# Nutrition App Using Gemini Pro : Your Comprehensive Guide to Healthy Eating and Well-being

Mentored by :
Siri

Prepared by :
Yash Pal Chawla (21BIT0167)
Sidhant Kumar (21BIT0622)
Adarsh Ojaswi Singh (21BPS1200)
Deepika R (22BEC1522)

# Nutrition App Using Gemini Pro : Your Comprehensive Guide To Healthy Eating And Well-Being

## 1) Project Initialisation and planning phase

### Problem Statement :

In today’s fast-paced world, maintaining a balanced and healthy diet tailored to individual needs and health conditions poses significant challenges. People often struggle to find personalized dietary recommendations that suit their unique preferences, health goals, and lifestyles. This issue is compounded for individuals with specific health conditions such as diabetes or for those with distinct fitness goals like weight loss or muscle building. Traditional dietary planning methods are often generic and fail to offer the level of customization needed for effective nutritional management and well-being.

### Objectives :

The Nutritionist AI aims to address this problem by leveraging the advanced capabilities of the Gemini Pro model. The app utilizes artificial intelligence to analyze user data, dietary preferences, and health goals, providing personalized meal plans, nutritional insights, and wellness tips. By offering tailored recommendations, the app promotes healthier eating habits and improves overall well-being.

### Problem Statement simple explanation :

| Problem Statement | I am (customer) | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|---|
| PS -1 | A health-conscious individual. | Improve my dietary habits. | I struggle with meal planning. | I lack the necessary tools. | Frustrated and uncertain. |

## 2) Initial Project Planning

| Spirint | Functional Requirement | Task | Priority | Start date | End date |
|---|---|---|---|---|---|
| 1 | Required Specification | Creat a requirements txt file | Less | July 1 | July 2 |
| 1 | Required Specification | Installing required libraries | High | July 2 | July 3 |
| 2 | Initialization of google API | Generate google API key | High | July 4 | July 5 |
| 2 | Initialization of google API | Initialize google API key | Medium | July 6 | July 8 |
| 3 | Interfacing with pre-trianed model | Load the gemini pro API | Medium | July 9 | July 11 |
| 3 | Interfacing with pre-trianed model | Implementation of function to get gemini response | Medium | July 12 | July 13 |

| Spirint | Functional Requirement | Task | Priority | Start date | End date |
|---|---|---|---|---|---|
| 3 | Interfacing with pre-trianed model | Implementing a function to read image and setting the format | Medium | July 14 | July 15 |
| 3 | Interfacing with pre-trianed model | Writing prompt for gemini model | High | July 16 | July 18 |
| 4 | Model Deployment | Ingrating with web framework | Medium | July 18 | July 19 |
| 4 | Model Deployment | Host the application | High | July 19 | July 20 |

# 3) Proposed Solution report :

# Project Overview :

### Objective and Scope

The objective of the Nutritionist AI project is to develop an innovative mobile application that provides personalized dietary recommendations and nutritional advice using the advanced capabilities of the Gemini Pro model. By leveraging artificial intelligence, the app aims to analyze user data, dietary preferences, and health goals to deliver tailored meal plans, nutritional insights, and wellness tips. The scope of the project encompasses the development of a user-friendly interface, integration with fitness trackers, and real-time feedback mechanisms to ensure comprehensive nutritional management. Nutritionist AI is designed to cater to various user scenarios, including weight loss, diabetes management, and muscle building, thereby promoting healthier eating habits and improving overall well-being through intelligent and data-driven recommendations. The project will require skills in Python, deep learning, and Streamlit to achieve its goals, ensuring a seamless and effective user experience.

## Problem Statement :

### Description

Nutritionist AI is an innovative mobile application that leverages the advanced Gemini Pro model to provide personalized dietary recommendations and nutritional advice. By analyzing user data, dietary preferences, and health goals, the app delivers tailored meal plans, nutritional insights, and wellness tips. This AI-powered solution promotes healthier eating habits and enhances overall well-being through intelligent, data-driven recommendations.

### Impact

Nutritionist AI empowers users to make informed dietary choices, supporting diverse health goals such as weight loss, diabetes management, and muscle building. By offering personalized, real-time feedback and comprehensive nutritional management, the app fosters sustainable healthy eating habits and improves overall well-being. This approach can lead to better health outcomes and a higher quality of life for users.

# Problem Solution :

## Approach

Nutritionist AI utilizes the Gemini Pro model to analyze user data, including dietary preferences, health goals, and physical activity. The app employs deep learning algorithms to generate personalized meal plans and nutritional insights. It integrates seamlessly with fitness trackers to provide comprehensive and real-time feedback, ensuring that users receive tailored and effective dietary recommendations.

