INDUSTRIAL ORIENTED MINI PROJECT

Report

On

NUTRIGUIDE.AI: PERSONALIZED NUTRITION AND HEALTH

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

By

K Tanvish Reddy - 22261A1228

V Sai Ruthik - 22261A1263

Under the guidance of

B. Meenakshi

Assistant Professor, Department of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to JNTUH, Hyderabad; Eight UG Programs Accredited by NBA; Accredited by NAAC with 'A++' Grade)

Gandipet, Hyderabad, Telangana, Chaitanya Bharati (P.O), Ranga Reddy District, Hyderabad– 500075, Telangana

2024-2025

CERTIFICATE

This is to certify that the **Industrial Oriented Mini Project** entitled **NUTRIGUIDE.AI: PERSONALIZED NUTRITION AND HEALTH MANAGEMENT** submitted by **K Tanvish Reddy(22261A1228), V Sai Ruthik (22261A1263)** in partial fulfillment of the requirements for the Award of the Degree of Bachelor of Technology in Information Technology as specialization is a record of the bona fide work carried out under the supervision of **B Meenakshi**, and this has not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Supervisor:	**IOMP Supervisor:**
B Meenakshi	**Dr. U. Chaitanya**
Assistant Professor	Assistant Professor
Dept. of IT	Dept. of IT

EXTERNAL EXAMINAR	**Dr. D. Vijaya Lakshmi**
	Professor and HOD
	Dept. of IT

DECLARATION

We hear by declare that the **Industrial Oriented Mini Project** entitled **NUTRIGUIDE.AI: PERSONALIZED NUTRITION AND HEALTH MANAGEMENT** is an original and bona fide work carried out by us as a part of fulfilment of Bachelor of Technology in Information Technology, Mahatma Gandhi Institute of Technology, Hyderabad, under the guidance of **B Meenakshi , Assistant Professor**, Department of IT, MGIT.
No part of the project work is copied from books /journals/ internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the project work done entirely by us and not copied from any other source.

K Tanvish Reddy - 22261A1228
V Sai Ruthik- 22261A1263

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the **Industrial Oriented Mini Project**.

We would like to express our sincere gratitude and indebtedness to our project guide **B Meenakshi , Assistant Professor**, Dept. of IT, who has supported us throughout our project with immense patience and expertise.

We are also thankful to our honourable Principal of MGIT **Prof. G. Chandramohan Reddy** and **Dr. D. Vijaya Lakshmi**, Professor and HOD, Department of IT, for providing excellent infrastructure and a conducive atmosphere for completing this **Industrial Oriented Mini Project** successfully.

We are also extremely thankful to our IOMP supervisor **Dr. U. Chaitanya**, Assistant Professor, Department of IT, and senior faculty **Mrs. B. Meenakshi** Department of IT for their valuable suggestions and guidance throughout the course of this project.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support all through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support for completion of this work.

K Tanvish Reddy - 22261A1228
V Sai Ruthik- 22261A1263

ABSTRACT

NutriGuide.AI is an innovative, AI-driven nutritional platform developed to assist individuals in making informed, personalized decisions about their diet and overall health. Leveraging the power of generative artificial intelligence and real-time data analysis, NutriGuide.AI offers a comprehensive suite of tools that support users in achieving their unique health goals—whether they aim to lose weight, build muscle, manage chronic conditions, or maintain a balanced lifestyle.

The platform provides customized meal plans based on personal health profiles, dietary preferences, and restrictions, ensuring users receive relevant and sustainable nutritional guidance. NutriGuide.AI features real-time nutritional feedback that analyzes food intake, helping users make immediate and informed choices. A key highlight is its image-based nutrient analysis, which uses AI-powered image recognition to scan and evaluate meals for accurate tracking of nutrient content.

To promote safe dietary practices, the platform includes food allergen detection, alerting users to ingredients that may trigger allergies. The system also offers dedicated support for individuals managing chronic health issues such as diabetes, hypertension, and cardiovascular diseases, providing tailored dietary advice aligned with medical guidelines.

With interactive tracking features, users can monitor their daily and weekly nutritional intake, assess progress, and adjust goals accordingly. By combining accessibility, scientific credibility, and user-friendly design, NutriGuide.AI serves as a reliable digital companion for anyone committed to leading a healthier, more informed lifestyle.

TABLE OF CONTENTS

Chapter No	Title	Page No
	CERTIFICATE	**i**
	DECLARATION	**ii**
	ACKNOWLEDGEMENT	**Iii**
	ABSTRACT	**iv**
	TABLE OF CONTENTS	**v**
	LIST OF FIGURES	**Vii**
	LIST OF TABLES	**viii**
1	**INTRODUCTION**	**1**
	1.1 MOTIVATION	1
	1.2 PROBLEM STATEMENT	2
	1.3 EXISTING SYSTEM	2
	1.3.1 LIMITATIONS	3
	1.4 PROPOSED SYSTEM	4
	1.4.1 ADVANTAGES	5
	1.5 OBJECTIVES	5
	1.6 HARDWARE AND SOFTWARE REQUIREMENTS	6
2	**LITERATURE SURVEY**	9
3	**ANALYSIS AND DESIGN**	12
	3.1 MODULES	13
	3.2 ARCHITECTURE	15
	3.3 UML DIAGRAMS	16
	3.3.1 USE CASE DIAGRAM	16
	3.3.2 CLASS DIAGRAM	18
	3.3.3 ACTIVITY DIAGRAM	21

Chapter No	**Title**	**Page No**
	3.3.4 SEQUENCE DIAGRAM	24
	3.3.5 COMPONENT DIAGRAM	27
	3.3.6 DEPLOYMENT DIAGRAM	29
	3.4 METHODOLOGY	32
4	**CODE AND IMPLEMENTATION**	**35**
	4.1 CODE	35
	4.2 IMPLEMENTATION	54
5	**TESTING**	**56**
	5.1 INTRODUCTION TO TESTING	56
	5.2 TEST CASES	57
6	**RESULTS**	58
7	**CONCLUSION AND FUTURE ENHANCEMENTS**	65
	7.1 CONCLUSION	**65**
	7.2 FUTURE ENHANCEMENTS	65
	REFERENCES	67

LIST OF FIGURES

Fig. 3.2.1 Architecture of Nutriguide.AI	13
Fig. 3.3.1.1 Use Case Diagram	15
Fig. 3.3.2.1 Class Diagram	16
Fig. 3.3.3.1 Activity Diagram	19
Fig. 3.3.4.1 Sequence Diagram	21
Fig. 3.3.5.1 Component Diagram	24
Fig. 6.1 Personalized Meal Plan Input Interface in NutriGuide.AI	58
Fig. 6.2 Personalized Meal Plan Output Interface in NutriGuide.AI	59
Fig. 6.3 One day Personalized Meal Plan Output in NutriGuide.AI	59
Fig. 6.4 Food Nutrient and Calorie Estimator interface in NutriGuide.AI	60
Fig. 6.5 Nutrient Analysis Output of Food image in NutriGUuide.AI	60
Fig.6.6 Meal Balance Assesment for Mangoes in NutriGuide.AI	61
Fig.6.7 Food Allergen Checker Interface in NutriGuide.AI	62
Fig.6.8 Allergen Check Result and Explanation	62
Fig.6.9 Diet Recommendation for Chronic Conditions-Input Interface	63
Fig.6.10 Cholestrol Management-Diet Recommendation Output	64

LIST OF TABLES

Table 2.1 Literature Survey of Research papers 10

Table 5.1 Test Cases of NutriGuide.AI 57

1. INTRODUCTION

1.1 MOTIVATION

In an era where health awareness is growing but dietary habits are increasingly compromised by fast-paced lifestyles, there is a critical need for intelligent systems that guide individuals toward better nutrition. Many people face challenges such as poor diet choices, unbalanced meals, lack of knowledge about nutrients, and difficulty accessing reliable health guidance tailored to their needs. This often leads to nutritional deficiencies, increased risk of chronic diseases, and overall reduced quality of life.

The **NutriGuide.AI** project was initiated to fill this gap by providing a smart, AI-powered platform that helps users understand and improve their daily dietary intake. Unlike generic meal planning apps, NutriGuide.AI uses image processing, personalized profiling, and generative AI to offer specific, actionable recommendations based on what users eat, their health conditions, preferences, and goals.

With features like real-time meal analysis, chatbot-assisted dietary advice, allergen detection, and health-aware suggestions, NutriGuide.AI transforms nutritional awareness from a passive to an interactive and intelligent experience. It adapts to the user's needs dynamically, factoring in chronic conditions, fitness targets, and personal goals.

This system empowers individuals to make informed dietary choices that are both health-conscious and personalized, ultimately promoting better eating habits, improved fitness, and long-term wellness.

1.2 PROBLEM STATEMENT

In today's digital age, individuals are becoming more conscious of the importance of nutrition and healthy eating. However, most people still face significant challenges in understanding what constitutes a balanced diet, recognizing the nutritional value of their meals, and receiving personalized dietary guidance. While numerous health and fitness applications exist, they often rely on static food databases and generic advice that fail to consider the user's unique dietary preferences, medical conditions, and health goals.

The problem is further compounded by the lack of intuitive systems that allow users to assess their meals in real-time, especially through everyday interactions like taking a photo of their food. Most existing solutions require manual logging or calorie counting, which is time-consuming and prone to errors. Additionally, these platforms often do not support dynamic adaptation to a user’s evolving health status or integrate contextual factors like allergies, body measurements, or lifestyle habits into their recommendations.

This project aims to bridge this critical gap by developing NutriGuide.AI, an AI-powered dietary analysis and recommendation platform. The system leverages advanced machine learning and computer vision to analyze food images, assess nutritional content, and generate personalized meal plans and suggestions. By collecting user-specific inputs such as age, weight, height, dietary restrictions, and fitness goals, NutriGuide.AI provides tailored advice on what to eat, what to avoid, and how to improve overall nutrition. The system is designed to be multilingual, adaptive, and accessible, enabling users to make informed decisions about their diet in real-time. Ultimately, the goal is to empower individuals to achieve long-term health and wellness through smarter, personalized nutritional guidance.

