Nutrisense Nutrition Assistant: Comprehensive Documentation

1. Introduction

The Nutrisense Nutrition Assistant is a sophisticated AI-powered application designed to provide personalized nutrition and fitness guidance. Built with Streamlit, it offers a user-friendly interface for managing dietary preferences, tracking food intake, finding suitable restaurants, and generating customized workout plans. This assistant leverages advanced AI models, specifically Mistral AI for intelligent advice and Exa API for real-time restaurant data, to deliver a holistic and personalized health management experience.

This document aims to provide a comprehensive overview of the project, including its features, underlying architecture, detailed setup and usage instructions, troubleshooting tips, and guidelines for contribution.

2. Features

The Nutrisense Nutrition Assistant is packed with several key features that cater to various aspects of personal health and wellness:

2.1. AI-Powered Nutrition Advice

This core feature provides users with personalized dietary recommendations and insights. The system is designed to understand individual needs and provide context-aware advice.

- **Personalized Recommendations**: The AI analyzes your unique profile, including age, gender, weight, height, activity level, and health goals, to offer nutrition advice that is specifically tailored to you. This ensures that the recommendations are relevant and effective for your personal health journey.

- **Dietary Restrictions Support**: The assistant is capable of handling a wide array of dietary restrictions and preferences. Whether you are vegetarian, vegan, gluten-free, or have specific food allergies, the AI adjusts its recommendations accordingly to ensure your safety and adherence to your chosen diet.

- **Health Goals Integration**: Your health objectives, such as weight loss, muscle gain, or general wellness, are integrated into the AI's advice generation. This means the recommendations are not generic but are aligned with helping you achieve your specific targets.
- **Real-time AI Responses**: Powered by the Mistral AI API, the assistant provides intelligent and contextual advice in real-time. This allows for dynamic and interactive conversations, making the nutrition guidance feel like a personal consultation.

2.2. Smart Restaurant Search

Finding suitable dining options that align with your dietary needs can be challenging. The Smart Restaurant Search feature simplifies this process.

- **Location-based Search**: Users can easily find restaurants in their vicinity, making it convenient to discover local dining spots.
- **Dietary Preference Filtering**: The search results are intelligently filtered based on your pre-set dietary restrictions and preferences, ensuring that you only see relevant options.
- **AI-Powered Recommendations**: Beyond simple filtering, the AI provides personalized restaurant suggestions, taking into account your overall profile and past interactions.
- **Real-time Data**: This feature utilizes the Exa API to fetch current and accurate restaurant information, including menus, reviews, and availability, ensuring you have the most up-to-date details.

2.3. Advanced Food Logging

Tracking food intake is crucial for managing nutrition. The assistant offers flexible and intuitive methods for logging your meals.

- **Natural Language Logging**: A standout feature is the ability to log food directly through chat using natural language. This eliminates the need for manual data entry and makes the process quick and seamless. For example, you can simply type: "I ate an apple," "Had grilled chicken for lunch," or "Just drank a coffee."

- **AI Calorie Estimation**: For common food items, the AI automatically estimates calorie content, saving you the effort of looking up nutritional information.
- **Daily Progress Tracking**: The system allows you to monitor your progress towards your daily calorie goals, providing a clear overview of your intake.
- **Macro Tracking**: For users who require more detailed nutritional insights, the assistant offers optional tracking for macronutrients (protein, carbohydrates, and fats).
- **Meal Type Categorization**: Food logs can be categorized by meal type (breakfast, lunch, dinner, and snacks), helping you analyze your eating patterns throughout the day.

2.4. Personalized Workout Plans

Beyond nutrition, the assistant also supports your fitness journey by generating customized workout routines.

- **AI-Generated Plans**: Custom workout routines are created based on your fitness level, health goals, and activity level, ensuring the plans are challenging yet achievable.
- **Nutrition Integration**: The workout plans are integrated with nutrition advice, providing recommendations for pre- and post-workout meals to optimize performance and recovery.
- **Progression Tracking**: The system offers structured plans for continuous improvement, helping you gradually increase intensity and achieve your fitness milestones.
- **Recovery Guidance**: The assistant also provides tips for rest day activities and recovery strategies, emphasizing the importance of holistic well-being.