## Key Features

- **Personalized Meal Plans**: Customizable meal plans based on individual dietary preferences and health goals.
- **Real-Time Feedback**: Immediate insights on calorie intake, nutritional balance, and meal suitability.
- **Integration with Fitness Trackers**: Syncs with fitness devices to incorporate physical activity data for holistic recommendations.
- **Educational Resources**: Provides informative content on nutrition and wellness to empower users in making informed decisions.

# Resource Requirements

## Hardware Requirements

1. **Server/Cloud Infrastructure**:
    - **CPU**: Multi-core processor (e.g., Intel Xeon or AMD EPYC)
    - **RAM**: Minimum 16 GB, recommended 32 GB or more for handling multiple requests efficiently.
    - **Storage**: SSD with at least 100 GB of space for storing models, logs, and user data.
    - **GPU**: NVIDIA GPU with CUDA support (e.g., NVIDIA Tesla or RTX series) for faster processing of AI models.
2. **Development Machines**:
    - **CPU**: Modern multi-core processor (e.g., Intel Core i7 or AMD Ryzen 7)
    - **RAM**: Minimum 8 GB, recommended 16 GB
    - **Storage**: SSD with at least 256 GB of space
    - **GPU**: Optional, but recommended for local model testing

## Software Requirements

1. **Operating System**:
   - Linux (Ubuntu 18.04 or later)
   - Windows 10 or later (for development)
2. **Programming Languages**:
   - Python 3.8 or later
3. **Libraries and Frameworks**:
   - TensorFlow or PyTorch (for deep learning)
   - Flask or Django (for web framework)
   - Streamlit (for user interface)
   - Requests (for API calls)
   - PDFPlumber or PyPDF2 (for reading PDF content)
4. **APIs**:
   - Google Cloud API
   - Gemini Pro API
5. **Tools**:
   - Git (for version control)
   - Docker (for containerization)
   - Nginx or Apache (for web server)

## Data Requirements

### 1. User-Uploaded Images

- **Format**: Images should be in a standard format such as JPEG, PNG, or TIFF.
- **Resolution**: High enough resolution to accurately identify food items, typically at least 300 DPI.
- **Size**: The file size should be manageable, ideally under 5MB to ensure quick upload and processing times.

## Activities Breakdown

### Requirements Specification

- Define detailed functional and non-functional requirements.
- Create `requirements.txt` file listing all necessary libraries.

### Initialization of Google API Key

- Generate a Google API key from the Google Cloud Platform.
- Initialize the Google API key within the application.

### Interfacing with Pre-trained Model

- Load the Gemini Pro pre-trained model.
- Implement a function to interact with the Gemini Pro model API.
- Implement a function to read content from PDFs.
- Write a prompt for the Gemini Pro model to generate responses.

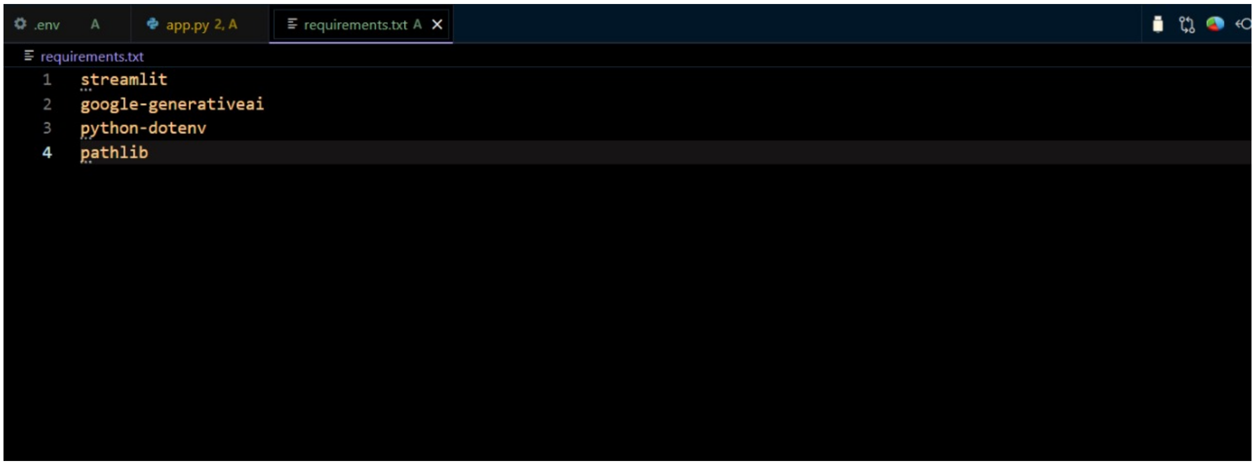
### Model Deployment

- Integrate with a web framework (Flask, Django, or Streamlit).
- Host the application on a server or cloud platform.
- Configure the web server (Nginx or Apache) to manage incoming requests.

By ensuring the proper hardware, software, and data requirements are met, this project can efficiently provide personalized dietary recommendations and nutritional advice through the Nutritionist AI application.

