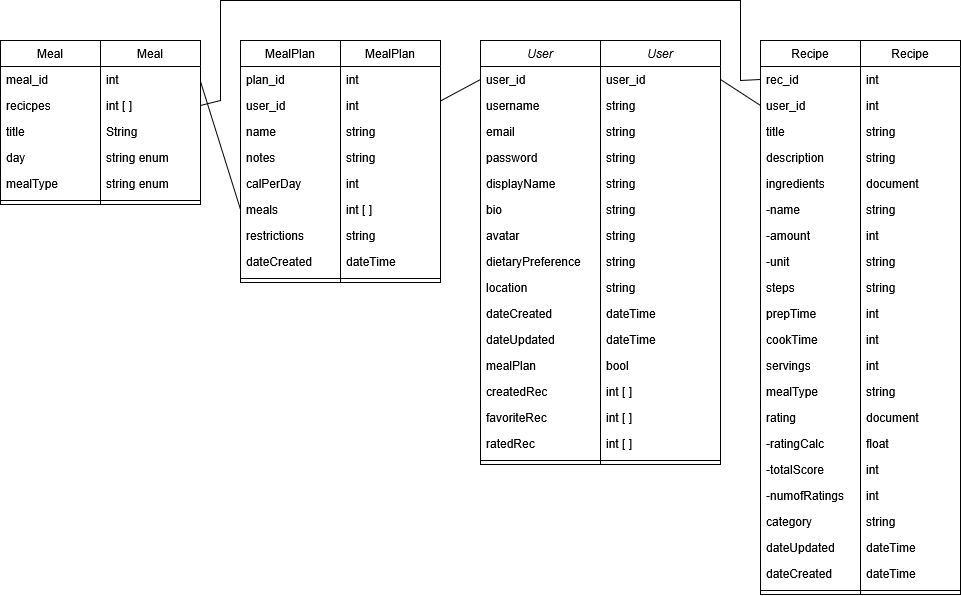
## 4. System Overview

### Recspicy Tech. stack:

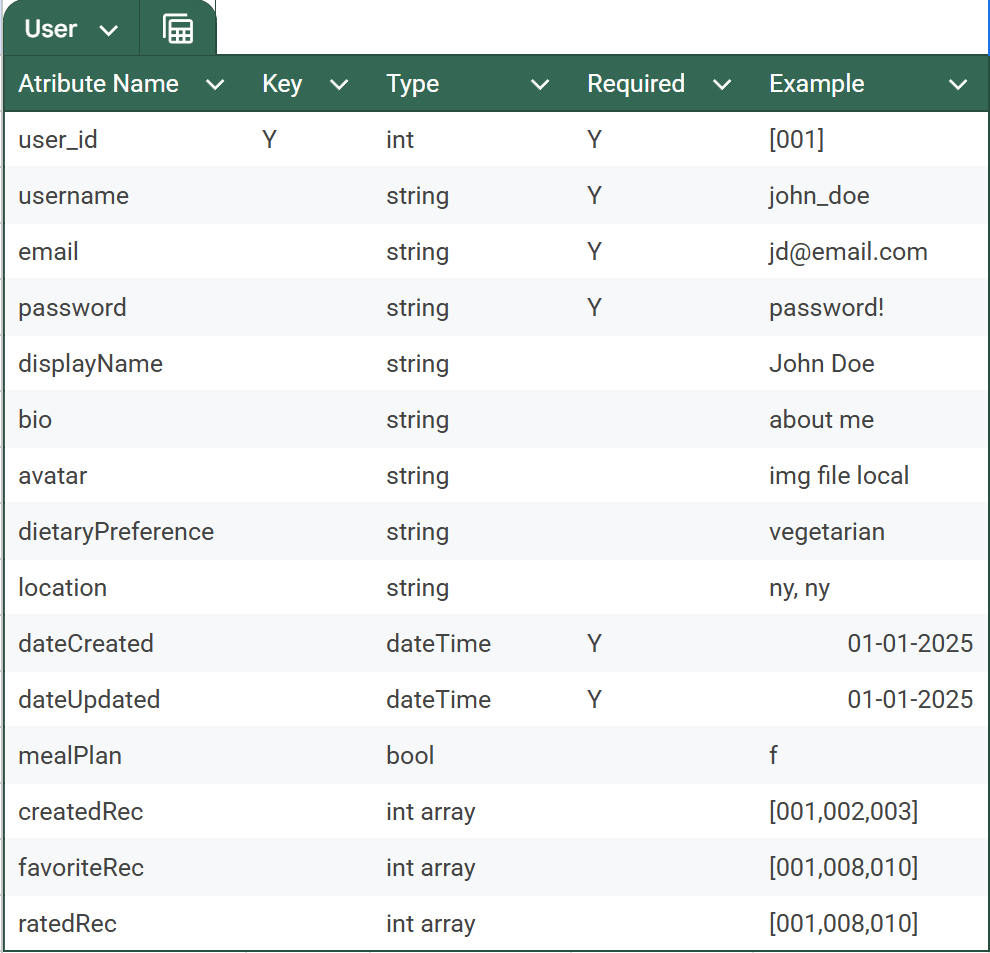
* **Frontend**: HTML5, CSS3, JavaScript (React.js)
* **Backend**: Node.js with Express.js
* **Database**: MongoDB for storing recipes, user profiles, and meal plans
* **Authentication**: JWT-based authentication system
* **External** API: Tasty API for recipe data

