

Nutritionist AI App Using Gemini Pro: Your Comprehensive Guide to Healthy Eating and Well-being

Project Description

The **Nutritionist AI** app is an innovative, AI-powered application designed to deliver personalized dietary recommendations and nutritional advice to its users. Utilizing Google's advanced **Gemini Pro** model, the app analyzes user data, dietary preferences, and health goals to generate custom meal plans, provide nutritional insights, and offer wellness tips. The primary objective is to guide users towards healthier eating habits and improved overall well-being through data-driven, customized recommendations.

Requirements Specification

1. **Python (v3.7+):**
 - Required for backend logic, data processing, and integrating the Google Gemini Pro API.
2. **Google API Key:**
 - Needed to authenticate requests to the Gemini Pro model from Google Generative AI API.
3. **Streamlit:**
 - Used to build the app's frontend, making it easy for users to interact with the Nutritionist AI and view their personalized recommendations in real-time.
4. **Deep Learning Model (Gemini Pro):**
 - A pre-trained model provided by Google, known for its generative capabilities, which the app uses for accurate dietary analysis and recommendations.
5. **Pillow (PIL):**
 - For handling image uploads, processing food images, and providing image data to the Gemini Pro model.
6. **Environment Setup:**
 - A `.env` file for secure storage of API keys.
 - A `requirements.txt` file listing the necessary Python libraries for the project (`streamlit`, `google-generativeai`, `dotenv`, `pillow`).

Initialization of Google API Key

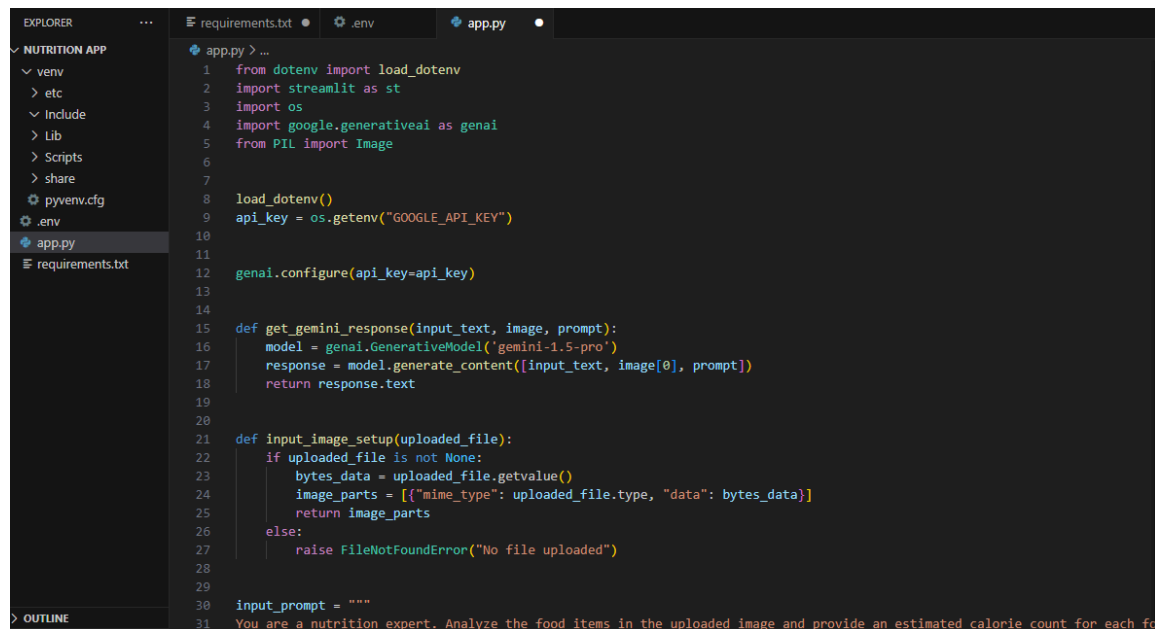
To use the Gemini Pro model, an **API key** from Google Generative AI is required.

1. **Sign up for Google Generative AI:**
 - Obtain your API key from the Google Cloud Console.
2. **Store the API Key:**
 - Create a `.env` file in the project root.
 - Add your API key in the `.env` file
3. **Load the API Key in Code:**
 - Use `dotenv` to load the API key securely.

Interfacing with Pre-trained Model

The **Gemini Pro** model is accessed through Google's API and provides the core generative capability for this app. This model interprets food-related data, analyzes dietary elements, and delivers precise calorie and nutritional estimates.