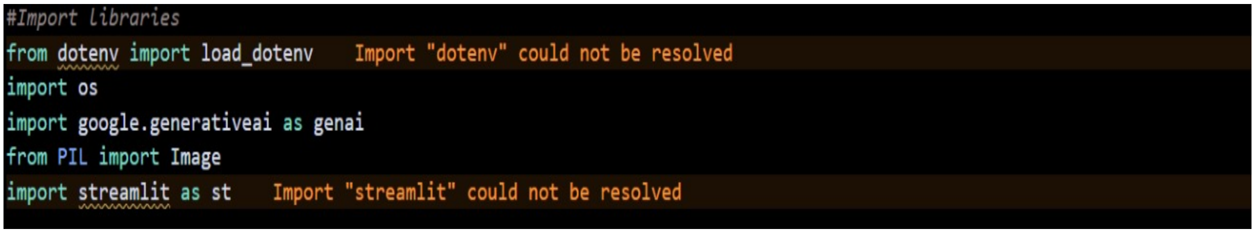
# 3) Nutrition AI application code :

Requirements.txt :



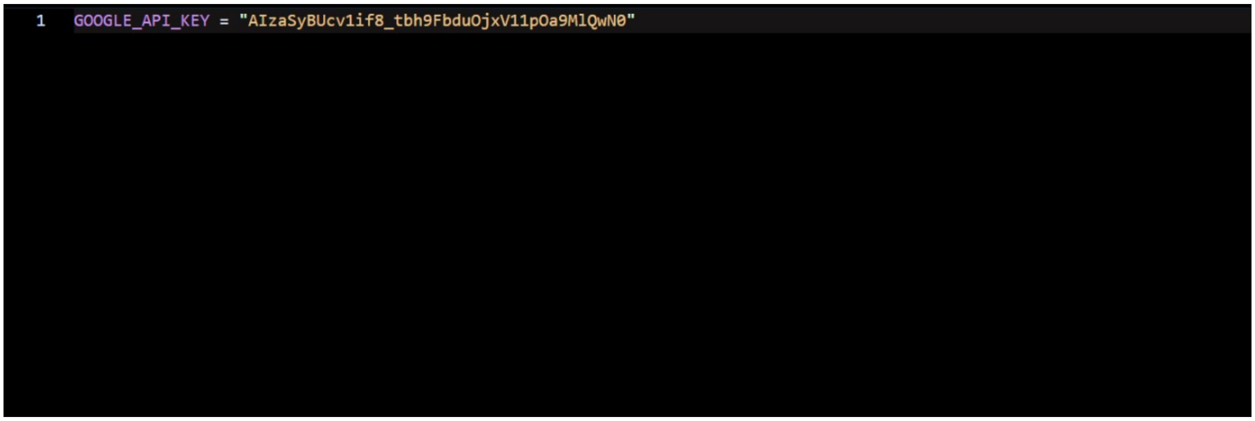
```
streamlit
google-generativeai
python-dotenv
pathlib
```

Used Libraries :



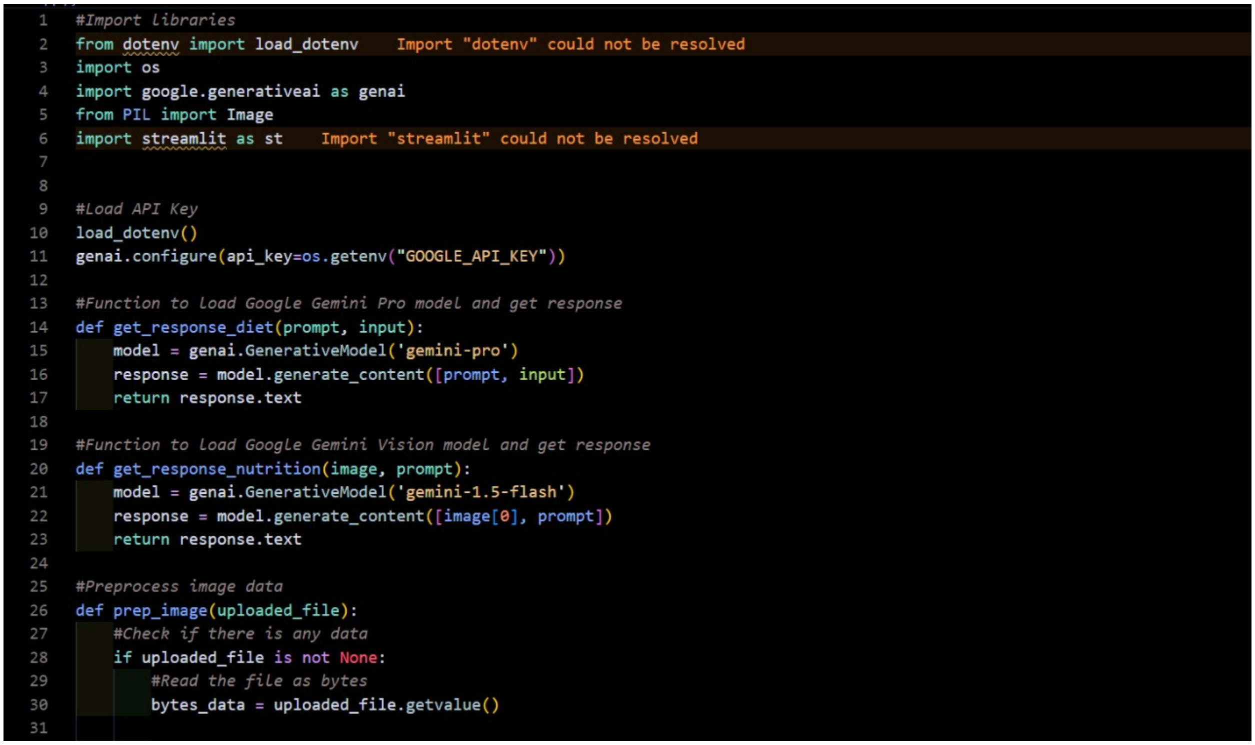
```
#Import libraries
from dotenv import load_dotenv    Import "dotenv" could not be resolved
import os
import google.generativeai as genai
from PIL import Image
import streamlit as st    Import "streamlit" could not be resolved
```