1.3 EXISTING SYSTEM

Existing systems for nutrition tracking and meal planning primarily focus on calorie counting, food logging, and providing general dietary guidelines. Applications such as **MyFitnessPal**, **Lose It!**, and **Lifesum** are widely used to help users log their meals and track macronutrient intake. These platforms typically rely on large food databases, allowing users to manually enter or search for foods they consume. While useful, this process can be time-consuming and inaccurate due to user input errors and database limitations.

Some advanced apps, like **Yazio** and **Cronometer**, offer personalized plans based on user goals (e.g., weight loss or muscle gain), but the personalization is often limited to predefined templates and does not dynamically adapt based on user feedback or real-time behavior. Moreover, they generally do not integrate computer vision to analyze food from images, making them less intuitive and harder to use for individuals who prefer visual input methods.

There are also some food recognition apps like **Bite.ai** or **Calorie Mama** that use AI to identify food items from images, but they often provide limited nutritional analysis and lack the integration with personal health profiles or broader lifestyle goals.

Additionally, most existing nutrition apps do not account for comprehensive user health profiles that include allergies, medical conditions, dietary preferences (e.g., vegan, gluten-free), and cultural food habits. They also lack multilingual support and fail to offer recommendations in diverse regional or linguistic contexts.

Overall, current systems focus either on **manual food tracking** or **basic visual recognition**, but rarely combine both with **deep personalization**, **real-time meal analysis**, and **adaptive planning**, which are essential for truly effective dietary guidance. NutriGuide.AI is designed to overcome these limitations by offering an AI-powered, image-based, and personalized nutrition assistant tailored to each user's unique health and lifestyle needs.

1.3.1 LIMITATIONS

Adaptability to Diverse Food Habits and Cultures: While NutriGuide.AI aims to provide personalized nutrition recommendations, the system may face challenges when dealing with diverse regional cuisines, cultural food practices, and locally specific dishes that are not present in the training dataset or food recognition model. This could limit the accuracy of the AI in correctly identifying and analyzing meals in non-standard contexts.

Reliance on Image Quality and User Input: The accuracy of meal recognition and nutrient estimation heavily depends on the quality of the uploaded images and the context provided by the user. Poor lighting, unclear angles, or incomplete meal captures may result in incorrect identification or partial nutritional analysis.

1.4 PROPOSED SYSTEM

The proposed system, **NutriGuide.AI**, aims to deliver intelligent, real-time nutrition and wellness recommendations by leveraging advanced AI techniques, including image recognition, nutritional analysis, and generative language models. The core objective of NutriGuide.AI is to assist users—especially those with chronic conditions or specific dietary needs—in identifying meals, analyzing nutritional content, and receiving tailored guidance for maintaining a balanced diet.

A key component of NutriGuide.AI is the **ESP32-CAM-based wearable device**, which allows users to capture images of their meals on the go. These images are transmitted to a backend server or cloud platform where **pre-trained deep learning models**, such as **MobileNetV2** or **EfficientNet**, perform **food recognition and classification**. The recognized food items are then evaluated using a nutrition database to estimate **macronutrients (calories, proteins, fats, carbohydrates)** and assess whether the food is **safe or harmful**, especially for individuals with allergies or health risks.

To enhance the personalization of meal guidance, NutriGuide.AI integrates **Google's Generative AI** or **Groq LLMs** to generate **context-aware dietary suggestions**, translated into audio feedback using **text-to-speech (TTS)** systems like gTTS. This makes the system accessible even to children and people with limited literacy.

The system's recommendations improve over time by maintaining **user profiles** that store dietary habits, food preferences, allergies, and past meals. Using this historical data, NutriGuide.AI adapts its suggestions, ensuring that each user receives **dynamic, health-appropriate advice** tailored to their lifestyle and goals.

To support testing and scalability, the project initially integrates image processing on a **laptop/cloud server** but is designed for future deployment on **edge devices** for real-time, offline capabilities. The architecture supports potential extension into a **mobile/web dashboard**, offering features such as diet tracking, calorie history, and AI chatbot-based queries for additional help.

Through a combination of **edge AI**, **cloud processing**, and **generative interaction**, NutriGuide.AI aspires to deliver a **next-generation dietary assistant**—personalized, responsive, and accessible—helping users make healthier choices and lead nutritionally balanced lives.

1.4.1 ADVANTAGES

• **Real-Time Nutritional Feedback:**

NutriGuide.AI provides immediate analysis of food items using image recognition and AI, allowing users to receive dietary guidance instantly after capturing an image. This real-time feedback helps users make better food choices on the spot, rather than relying on delayed or generic advice.

• **Personalized Dietary Recommendations:**

The system dynamically adapts to each user's unique nutritional needs, preferences, and health conditions. Using AI-powered models and a growing dataset, NutriGuide.AI tailors recommendations for balanced diets, harmful food alerts, and healthy alternatives, ensuring individualized care.

• **Holistic Health Monitoring:**

Beyond just calorie tracking, NutriGuide.AI considers ingredients, allergies, potential health risks, and long-term eating patterns. This comprehensive analysis ensures users gain deeper insights into how their diet impacts overall health, not just weight.

• **Enhanced Accessibility with Voice Feedback:**

By converting nutritional insights into audio responses, NutriGuide.AI makes dietary guidance accessible to children, elderly individuals, or users with limited literacy. This voice-enabled feature improves usability across diverse demographics.

• **Constantly Improving Intelligence:**

The platform learns over time from user behavior and new food data, improving its accuracy and depth of recommendations. This continuous learning loop makes NutriGuide.AI smarter and more effective the more it's used.

1.5 OBJECTIVES

• Provide Real-Time Nutritional Feedback using image-based food recognition and AI analysis to help users make healthier dietary choices instantly.

• Deliver Personalized Meal Plans tailored to individual user profiles, including dietary preferences, allergies, health goals, and chronic medical conditions.

• Incorporate Cybernetic Health Principles to adapt dynamically based on changing food habits, lifestyle behaviors, and real-time data inputs.

• Enable Continuous Learning through user interaction and historical meal data, improving recommendation accuracy and dietary suggestions over time.
• Offer Multimodal Accessibility, including chatbot-based queries and voice-based responses, ensuring inclusive usage across all user types.

1.6 HARDWARE AND SOFTWARE REQUIREMENTS

1.6.1 SOFTWARE REQUIREMENTS

•Software:
The system is developed using Visual Studio Code (VS Code) as the integrated development environment. VS Code offers a lightweight yet powerful platform for writing, testing, and debugging the Python code that powers NutriGuide.AI’s backend services and machine learning components.

•Primary Langugae:
The core development is done in Python, chosen for its extensive ecosystem and support for machine learning, data processing, and web development. Key libraries include:

- pandas, numpy for data handling
- scikit-learn, TensorFlow for machine learning model training and inference
- gTTS for audio generation
- google.generativeai or groq for language model-based recommendations

•Frontend Framework:
The frontend is built using Streamlit, a Python-based web application framework that enables rapid development of interactive user interfaces. It allows users to upload meal images, view analysis, and receive real-time AI feedback.

•Backend Framework:
All backend services are implemented in Python. The logic for dietary analysis, object classification (safe/harmful), recommendation generation, and audio feedback is executed via Flask or Streamlit backends running on a local or cloud server.

•Database:
The project uses SQLite3 for data storage, which is a lightweight, file-based database system. It stores:

- User profiles and health data
- Historical food intake logs
- Generated recommendations and analysis results

•Frontend Technologies:
For custom interface elements, the project also uses:

- HTML for page structure
- CSS for design and styling
- JavaScript for dynamic interactions
- Bootstrap 4 to ensure responsive, mobile-friendly layouts

1.6.2 HARDWARE REQUIREMENTS

•**Operating System:**
The system is compatible with both **Windows** and **macOS** environments, ensuring smooth development and execution of the application across commonly used platforms.
• **Processor:**
A processor with a minimum specification of **Intel Core i5** (or equivalent AMD Ryzen) is

recommended. This ensures efficient handling of real-time data processing, machine learning model execution, and user interaction without performance lag.

• **RAM:**

A minimum of **8 GB RAM** is required to support multitasking, model inference, and real-time nutritional analysis. For enhanced performance, especially when handling larger datasets or running image-based recognition models, **16 GB or higher** is recommended.

2. LITERATURE SURVEY

K. Azzimani, H. Bihri, A. Dahmi, S. Azzouzi, and M. E. H. Charaf, "An AI Based Approach for Personalized Nutrition and Food Menu Planning," 2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Fez, Morocco, 2022, pp. 1–5, doi: 10.1109/ICECOCS55148.2022.9983099. [1]

F. Rad, M. Amiri, and J. Li, "Optimizing Nutritional Decisions: A Particle Swarm Optimization–Simulated Annealing-Enhanced Analytic Hierarchy Process Approach for Personalized Meal Planning," Nutrients, vol. 16, no. 18, 2024, p. 3117. [2]

R. D. Gutiérrez, M. García-Torres, P. M. Sánchez, and A. Berlanga, "ChatDiet: Empowering Personalized Nutrition-Oriented Food Recommender Systems," Smart Health, vol. 34, 2024, p. 100821, ISSN 2352-6483, doi: 10.1016/j.smhl.2024.100821. [3]

R. Arora, D. Sharma, and P. Nair, "An AI Based Approach for Personalized Nutrition and Food Menu Planning," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), IEEE, 2022, pp. 1197–1201, doi: 10.1109/ICSCDS53736.2022.9760970. [4]

E. Kaldirim, T. Ucar, and M. Ince, "A Deep Learning Based Food Image Recognition System for Calorie Estimation," 2020 28th Signal Processing and Communications Applications Conference (SIU), IEEE, 2020, pp. 1–4, doi: 10.1109/SIU49456.2020.9302277. [5]

F. Farahani, M. Firouzi, and A. Sarrafzadeh, "Towards Personalized Nutrition Recommendation Using Federated Learning: Opportunities and Challenges," IEEE Internet of Things Journal, vol. 10, no. 6, pp. 4956–4968, March 2023, doi: 10.1109/JIOT.2022.3220000. [6]

S. Chun, J. Kim, and D. Oh, "FoodAI: An Intelligent Food Recognition System Based on Deep Convolutional Neural Networks," Sensors, vol. 21, no. 10, 2021, p. 3363, doi: 10.3390/s21103363. [7]