## 5. Detailed Design - Database Models and Design

### Data Model:



### The User Model:



Our user model is responsible for storing and managing user-related data. It includes essential authentication credentials such as email, username, and a securely hashed password, ensuring account security. The user model stores user preferences, including dietary restrictions, and favorite cuisines, allowing us to provide a personalized experience.

Users may also have profile-related metadata, such as profile pictures, account creation dates, and activity logs, which enhance user engagement and account management. The system must support authentication with traditional login methods as well as OAuth which is Google Sign-In for ease of access. To protect user data, JWT-based authentication is implemented and a secure password hashing technique like bcrypt will be used to ensure credentials remain encrypted and private.

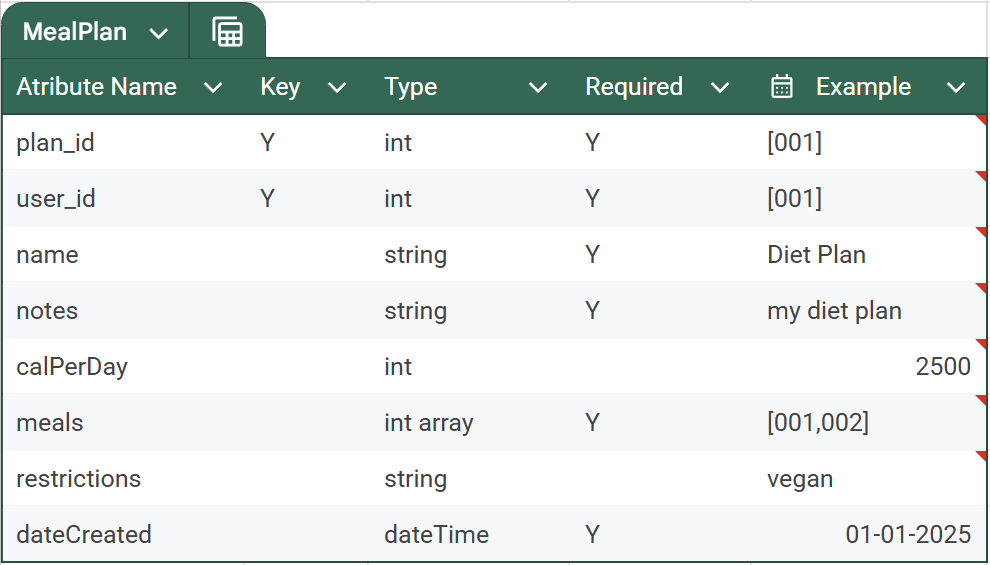
### Recipe Model:



Our recipe model includes recipe details such as ingredients, preparation steps, and ratings. The Recipe Model serves as the core database structure for storing and organizing recipe-related data. Each recipe entry includes details like title, description, ingredients, and preparation steps, which means users have all necessary information to follow a recipe accurately. The model supports optional metadata such as cooking time, serving size, and nutritional information, which allows users to make informed meal choices. User-generated recipes are stored with API-sourced recipes creating a diverse collection of recipe options.

This system lets users interact with recipes through features like rating, and favoriting to establish credibility and popularity rankings which we can use on our end to create featured filtering systems and best recipes. Each recipe is linked to its creator, allowing users to manage, update, or delete their submissions. Tags and categorization such as cuisine type, dietary labels, and allergies are used to improve searchability and filtering within the present system.

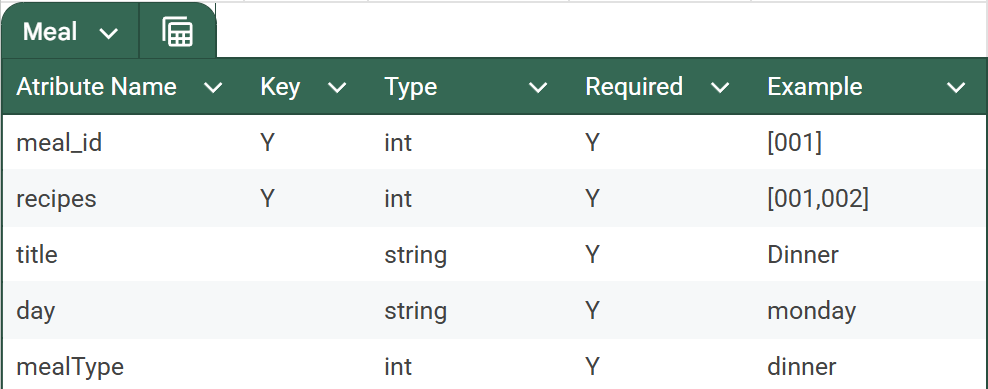
### Meal Plan Model:



The Meal Plan Model allows users to create structured meal plans linked to recipes. The Meal Plan Model is designed to help users organize their meals for a specified time, typically a week. Each meal plan entry contains references to the users created recipes, allowing users to structure their daily meals easily and efficiently.

Users can add breakfast, lunch, dinner, and snack options for each day of the week, adjusting their plans to dietary needs or fitness goals. This model will also use nutritional tracking, aggregating calorie counts and macronutrient value estimates from selected recipes to help users maintain balanced diets. Users can create, modify, or delete meal plans at any time, providing flexibility in meal plan organization and management.

### Meal Model:



The meal model manages saved recipes for easy retrieval in meal plans. The meal model also allows users to save and quickly access their saved meals from their profile. This model uses id’s to link a user’s meals to their meal plans, this allows them to build a personalized collection without modifying content. Users can add or remove recipes from their meals. We are doing it this way so the simplicity and convenience of this works in the users favor as a way to revisit preferred meals.

The model supports functionalities like categorizing meal into meal plans on the user's profile for better organization. Meal Plans influence the platform's recommendation system over time, helping current and new users discover dishes based on their preference.

## 6. User Interface - Structure and Navigation

The Recspicy User Interface is designed for simplistic navigation, this intention is for new users to quickly access features through an easy to navigate and well structured layout. The top navigation bar remains consistent across all accessible pages allowing for seamless transitions between sections.