2.5. User Profile Management

Effective personalization relies on accurate user data. The assistant provides robust tools for managing your profile.

- **Comprehensive Profiles**: Users can input detailed personal information, including age, weight, height, activity level, and health goals, to create a complete profile.

- **Dietary Preferences**: You can specify allergies, dietary restrictions, and food preferences, which the AI uses to tailor its recommendations.
- **Health Goals**: Clearly define your health objectives, such as weight loss, muscle gain, or maintenance, to receive targeted advice.
- **Persistent Storage**: All your preferences and logged data are securely saved in a local SQLite database, ensuring data persistence across sessions and maintaining privacy.

3. Architecture

The Nutrisense Nutrition Assistant is built upon a modular and scalable architecture, leveraging modern technologies to deliver its diverse functionalities.

3.1. Core Components

- **Streamlit Frontend**: The user interface is developed using Streamlit, a powerful open-source app framework for Machine Learning and Data Science. Streamlit enables the creation of modern, responsive web applications with minimal code, providing an intuitive and interactive experience for users.
- **Mistral AI API**: This is the brain of the nutrition assistant. The Mistral AI API powers the advanced AI capabilities, including personalized nutrition advice, natural language food logging, and AI-generated workout plans. It allows for intelligent, contextual, and real-time responses based on user queries and profiles.
- **Exa API**: For real-time restaurant search and recommendations, the application integrates with the Exa API. This external service provides up-to-date information on restaurants, enabling the assistant to offer accurate and relevant dining suggestions based on location and dietary preferences.
- **SQLite Database**: User preferences, food logs, and other persistent data are stored in a local SQLite database. SQLite is a lightweight, file-based database system that is ideal for local storage, ensuring data privacy and easy management without requiring a separate database server.

- **Session Management**: The application incorporates robust session management to maintain persistent user states across interactions. This means your preferences and past activities are remembered, providing a seamless and continuous user experience.

3.2. Key Features (Technical Perspective)

- **Natural Language Processing (NLP)**: A critical technical feature is the assistant's ability to understand and process natural language inputs, particularly for food logging. This is achieved through the Mistral AI API, which interprets user queries like "I ate an apple" and extracts relevant information for logging.
- **Personalized AI Responses**: The system dynamically generates context-aware responses by combining user profile data with AI model capabilities. This personalization extends to nutrition advice, restaurant suggestions, and workout plans, making each interaction highly relevant to the individual user.
- **Real-time Data Integration**: The integration with the Exa API for restaurant search demonstrates the application's ability to fetch and utilize real-time external data. This ensures that the information provided to the user is always current.
- **Local Data Storage**: The use of SQLite for local data storage emphasizes a privacy-focused approach. User data is kept on the local machine, giving users more control over their personal information.
- **Responsive Design**: The Streamlit frontend inherently supports responsive design, ensuring that the application is accessible and user-friendly across various devices, including desktops and mobile phones.

4. Setup and Installation

To get the Nutrisense Nutrition Assistant up and running on your local machine, follow these steps:

4.1. Prerequisites

Before you begin, ensure you have the following installed:

- **Python 3.8+**: The application is developed in Python, and version 3.8 or higher is required.
- **Mistral AI API Key**: You will need an API key from Mistral AI to enable the AI-powered features. Obtain your key from https://console.mistral.ai/api-keys.
- **Exa API Key**: For the restaurant search functionality, an API key from Exa is necessary. You can get your key from https://exa.ai/.

4.2. Installation Steps

1. **Clone the repository**:
   Open your terminal or command prompt and clone the GitHub repository using the following command:
2. **Install dependencies**:
   Navigate into the cloned directory and install the required Python packages. It's recommended to use a virtual environment to manage dependencies.
3. **Set up environment variables**:
   The application requires API keys to function correctly. Create a `.env` file in the root directory of the project by copying the provided template:
   - **MISTRAL_API_KEY**: Your 32-character alphanumeric Mistral AI API key.
   - **EXA_API_KEY**: Your UUID format Exa API key.
   - **DATABASE_URL**: (Optional) Path to your SQLite database file. Defaults to `nutrisense_data.db`.
   - **LOG_LEVEL**: (Optional) Logging level (e.g., INFO, DEBUG, WARNING, ERROR). Defaults to `INFO`.
4. **Run the application**:
   Once the dependencies are installed and environment variables are set, you can run the Streamlit application:

5. How to Use

This section guides you through the main functionalities of the Nutrisense Nutrition Assistant.