M. Jayaraman, R. Rath, and V. S. Hegde, "DietLens: Mobile Application for Nutritional Analysis of Food Using Machine Learning," 2022 IEEE World AI IoT Congress (AIIoT), pp. 118–123, doi: 10.1109/AIIoT54504.2022.9880987. [8]

M. Elbadawi, A. Abuarqoub, and Y. Jararweh, "Nutrition-Aware IoT-Enabled Food Recommendation System," Journal of Ambient Intelligence and Humanized Computing, vol. 12, 2021, pp. 9535–9550, doi: 10.1007/s12652-020-02591-2. [9]

H. Mousavi and N. Shirmohammadi, "AI-Powered Nutrition Assistant: Combining NLP and Recommendation Systems for Personalized Dietary Advice," Procedia Computer Science, vol. 198, 2022, pp. 191–198, doi: 10.1016/j.procs.2021.12.220. [10]

Table 2.1 Literature Survey of NutriGuide.AI

Ref	**Author& year of publication**	**Journal / Conference**	**Method / Approach**	**Merits**	**Limitations**	**Research Gap**
[1]	Mavroudi et al., (2023)	*Nutrition*, Elsevier	AI-driven dietary analysis and mobile application evaluation	Personalized dietary recommendations validated by experts	Limited to a specific demographic	Lacks adaptation for chronic condition management
[2]	Han et al., (2023)	*Healthcare Analytics*, Science Direct	Scoping review of AI methods in precision nutrition	Covers broad AI techniques in nutrition research	Mostly theoretical; fewer practical models	Implementation in real-world diet systems is minimal
[3]	Suresh et al., (2022)	*ACM Computing Surveys*, ACM Digital Library	Comparative study of nutrition recommender systems sleep quality	Comprehensive survey of AI recommendation models	No standard framework	Absence of unified architecture for nutrition platforms
[4]	Kuhn et al., (2021)	*Sensors*, MDPI	Smartphone-based image recognition for food intake	Enables automated nutrient analysis from images	Accuracy affected by lighting and food presentation	Lack of robust training datasets

[6]	Azzimani et al., (2022)	*IEEE ICECOCS*	AI-enabled food menu planning	Personalized nutrition using user inputs	Simpler model; no dynamic feedback loop	Doesn't support real-time updates or adaptive diets
[7]	Rad et al., (2024)	*Nutrients*	Hybrid PSO–Simulated Annealing–AHP for meal planning	Optimized meal suggestions based on multiple criteria	Computationally expensive	High complexity for user-side mobile use
[8]	Gutiérrez et al., (2024)	*Smart Health*	ChatDiet – food recommender with personalized nutrition focus	Integrates NLP with food databases	Limited food diversity handling	Needs broader food databases and allergy awareness
[9]	Elbadawi et al., (2021)	*Journal of Ambient Intelligence and Humanized Computing*	IoT-enabled food recommendation system	Real-time environment-based suggestions	Requires full IoT infrastructure	Poor generalization without sensor data
[10]	Mousavi & Shirmohammadi (2022)	*Procedia Computer Science*	AI-powered nutrition assistant using NLP + recommender systems	Personalized dietary advice through AI	Incomplete health context integration	Lack of chronic condition data fusion

3. ANALYSIS AND DESIGN

In the modern health-tech landscape, users often lack accurate, real-time, and personalized nutrition advice that adapts to their preferences, health goals, and dietary restrictions. Existing platforms generally rely on manual food logging and generic diet plans, which can be tedious, inaccurate, and impersonal. **NutriGuide.AI** addresses these limitations through a novel approach that combines **AI-powered image recognition**, **real-time nutrient analysis**, and **personal health profiling** to provide instant, meaningful dietary recommendations.

At the core of the NutriGuide.AI system is its ability to analyze meals from user-captured images using **deep learning models**. The frontend, built using **Streamlit**, provides an intuitive interface where users can upload or take photos of their meals. These images are processed by a **CNN-based food classification model** (e.g., MobileNet or ResNet) that identifies the type of food, which is then mapped to a nutritional database for estimating calorie and macro/micronutrient values.

The system also integrates user data such as:

- Age, gender, weight, height
- Health conditions (e.g., diabetes, hypertension)
- Dietary preferences (e.g., vegetarian, low-carb, gluten-free)
- Allergen sensitivities

This profile data allows the backend logic to **personalize meal recommendations** by applying rule-based logic and machine learning inference. For example, if a food contains peanuts and the user has a peanut allergy, the system will flag it as harmful and suggest an alternative.

Additionally, NutriGuide.AI supports:

- **Real-time dietary feedback**: Offers insights like “high in sodium,” “good protein source,” or “suitable for diabetics”
- **Audio feedback**: Using tools like gTTS, the system provides **spoken responses** to guide users, especially useful for visually impaired users or children
- **Progress tracking**: Users can review their daily/weekly food log and evaluate if they are meeting nutritional goals

3.1 MODULES

User Interaction Module

This module allows users to interact with the system by uploading meal images and entering personal health data. It also enables users to receive nutritional analysis, personalized food recommendations, and audio feedback for ease of understanding.

- Input:
  - Food image (captured via camera or uploaded)
  - User profile data (age, weight, height, allergies, dietary preferences, chronic conditions)
- Output:
  - Real-time meal analysis
  - Personalized dietary recommendations
  - Audio/text-based feedback
  - Meal history and nutrition logs

2. **Image Processing & Food Detection Module**

Handles the preprocessing and classification of the uploaded meal image. It uses a lightweight convolutional neural network (e.g., MobileNetV2) to detect and classify food items from the image.

- Input:
  - Raw image of food from ESP32-CAM or user upload
- Output:
  - Detected food item label(s)
  - Bounding boxes (if object detection is applied)
  - Food item class for further nutrient lookup

3. **Nutritional Analysis & Allergen Detection Module**

Fetches nutritional details based on the classified food item from a database and checks against the user’s allergy list or chronic health risks.

- Input:
  - Detected food items
  - User dietary preferences and allergy info

- Output:
    - Caloric value, macronutrient breakdown (protein, fats, carbs)
    - Flags for allergens, high-sugar, or high-sodium foods
    - Warnings for medically unsuitable items

4. **Recommendation Engine Module**

Applies rule-based and ML-driven logic (e.g., logistic regression, decision trees) to suggest healthier alternatives or portion modifications based on the user's goal (e.g., weight loss, muscle gain, diabetic-safe diet).

- Input:
    - Nutritional values and user profile
    - Health goal and risk factors
- Output:
    - Personalized meal suggestions
    - Tips like “reduce sodium,” “add protein,” or “avoid fried foods”

5. **Audio Feedback Module**

Converts the AI-generated recommendation into a speech response using TTS (e.g., gTTS), making the platform more accessible for children, visually impaired users, and people with low literacy.

- Input:
    - AI recommendation text
- Output:
    - Audio output played through speaker (ESP32 or PC/mobile)

6. **Progress Tracking & Visualization Module**

Maintains logs of user meals and recommendations. Displays nutritional progress using simple graphs, such as calorie intake trends, nutrient balance, and frequency of harmful foods.

- Input:
    - Daily/weekly meal data and recommendations
- Output:
    - Charts showing calorie history, safe/harmful food ratio, and goal achievement progress

3.2 ARCHITECTURE

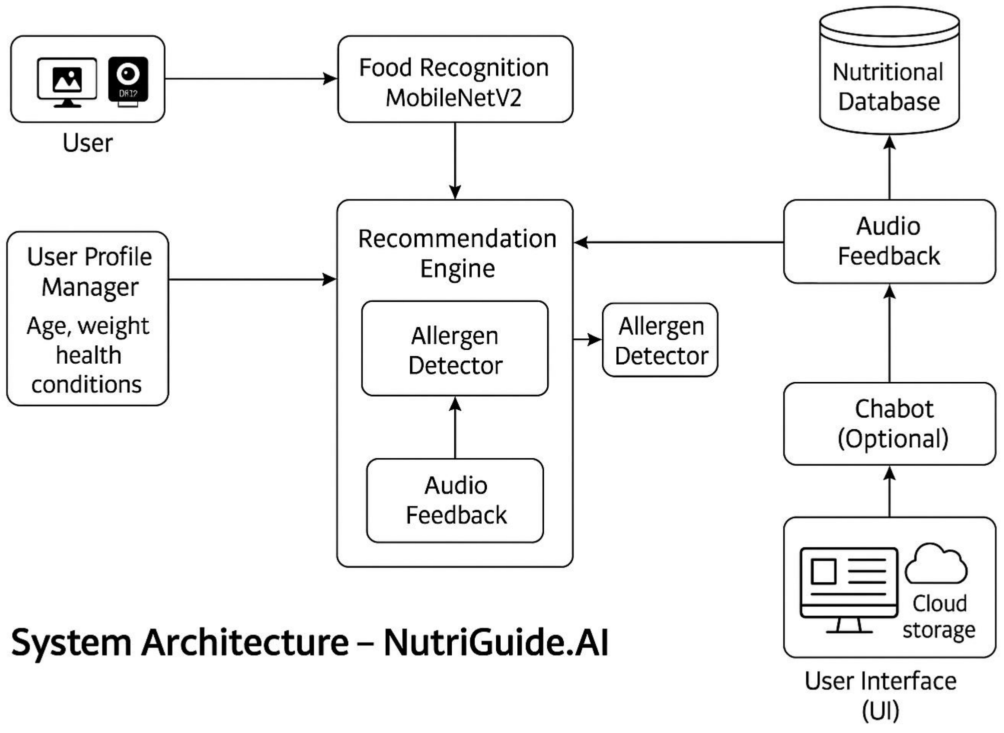


Fig. 3.2.1 Architecture of Nutriguide.AI

The architecture of NutriGuide.AI, as illustrated in Fig. 3.2.1, is designed to deliver personalized, AI-driven nutritional guidance that adapts to users' individual health goals and dietary needs. The system workflow begins with the Data Collection component, where users input personal health profiles, dietary preferences, restrictions, and upload meal images via an intuitive interface. This data is securely stored in a structured format to support real-time analysis.

Next, the system performs Image-Based Nutrient Analysis by applying AI-powered image recognition models to scan uploaded meal photos. This analysis extracts detailed nutritional information, such as calorie content, macronutrients, and micronutrients, providing an accurate breakdown of the user's food intake.

