

Final Project Report Template

1. Introduction
 - 1.1. Project overviews
 - 1.2. Objectives
2. Project Initialization and Planning Phase
 - 2.1. Define Problem Statement
 - 2.2. Project Proposal (Proposed Solution)
 - 2.3. Initial Project Planning
3. Data Collection and Preprocessing Phase
 - 3.1. Data Collection Plan and Raw Data Sources Identified
 - 3.2. Data Quality Report
 - 3.3. Data Preprocessing
4. Model Development Phase
 - 4.1. Model Selection Report
 - 4.2. Initial Model Training Code, Model Validation and Evaluation Report
5. Model Optimization and Tuning Phase
 - 5.1. Tuning Documentation
 - 5.2. Final Model Selection Justification
6. Results
 - 6.1. Output Screenshots
7. Advantages & Disadvantages
8. Conclusion
9. Future Scope
10. Appendix
 - 10.1. Source Code

10.2. GitHub & Project Demo Link

1. Introduction

1.1 Project Overview

A smartphone app called Nutritionist AI was created to give customers individualised food suggestions and nutritional guidance. It makes use of cutting-edge AI capabilities—more precisely, the Google AI Gemini Pro model—to evaluate user data and offer data-driven recommendations for reaching a range of wellness and health objectives.

1.2 Objectives

- **Personalized Nutrition:** Develop a system that analyzes user data (food intake, health goals, preferences) and provides tailored dietary recommendations using Gemini Pro's image recognition and text generation capabilities.
- **Dietary Analysis:** Utilize Gemini Pro to analyze food images captured by users, identifying food items and estimating calorie intake.
- **Goal-Focused Advice:** Craft personalized meal plans and suggestions aligned with user-specified health goals (weight management, disease prevention, improved energy levels).
- **Nutritional Education:** Integrate educational resources and answer user questions about nutrition within the app, leveraging Gemini Pro's text generation capabilities.
- **Accessibility and Convenience:** Design a user-friendly mobile application that is accessible and convenient for users to integrate into their daily routines.

2. Project Initialization and Planning Phase

2.1 Define Problem Statement:

People sometimes find it difficult to understand complicated nutritional information and adopt wholesome eating habits. Guidance that is accessible and customised to meet the needs and objectives of each individual is required. Conventional nutritional counselling can be costly and time-consuming.

2.2 Project Proposal (Proposed Solution):

Nutritionist AI resolves these difficulties by offering a smartphone application that serves as a virtual nutritionist. The app uses powerful AI and user data to provide specific dietary advice and educational tools, supporting educated and healthy eating habits.

2.3 Initial Project Planning:

- **Target Audience:** Health-conscious individuals seeking personalized dietary guidance for weight management, disease prevention, or overall health improvement.
- **Technology Stack:** Mobile app development framework (e.g., Streamlit Native, Flutter), Google Cloud Platform for deploying Gemini Pro, secure database for user information.
- **Project Timeline:** Define development milestones for data collection, model training, app development, and testing phases.
- **Resource Allocation:** Allocate resources for development, cloud computing costs, and ongoing maintenance.

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources

Identified:

- **User Data:** Develop a user onboarding process collecting information on demographics, health goals, dietary preferences (allergies, dislikes), and activity levels.
- **Food Images:** Allow users to capture images of their meals using the mobile app's camera functionality.
- **Nutritional Information:** Utilize established nutritional databases (e.g., USDA database) to access detailed information on various food items.

3.2 Data Quality Report:

Define strategies to address potential data quality issues:

- **Inaccuracy in User Input:** Implement validation checks for user-entered data and provide educational resources on portion control.
- **Missing Data:** Develop strategies for handling missing data points (e.g., imputing average values, prompting users for clarification).
- **Image Quality Issues:** Implement image pre-processing techniques (resizing, noise reduction) to ensure optimal analysis by Gemini Pro.

- **Third-Party Data Inconsistencies:** Validate nutritional data against established databases and allow users to flag inconsistencies.

3.3 Data Preprocessing:

- **User Data:** Cleanse and format user-entered data for compatibility with Gemini Pro's input requirements.
- **Food Images:** Preprocess food images by resizing, applying noise reduction, and color space conversion for consistent analysis.
- **Nutritional Information:** Standardize and format nutritional data from various sources for integration with user data and recommendations.

