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

## **Description**

The Personalized Recipe Finder & Meal Planner is a full-stack web application that helps users discover recipes tailored to their dietary needs and available ingredients. The application goes beyond simple recipe search by allowing users to build weekly meal plans, save favorite recipes, and generate shopping lists automatically. This project aims to simplify the process of meal planning and promote healthier eating habits by offering personalized recipe recommendations and nutrition insights.

## **Tech Stack**

* **Frontend:** React
* **Backend:** Python with Flask
* **Database:** SQL-based solution (SQLite for development and PostgreSQL for production) using SQLAlchemy as the ORM
* **External APIs:**
  + Recipe and nutritional data will be sourced from APIs such as Spoonacular or Edamam.
* **Authentication & Security:**
  + JSON Web Tokens (JWT) for user authentication
  + Environment variables for API keys and sensitive configuration data

## **Focus**

This project is an evenly balanced full-stack application with focus on both the front-end UI and back-end functionality. Key components include:

* **Backend:**
  + Creating Flask endpoints to integrate with external recipe APIs and to manage user data, including authentication, saved recipes, and meal plans.
* **Frontend:**
  + Building a responsive and intuitive React interface that enables users to search for recipes, view details, manage their meal plans, and generate shopping lists.

## **Type**

This project will be delivered as a **responsive website**, optimized for both desktop and mobile devices.

## **Goal**

The primary goal of this project is to offer users a comprehensive and personalized platform that:

* Enables discovery of recipes that meet individual dietary requirements.
* Simplifies meal planning by allowing users to schedule recipes over a weekly calendar.
* Automatically generates shopping lists based on selected recipes, thereby saving time and effort.

## **Users**

The application is targeted toward:

* Health-conscious individuals and families who want to improve their diet.
* Busy professionals who need an efficient solution for meal planning.
* Cooking enthusiasts looking for creative recipes and meal ideas.
* Users with specific dietary restrictions (e.g., vegan, gluten-free, low-carb) who need customized recipe recommendations.

## **Data and Data Collection**

* **External Data:**
  + The app will fetch recipe details such as ingredients, cooking instructions, nutritional information, and images from external APIs (Spoonacular or Edamam).
* **User-Generated Data:**
  + User profiles, preferences, saved recipes, and meal plans will be stored in a SQL database.
* **Data Collection Strategy:**
  + Recipe data will be retrieved dynamically when users perform searches.
  + Flask endpoints will handle CRUD operations for user data, ensuring a personalized and secure experience.

## **Approach and Functionality**

### **Database Schema**

* **Users Table:**
  + id (Primary Key)
  + username (String)
  + email (String)
  + password (Hashed String using bcrypt)
  + preferences (JSON field for dietary restrictions, favorite cuisines, etc.)
* **Recipes Table (for caching purposes):**
  + id (Primary Key)
  + recipe\_id (External API recipe identifier)
  + title (String)
  + ingredients (JSON or Text)
  + instructions (Text)
  + nutritional\_info (JSON)
  + image\_url (String)
* **MealPlans Table:**
  + id (Primary Key)
  + user\_id (Foreign Key referencing Users)
  + date\_range (Start Date and End Date)
  + recipes (JSON list of recipe IDs)
  + shopping\_list (JSON or Text)

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

* **Challenges:**
  + Handling API rate limits and ensuring the secure management of API keys.
  + Dealing with changes in external API structures and data formats.
* **Mitigation:**
  + Implement caching strategies to reduce frequent API calls.
  + Regularly monitor and update API integration code as per the API documentation.

### **Security Considerations**

* **Sensitive Information:**
  + User credentials and API keys.
* **Approach:**
  + Use bcrypt for password hashing and JWT for route protection.
  + Store sensitive data in environment variables, and ensure all endpoints handling user data are secure.

### **User Flow**

1. **Onboarding:**
   * Users sign up or log in and set their dietary preferences.
2. **Recipe Search:**
   * Users input ingredients or keywords; the app queries the external API to fetch relevant recipes.
3. **Recipe Details:**
   * Users can view detailed recipe information, save favorite recipes, and review nutritional data.
4. **Meal Planning:**
   * Users create weekly meal plans by selecting recipes, with the option to rearrange as needed.
5. **Shopping List Generation:**
   * The app automatically generates a shopping list based on the recipes in the meal plan.
6. **Dashboard:**
   * A central hub where users manage their saved recipes, meal plans, and view nutritional insights.

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

* **Personalized Recommendations:**
  + Leverage user interaction data and preferences to suggest recipes.
* **Grocery Delivery Integration:**
  + Explore integration with grocery delivery services for direct online shopping.
* **Nutritional Tracking:**
  + Incorporate visualizations (charts/graphs) to track users' nutritional intake over time.
* **Social Sharing:**
  + Allow users to share their meal plans and favorite recipes on social media.

## **Conclusion**

The Personalized Recipe Finder & Meal Planner is designed to be a comprehensive solution for meal planning and healthy eating. By leveraging Python/Flask for the backend, React for the frontend, and a robust SQL database, this capstone project will showcase my ability to build a scalable, secure, and user-friendly web application. With a clear user flow, defined data strategy, and potential stretch features, I believe this project is both challenging and achievable within the expected 45-65 hour timeframe.