The extracted data, combined with user profile information and real-time nutritional feedback, is fed into the Personalized Recommendation Engine. This engine employs advanced generative AI models along with domain-specific nutritional guidelines to create customized meal plans aligned with the user's health objectives—whether for weight loss, muscle gain, chronic condition management, or balanced living.

In parallel, the system integrates a Food Allergen Detection module that scans meal ingredients and alerts users to potential allergens, ensuring safe dietary practices. For users with chronic health conditions like diabetes, hypertension, or cardiovascular diseases, the platform offers tailored advice grounded in medical best practices.

The Progress Tracking and Feedback component visualizes nutritional intake trends and goal progress through interactive dashboards, enabling users to monitor their adherence and adjust targets dynamically.

Users interact continuously through a friendly interface that supports input updates, image uploads, and receipt of actionable insights such as meal recommendations, portion adjustments, and lifestyle tips. The system's real-time adaptability is sustained by continuous learning from new data, ensuring recommendations remain relevant and effective over time.

3.3 UML DIAGRAMS

3.3.1 USE CASE DIAGRAMS

A use case diagram is a visual representation that depicts the interactions between various actors and a system, capturing the ways in which users or external entities interact with the system to achieve specific goals. It is an essential tool in system analysis and design, often used in software engineering and business analysis. In a use case diagram, actors are entities external to the system that interact with it, and use cases are specific functionalities or features provided by the system as seen in Fig. 3.3.1.1. These interactions are represented by lines connecting actors to use cases. The diagram helps to illustrate the scope and functionality of a system, providing a high-level view of how users or external entities will interact with it.

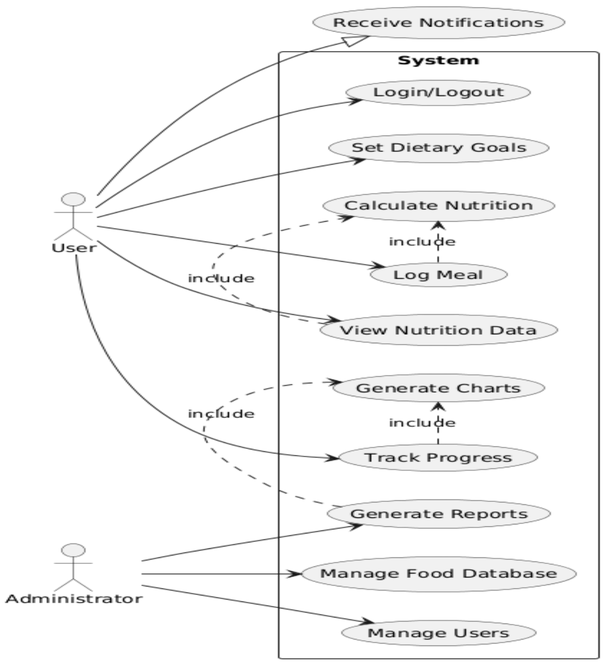


Fig. 3.3.1.1 Use Case Diagram

Actors

1. **User:** The individual interacting with the system to set dietary goals, log meals, view nutrition data, and track progress.
2. **Administrator**: The privileged user managing food databases, user accounts, and system reports.

Use Cases

1. **Login/Logout:** The user authenticates into or exits the system to access personalized dietary features.
2. **Set Dietary Goals**: The user defines nutritional targets (e.g., calorie limits, macronutrient ratios).
3. **Calculate Nutrition:** The system computes nutritional values (calories, proteins, carbs) for logged meals or user goals.
4. **Log Meal**: The user records consumed food items, triggering automatic nutrition calculation.

5. **View Nutrition Data:** The user examines summarized nutritional information (e.g., daily intake, goal progress).
6. **Generate Charts:** The system visualizes nutrition data (e.g., pie charts for macros, trend graphs over time).
7. **Track Progress:** The system compares user intake against dietary goals and highlights achievements/deficits.
8. **Generate Reports:** The administrator exports structured summaries (e.g., user progress, nutrition analytics).
9. **Manage Food Database:** The administrator adds, edits, or removes food items and their nutritional profiles.
10. **Manage Users:** The administrator handles user accounts (e.g., registration, permissions, data access).

3.3.2 CLASS DIAGRAM

A class diagram is a visual representation that models the static structure of a system, showcasing the system's classes, their attributes, methods (operations), and the relationships between them as seen in Fig. 3.3.2.1. It is a key tool in object-oriented design and is commonly used in software engineering to define the blueprint of a system.

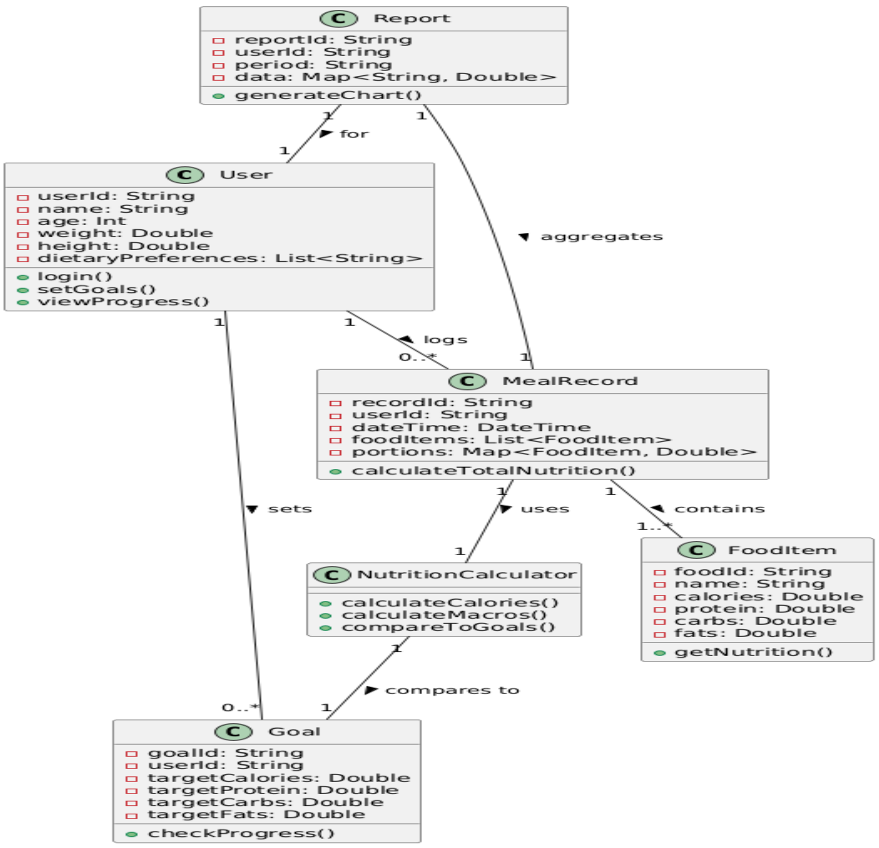


Fig. 3.3.2.1 Class Diagram

Relationships:

1. **User → MealRecord**

- The User class logs multiple MealRecords.
- Each MealRecord stores details about consumed food items and portion sizes.
- Relationship:

  User (1) ↔ (0..*) MealRecord

2. **MealRecord → FoodItem**

- Each MealRecord contains one or more FoodItem entries along with portion quantities.
- These food items are used to calculate the nutritional value of a meal.
- Relationship:

  MealRecord (1) ↔ (1..*) FoodItem

3. **MealRecord → NutritionCalculator**

- The NutritionCalculator is used to process the food items in a meal and calculate:
    - Calories
    - Macronutrients (Protein, Carbs, Fats)
- Relationship:
  MealRecord (1) uses (1) NutritionCalculator

4. **User → Goal**

- A User sets one or more Goals related to daily nutritional targets (e.g., calories, carbs).
- Goals are used to guide food intake and track performance.
- Relationship:
  User (1) ↔ (0..*) Goal

5**. Goal → NutritionCalculator**

- The NutritionCalculator compares actual nutrition intake from MealRecords with the user's Goal.
- This helps users track how closely their eating habits align with their objectives.
- Relationship:
  Goal (0..*) compared to (1) NutritionCalculator

6. **User → Report**

- Users receive a Report summarizing their nutrition data over a defined period.
- The report includes graphs, trends, and comparisons with set goals.
- Relationship:
  User (1) ↔ (1) Report

7. **Report → MealRecord**

- The Report aggregates data from the user's MealRecords for analytics and visualization.
- Relationship:
  Report (1) aggregates (1..*) MealRecord

System Flow:

1. User Interaction

- The user logs in, sets dietary goals, and begins logging food through MealRecords.
- Users also set dietary preferences and restrictions during profile setup.

2. Meal Logging

- Users record meals, specifying food items and portion sizes.
- These are stored as MealRecords that are linked to the user.

3. Nutritional Calculation

- The NutritionCalculator processes food data in the MealRecord, calculates calories, macros, and total intake.
- It compares the intake with the user's Goals to determine deviations.

4. Goal Tracking

- Users set targets (e.g., daily calories = 1800 kcal).
- The system continuously compares actual consumption to targets using the compareToGoals() method in the NutritionCalculator.

5. Reporting

- The Report class collects user data over a specified time range and generates charts or summaries using generateChart().
- This gives the user visual feedback on their nutrition journey.

3.3.3 ACTIVITY DIAGRAM FOR NUTRIGUIDE.AI

An Activity Diagram is a type of behavioral diagram used in Unified Modeling Language (UML) to represent the flow of control or data through the system as seen in Fig. 3.3.3.1. It focuses on the flow of activities and actions, capturing the sequence of steps in a particular process or workflow. Activity diagrams are commonly used to model business processes, workflows, or any sequential activities in a system.

