

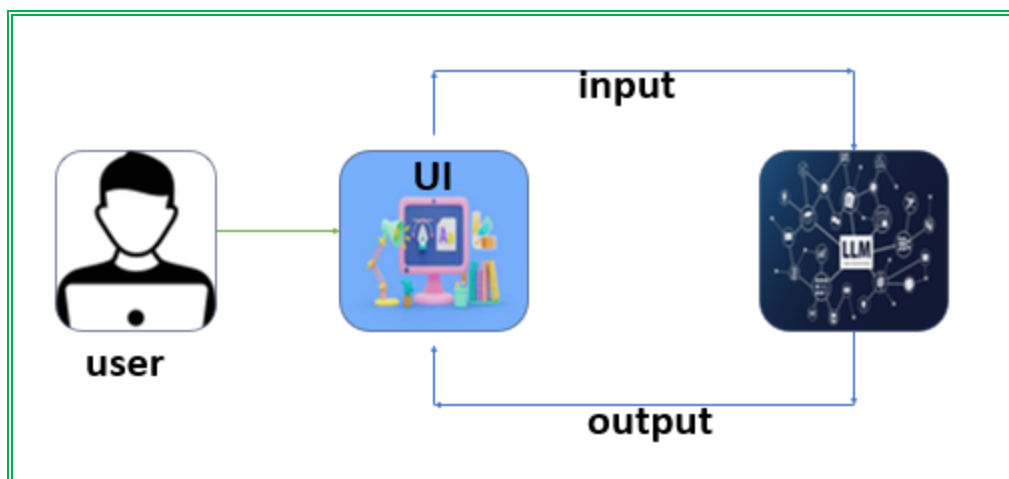
# Nutrition App Using Gemini Pro : Your Comprehensive Guide To Healthy Eating

## And Well- Being

### 1.Description:

Nutritionist AI is an innovative mobile application designed to provide personalized dietary recommendations, meal planning, nutritional advice and general well-being insights using the advance capabilities of the gemini pro model. In this project I use gemini-1.5-flash model for designing the nutrition app.

### Technical Architecture:



### 2.Requirements:

- **Streamlit:** Streamlit is a powerful framework for building interactive web application with python.it is an open-source python library that makes it easy to create and share custom web apps.By using Streamlit we can quickly build and deploy powerful data applications.
- **Google-generativeai:** It is a Python client library for accessing the GenerativeAI API, providing interactions with pre-trained language models like Gemini.
- **Python-dotenv:** it allows you to manage environment variables stored in a .env file for our python projects.

- **PyPDF2:** It is a Python library for extracting text and manipulating PDF documents.
- **Pillow:** Pillow is a Python Imaging Library (PIL) fork that adds support for opening, manipulating, and saving many different image file formats.
- **API Key:** for api key (Link: <https://ai.google.dev/gemini-api/docs/api-key>).

### 3. Interfacing with Pre-Trained Model:

- **Load The Gemini Pro API:**

```
### AI Nutritionist App
from dotenv import load_dotenv
load_dotenv() ## load all the environment variables
import streamlit as st
import os
import google.generativeai as genai
from PIL import Image
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
```

- the script starts by loading environment variables from a .env file using the load\_dotenv() function from the dotenv package.
- streamlit for creating the web app interface.
- os for accessing environment variables.
- google.generativeai for utilizing Google's Generative AI capabilities.
- PIL->image for image processing.
- The genai.configure() function is called to set up the google generative AI API with the API key retrieved from the environment variables, ensuring secure and authorized access to the AI services.

- **implementation:**

- ◆ **A Function to get Gemini Response**

```
def get_gemini_response(input, image, prompt):
    model = genai.GenerativeModel(model_name='gemini-1.5-flash')
    response=model.generate_content([input, image[0], prompt])
    return response.text
```

The function get\_gemini\_response takes an input text as a parameter. it calls the generate\_content method of the model object to generate a response.

The generated response is returned as text.

#### ◆ A function to read The Image And Set The Image Format for Gemini Model

Input:

code:

```
def input_image_setup(uploaded_file):  
    # Check if a file has been uploaded  
    if uploaded_file is not None:  
        # Read the file into bytes  
        bytes_data = uploaded_file.getvalue()  
        image_parts = [  
            {  
                "mime_type": uploaded_file.type, # Get the  
                mime type of the uploaded file  
                "data": bytes_data  
            }  
        ]  
        return image_parts  
    else:  
        raise FileNotFoundError("No file uploaded")
```

The function `input_image_setup` processes an uploaded image file for a health management application. It first checks if a file has been uploaded.