Google API key initialisation :

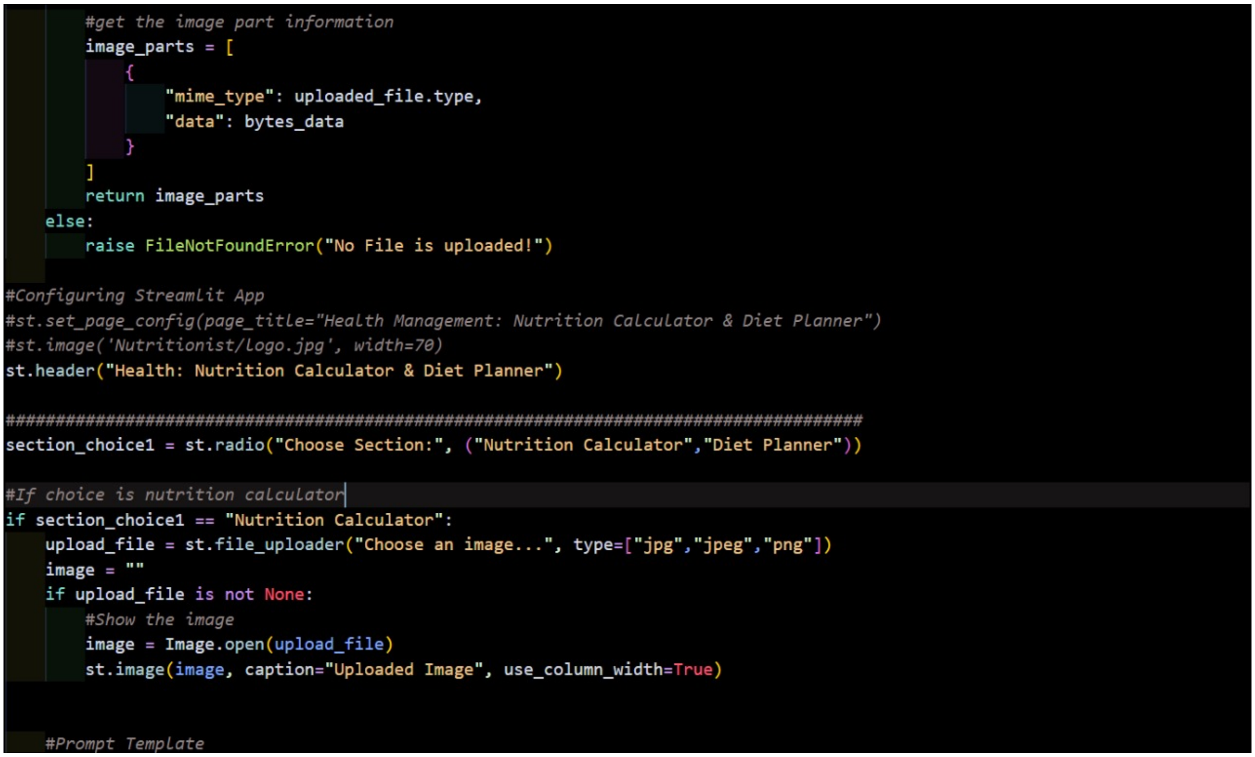


```
GOOGLE_API_KEY = "AIzaSyBUcv1if8_tbh9FbduOjxV11pOa9MlQwN0"
```