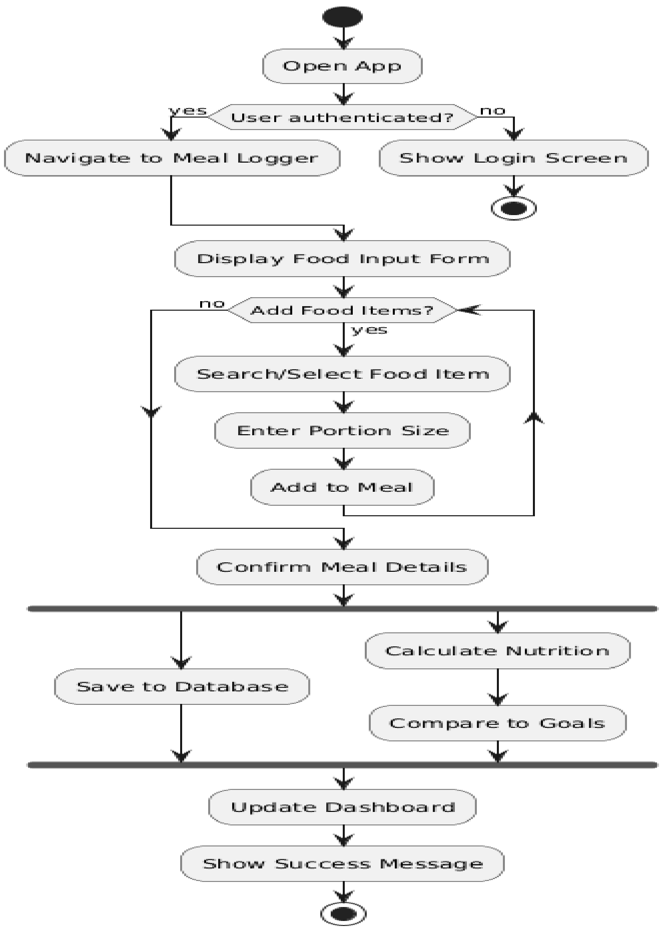


Fig. 3.3.3.1 Activity Diagram

Flow Explanation:

1. Open Application

The user launches the NutriGuide.AI application from a desktop or mobile interface.

2. User Authentication

- If the user is already registered, they are authenticated and navigated to the Meal Logger interface.
- If not authenticated, the system redirects the user to the Login Screen to enter credentials or register.

3. Display Food Input Form

Once authenticated, the user is shown the Meal Logging Form which allows them to begin logging their meal.

4. Add Food Items?

The system prompts the user to add at least one food item to their meal:

- If yes, the user proceeds to select a food item.
- If no, the system proceeds with an empty meal or prompts again for input.

5. Search/Select Food Item

The user searches or selects the desired food item from a list or via image-based input (in future versions, AI-based food recognition may auto-fill this).

6. Enter Portion Size

The user enters the portion size of the selected food item (e.g., grams, slices, cups).

7. Add to Meal

The selected item along with its portion is added to the current meal list.

8. Repeat or Confirm Meal

- The user may continue adding more items by looping back.
- Once done, they Confirm Meal Details to move forward.

9. Nutrition Calculation

The system uses the NutritionCalculator module to:

- Calculate total calories, protein, fats, carbs
- Analyze macros and micronutrients

10. Compare with Goals

The calculated meal values are compared with the user's personalized nutrition goals to identify gaps or overages.

11. Save to Database

All meal details, nutritional values, and comparison results are stored in the user’s record for future tracking and reporting.

12. Update Dashboard

The user dashboard is updated in real-time, showing:

- Daily calorie intake
- Macronutrient breakdown
- Goal alignment

13. Show Success Message

A confirmation message is shown, notifying the user that the meal has been successfully logged and analyzed.

14. End of Workflow

At this stage, the user can:

- Log additional meals
- Review historical data
- Or exit the app

3.3.4 SEQUENCE DIAGRAM

A sequence diagram illustrates the flow of interactions between actors and system components over time as seen in Fig. 3.3.4.1, emphasizing the order in which messages are exchanged to achieve specific functionalities. Actors represent external entities that interact with the system, while lifelines depict the system components involved in the process. Messages are shown as arrows, indicating the flow of information or actions between these elements. By providing a step-by-step view of workflows, sequence diagrams help in understanding and designing the dynamic behaviour of a system.

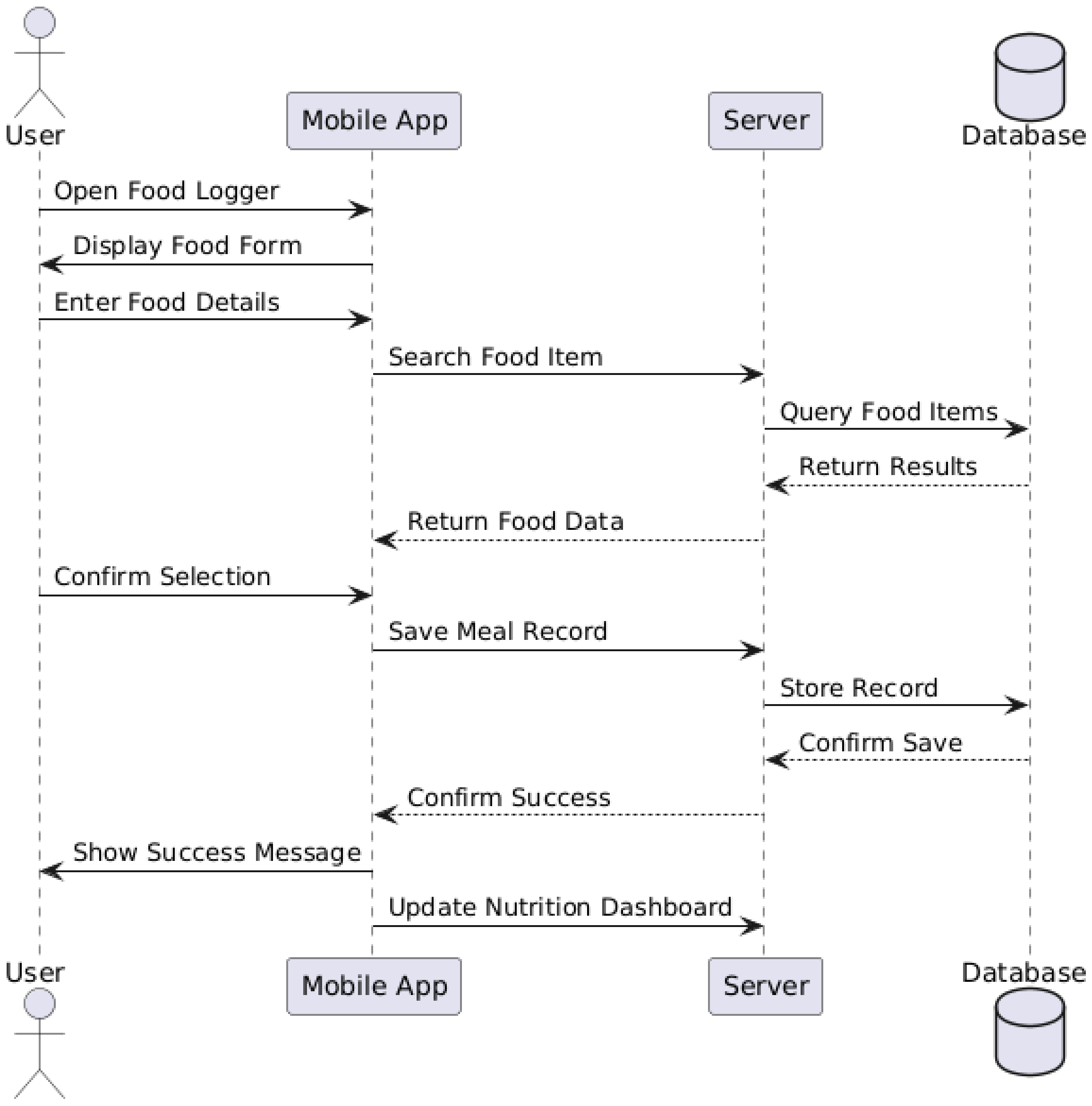


Fig. 3.3.4.1 Sequence Diagram

Key Interactions and Relationships

User and Web Interface:

- Open Food Logger:

  The user opens the NutriGuide.AI web app (built with Streamlit or similar) and navigates to the food logging section.
- Display Food Input Form:

  The system presents a web-based form where the user can either upload an image of their meal or manually enter food details.

- Enter Food Details:
  The user inputs the food name, selects from dropdowns or image recognition suggestions, and enters portion size.
- Confirm Selection:
  Once the meal details are entered, the user submits the form to generate nutritional analysis and recommendations.

Web App and Server

- Search Food Item:
  The web frontend sends a request to the backend (Python/Flask/Streamlit) to search for the food item either in a local nutrient database or via API.
- Save Meal Record:
  The confirmed meal information, along with its nutritional value, is sent to the backend to be stored in the database (SQLite/PostgreSQL).

Server and Database

- Query Food Items:
  The backend server queries the food database for nutritional information related to the selected food items (e.g., calories, macros, allergens).
- Store Record:
  After successful analysis, the backend saves the meal log along with the user ID and timestamp in the database.
- Confirm Save:
  The database confirms successful storage, and the backend returns a confirmation to the web interface.

Groq API Integration

- Personalized Feedback Request:
  The backend sends user input and food analysis context to the Groq API (via LLM like LLaMA-3), asking for intelligent recommendations or insights (e.g., "Is this meal healthy for a diabetic?").
- Groq Response Returned:
  The Groq API returns a smart, natural language response tailored to the user's

health profile (e.g., “Consider reducing the rice portion if your goal is weight loss”).

Web Interface and User

- Confirm Success:
  The web app shows a confirmation that the meal was logged successfully.
- Update Nutrition Dashboard:
  User’s dashboard is updated to reflect changes in calorie intake, macro targets, and adherence to dietary goals.
- Show AI Recommendations (from Groq):
  The response from Groq is displayed as a friendly, informative suggestion (text or text-to-speech output).

3.3.5 COMPONENT DIAGRAM

A Component Diagram is a type of structural diagram used in software engineering to represent the components of a system and how they interact or depend on each other. It shows how the components (which could be software modules, subsystems, or other significant parts) are organized and connected within a system. In this diagram, each component encapsulates a set of related functionalities and interfaces as shown in Fig. 3.3.5.1.

