# **Personalized Recipe Finder & Meal Planner - Capstone Project Proposal**

## **Description**

The Personalized Recipe Finder & Meal Planner is a full-stack web application designed to help users discover recipes tailored to their dietary needs and available ingredients. In addition to recipe discovery, the app enables users to create and manage weekly meal plans, save their favorite recipes, and generate shopping lists. The goal is to simplify meal planning and encourage healthy eating habits through personalized suggestions.

## **Tech Stack**

* **Frontend:** React
* **Backend:** Python with Flask
* **Database:** SQL-based solution (e.g., SQLite for development, PostgreSQL for production) with SQLAlchemy as the ORM
* **External APIs:**
  + Recipe and nutritional data will be sourced from an external API such as Spoonacular or Edamam.
* **Additional Tools:**
  + Git & GitHub for version control and repository management.
  + JWT (JSON Web Tokens) for secure user authentication.
  + A CSS framework (such as Bootstrap or Material-UI) for building a responsive UI.

## **Focus**

This project is a balanced full-stack application with attention on both frontend and backend components:

* **Backend:** Implement a Flask-based API for integrating external recipe data, handling user authentication, and managing data storage.
* **Frontend:** Develop a responsive React interface that allows users to search for recipes, manage meal plans, and interact with their personalized dashboard.

## **Type**

This project will be a **responsive website** designed to work on both desktop and mobile devices.

## **Goal**

The primary goal of the project is to provide a user-friendly platform that:

* Helps users discover recipes that meet their specific dietary preferences.
* Simplifies meal planning and grocery shopping through personalized features.
* Encourages healthier eating by offering tailored nutritional insights.

## **Users**

The target demographic includes:

* Health-conscious individuals and families aiming to improve their diet.
* Busy professionals looking for efficient meal planning solutions.
* Cooking enthusiasts who want to explore new recipes.
* Users with specific dietary requirements (e.g., gluten-free, vegan, low-carb) who need tailored meal suggestions.

## **Data and Data Collection**

* **External Data:**
  + Recipe details (ingredients, instructions, nutritional information, images) will be fetched from APIs such as Spoonacular or Edamam.
* **User-Generated Data:**
  + User profiles, meal plans, and saved recipes will be stored in our SQL database using SQLAlchemy for data modeling.
* **Data Collection Strategy:**
  + Recipe data will be retrieved dynamically from the external API when users search for recipes.
  + Custom Flask endpoints will handle user data management (e.g., registration, login, saving recipes, and creating meal plans).
  + Implement caching strategies if necessary to manage API rate limits and improve performance.

## **Approach and Functionality**

### **Database Schema**

* **Users Table:**
  + id (Primary Key)
  + username (String)
  + email (String)
  + password (Hashed String)
  + preferences (JSON or separate columns for dietary restrictions, favorite cuisines, etc.)
* **Recipes Table (for caching purposes, if needed):**
  + id (Primary Key)
  + recipe\_id (String from the external API)
  + title (String)
  + ingredients (Text or JSON)
  + instructions (Text)
  + nutritional\_info (JSON)
  + image\_url (String)
* **MealPlans Table:**
  + id (Primary Key)
  + user\_id (Foreign Key to Users)
  + date\_range (Date Range/Start & End Dates)
  + recipes (Relationship to Recipes or a JSON list of recipe IDs)
  + shopping\_list (Text or JSON)

### **API Integration & Potential Challenges**

* **Integration Issues:**
  + Managing API rate limits and ensuring the security of API keys.
  + Handling changes in external API data structures or formats.
* **Mitigation Strategies:**
  + Use caching or local storage of frequently used data.
  + Monitor external API documentation for updates and adjust the integration code as needed.

### **Security Considerations**

* **Sensitive Information:**
  + User credentials and authentication tokens.
* **Approach:**
  + Store passwords securely using hashing (e.g., bcrypt).
  + Protect sensitive endpoints with JWT-based authentication.
  + Use environment variables to secure API keys and other sensitive configuration data.

### **User Flow**

1. **Onboarding:**
   * Users sign up or log in and set their dietary preferences.
2. **Recipe Search:**
   * Users search for recipes by entering ingredients or keywords; the app calls the external API to fetch matching recipes.
3. **Recipe Details:**
   * Users can view detailed recipe information, including nutritional data and preparation instructions, and save their favorite recipes.
4. **Meal Planning:**
   * Users can create and customize weekly meal plans by selecting recipes.
5. **Shopping List Generation:**
   * The app automatically generates a shopping list based on the selected recipes in the meal plan.
6. **Dashboard:**
   * Users access a dashboard displaying their saved recipes, meal plans, and nutritional progress.

### **Features Beyond CRUD (Stretch Goals)**

* **Personalized Recommendations:**
  + Suggest recipes based on user preferences and historical interactions.
* **Grocery Delivery Integration:**
  + Explore integration with grocery delivery services for a seamless shopping experience.
* **Nutritional Tracking:**
  + Provide visualizations (charts/graphs) to track nutritional intake over time.
* **Social Sharing:**
  + Enable users to share their meal plans and favorite recipes on social media.

## **Conclusion**

The Personalized Recipe Finder & Meal Planner is designed to offer a comprehensive solution for meal planning and healthy eating. By leveraging the Python/Flask backend alongside a React frontend, the project will integrate real-time external recipe data with personalized user management features. This capstone project will showcase the ability to build a scalable, secure, and user-focused web application, aligning well with the skills and technologies emphasized in the bootcamp.