SMART GEN AI PROJECT

AI NUTRIONIST

Project Overview

This project aims to create an AI-powered nutritionist application using Streamlit, Google Generative AI (Gemini), and a Google API key. The app will leverage Gemini's capabilities to analyze food images, estimate calorie counts, and provide nutritional information.

Technologies and Tools

- Streamlit: A Python library for building interactive web applications.
- Google Generative AI (Gemini): A powerful language model capable of generating text, translating languages, writing different kinds of creative content, and answering your questions in an informative way.
- Google Cloud Platform (GCP): To manage API keys and access Gemini.
- Python: The programming language used for development.
- PIL (Pillow): Python Imaging Library for image processing.

Project Structure

project_directory

- app.py: The main Python script containing the application logic.
- requirements.txt: A file listing the required Python packages.
- .env (optional): A file for storing sensitive information like the Google API key.

Step-by-Step Guide

1. **Install required packages:**

   pip install streamlit google-generativeai Pillow

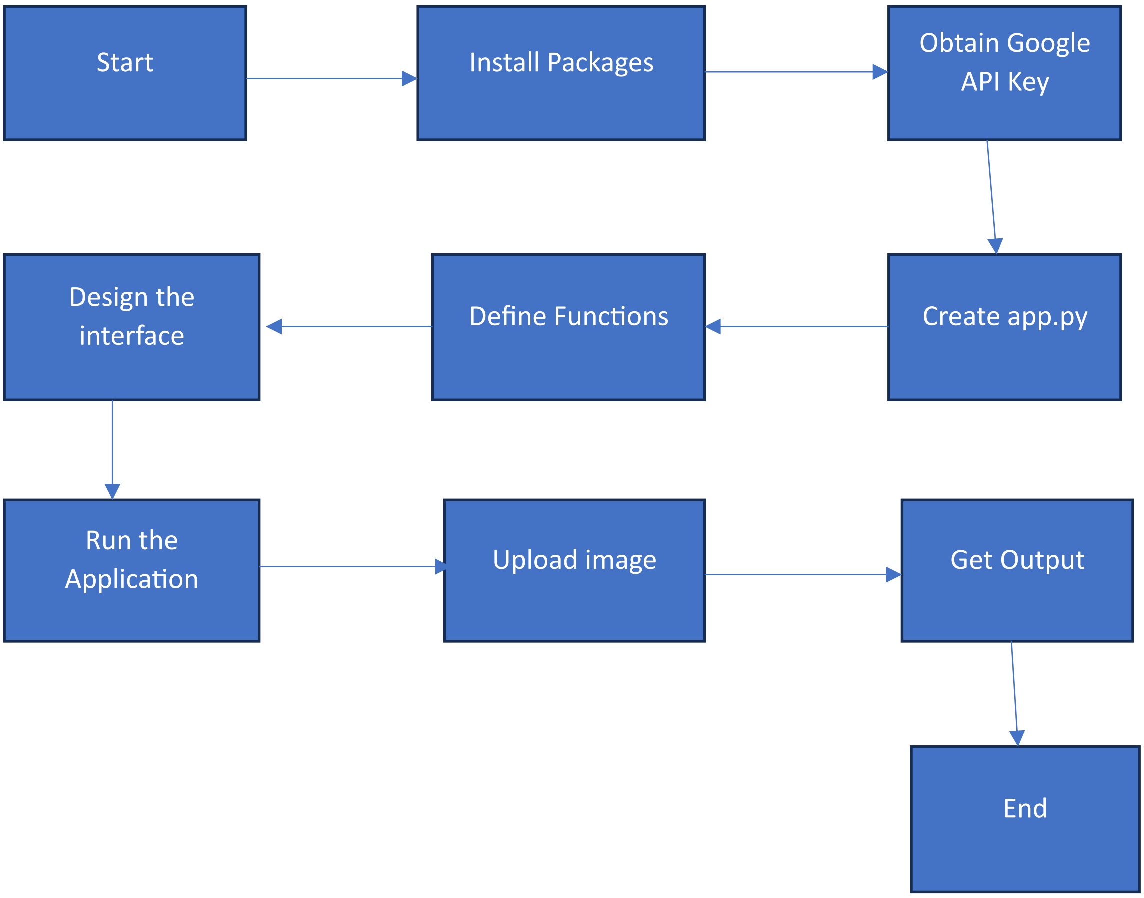
2. **Obtain Google API Key:**

   - Create a GCP project and enable the Generative AI API.
   - Generate a service account key and store it securely (e.g., in a .env file).

3. **Create Streamlit App:**
   - Create a new Python file (e.g., app.py) and import necessary libraries:
   - Configure Gemini:
   - Load the API key from the environment.
4. **Define Functions:**
   - Create functions to handle image upload, process the image with Gemini, and display the results.
5. **Build the Streamlit Interface:**
   - Use Streamlit components to create a user interface with input fields for image upload and a display area for the results.
6. **Run the Application:**
   - Execute the app.py script:

     streamlit run app.py

Flow Chart



Code

```
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"))


def get_gemini_repsonse(input, image, prompt):

        model=genai.GenerativeModel('gemini-pro-vision')

        response=model.generate_content([input, image [0], prompt])

        return response.text

def input_image_setup(uploaded_file):

        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

                                  "data": bytes_data

                                } ]

                return image_parts

        else:

                raise FileNotFoundError("No file uploaded") ##initialize our streamlit app

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

                - - - - -- -
```

```
                    - - - - -- -
"""


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:
        images = Image.open(uploaded_file)
        st.image(image, caption="Uploaded Image.", use_column_width=True)
submit=st.button("Tell me the total calories")


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)
```