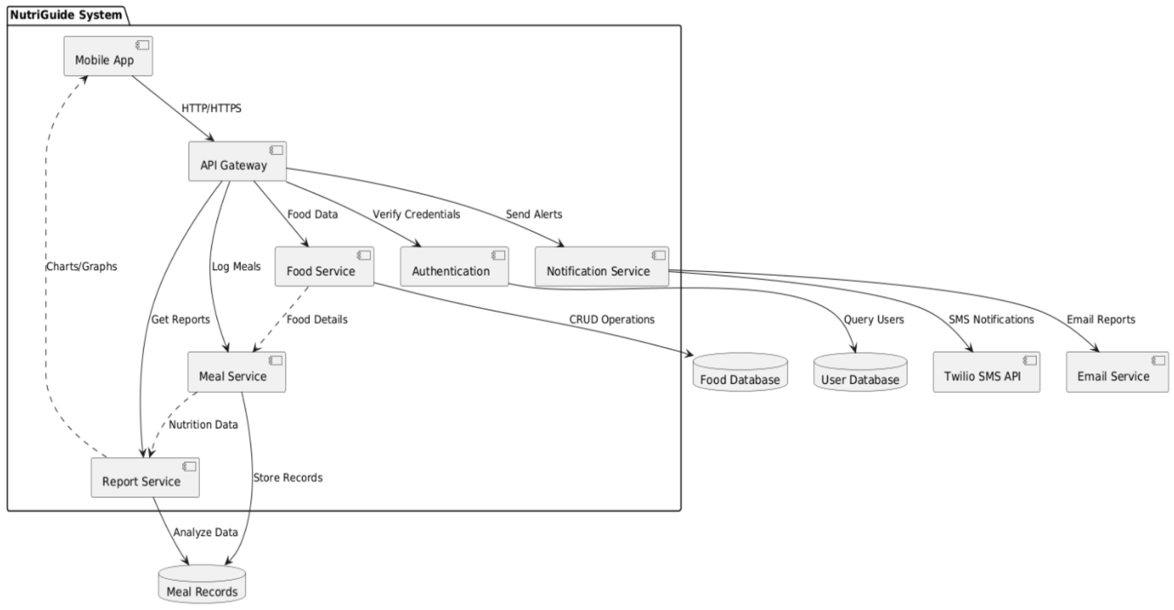


Fig. 3.3.5.1 Component Diagram

Web Interface (User Interaction Layer)

Description:
This is the primary front-end interface developed using Streamlit, where users interact with the platform. It allows users to log meals (either manually or via image upload), view nutritional feedback, and receive AI-generated recommendations.

- Features: Upload meal photo, enter portion size, receive text/audio feedback, view progress charts.

2. API Gateway

Description:
Handles all HTTP/HTTPS communication between the front-end and backend services. It routes requests to the correct internal service (e.g., authentication, food data retrieval, report generation).

- Ensures secure, centralized control for all external API calls (Groq, food database).
- Decouples frontend from backend logic.

3. Authentication Service

Description:
Verifies user credentials during login or registration. Works with the User Database to validate access, manage sessions, and enforce secure access control across the system.

4. Food Service

Description:
Manages the retrieval of nutritional information based on food names or image recognition results. Interacts with the Food Database to fetch macros (calories, proteins, fats, carbs) and allergen flags.

- Optionally integrates with Groq to support AI explanation of food health impact.

5. Meal Service

Description:
Handles user meal logging — storing each meal entry with timestamp, item, portion, and calculated nutrition data. Sends processed information to the Report Service and Database.

- Responsible for saving meal records and handling data preprocessing for the recommendation engine.

6. Report Service

Description:
Generates charts, insights, and weekly/monthly nutrition summaries for users. It collects analyzed nutrition data and feeds it back to the user interface as graphs or status badges.

- Uses inputs from Meal Records and Nutrition Goals to assess progress.

3.3.6 DEPLOYMENT DIAGRAM

The Deployment Diagram illustrates the physical deployment of software components across hardware nodes in the **Rhythm Restore** system. It captures the interactions between the **user device**, **server**, and **database**, highlighting how system components communicate and are distributed in a real-world deployment scenario.

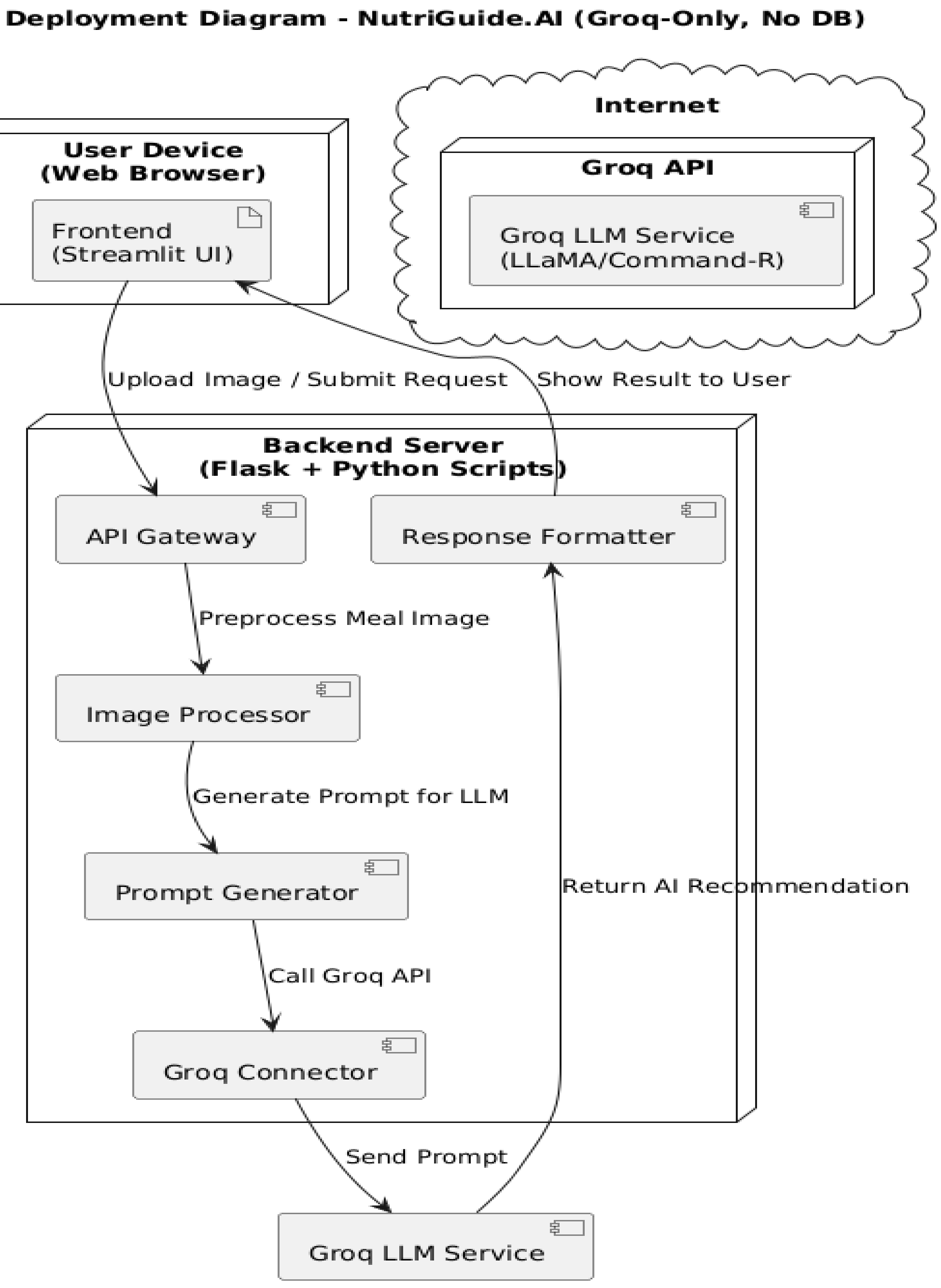


Fig. 3.3.6.1 Deployment Diagram

User Device:

- Hosts the **web interface**, built using **Streamlit**, which runs in the user's browser.
- Users can upload food images, enter dietary details, and view personalized recommendations.

- All interactions (form submission, meal analysis) are sent to the server over secure HTTP/HTTPS.
- The frontend displays AI-generated nutrition feedback and suggestion messages, optionally as audio output.

• Server (Backend Flask API):

- Acts as the core processing hub, hosting:
    - **Image Processing Module:** Prepares the uploaded food images (resize, format).
    - **Prompt Generation Engine:** Converts image/text inputs into structured prompts.
    - **Groq Connector:** Handles secure communication with **Groq LLM API** (e.g., LLaMA-3 via GroqCloud).
    - **Response Formatter:** Receives the AI-generated reply and converts it into user-friendly output.
- The server is stateless and does not store any data locally. All logic and recommendation generation are handled in real-time using Groq's inference service.

• Groq API (Cloud AI Engine):

- A cloud-based large language model (LLM) accessed via API.
- Processes the prompt from the backend and returns intelligent, context-aware dietary suggestions.
- Powered by models like **LLaMA-3 or Mixtral**, optimized for fast, cost-effective inference via **GroqCloud**.
- No persistent state is maintained — all results are generated dynamically per request.

This lightweight, serverless-ready architecture removes the need for internal databases or storage, enabling NutriGuide.AI to offer instant, AI-generated dietary insights with minimal infrastructure. It ensures high scalability, low latency, and ease of deployment across platforms

3.4 METHODOLOGY

Data Acquisition

User Input (Live Data):
NutriGuide.AI captures real-time input from users through a web interface. This includes:

- Meal images (uploaded or captured via camera)
- Manual food entries
- User-specific metadata (age, weight, allergies, health goals)

• Purpose:
These inputs form the core data used to:

- Generate immediate AI-powered dietary feedback
- Trigger prompt generation for Groq API inference
- Tailor nutritional recommendations based on user profiles and food content

Model Steps

• Prompt Generation:
Submitted meal inputs are converted into structured prompts. These prompts may include:

- Food item description or classification result (e.g., "1 bowl of white rice with curry")
- User profile data (e.g., diabetic, weight loss goal, peanut allergy)

• LLM Inference with Groq:

Prompts are sent to Groq's LLM (e.g., LLaMA-3) via API. The model processes the prompt to:

- Evaluate the nutritional safety of the meal
- Suggest healthier alternatives if necessary
- Align advice with the user's dietary restrictions and goals

• Response Parsing:

The Groq-generated response is formatted by the backend to deliver:

- Readable and relevant dietary recommendations
- Optional text-to-speech audio output using TTS engines (e.g., gTTS)

• Feedback Loop (Optional):

While NutriGuide.AI is stateless in its current form, future enhancements may include:

- Feedback collection to improve AI prompts
- User progress tracking based on recommendation follow-through

Models Used

1. **Personalized Nutritional Recommendation Model (Transformer-based):**

- Description: This model leverages transformer architecture to analyze user dietary preferences, health goals, and nutritional data to generate personalized meal plans and nutrition advice.
- How it Works: The model is trained on a dataset containing diverse dietary patterns, nutritional values, and user feedback. It processes input features like age, weight, activity level, and dietary restrictions, then generates tailored food recommendations and nutrient intake targets.
- Why it's Used: Transformers excel at understanding complex relationships in sequential and structured data, making them well-suited for capturing personalized nutritional needs and evolving user behavior.

