|  |  |
| --- | --- |
| **Date** | 7 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**PROBLEM STATEMENT**

Many individuals find it difficult to track their daily calorie intake, especially when consuming meals with multiple food items. This lack of visibility into nutritional information affects people trying to maintain or improve their health, especially those managing weight, diabetes, or other dietary conditions. Manually logging each item is time-consuming and often inaccurate. A more intuitive and automated solution could drastically improve users’ health monitoring experiences and adherence to diet plans.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I am | I am trying to | But | Because | Which makes me feel |
| **A health-conscious individual** | Track calorie from food images | I don't know the names or calories of all food items | It's hard to manually estimate nutrition from diverse meals | Frustrated and less motivated to maintain a healthy diet |

|  |  |
| --- | --- |
| **Date** | 7 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**Product Backlog, Sprint Schedule, and Estimation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Priority** | **Team Members** | **Sprint Start Date** | **Sprint End Date (Planned)** |
| Sprint-1 | App Design & Environment Setup | AN-1 | Design UI layout using Streamlit | Medium | Yogesh | 2025/08/07 | 2025/08/10 |
| Sprint-1 | App Design & Environment Setup | AN-2 | Set up virtual environment & install dependencies | High | Yogesh | 2025/08/07 | 2025/08/10 |
| Sprint-2 | Gemini AI Integration | AN-3 | Configure Google Gemini API key | High | Yogesh | 2025/08/10 | 2025/08/12 |
| Sprint-2 | Gemini AI Integration | AN-4 | Write function to send image + prompt to Gemini | High | Yogesh | 2025/08/10 | 2025/08/12 |
| Sprint-2 | Gemini AI Integration | AN-5 | Parse response from Gemini and display output | Medium | Yogesh | 2025/08/10 | 2025/08/12 |
| Sprint-3 | Image Input & Preprocessing | AN-6 | Upload image and convert to correct format | Medium | Yogesh | 2025/08/12 | 2025/08/14 |
| Sprint-3 | Image Input & Preprocessing | AN-7 | Handle unsupported image types | Low | Yogesh | 2025/08/12 | 2025/08/14 |
| Sprint-4 | Testing & Deployment | AN-8 | Test app locally with different inputs | High | Yogesh | 2025/08/14 | 2025/08/16 |
| Sprint-4 | Testing & Deployment | AN-9 | Deploy app to Streamlit Cloud or local server | Medium | Yogesh | 2025/08/14 | 2025/08/16 |
| Sprint-4 | Documentation | AN-10 | Create user manual & final project documentation | Medium | Yogesh | 2025/08/14 | 2025/08/17 |

|  |  |
| --- | --- |
| **Date** | 7 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**Project Proposal (Proposed Solution) Report**

The proposal aims to transform dietary tracking using AI and computer vision. It simplifies nutritional monitoring by allowing users to upload food images and receive estimated calorie counts in real time. This solution enables healthier living through AI-driven analysis without the need for manual input. Key features include Google Gemini Vision API integration, real-time calorie analysis, and a user-friendly Streamlit interface.

**Project Overview**

**Objective**

To create an AI-powered web app that estimates total calories from food images by analyzing each food item using a multimodal model.

**Scope**

The project focuses on helping health-conscious users track calorie intake effortlessly using image analysis. It will integrate Google’s Gemini model and be built with Python and Streamlit, allowing real-time estimation with a simple user interface.

**Problem Statement**

**Description**

Manually tracking food intake is time-consuming and often inaccurate. Many users don’t know the nutritional values of diverse meals, especially when dining out or eating complex dishes. This affects health goals and dietary planning.

**Impact**

Solving this problem enhances user health tracking, improves dietary awareness, and promotes better decision-making by providing AI-driven, instant calorie breakdowns of meals using just an image.

**Proposed Solution**

**Approach**

Leverage Google Gemini’s multimodal capabilities to identify food items from uploaded images and generate accurate calorie estimates using a predefined prompt.

**Key Features**

* Integration with Gemini Vision API (Gemini 2.0 Flash)
* Upload any food image (jpg, jpeg, png)
* AI-generated itemized calorie report
* Streamlit-based user interface
* Real-time response and food visualization

**Resource Requirements**

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification / Allocation** |
| **Hardware** |  |  |
| Computing Resources | GPU for inference | Google-hosted Gemini API (cloud-based) |
| Memory | RAM | 8 GB (local) |
| Storage | Disk for logs and images | 100 GB SSD |
| **Software** |  |  |
| Frameworks | Python framework for web app | Streamlit |
| Libraries | Required packages | google.generativeai, PIL, dotenv, streamlit |
| Development Environment | IDE | VS Code / Jupyter Notebook |
| **Data** | Source for food analysis training | Google Gemini's pre-trained data |

|  |  |
| --- | --- |
| **Date** | 10 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**Model Development Plan & Integration Report**

This phase covers the development and integration of the **Google Gemini API** into the AI Nutritionist App for real-time food recognition and calorie estimation.  
Rather than building and training a custom model locally, the application leverages Gemini’s multi-modal capabilities to process both text and image inputs.