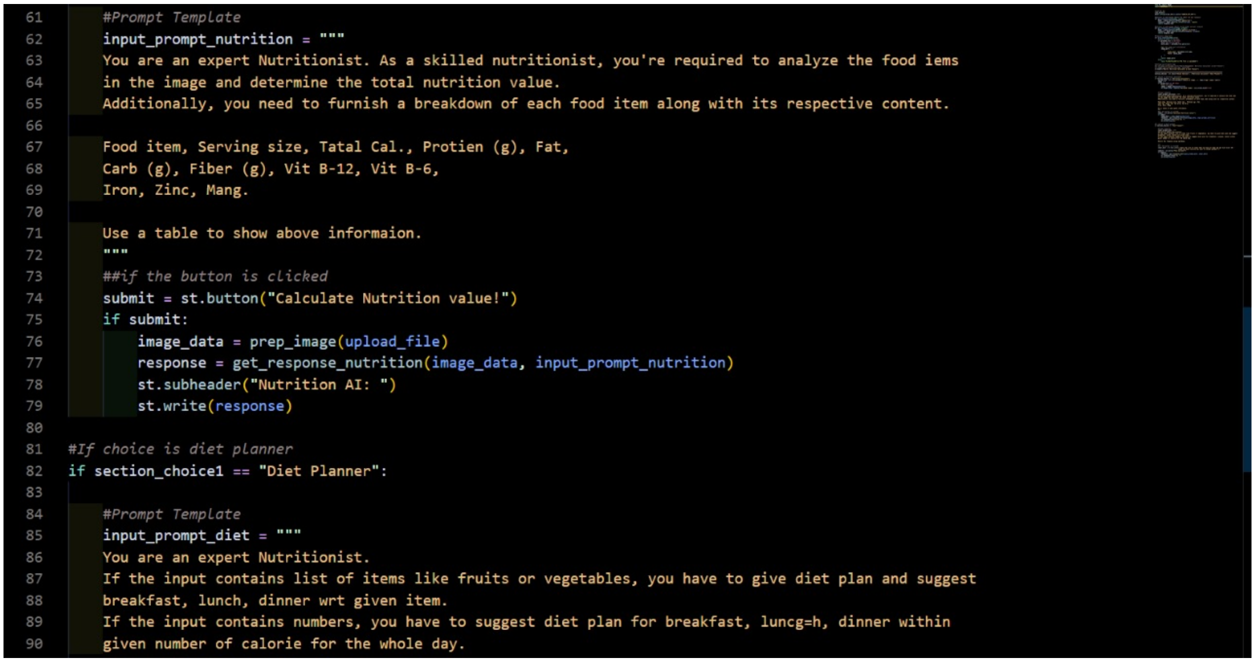
Function generation :



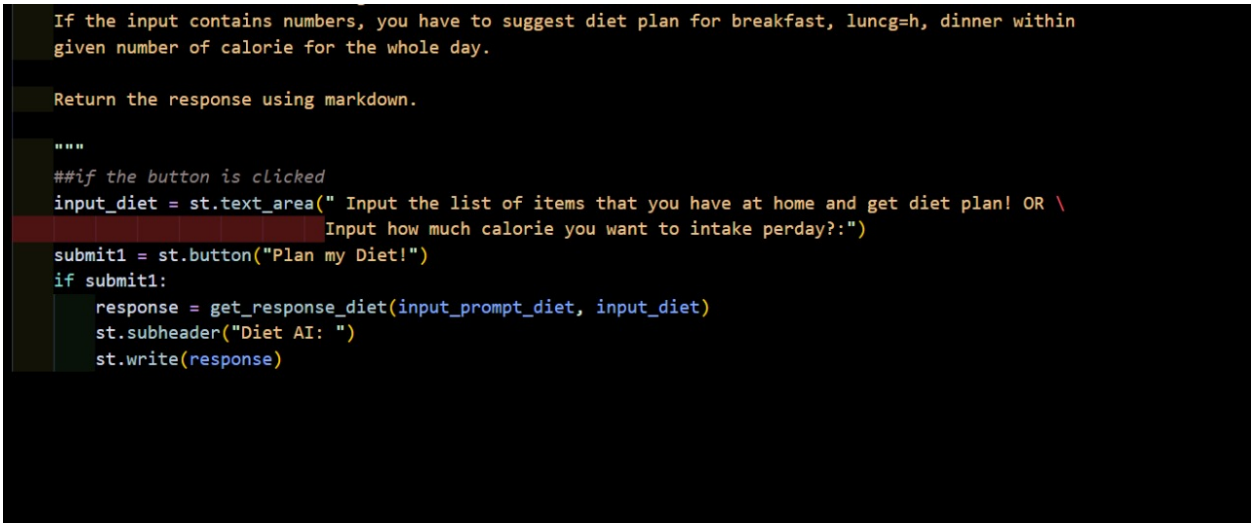
```
#Import libraries
from dotenv import load_dotenv    Import "dotenv" could not be resolved
import os
import google.generativeai as genai
from PIL import Image
import streamlit as st    Import "streamlit" could not be resolved


#Load API Key
load_dotenv()
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

#Function to load Google Gemini Pro model and get response
def get_response_diet(prompt, input):
    model = genai.GenerativeModel('gemini-pro')
    response = model.generate_content([prompt, input])
    return response.text

#Function to load Google Gemini Vision model and get response
def get_response_nutrition(image, prompt):
    model = genai.GenerativeModel('gemini-1.5-flash')
    response = model.generate_content([image[0], prompt])
    return response.text

#Preprocess image data
def prep_image(uploaded_file):
    #Check if there is any data
    if uploaded_file is not None:
        #Read the file as bytes
        bytes_data = uploaded_file.getvalue()