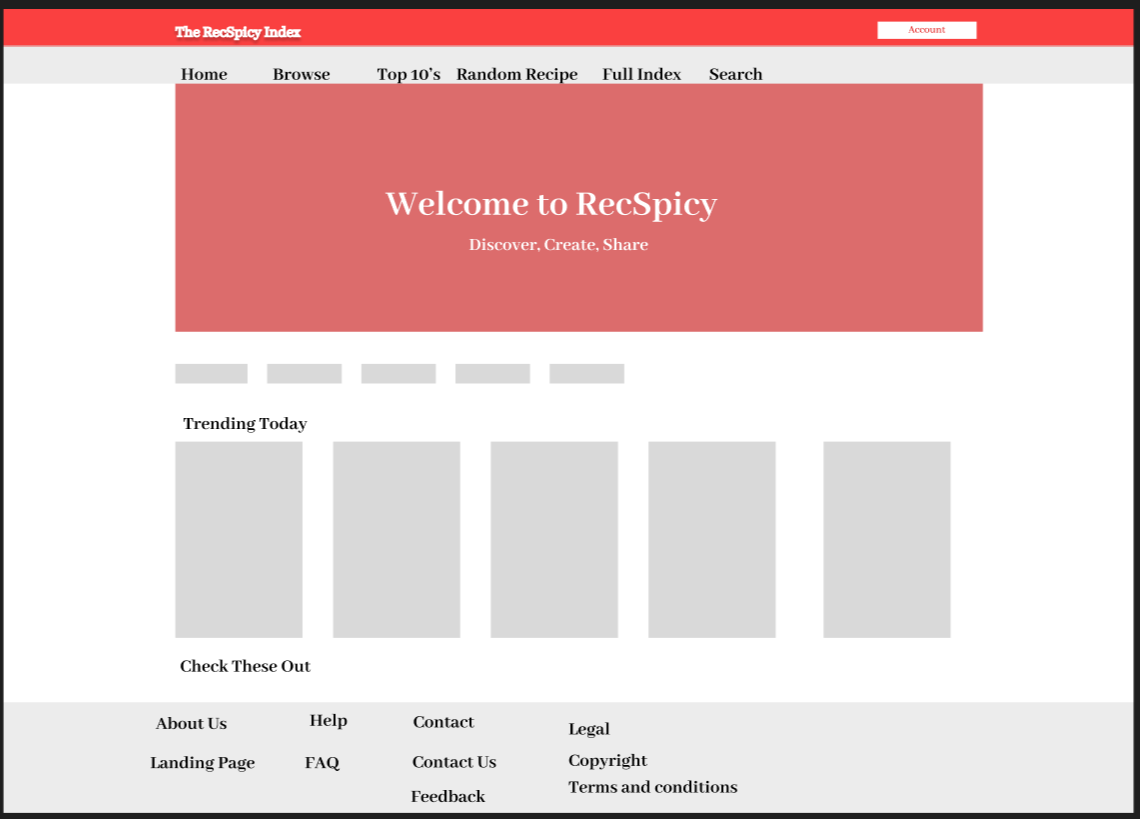
### Navigation Bar & Key Sections

* **Home** – Displays trending recipes, featured collections, and a quick search bar.
* **Recipe Discovery** – Allows users to explore recipes with advanced filtering options.
* **Meal Planning** – Allows users to create, modify, and track meal plans.
* **User Profile** – Stores user details, favorite recipes, and personalized settings.