2. **Food Image Recognition Model (Convolutional Neural Network - CNN):**

- Description: A CNN model designed to identify food items from images uploaded by users, enabling automatic tracking of meal components and calories.
- How it Works: The CNN is trained on thousands of labeled food images to recognize different food categories and portion sizes. When a user uploads a meal photo, the model detects the food types and estimates nutritional content.
- Why it’s Used: CNNs are highly effective at image classification tasks, allowing NutriGuide to simplify food logging and improve the accuracy of dietary tracking.

3. **Nutrient Impact Predictor (Gradient Boosting Regressor):**

- Description: This model predicts the likely impact of specific nutrient intakes on user health metrics like energy levels, weight changes, or blood sugar.
- How it Works: Using historical user data and health outcomes, the Gradient Boosting Regressor learns to map nutrient intake patterns to measurable health effects. It helps provide actionable feedback on dietary adjustments.
- Why it’s Used: Gradient boosting enhances prediction accuracy by combining weak learners and focusing on errors, which improves personalized health insights and recommendations.

4. CODE AND IMPLEMENTATION

4.1 CODE

main.py

```
import streamlit as st
import datetime
import os
import matplotlib.pyplot as plt
from groq import Groq
import google.generativeai as genai
from PIL import Image

API_KEY_GROQ =
"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"
API_KEY_GENAI = "AIzaSyCOkrbREKz9zvKo2mD1UrDS_s-u3WoFzKY"

client = Groq(api_key=API_KEY_GROQ)
genai.configure(api_key=API_KEY_GENAI)

st.set_page_config(page_title="NutriGuide.AI", page_icon="🍎", layout="wide")

app_mode = st.sidebar.selectbox(
    "Choose a feature",
    [
        "Home",
        "Personalized Meal Plan",
        "Food Nutrient and Calorie Estimator",
        "Food Allergen Checker",
        "Diet Recommendation for Chronic Conditions",
        "Personalized Diet and Fitness Syncing"
    ]
)
```

```
def load_dynamic_css():
    st.markdown(
        """
        <style>
        /* General Dark Mode Setup */
        .stApp {
            background-color: #2c3e50;  /* Dark background */
        }

        /* Sidebar Styling */
        .css-1d391kg {
            background-color: #34495e;  /* Darker sidebar */
        }

        /* Main Content Styling */
        .main-bg {
            background: linear-gradient(45deg, #16a085, #1abc9c, #f39c12);
            background-size: 400% 400%;
            color: #ecf0f1;
            padding: 40px;
            border-radius: 12px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

        /* Centering Title */
        .main-bg h1 {
            color: #ffffff;  /* White text color for title */
            text-align: center;
            font-size: 3rem;
            font-weight: 600;
```

```
}

/* Button Animation */
.stButton>button {
   background-color: #16a085;
   color: #2c3e50;
   border-radius: 5px;
   transition: transform 0.3s ease-in-out;
}

.stButton>button:hover {
   transform: scale(1.05);
}

/* List Styling and Hover Effects */
ul {
   margin: 20px 0;
   padding-left: 30px;
}

ul li {
   color: #bdc3c7;
   font-size: 1.1rem;
   padding: 5px 0;
   transition: color 0.3s ease;
}

ul li:hover {
   color: #1abc9c;  /* Green color on hover */
}

/* About Us Section Styling */
.feature-bg {
   background-color: #34495e;
```

```
    padding: 25px;
    border-radius: 10px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    margin-bottom: 30px;
}

.feature-bg h2 {
    color: #1abc9c;
    font-size: 1.8rem;
    margin-bottom: 10px;
}

.feature-bg p {
    color: #bdc3c7;
    font-size: 1.1rem;
}

/* Section Styling */
.feature-section {
    padding: 30px;
    margin-top: 40px;
    border-radius: 12px;
    background-color: #34495e;
    box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
}
.feature-section ul {
    padding-left: 20px;
}

/* Footer Styling */
.footer {
    text-align: center;
    padding: 20px;
    margin-top: 50px;
```

```
            background-color: #34495e;
            border-radius: 10px;
            color: #ecf0f1;
        }
        </style>
        """,
        unsafe_allow_html=True
    )
load_dynamic_css()

if app_mode == "Home":
    st.markdown('<div class="main-bg">', unsafe_allow_html=True)

    st.markdown('<h1>Welcome to NutriGuide.AI</h1>', unsafe_allow_html=True)

    st.markdown(
        """
        <div class="feature-bg">
            <h2>Your Personalized Path to Better Nutrition</h2>
            <p>
                NutriGuide.AI is your ultimate personal health and diet assistant powered by AI.
Our advanced algorithms provide personalized solutions for your health and nutrition
needs. With NutriGuide.AI, you get accurate recommendations to help you achieve a
healthier, happier lifestyle!
            </p>
            <p>
                Here’s what we offer:
            </p>
            <ul>
                <li><b>Personalized Meal Plans:</b> Tailored to your dietary preferences,
goals, and health requirements.</li>
                <li><b>Food Calorie Estimation:</b> Instantly estimate the calories in your
meals, helping you track your daily intake and stay on top of your nutrition.</li>
```

```
                <li><b>Allergen Checker:</b> Our AI-powered allergen checker helps you avoid harmful ingredients and allergens in your food.</li>
                <li><b>Chronic Conditions Support:</b> Tailored diet recommendations based on your specific health conditions, such as diabetes, heart disease, and more.</li>
                <li><b>Fitness Syncing:</b> Sync your fitness routine with your meal plan, ensuring that your nutrition supports your fitness goals.</li>
            </ul>
        </div>

        <div class="feature-bg">
            <h2>About Us</h2>
            <p>
                NutriGuide.AI was founded with the mission to make healthy eating easier, smarter, and personalized for everyone. Our team of experts in nutrition, artificial intelligence, and health technology have come together to create an app that helps you make informed food choices, track your meals, and stay on top of your health goals.
            </p>
        </div>

        <div class="feature-section">
            <h2>Why Choose NutriGuide.AI?</h2>
            <ul>
                <li><b>Personalization at its Core:</b> Every recommendation is tailored to your preferences and health needs, ensuring you get the best advice.</li>
                <li><b>AI-powered Insights:</b> Our advanced AI-driven algorithms analyze your dietary habits, providing data-driven solutions to optimize your health.</li>
                <li><b>Comprehensive Approach:</b> We offer features that go beyond simple meal tracking, including fitness syncing, allergen checking, and chronic conditions support.</li>
                <li><b>User-friendly Interface:</b> NutriGuide.AI has a sleek, intuitive design that ensures easy navigation and effortless use for everyone, from beginners to experts.</li>
            </ul>
        </div>
        <div class="feature-section">
            <h2>Features</h2>
            <ul>
                <li><b>Personalized Meal Plans</b> – Tailored to your dietary preferences,
lifestyle, and health goals.</li>
                <li><b>Food Calorie Estimator</b> – Quickly calculate the nutritional value of
your meals.</li>
                <li><b>Food Allergen Checker</b> – Scan your food and check for potential
allergens.</li>
                <li><b>Chronic Conditions Support</b> – Get diet recommendations for
specific conditions like diabetes, hypertension, and more.</li>
                <li><b>Fitness Syncing</b> – Sync with your fitness routine to optimize diet
and exercise together.</li>
            </ul>
        </div>

        <div class="feature-bg">
            <h2>Ready to Start Your Nutrition Journey?</h2>
            <p>Join the thousands of users already using NutriGuide.AI to take control of their
health. Try it today!</p>
            <div class="stButton"><button>Get Started</button></div>
        </div>
    """,
    unsafe_allow_html=True
)

    # Footer Section
    st.markdown(
        """
        <div class="footer">
            <p>🍎 NutriGuide.AI © 2024 | All Rights Reserved</p>
        </div>
        """,
        unsafe_allow_html=True
    )

# Feature 1: Personalized Meal Plan
if app_mode == "Personalized Meal Plan":
    st.markdown('<div class="feature-bg"><h2>Personalized Meal Plan</h2></div>',
unsafe_allow_html=True)



    age = st.number_input("Enter your age:", min_value=0, max_value=120)
    sex = st.selectbox("Select your sex:", ["Male", "Female", "Other"])
    weight_goal = st.selectbox("Weight goal:", ["Gain", "Loss"])
    eating_habits = st.selectbox("Eating habits:", ["Omnivore", "Flexitarian", "Pescatarian",
"Vegetarian", "Vegan"])
    dietary_restrictions = st.selectbox("Dietary restrictions:", ["None", "Mediterranean",
"Paleo", "Whole30", "Low Carb", "High Carb", "Gluten-Free", "Lactose-Free", "Raw
Food", "Alkaline Food"])
    intermittent_fast_timing = st.selectbox("Select your intermittent fasting schedule:",
["None", "16/8", "18/6", "20/4"])
    traditional_diet = st.selectbox("Choose traditional diet:", ["None", "Ayurvedic",
"Macrobiotic", "Halal", "Kosher"])
    duration = st.selectbox("Meal plan duration:", ["One day", "One week", "One month"])

    if st.button("Get Meal Plan", key="meal_plan_btn"):
        try:
            messages = [
                {
                    "role": "user",
                    "content": f"Write a personalized meal generation for the following
parameters\n"
                               f"1. Age: {age}\n"
                               f"2. Sex: {sex}\n"
                               f"3. Weight goal: {weight_goal}\n"
                f"4. Food eating habits: {eating_habits}\n"
                f"5. Dietary restrictions: {dietary_restrictions}\n"
                f"6. Intermittent fasting :{intermittent_fast_timing}\n"
                f"7. Traditional diet: {traditional_diet}\n"
                f"8. duration :{duration}\n"
            },
            {
                "role": "assistant",
                "content":"dont inlucde to consult wiht dietictian message ,start with your
personlized meal plan is\n"
                "Provide a meal plan based on the above parametersand also include the
cost of the meals in INR\n"
                "make sure first the dish must be mentioned followed by ingeredints
and its cost then approximation of entire dish\n"
                "when its intermittent fasting make sure to take time which user
provides\n"
                "provide deatils of the paramter choosen by the user before giving the
diet \n"
                "If i ask in days give me in the complete day. i want youto give me all
the data which asked "