```



```
        #get the image part information
        image_parts = [
            {
                "mime_type": uploaded_file.type,
                "data": bytes_data
            }
        ]
        return image_parts
    else:
        raise FileNotFoundError("No File is uploaded!")

#Configuring Streamlit App
#st.set_page_config(page_title="Health Management: Nutrition Calculator & Diet Planner")
#st.image('Nutritionist/logo.jpg', width=70)
st.header("Health: Nutrition Calculator & Diet Planner")

#########################################################################################
section_choice1 = st.radio("Choose Section:", ("Nutrition Calculator","Diet Planner"))

#If choice is nutrition calculator
if section_choice1 == "Nutrition Calculator":
    upload_file = st.file_uploader("Choose an image...", type=["jpg","jpeg","png"])
    image = ""
    if upload_file is not None:
        #Show the image
        image = Image.open(upload_file)
        st.image(image, caption="Uploaded Image", use_column_width=True)


    #Prompt Template
```

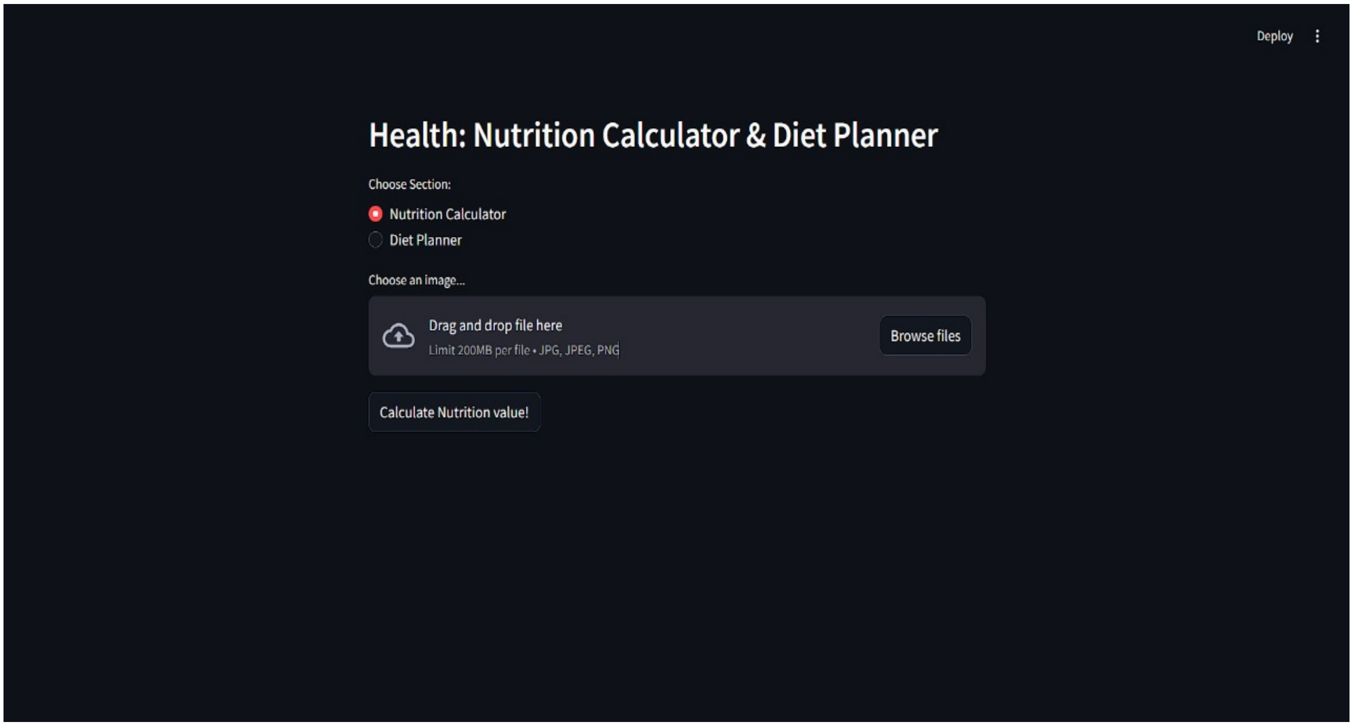


```
    #Prompt Template
    input_prompt_nutrition = """
    You are an expert Nutritionist. As a skilled nutritionist, you're required to analyze the food iems
    in the image and determine the total nutrition value.
    Additionally, you need to furnish a breakdown of each food item along with its respective content.

    Food item, Serving size, Tatal Cal., Protien (g), Fat,
    Carb (g), Fiber (g), Vit B-12, Vit B-6,
    Iron, Zinc, Mang.

    Use a table to show above informaion.
    """
    ##if the button is clicked
    submit = st.button("Calculate Nutrition value!")
    if submit:
        image_data = prep_image(upload_file)
        response = get_response_nutrition(image_data, input_prompt_nutrition)
        st.subheader("Nutrition AI: ")
        st.write(response)

#If choice is diet planner
if section_choice1 == "Diet Planner":

    #Prompt Template
    input_prompt_diet = """
    You are an expert Nutritionist.
    If the input contains list of items like fruits or vegetables, you have to give diet plan and suggest
    breakfast, lunch, dinner wrt given item.
    If the input contains numbers, you have to suggest diet plan for breakfast, luncg=h, dinner within
    given number of calorie for the whole day.
```



```
    If the input contains numbers, you have to suggest diet plan for breakfast, luncg=h, dinner within
    given number of calorie for the whole day.

    Return the response using markdown.

    """
    ##if the button is clicked
    input_diet = st.text_area(" Input the list of items that you have at home and get diet plan! OR \
                              Input how much calorie you want to intake perday?:")
    submit1 = st.button("Plan my Diet!")
    if submit1:
        response = get_response_diet(input_prompt_diet, input_diet)
        st.subheader("Diet AI: ")
        st.write(response)
```