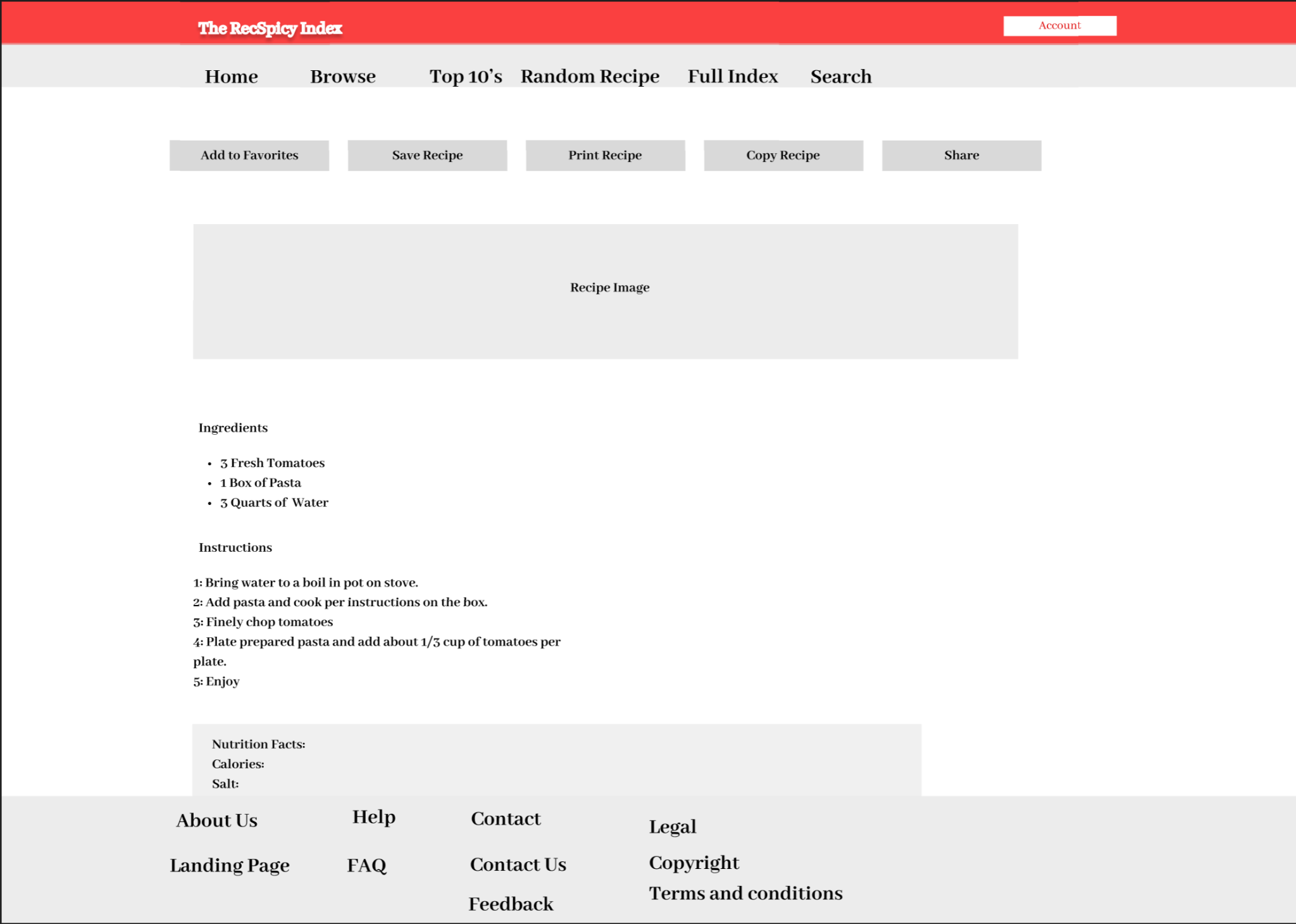
### Additional UI Features

* **Sticky Navigation** – Keeps the menu accessible while scrolling.
* **Search & Filters** – Allows users to refine recipes queries by ingredients, cooking time, and dietary needs.