4. Model Development Phase

4.1 Model Selection Report:

Evaluate potential AI models for the project, considering factors like performance, complexity, and suitability for mobile application integration. Analyze these models:

- **Gemini Pro (Baseline):** Pre-trained model offering object recognition and text generation capabilities, potentially reducing custom training needs.

4.2 Initial Model Training Code, Model Validation and Evaluation Report:

- Develop an initial model training script utilizing the chosen model (e.g., Gemini Pro) and the preprocessed data.

5. Results

5.1 Output Screenshots

Educational Resources

1. [Balanced Diets](#)

2. [Healthy Eating Tips](#)

3. [Macronutrients and Micronutrients](#)

Gemini Health App

Choose an image...

Drag and drop file here


Limit: 200MB per file • JPG, JPEG, PNG

Browse files

indiafood.jpg

10.7KB

×




← Manage app

Educational Resources

1. [Balanced Diets](#)

2. [Healthy Eating Tips](#)

3. [Macronutrients and Micronutrients](#)



Uploaded image.

Ask a question about the food in the image:

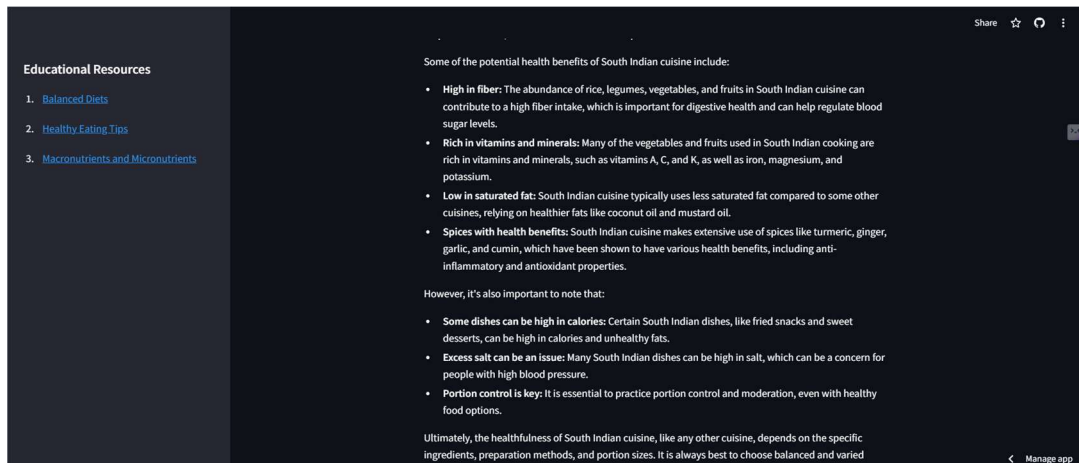
is this food is good to health

Analyze Image

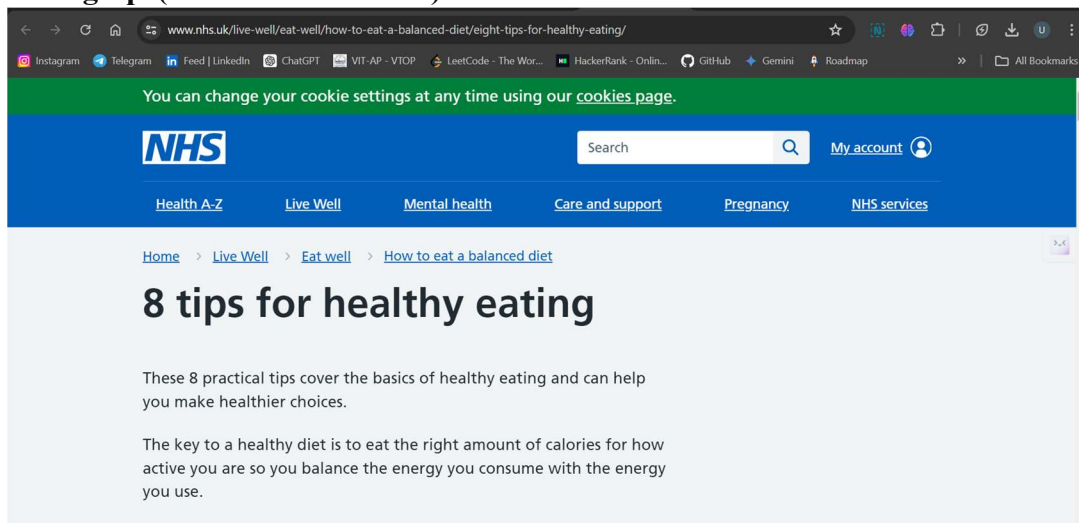
Analysis Results:

← Manage app

1. Rice - 200 calories
2. Sambar - 100 calories
3. Rasam - 50 calories
4. Avial - 100 calories
5. Thoran - 100 calories
6. Olan - 100 calories
7. Puliserry - 100 calories
8. Moru Kachiyathu - 100 calories
9. Banana - 100 calories
10. Buttermilk - 50 calories



Eating tips(Education resource):



6. Advantages and Disadvantages

Advantages

- **Personalized approach:** Utilizing user data on health goals, dietary preferences, and food intake (through image recognition), Nutritionist AI tailors recommendations to individual needs. This personalized approach goes beyond one-size-fits-all dietary plans, promoting sustainable and effective behavior change.
- **AI-powered analysis:** The Gemini Pro model analyzes food images captured by users, providing insights into calorie content and potential ingredient breakdowns. This eliminates the need for manual calorie counting and fosters a more accurate understanding of dietary intake.
- **Educational guidance:** Nutritionist AI doesn't just provide recommendations; it also educates users about healthy eating

practices. The Gemini Pro model can answer user questions about specific foods or dietary strategies, promoting long-term knowledge and empowering informed decision-making.

- **Accessibility and convenience:** As a mobile application, Nutritionist AI is readily available anytime, anywhere. This accessibility removes geographical barriers and allows users to receive on-demand nutritional support, fostering consistency and adherence to dietary plans.
- **Data-driven insights:** By collecting and analyzing user data, Nutritionist AI provides valuable insights into dietary patterns and progress over time. This data can be used to adjust recommendations and personalize the user experience further, leading to better long-term outcomes.

Disadvantages

- **Data privacy concerns:** Collecting and analyzing user data, including food images, raises privacy concerns. Nutritionist AI needs to prioritize robust data security measures and ensure user data remains anonymized and secure.
- **Accuracy limitations:** While the Gemini Pro model is powerful, it relies on the accuracy of image recognition and user-provided information. Factors like lighting and image quality can affect recognition, and users might unintentionally misreport dietary intake.
- **Limited scope:** While Nutritionist AI provides valuable guidance, it's not a substitute for professional medical advice. Users with specific health conditions might require personalized consultations with doctors or registered dietitians for tailored dietary plans.
- **Computational requirements:** The underlying AI model, Gemini Pro, might have specific computational requirements for running on mobile devices. Battery life and processing power optimization are crucial for a seamless user experience.
- **Accessibility limitations:** Despite the mobile platform, access to smartphones and internet connectivity might vary among users. Addressing accessibility barriers through alternative interfaces or offline functionalities can be a future consideration.

7. Conclusion

Nutritionist AI proposes a potential method regarding individual nutrition advice. It simplifies nutritional tracking, provides educational suggestions, and enables users to make more educated food choices by employing powerful AI capabilities. Addressing data privacy issues, properly communicating accuracy limitations, and identifying the

need for expert medical guidance in specific circumstances are all critical components of responsible development. Overall, Nutritionist AI having the ability to promote better lives and help people accomplish their wellness objectives through targeted nutrition.

8. Future Scope

- **Integration with wearable devices:** Integrating with wearable devices like fitness trackers or smartwatches can provide a more holistic picture of health data, enabling the app to personalize recommendations further.
- **Advanced recipe generation:** By incorporating user preferences and dietary restrictions, the app could generate personalized recipe suggestions, bridging the gap between recommendations and implementation.
- **Community features:** Building a supportive online community within the app can offer users a platform to share experiences, find motivation, and connect with others on similar wellness journeys.
- **Multilingual support:** Expanding language capabilities can make Nutritionist AI accessible to a wider audience and remove cultural barriers to healthy eating habits.
- **Offline functionality:** Developing functionalities that work without an internet connection can cater to users in areas with limited connectivity, ensuring continuous access to basic guidance.