# 4) Project Deployment :



## Health: Nutrition Calculator & Diet Planner

Choose Section:

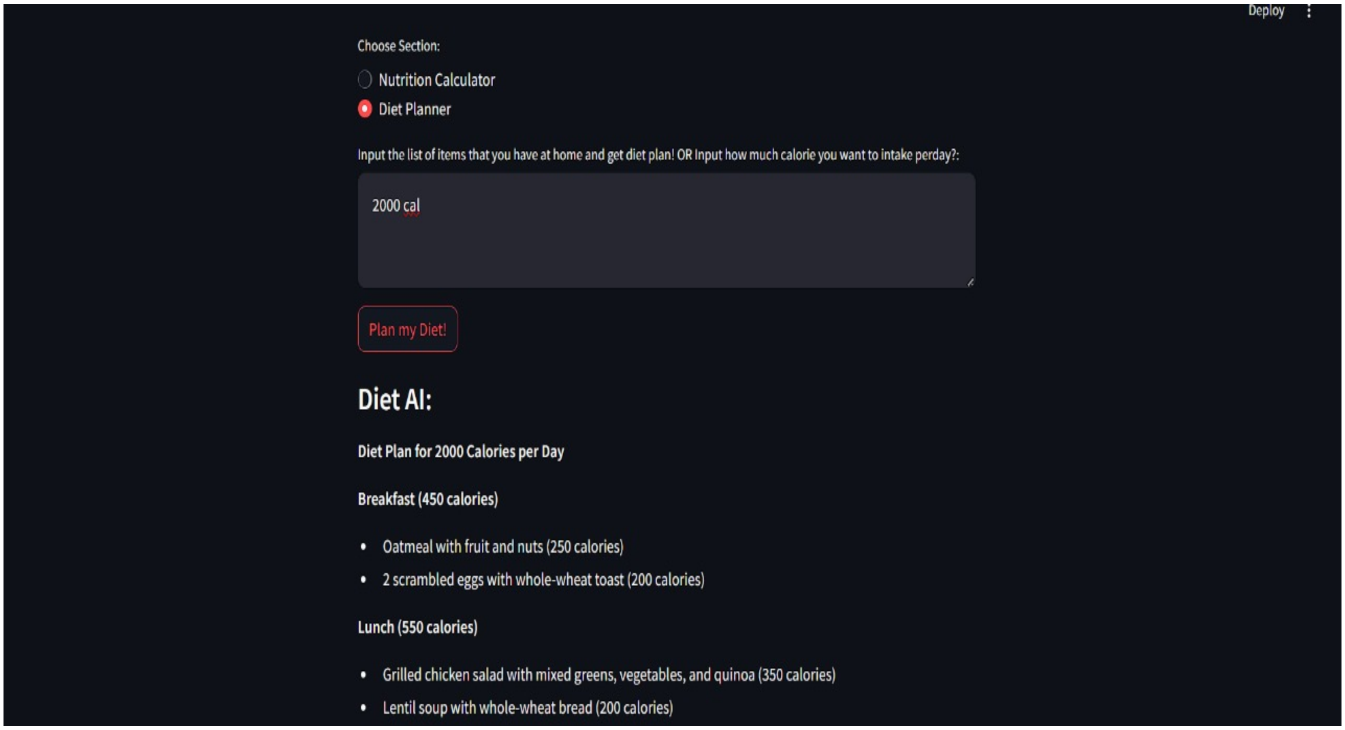
● Nutrition Calculator

○ Diet Planner

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG



Choose Section:

○ Nutrition Calculator

● Diet Planner

Input the list of items that you have at home and get diet plan! OR Input how much calorie you want to intake perday?:

2000 cal

### Diet AI:

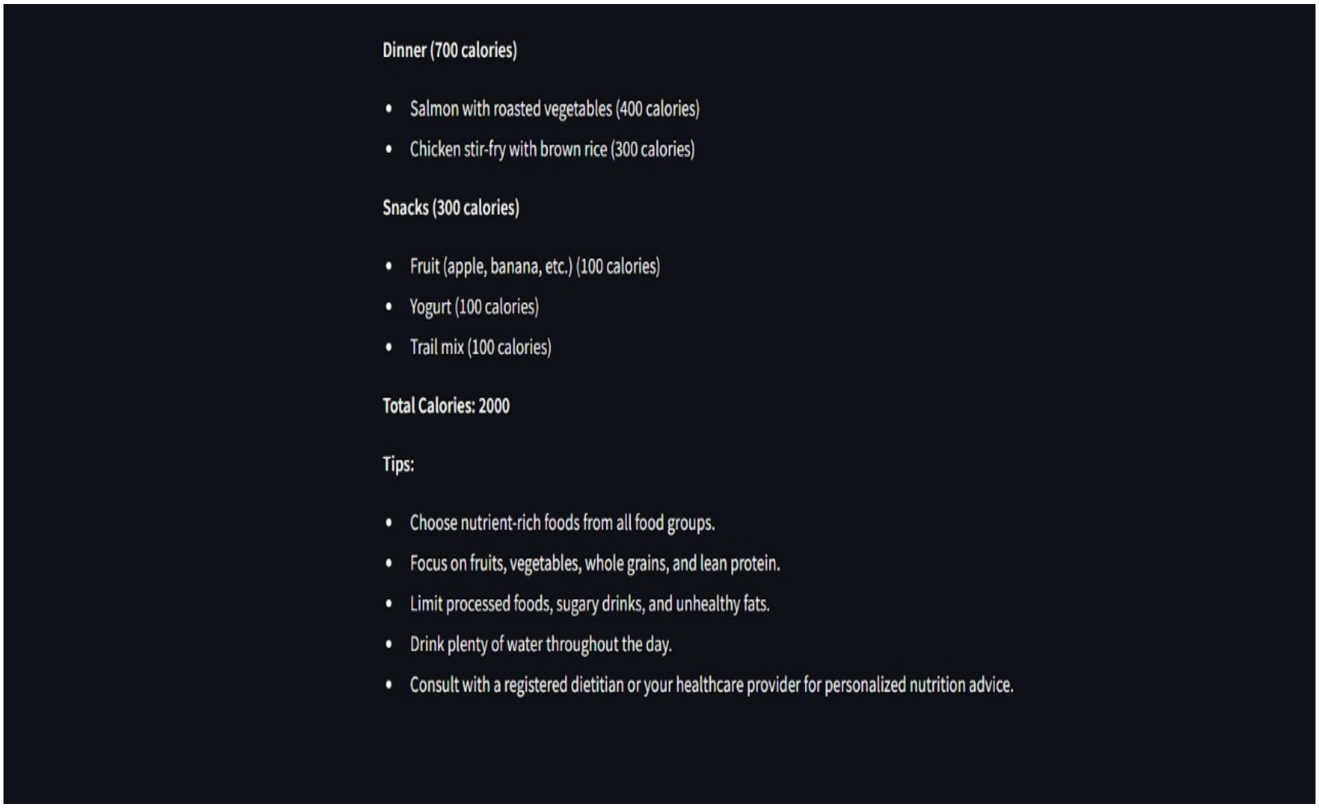
**Diet Plan for 2000 Calories per Day**

**Breakfast (450 calories)**

- Oatmeal with fruit and nuts (250 calories)
- 2 scrambled eggs with whole-wheat toast (200 calories)

**Lunch (550 calories)**

- Grilled chicken salad with mixed greens, vegetables, and quinoa (350 calories)
- Lentil soup with whole-wheat bread (200 calories)



**Dinner (700 calories)**

- Salmon with roasted vegetables (400 calories)
- Chicken stir-fry with brown rice (300 calories)

**Snacks (300 calories)**

- Fruit (apple, banana, etc.) (100 calories)
- Yogurt (100 calories)
- Trail mix (100 calories)

**Total Calories: 2000**

**Tips:**

- Choose nutrient-rich foods from all food groups.
- Focus on fruits, vegetables, whole grains, and lean protein.
- Limit processed foods, sugary drinks, and unhealthy fats.
- Drink plenty of water throughout the day.
- Consult with a registered dietitian or your healthcare provider for personalized nutrition advice.