**Revised Project Spec: Personalized Meal Planning and Nutrition Tracker**

**Project Description:**

The **Personalized Meal Planning and Nutrition Tracker** app is designed to help users plan meals based on their dietary preferences, nutritional needs, and health goals. Users can input their dietary restrictions, preferred cuisines, and specific health goals such as weight loss or muscle gain. The app generates meal plans and grocery lists based on this input. Additionally, the app tracks the nutritional content of meals and provides insights into the user's diet over time. It also integrates with fitness trackers to give a comprehensive view of the user's health progress.

**Key Features (Revised):**

**Personalized Meal Planning:**

* Users can input their dietary preferences, restrictions, and health goals.
* The app generates weekly meal plans tailored to the user input.
* Grocery lists are automatically created based on the meal plan.

**Nutritional Tracking (Revised):**

* **Revised**: Transition from using CSV files for storing nutritional data to **SQLite** database for more scalable and efficient data management.
* Tracks users' daily nutritional intake.
* Provides insights into nutritional trends over time.
* Offers recommendations for improving diet based on tracked data.

**Fitness Tracker Integration (Revised):**

* **Revised**: The fitness tracker integration will focus only on **Fitbit** to simplify the data fetching and sync process.
* Syncs with Fitbit to track exercise and activity levels.
* Provides a holistic view of user health by combining nutrition and fitness data.

**Use Case (Revised):**

Users input their dietary preferences (e.g., avoiding dairy) and health goals (e.g., weight loss, muscle gain). The app generates a weekly meal plan based on these inputs, preferring Mediterranean cuisine. It provides a grocery list for easy shopping. As the user follows the meal plan, the app tracks their nutrient intake, and after a month, shows improvement in their overall nutrition. The app then suggests refining their diet, such as increasing protein intake or reducing sugar consumption. With Fitbit integration, the user can also track physical activity and see a combined view of fitness and nutrition data.

**Challenges (Revised):**

1. **Meal Plan Algorithm**:
   * Challenge: Developing a flexible and accurate algorithm to generate personalized meal plans based on dietary restrictions and preferences.
   * Solution: Focus on refining the meal plan generation through iterative testing and user feedback.
2. **Nutritional Data Accuracy**:
   * Challenge: Ensuring accurate nutritional data with the integration of the Edamam API.
   * Solution: Optimize API calls and ensure data accuracy by cross-referencing API documentation.
3. **Fitness Tracker Integration** (Revised):
   * Challenge: Integrating the Fitbit API, including managing data flows and real-time syncing.
   * **Revised** Solution: Narrow focus on Fitbit to simplify the integration process and allow for successful data syncing and analysis.

**Technology Stack (Revised):**

1. **Backend**:
   * **Python** using **Flask** for handling routing, meal planning logic, and integration with external services like nutrition and fitness APIs.
2. **Frontend**:
   * **Flask Templates** to handle user input and display meal plans, grocery lists, and nutritional insights.
3. **Database (Revised)**:
   * **Revised**: **SQLite** for storing user profiles, meal plans, and nutritional data. Replacing CSV files with SQL to allow for better scalability and data management.
4. **APIs**:
   * **Nutrition API**: Edamam API for retrieving accurate food and nutrition data.
   * **Fitness Tracker API**: Fitbit for syncing user activity data.

**GitHub Repository Structure (Revised):**

* **README.md**: A brief project overview explaining its core functionality (meal planning, nutrition tracking).
* **.gitignore**: Python-style .gitignore to ensure that unnecessary files (e.g., \_\_pycache\_\_, .env) are not tracked.
* **LICENSE**: MIT License.
* **docs** folder: For storing the project spec and any additional documentation related to the app (e.g., API references, sketches, or mockups).
* **Project Root**:
  + All Python code (e.g., main.py and supporting modules) will reside in the root folder.
  + A **data** folder will store static files like example datasets.
  + An **img** folder will store any images or diagrams relevant to the project documentation.

**Revisions Summary:**

* **Revised**: Replaced the plan to use CSV files with SQLite database migration for storing nutritional data and user profiles.
* **Revised**: Fitness tracker integration will focus solely on Fitbit to simplify data fetching and syncing.
* **Proposed**: Added optional feature to provide more detailed nutrition insights, including tracking macros and trends over time, if time permits.