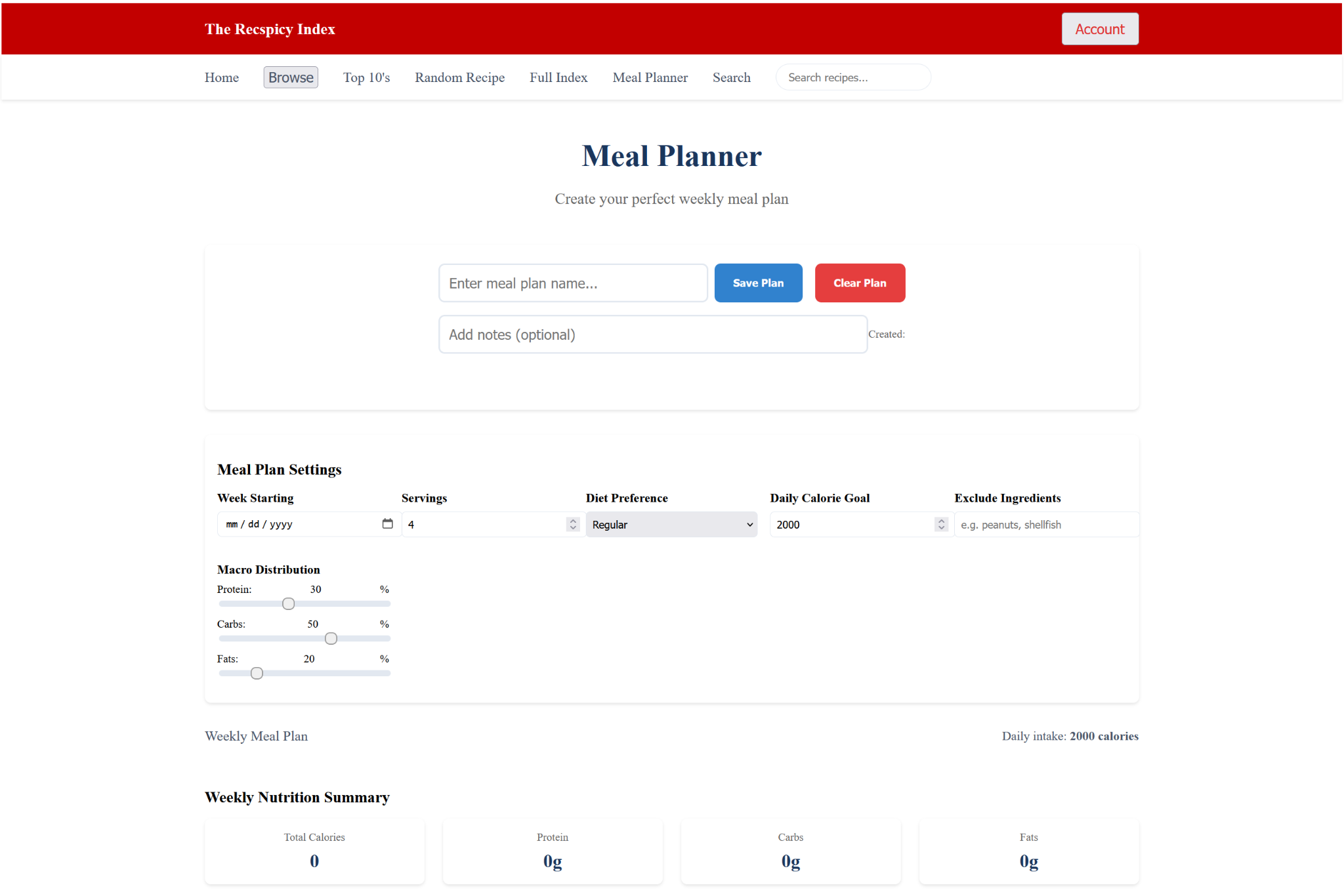
### Wireframe 1: Home Page



### Wireframe 2: Recipe Detail Page



### Wireframe 3: Meal Planning Interface



## 7. API Development

### API Functions

* **User Authentication** – Uses JWT-based authentication for secure login and session management.
* **Recipe Management** – Allows users to create, update, delete, and fetch recipes from the database.
* **Meal Planning** – Allows for users to organize, modify, and retrieve structured meal plans.
* **Favorites System** – Supports saving, retrieving, and managing favorite recipes.
* **Admin Controls** – Provides endpoints for content moderation, user management, and system monitoring.