1. **Set up Google Generative AI:**
 - Install the `google-generativeai` package to access Gemini Pro.
2. **Configure Gemini Pro API:**
 - Use the configured API key to authenticate and interact with Gemini Pro.
3. **Generate Responses:**
 - Define a function that interacts with the model and processes user data and images to generate recommendations.



```
EXPLORER  ...  requirements.txt  .env  app.py  ...  
NUTRITION APP  
  venv  
  > etc  
  Include  
  Lib  
  Scripts  
  share  
  pyvenv.cfg  
  .env  
  app.py  
  requirements.txt  
OUTLINE  
app.py > ...  
1  from dotenv import load_dotenv  
2  import streamlit as st  
3  import os  
4  import google.generativeai as genai  
5  from PIL import Image  
6  
7  
8  load_dotenv()  
9  api_key = os.getenv("GOOGLE_API_KEY")  
10  
11  
12  genai.configure(api_key=api_key)  
13  
14  
15  def get_gemini_response(input_text, image, prompt):  
16      model = genai.GenerativeModel('gemini-1.5-pro')  
17      response = model.generate_content([input_text, image[0], prompt])  
18      return response.text  
19  
20  
21  def input_image_setup(uploaded_file):  
22      if uploaded_file is not None:  
23          bytes_data = uploaded_file.getvalue()  
24          image_parts = [{"mime_type": uploaded_file.type, "data": bytes_data}]  
25          return image_parts  
26      else:  
27          raise FileNotFoundError("No file uploaded")  
28  
29  
30  input_prompt = ""  
31  You are a nutrition expert. Analyze the food items in the uploaded image and provide an estimated calorie count for each fo
```

Model Deployment

1. Deploying with Streamlit:

- The app uses Streamlit to create an interactive interface for uploading images and receiving nutritional insights.

2. Handling User Inputs:

- Users provide input text and food images that the model uses to generate a response.

3. Displaying Results:

- Once processed, the app displays the estimated calorie intake and nutritional breakdown to guide users on their dietary choices.

4. Deploy on Streamlit Cloud:

- For live deployment, the app can be hosted on **Streamlit Cloud** for easy access and sharing.

Deploy :

AI Nutritionist App

Input Prompt:

tell calories

Choose an image...



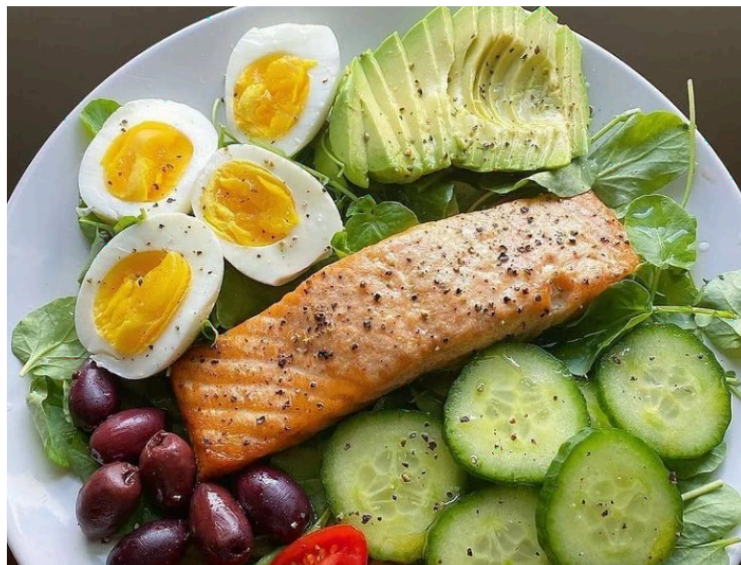
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



img7.jpg 126.0KB





Uploaded Image.

Tell me the total calories

Estimated Calorie Intake

Here's an estimated calorie breakdown of the meal:

- Salmon (4 oz): 200 calories
- Eggs (4 medium): 280 calories
- Avocado (1/2 medium): 160 calories
- Cucumber (1 cup, sliced): 15 calories
- Cherry Tomatoes (1/2 cup): 15 calories
- Olives (10): 40 calories
- Watercress (2 cups): 8 calories

Total Calories: **Approximately 718 calories**

Future Enhancements

1. **User Authentication:**
 - Add secure user authentication to store dietary preferences and provide personalized insights over time.
2. **Detailed Nutritional Insights:**
 - Expand to include micronutrient analysis (vitamins, minerals) based on meal contents.
3. **Multi-language Support:**
 - Include support for multiple languages to make the app accessible to a global audience.
4. **Data Storage:**
 - Implement a backend database to store user dietary history and enable data-driven insights.