5.1. Set Your Preferences

Before using the AI features, it's essential to set up your personal preferences. This allows the assistant to provide tailored advice.

1. **Navigate to "🎯 Set Preferences"**: In the sidebar of the application, click on the "🎯 Set Preferences" option.
2. **Enter User ID**: Provide a unique User ID. This ID links all your data and ensures your preferences and logs are associated with your profile.
3. **Fill in Profile Information**: Complete the form with your details:
    - **Basic Info**: Age, gender, current weight, and height.
    - **Activity Level and Health Goals**: Select your activity level and define your health objectives (e.g., weight loss, muscle gain, maintenance).
    - **Dietary Restrictions and Allergies**: Specify any dietary restrictions (e.g., vegetarian, vegan, gluten-free) or food allergies you may have.
4. **Save Preferences**: Click the "Save Preferences" button to store your information in the local database.

5.2. Get AI Nutrition Advice

Once your preferences are set, you can start receiving personalized nutrition advice.

1. **Navigate to "💬 Nutrition Chat"**: Go to the "💬 Nutrition Chat" section in the sidebar.
2. **Enter User ID**: Enter the same User ID you used when setting your preferences.
3. **Ask Nutrition Questions**: Type your nutrition-related questions into the chat interface. The AI will provide comprehensive and personalized advice based on your profile. Examples:
    - "What should I eat for breakfast?"

    - "How many calories should I eat to lose weight?"
    - "What are good protein sources for vegetarians?"

5.3. Food Logging

Track your food intake using either the chat-based method or the dedicated food logger.

Option A: Chat-based Logging (Recommended)

This is the most convenient way to log your food.

- **In the chat interface**: Simply describe what you ate using natural language. The AI will automatically process and log the food item, estimating calories and categorizing it.
    - Examples: "I ate an apple," "Had grilled chicken breast for lunch," "Just drank a coffee," "Ate oatmeal with berries for breakfast."

Option B: Dedicated Food Logger

For more manual control, use the dedicated food logger.

1. **Navigate to "🍎 Food Logger"**: Select this option from the sidebar.
2. **Fill in Food Details**: Manually enter the food item, estimated calories, and other relevant details.
3. **Log Food**: Click "Log Food" to save the entry.

5.4. Restaurant Search

Find restaurants that match your dietary needs and location.

1. **In the chat interface**: Ask for restaurant recommendations. The AI will use your location and preferences to provide suggestions.
    - Examples: "Find restaurants in Mumbai," "Show me vegetarian restaurants in Delhi," "Best healthy restaurants near me."

5.5. Workout Plans

Request personalized workout routines based on your fitness goals.

1. **In the chat interface**: Ask for workout advice or a specific plan.
   - Examples: "Create a workout plan for weight loss," "Give me a strength training routine," "How should I exercise for muscle gain?"

6. Troubleshooting

This section addresses common issues you might encounter and provides solutions.

6.1. Common Issues

1. **API Key Errors (401/403)**:
   - **Symptom**: The application returns authentication errors or fails to use AI/restaurant search features.
   - **Solution**: Verify that your Mistral AI and Exa API keys are correct and properly set in your `.env` file. Ensure the Mistral API key is a 32-character alphanumeric string and the Exa API key is in UUID format. Double-check for any typos or extra spaces.
   - **Testing**: You can use the provided test scripts to verify API functionality:
2. **Environment Variables Not Loading**:
   - **Symptom**: The application behaves as if API keys or other configurations are missing, even if you've set them in `.env`.
   - **Solution**: Ensure that the `.env` file is located in the same directory as `app.py` (the project root). Check file permissions to ensure the application can read the `.env` file.
   - **Testing**: Use the debug script to check environment variable loading:
3. **Food Logging Not Working**:
   - **Symptom**: Food entries are not saved or processed correctly.
   - **Solution**: Make sure you have set a User ID in the "🎯 Set Preferences" section and are using the *same* User ID consistently in both the chat interface and the dedicated

food logger. When using chat-based logging, try phrasing your input naturally, for example: "I ate [food name]."