10. Appendix

10.1 Source Code

```
import streamlit as st
from dotenv import load_dotenv
import os
import google.generativeai as genai
from PIL import Image
from pathlib import Path

# Load environment variables (likely your Google API key)
load_dotenv()
```

```

api_key = os.getenv("GOOGLE_API_KEY")

# Check if the API key is set; if not, display an error message
if not api_key:
    st.error("API key is not set. Please check your .env file.")
else:
    # Configure the generative AI model with the provided API
    key
    genai.configure(api_key=api_key)

    # Function to get the response from the generative AI
    model
    def get_gemini_response(image_data, input_prompt):
        model = genai.GenerativeModel('gemini-1.5-flash')
        response = model.generate_content([image_data[0],
input_prompt])
        return response.text

    # Function to process the uploaded image
    def input_image_setup(uploaded_file):
        if uploaded_file is not None:
            bytes_data = uploaded_file.getvalue()
            mime_type = uploaded_file.type
            image_parts = [
                {
                    "mime_type": mime_type,
                    "data": bytes_data
                }
            ]
            return image_parts
        else:
            raise FileNotFoundError("No file uploaded")

    # Set the page configuration for the Streamlit app
    st.set_page_config(page_title="Gemini Health App")
    st.header("Gemini Health App")

```

```

# File uploader widget for users to upload an image
uploaded_file = st.file_uploader("Choose an image...",
type=["jpg", "jpeg", "png"])
if uploaded_file is not None:
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Image.",
use_column_width=True)

# Predefined prompt for calorie calculation
calorie_prompt = """
You are an expert nutritionist analyzing the food items in
the image.
Please calculate the total calories and provide details of
each item,
with calorie intake, in the following format:

1. Item 1 - Calories
2. Item 2 - Calories
...
"""

# Text area for users to ask questions about the food in the
image
food_question = st.text_area("Ask a question about the food
in the image:", key="food_question")

# Button to analyze the uploaded image
if st.button("Analyze Image"):
    if uploaded_file is not None:
        try:
            image_data = input_image_setup(uploaded_file)

            # Get the response for the calorie prompt
            calorie_response =
get_gemini_response(image_data, calorie_prompt)
            st.session_state.calorie_response = calorie_response

```

```

        # Get the response for the user's question
        question_response = None
        if food_question:
            question_response =
get_gemini_response(image_data, food_question)
            st.session_state.question_response =
question_response
        else:
            st.session_state.question_response = None

        # Display the analysis results
        st.subheader("Analysis Results:")
        st.write("Total Calories:")
        st.write(calorie_response)
        if question_response:
            st.write("Answer to your question:")
            st.write(question_response)

    except FileNotFoundError as e:
        st.error(str(e))
    else:
        st.error("Please upload an image.")

# Button to save the report
if st.button("Save Report"):
    if 'calorie_response' in st.session_state:
        try:
            home_dir = Path.home()
            report_path = home_dir / "nutrition_report.txt"
            with open(report_path, "w") as report_file:
                report_file.write("Total Calories:\n")
                report_file.write(st.session_state.calorie_respons
e + "\n")
                if 'question_response' in st.session_state and
st.session_state.question_response:
                    report_file.write("\nAnswer to your
question:\n")

```

```

report_file.write(st.session_state.question_response)

st.sidebar.write(f'Report saved successfully to {report_path}!')
except Exception as e:
    st.error(f'Error saving the report: {str(e)}')
else:
    st.error("Please analyze an image first to generate a report.")

# Additional sidebar functionalities
st.sidebar.header("Educational Resources")
st.sidebar.write("1. [Balanced Diets](https://www.nhs.uk/live-well/eat-well/how-to-eat-a-balanced-diet/eating-a-balanced-diet/)")
st.sidebar.write("2. [Healthy Eating Tips](https://www.nhs.uk/live-well/eat-well/how-to-eat-a-balanced-diet/eight-tips-for-healthy-eating/)")
st.sidebar.write("3. [Macronutrients and Micronutrients](https://www.healthline.com/nutrition/micronutrients)")

```

10.2 GitHub & Project Demo Link

GitHub:

<https://github.com/uttejkuvaro/NutritionAppUsingStreamlit-GenAI.git>

Project Demo Link:

<https://drive.google.com/file/d/1LUKzavQrk7YSJacBHPf1BbTQ6wzxRLiJ/view?usp=sharing>

Project Link: <https://nutritionalapp-genai.streamlit.app/>