**Model Objective**

* Accept food images uploaded by the user.
* Identify all visible food items.
* Estimate per-item and total calorie content.
* Return results in a **structured, human-readable format**.

**Why Gemini?**

* **Multi-modal input**: Can process both image and text prompts together.
* **Pretrained** on diverse datasets, enabling recognition across cuisines.
* **Low latency**: Delivers results in seconds without heavy local computation.
* **Flexibility**: Output can be tailored using prompt engineering.

|  |  |
| --- | --- |
| **Date** | 10 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**System Architecture in app.py**

1. **Environment Configuration**
   * API key stored in .env file.
   * Loaded securely using dotenv.

1. **Prompt Engineering**
   * A **system prompt** clearly defines the role:
   * text
   * CopyEdit
   * You are an expert nutritionist. From the image, identify all the food items and:  
     - Estimate total calories  
     - List each item with estimated calories in this format:  
     1. Item 1 - X calories  
     2. Item 2 - Y calories  
     ...

1. **Image Preprocessing**
   * Uploaded file is read as raw bytes.
   * Converted into a format Gemini API accepts:
   * python
   * CopyEdit
   * image\_parts = [{  
         "mime\_type": uploaded\_file.type,  
         "data": uploaded\_file.getvalue()  
     }]

1. **Model Call**
   * Model: "models/gemini-2.0-flash".
   * Input: [system\_prompt, image\_parts[0], user\_extra\_text].
   * API returns .text with recognized items and calories.

1. **UI Integration (Streamlit)**
   * User uploads image via st.file\_uploader.
   * Gemini output displayed in st.write().

|  |  |
| --- | --- |
| **Date** | 10 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**Model Workflow**

**Step 1 – User Input**  
User uploads a JPG/PNG image of their meal and optionally provides extra instructions (e.g., “For 2 servings”).

**Step 2 – Prompt Assembly**  
The app combines:

* System nutritionist instructions.
* Uploaded image bytes.
* Optional user text.

**Step 3 – Gemini Inference**  
Gemini analyses the image and returns:

* List of detected food items.
* Estimated calories per item.
* Total estimated calories.

**Step 4 – Display Results**  
The structured response is shown directly in the Streamlit app.

|  |  |
| --- | --- |
| **Date** | 10 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

**Performance Targets**

* **Response Time**: ≤ 5 seconds for standard image.
* **Recognition Coverage**: Common global cuisines + Indian regional foods.
* **Calorie Accuracy**: Within ±15% of standard USDA values for well-lit images.

**Advantages of This Approach**

* **No model training required** — faster development cycle.
* **Cloud-based inference** — low local hardware requirements.
* **Scalable** — same app works for many users without local GPU.
* **Easily updatable** — improve accuracy by updating prompts.

|  |  |
| --- | --- |
| **Date** | 10 August 2025 |
| **Project Name** | Nutrition App using Gemini Pro |

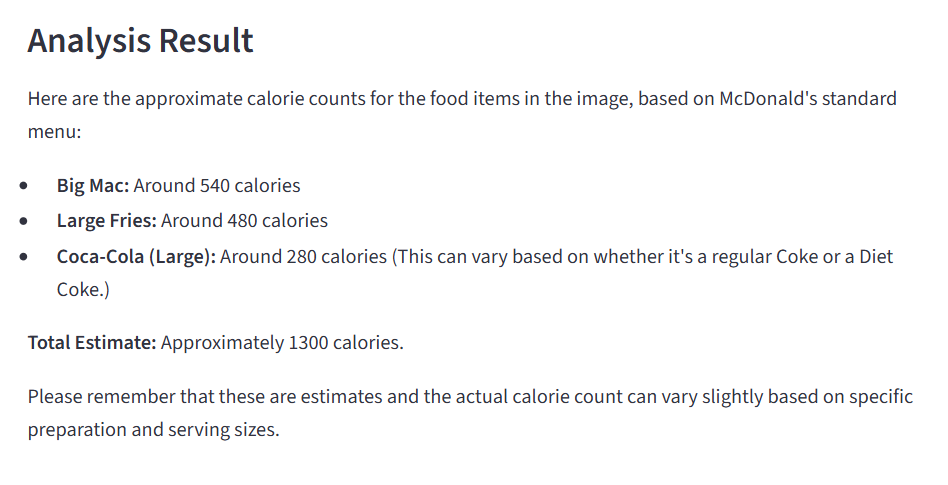
**OUTPUT:**

**Image:**



Then we click, tell me the calories button to check the calories in the food and give the prompt to tell the calories of foods in images.

**Output:**



Hence, the project completes.

The github link and implementation code of project is here: https://github.com/csdsyogesh/Doctor