- **Testing**: Verify food logging functionality with the dedicated test script:

6.2. Testing

The project includes several utility scripts to help you test different components:

- `test_mistral.py` : Verifies the connection and functionality of the Mistral AI API.
- `debug_env.py` : Helps debug issues related to environment variable loading.
- `test_food_logging.py` : Tests the food logging mechanism.

To run these tests, navigate to the project root directory in your terminal and execute the respective Python script.

7. Database Schema

The Nutrisense Nutrition Assistant uses a local SQLite database to store user-specific data. The database is initialized with two main tables:

7.1. Tables

- **`user_preferences`** : This table stores comprehensive user profile information and dietary preferences. It is designed to hold all the data necessary for personalizing AI responses and recommendations.
- **`nutrition_logs`** : This table is dedicated to tracking the user's food intake. Each entry records a specific food item consumed, along with its estimated calorie content and the timestamp of the log.

7.2. Key Fields

Here are some of the key fields within these tables:

- **`user_id`**: A unique identifier that links all user data across both `user_preferences` and `nutrition_logs` tables. This ensures that all activities and preferences are correctly attributed to the respective user.
- **`food_item`**: (in `nutrition_logs`) The name or description of the food item consumed.
- **`calories`**: (in `nutrition_logs`) The estimated calorie content of the logged food item.
- **`timestamp`**: (in `nutrition_logs`) The date and time when the food item was logged, allowing for chronological tracking of intake.
- **`dietary_restrictions`**: (in `user_preferences`) Stores dietary restrictions as a JSON array (e.g., `["vegetarian", "gluten-free"]`).
- **`food_allergies`**: (in `user_preferences`) Stores food allergies as a JSON array (e.g., `["peanuts", "dairy"]`).
- **`health_goals`**: (in `user_preferences`) Stores health goals as a JSON object (e.g., `{"weight_loss": true, "muscle_gain": false}`).
- **`weight_goal`**: (in `user_preferences`) The user's target weight.
- **`current_weight`**: (in `user_preferences`) The user's current weight.
- **`activity_level`**: (in `user_preferences`) Describes the user's physical activity level (e.g., "sedentary", "light", "moderate", "active", "very_active").
- **`age`**: (in `user_preferences`) The user's age.
- **`gender`**: (in `user_preferences`) The user's gender.
- **`height_cm`**: (in `user_preferences`) The user's height in centimeters.
- **`daily_calorie_target`**: (in `user_preferences`) The user's calculated or desired daily calorie intake.
- **`protein_target`**, **`carb_target`**, **`fat_target`**: (in `user_preferences`) Optional target values for macronutrient intake.

8. Contributing

We welcome contributions to the Nutrisense Nutrition Assistant project! If you're interested in improving the application, please follow these guidelines:

1. **Fork the repository**: Start by forking the `harishmogili21/Nutrisense-Nutrition-Assistant` repository on GitHub.
2. **Create a feature branch**: Create a new branch for your feature or bug fix. Use a descriptive name (e.g., `feature/add-new-meal-type` or `bugfix/api-key-error`).
3. **Make your changes**: Implement your changes, ensuring that your code adheres to the existing style and conventions.
4. **Test thoroughly**: Before submitting, thoroughly test your changes to ensure they work as expected and do not introduce new issues. Utilize the provided test scripts (`test_mistral.py` , `debug_env.py` , `test_food_logging.py`) and add new tests if necessary.
5. **Submit a pull request**: Once your changes are complete and tested, push your branch to your forked repository and open a pull request to the `main` branch of the original repository. Provide a clear and concise description of your changes.

9. License

This project is licensed under the MIT License. You can find the full text of the license in the `LICENSE` file within the repository. This permissive license allows you to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the software.

10. Acknowledgments

We extend our gratitude to the following organizations and projects for their invaluable contributions to the Nutrisense Nutrition Assistant:

- **Mistral AI**: For providing the powerful AI capabilities that drive the personalized nutrition advice and natural language processing.

- **Exa**: For enabling the real-time restaurant search functionality with their comprehensive API.
- **Streamlit**: For the intuitive and efficient web framework that made building the user interface a seamless experience.
- **SQLite**: For providing a lightweight and reliable local data storage solution, ensuring user privacy and data persistence.

Author: Manus AI
Date: July 26, 2025