If a file is present, it reads the file's content into bytes and creates a dictionary containing the file's MIME type and its byte data. This dictionary is then stored in a list named `image_parts`, which is returned by the function.

If no file is uploaded, the function raises a `FileNotFoundError`, indicating that an image file is required but not provided.

This setup ensures that the uploaded image is correctly formatted and ready for further processing or analysis in the application.

#### ● Write A prompt For Gemini Model:

```

input_prompt="""
You are an expert in nutritionist where you need to see the food items from the image
and calculate the total calories, also provide the details of every food items with calories intake
is below format
1. Item 1 - no of calories
2. Item 2 - no of calories
----
----
After that mention that the meal is healthy meal or not and also mention the percentage split of ratio of
carbohydrates, proteins, fats, sugar and calories in meal.
finally give suggestion which item should be removed and which items should be added to meal to make the
meal healthy if it's unhealthy
"""

```

The variable `input_prompt` is a multi-line string designed as a prompt for a nutritionist AI model. It instructs the model to analyze an image of food items, identify each food item, and calculate the total calories. Additionally, the model is to provide a detailed breakdown of each food item with its respective calorie count. The expected output format is a numbered list where each item is listed alongside its calorie content, ensuring clarity and structured information for the user. This prompt is likely used in conjunction with an AI service that can process images and generate nutritional information based on the visual data provided.

## 4. Deployment:

We deploy our model using the Streamlit framework, a powerful tool for building and sharing data applications quickly and easily. With Streamlit, we can create interactive web applications that allow users to interact with our models in real-time, providing an intuitive and seamless experience.

- **integrate with web Framework:**

```

##initialize our streamlit app
st.set_page_config(page_title="AI Nutritionist App")
st.header("AI Nutritionist App 🍽️")
input=st.text_input("Input Prompt: ",key="input")
uploaded_file = st.file_uploader("Choose an image..", type=["jpg",
"jpeg", "png"])
image=""
if uploaded_file is not None:
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Image.",
    use_column_width=True)
submit=st.button("Tell me the total calories")

```

If "Tell me the total calories" button is clicked

```

if submit:
    image_data=input_image_setup(uploaded_file)
    #
    response=get_gemini_repsonse(input_prompt,image_data,input)
    st.subheader("The Response is")
    st.write(response)

```

Above code initializes a Streamlit application titled "AI Nutritionist App".It include a text input feild for users to enter a custom prompt and file upload an image in JPG, JPEG, or PNG format. If the image is uploaded, it is opened using the [PIL library](#) and displayed within the app with a caption.

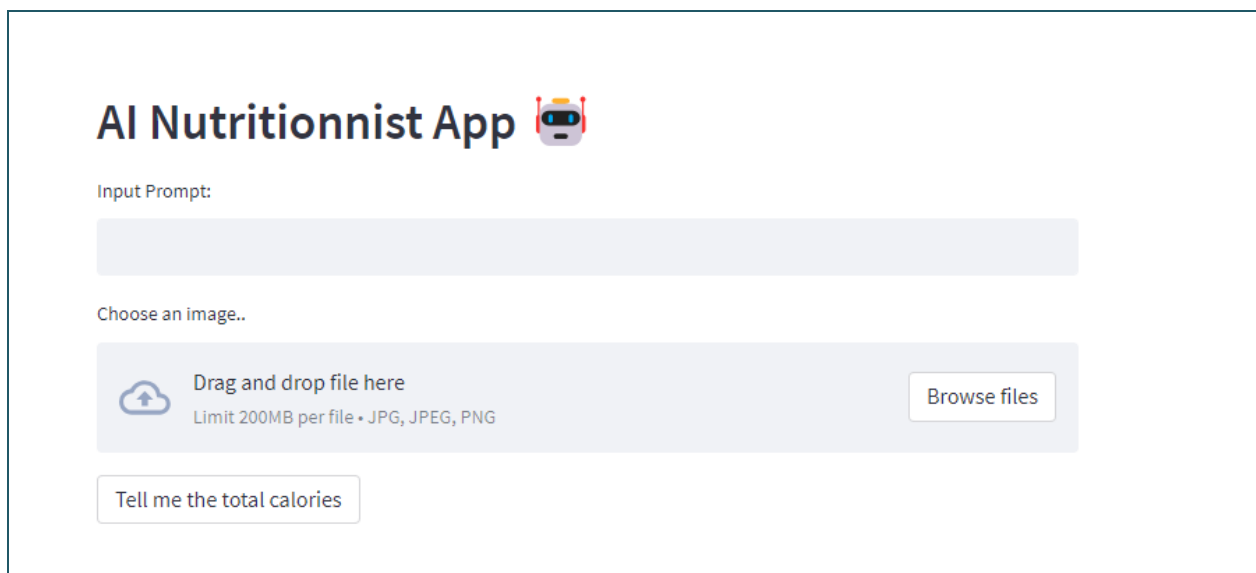
- **Host The Application:**

```
PS C:\Users\mehta\OneDrive\Desktop\nutrition app> streamlit run app.py
```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8503>