### External API Integration

* **Tasty API** – Enhances recipe discovery by importing popular recipes from an external database.
* **Nutrition API** – Retrieves nutritional data for meal plans and individual recipes.
* **Search & Filtering API** – Utilizes search queries, enabling filtering by ingredients, dietary preferences, and cooking time.

## 8. Search Engine and Filtering

Users can search recipes by title, ingredient, dietary restriction, and cooking time.Advanced filtering refines search results based on user preference, nutritional content, and meal type. Recspicy's search engine is made to provide fast and accurate recipe discovery allowing the users to search based on multiple criteria.

Users can refine results with advanced filtering options to make sure they find recipes that match their preferences, or cooking constraints. Database queries and search algorithms are developed and refined so the system provides relevant results quickly and efficiently.

### Search Parameters

* **Title & Ingredient Search** – Users can search for recipes by name or key ingredients.
* **Dietary Restrictions** – Filters allow users to exclude allergens or select specific diets like vegan or keto.
* **Cooking Time & Difficulty** – Users can sort recipes based on preparation time and skill level.
* **Cuisine Type** – Filters refine searches based on regional or cultural cuisine preferences.

### Search Optimization

* **Full-Text Indexing** – Speeds up keyword-based searches for quick results.
* **Filtering** – Allows users to combine multiple search criteria for recipe precision.
* **Sorting & Ranking** – Displays popular, highly rated, or recently added recipes at the top.

## 9. Meal Planning System

Users can create meal plans, assign to specific days, and track nutrition. Integration with a Nutrition API provides calorie and macronutrient data for estimated results. Recspicy’s meal planning system is designed to allow users to organize their meals for the week by assigning recipes to specific days.

Nutritional tracking provides insights into calorie intake and macronutrients, helping users make informed dietary choices. The specific system is designed to enable users to modify, save, and reuse meal plans based on their preferences.

### Meal Plan Features

* **Meal Scheduling** – Users can assign breakfast, lunch, dinner, and snacks to each day.
* **Nutrition Tracking** – Aggregates calorie, protein, fat, and carbohydrate data from chosen recipes.
* **Customizable Plans** – Users can add, edit, or delete meal plans as needed.

### Enhancements

* **Portion Adjustments** – Allows users to scale recipes for different serving sizes.
* **Meal Plan Sharing** – Enables users to share meal plans with friends or family.

## 10. Admin Controls and Moderation

Admin dashboard allows admins to utilize tools for managing flagged recipes, user reports, and moderation tasks.Recipe tracking analytics provide insights into user engagement and preferences.

Admins can manage flagged recipes, review user reports, and enforce policies to prevent inappropriate content. A built in analytics system offers insights into recipe popularity and user engagement, helping to improve the overall website experience.

### Moderation

* **Flagged Recipe Review** – Admins can inspect and take action on reported recipes.
* **User Reports Management** – Handles complaints related to inappropriate content or user behavior.
* **Content Approval & Removal** – Allows admins to edit or remove recipes violating guidelines.
* **User Account Actions** – Enables suspension or banning of users engaging in policy violations.

### Analytics & Monitoring

* **Recipe Popularity Tracking** – Identifies frequently viewed and highly rated recipes.
* **User Engagement Metrics** – Tracks active users, favorite recipes, and interaction trends.
* **System Health Monitoring** – Detects performance issues and API requests.
* **Feedback Collection** – Reviews user suggestions to enhance platform functionality.