Network URL: <http://192.168.40.177:8503>



**AI Nutritionist App** 🤖

Input Prompt:

Choose an image..

Drag and drop file here  
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Tell me the total calories

# AI Nutritionnist App 🤖

Input Prompt:

tell me calories

Choose an image..



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



healthy2 meal.jpg 58.2KB



Uploaded Image.

## The Response is

The image shows a bowl with the following ingredients:

1. Red Cabbage - 20 calories
2. Green Lettuce - 10 calories
3. Cucumber - 15 calories
4. Carrots - 40 calories
5. Chickpeas - 120 calories
6. Green Peas - 40 calories
7. Roasted Sweet Potatoes - 180 calories
8. Brown Rice - 150 calories
9. Cashew nuts - 100 calories
10. Tomatoes - 20 calories

Total calories in the bowl - 695 calories

This is a healthy meal. It contains a good balance of carbohydrates, proteins, and fats.

The percentage split of the ratio of carbohydrates, proteins, fats, sugar, and calories in the meal is as follows:

- Carbohydrates: 50%
- Proteins: 15%
- Fats: 25%
- Sugar: 5%
- Calories: 100%

However, there are some items that can be removed or added to make this meal even healthier:

### To remove:

- Cashew nuts: They are high in calories and fat.
- Roasted Sweet Potatoes: It already has a high amount of carbs which might be more than necessary

### To add:

- Lean protein: Add some lean protein, such as grilled chicken or tofu.

By making these changes, you can create a healthy and balanced meal that will provide you with the



unhealthy meal.jpg 190.5KB



Uploaded Image.

Tell me the total calories



## The Response is

This meal is very unhealthy and should not be consumed on a regular basis. Here's a breakdown of calories and nutritional content, along with suggestions for making it healthier:

1. **Burger 1** - 500 calories
2. **Burger 2** - 500 calories
3. **Burger 3** - 500 calories
4. **French Fries 1** - 350 calories
5. **French Fries 2** - 350 calories
6. **Chicken Nuggets** - 250 calories
7. **Mayonnaise 1** - 100 calories
8. **Mayonnaise 2** - 100 calories
9. **Mustard** - 50 calories
10. **Ketchup** - 20 calories

**Total Calories:** Approximately 2620 calories

**Nutritional Breakdown (Approximate):**

- **Carbohydrates:** 60%

- **Protein:** 20%
- **Fats:** 20%
- **Sugar:** 10%

**Suggestions for a Healthier Meal:**

**Remove:**

- **French Fries:** They are high in unhealthy fats and calories.
- **Chicken Nuggets:** These are processed and high in saturated fat.
- **Mayonnaise:** This is high in calories and fat.

**Add:**

- **Green Salad:** A fresh salad with a light dressing provides essential vitamins, minerals, and fiber.
- **Grilled Chicken:** A lean protein source that can replace the burgers.
- **Baked Sweet Potato:** A healthier alternative to French fries with fiber and vitamins.

**Important Note:** This is an approximate calorie breakdown, and actual calorie content may vary depending on the ingredients and preparation methods. It's important to consult with a registered dietitian for personalized nutrition advice.