## 11. Non-Functional Requirements

### Scalability

* **Caching** – Reduces database load by storing frequently accessed data.
* **Optimized Queries** – Uses indexing and query optimization for faster data retrieval.
* **Load Balancing** – Distributes traffic to prevent server overload.
* **Modular Architecture** – Allows easy expansion as new features are added.

### Security

* **JWT Authentication** – Secures user sessions with token-based authentication.
* **Password Hashing** – Encrypts passwords using bcrypt to prevent data leaks.
* **Role-Based Access Control** – Restricts admin and user privileges appropriately.
* **Data Encryption** – Protects sensitive user data both in transit and at rest.

### Performance

* **Database Indexing** – Speeds up searches and filtering operations for improved user experience.

## 12. Functional Requirements

### User Authentication & Profiles

* **User Registration & Login** – Allows users to create accounts and sign in securely.
* **OAuth Integration** – Supports Google Sign-In for quick authentication.
* **Profile Customization** – Enables users to update their bio, avatar, and dietary preferences.
* **Password Recovery** – Provides a secure way to reset forgotten passwords.

### Recipe Management

* **Create & Edit Recipes** – Users can add, modify, or delete their own recipes.
* **Recipe Ratings & Favorites** – Allows users to rate and save recipes for future use.
* **Commenting System** – Enables users to leave feedback and interact with recipe creators.
* **Search & Filters** – Users can find recipes based on ingredients, cuisine, or dietary needs.

### Meal Planning System

* **Weekly Meal Plans** – Users can assign recipes to specific days for structured planning.
* **Nutrition Integration** – Displays calorie and macronutrient breakdowns for meal plans.
* **Plan Sharing** – Allows users to share meal plans with others.

### Admin & Moderation Controls

* **Content Moderation** – Admins can review and remove inappropriate recipes or comments.
* **User Reports Handling** – Enables action on flagged content and user complaints.
* **Analytics Dashboard** – Tracks user engagement, popular recipes, and platform usage.
* **System Logs & Monitoring** – Ensures stability and detects performance issues.

## 13. Testing Strategy

**Unit Testing**: Validates functions and database interactions.

**Integration Testing**: Ensures API interactions.

**UI Testing**: Confirms responsiveness and usability across devices.

### Unit Testing

* **Function Validation** – Tests individual backend functions for expected behavior.
* **Database Operations** – CRUD (Create, Read, Update, Delete) actions for user and recipe data.
* **Authentication Security** – Ensures JWT authentication and password encryption work correctly.

### Integration Testing

* **API Communication** – Confirms that frontend and backend interactions.
* **Third-Party API Integration** – Verifies stable connections with services like the Tasty API.
* **Data Flow Consistency** – Meal planning and recipe management function as expected.

### UI Testing

* **Responsive Design** – Confirms proper display of our website on multiple browsers.
* **User Experience Checks** – Tests navigation, buttons, and interactive elements for operation.
* **Accessibility Compliance** – The platform meets usability standards for all users.

## 14. Appendix & References

The appendix provides materials, diagrams, and additional documentation that help in the understanding of Recspicy’s system architecture and functionality.

References include API documentation, and security guidelines followed during development.

### Appendix

* **System Diagrams** – Includes database schema, API structure, and data flow charts.
* **Wireframes & UI Mockups** – Displays layout concepts for core pages like Home, Recipe Detail, and Meal Planning.
* **Use Case Scenarios** – Provides examples of how users interact with the platform.
* **Data Models** – Shows structured representations of users, recipes, meal plans, and favorites.

### References

* **API Documentation** – References for third-party APIs like Tasty API and Nutrition API.
* **Security Standards** – Guidelines for JWT authentication, password hashing, and data encryption.
* **Performance Optimization Techniques** – Resources on caching, query indexing, and load balancing.
* **Web Accessibility Guidelines** – Ensures compliance with usability standards for all users.
* **Development Frameworks & Libraries** – Includes documentation for React.js, Node.js, Express, and MongoDB.