            }
        ]
        completion = client.chat.completions.create(
            model="llama3-70b-8192"
,
            messages=messages,
            temperature=1,
            max_tokens=1024
        )
        meal_plan = completion.choices[0].message.content
        st.write(meal_plan)
    except Exception as e:
        st.error(f"Error: {e}")
```

```
# Feature 2: Food Nutrient and Calorie Estimator
if app_mode == "Food Nutrient and Calorie Estimator":
    st.markdown('<div class="feature-bg"><h2>Food Nutrient and Calorie
Estimator</h2></div>', unsafe_allow_html=True)

    st.write("""
        Upload an image of your food, and our AI will analyze the nutrient composition,
including calories, protein, fats, carbohydrates, vitamins, and minerals.
        Additionally, you'll receive a meal balance assessment.
    """)

    image_file = st.file_uploader("Upload an image of food:", type=["jpg", "jpeg", "png"])

    if image_file:
        # Ensure the 'temp' directory exists
        temp_dir = "temp"
        if not os.path.exists(temp_dir):
            os.makedirs(temp_dir)

        temp_image_path = os.path.join(temp_dir, image_file.name)
        with open(temp_image_path, "wb") as f:
            f.write(image_file.getbuffer())

        st.image(image_file, caption="Uploaded Image", use_column_width=True)

        if st.button("Analyze Nutrients"):
            try:

                myfile = genai.upload_file(temp_image_path)
                model = genai.GenerativeModel("gemini-1.5-flash")

                result = model.generate_content(
                    [
                myfile,
                "\n\n",
                """You are an expert nutritionist. Analyze the food dish or dishes in this
image and provide a detailed breakdown of the nutrient composition, and you do not need
to provide exact composition; you can give expected ranges including expected values for
calories, protein, fats, carbohydrates, vitamins, and minerals.
                Provide the analysis in the following format in bullet points:
                - Food Item:
                  - Calories: X
                  - Protein: Yg
                  - Fats: Zg
                  - Carbohydrates: Ag
                  - Vitamins: [List]
                  - Minerals: [List]
                """
            ]
        )

        st.success("Nutrient analysis completed!")
        st.write(result.text)

        balance_result = model.generate_content(
            [
                result.text,
                "\n\n",
                """Based on the nutrient analysis provided, categorize the meal as one of
the following:
                - Not Healthy
                - Healthy
                - Nutritious
                - Somewhat balanced
                - Highly balanced and nutritious
                Provide a short explanation for your assessment and, if required, give tips to
make it more balanced.
```

```
                    """
                ]
            )

            st.markdown("<div style='background-color:#000000; padding: 10px; border-
radius: 5px;'>"
                        "<strong>Meal Balance Assessment:</strong><br>"
                        f"{balance_result.text}"
                        "</div>", unsafe_allow_html=True)

        except Exception as e:
            st.error(f"An error occurred: {e}")

    if os.path.exists(temp_image_path):
        os.remove(temp_image_path)

# Feature 3: Food Allergen Checker
elif app_mode == "Food Allergen Checker":
    st.markdown('<div class="feature-bg"><h2>Food Allergen Checker</h2></div>',
unsafe_allow_html=True)
    food_item = st.text_input("Enter the food dish you want to check (e.g., Pizza):")
    allergen = st.text_input("Enter any specific allergens if you have (optional):")
    image_file = st.file_uploader("Upload an image of the food dish (optional):",
type=["jpg", "jpeg", "png"])

    if st.button("Check Allergens"):
        temp_image_path = None  # Initialize variable for temporary image path
        if food_item or image_file:
            prompt = ""
            if allergen:
                if food_item:
                    prompt += f"The dish '{food_item}' is being analyzed. I am allergic to
'{allergen}'. Does this dish contain my allergen? Please provide a simple 'yes', 'no', 'high
```

```
probability', or 'low chances', and 4-5 lines of explanation with a warning about avoiding
'{allergen}' in the dish."
        if image_file:
            temp_image_path = os.path.join("temp", image_file.name)
            with open(temp_image_path, "wb") as f:
                f.write(image_file.getbuffer())
            prompt += f"\nThis is the food '{temp_image_path}' I am going to eat. I am
allergic to '{allergen}'. Just tell if the dish can generally contain '{allergen}'. Give a one-
word answer followed by a detailed explanation."
    else:
        if food_item:
            prompt += f"Analyze the dish '{food_item}' for potential allergens. What
ingredients might cause allergic reactions? Please provide a summary."
        if image_file:
            temp_image_path = os.path.join("temp", image_file.name)
            with open(temp_image_path, "wb") as f:
                f.write(image_file.getbuffer())
            prompt += f"\nThis is the food '{temp_image_path}' I am going to eat. What
ingredients might cause allergic reactions? Please provide a summary."

    # Execute AI model
    if prompt:
        model = genai.GenerativeModel("gemini-1.5-flash")
        try:
            if temp_image_path:
                myfile = genai.upload_file(temp_image_path)
                result = model.generate_content([myfile, "\n\n", prompt])
            else:
                result = model.generate_content(prompt)


            response_lines = result.text.splitlines()
            simple_response = response_lines[0] if response_lines else "No response
received."
            explanation = "\n".join(response_lines[1:]) if len(response_lines) > 1 else ""
```

```
                # Display responses in colored boxes
                if "yes" in simple_response.lower():
                    response_color = "#FF4D4D"
                elif "high probability" in simple_response.lower():
                    response_color = "#FF4D4D"
                elif "no" in simple_response.lower():
                    response_color = "#ADD8E6"
                elif "low chances" in simple_response.lower():
                    response_color = "#B0E0E6"
                else:
                    response_color = "#D3D3D3"

                st.success("Allergen check completed!")
                st.markdown(
                    f"<div style='border: 2px solid {response_color}; padding: 10px; border-
radius: 5px; background-color: {response_color};'>Response: {simple_response}</div>",
                    unsafe_allow_html=True
                )
                if explanation:
                    st.markdown(
                        f"<div style='border: 2px solid #2196F3; padding: 10px; border-radius:
5px;'><strong>Explanation:</strong><br>{explanation}</div>",
                        unsafe_allow_html=True
                    )
            except Exception as e:
                st.error(f"An error occurred during the allergen check: {e}")
            finally:
                if temp_image_path and os.path.exists(temp_image_path):
                    os.remove(temp_image_path)
        else:
            st.error("Unable to generate a prompt for the AI. Please check your inputs.")
    else:
        st.error("Please enter a food dish or upload a photo to check.")
```

```
# Feature 4: Diet Recommendation for Chronic Conditions
elif app_mode == "Diet Recommendation for Chronic Conditions":
    st.markdown('<div class="feature-bg"><h2>Diet Recommendation for Chronic
Conditions</h2></div>', unsafe_allow_html=True)

    condition = st.text_input("Enter your chronic condition (e.g., Diabetes, Hypertension):")
    severity = st.selectbox("Select the severity level:", ["Mild", "Moderate", "Severe"])
    goals = st.text_input("Health goals related to this condition (e.g., lower blood pressure,
manage blood sugar)")
    food = st.selectbox("What type of meal do you like?", ["Veg", "Non-Veg", "Vegan"])

    if st.button("Get Diet Recommendation"):
        try:
            messages = [
                {
                    "role": "user",
                    "content": (
                        f"Provide detailed information for
{condition},{severity},{goals}{food}.\n"
                        "1. Conditional-specific nutritional guidelines\n"
                        "2. Customized meal planning. Include options for day, week, or month.\n"
                        "3. Symptom management tips\n"
                        "4. Educational content and lifestyle tips.\n"
                        "Please format the response accordingly."
            )
                }
            ]
            completion = client.chat.completions.create(
                model="llama3-70b-8192",
                messages=messages,
                temperature=1,
                max_tokens=1024
            )
            diet_recommendation = completion.choices[0].message.content
            st.write(diet_recommendation)
        except Exception as e:
            st.error(f"Error: {e}")

# Feature 5: Personalized Diet and Fitness Syncing
elif app_mode == "Personalized Diet and Fitness Syncing":
    st.markdown('<div class="feature-bg"><h2>Personalized Diet and Fitness
Syncing</h2></div>', unsafe_allow_html=True)
    # User Inputs
    age = st.number_input("Age", min_value=1, max_value=100, value=30)
    gender = st.selectbox("Gender", ["Male", "Female", "Other"])
    weight = st.number_input("Weight (kg)", min_value=20, max_value=200, value=70)
    height = st.number_input("Height (cm)", min_value=100, max_value=250, value=165)
    body_fat_percentage = st.number_input("Body Fat Percentage (%)", min_value=1,
max_value=50, value=20)
    sleep_hours = st.number_input("Average Sleep Hours per Night", min_value=1,
max_value=24, value=7)
    stress_level = st.selectbox("Stress Level", ["Low", "Medium", "High"])

    goal = st.selectbox("Diet Goal", ["Weight Loss", "Weight Gain", "Maintenance"])
    activity_level = st.selectbox("Activity Level", ["Sedentary", "Lightly Active",
"Moderately Active", "Very Active"])

    protein_goal = st.number_input("Daily Protein Goal (g)", min_value=10,
max_value=300, value=100)
    carb_goal = st.number_input("Daily Carbohydrate Goal (g)", min_value=20,
max_value=500, value=250)
    fat_goal = st.number_input("Daily Fat Goal (g)", min_value=10, max_value=200,
value=70)

    dietary_preferences = st.selectbox("Dietary Preferences", ["No Preference",
"Vegetarian", "Vegan", "Low-Carb", "High-Protein"])
```

```
    dietary_restrictions = st.multiselect("Dietary Restrictions", ["Gluten-Free", "Dairy-
Free", "Nut-Free", "Shellfish-Free"])

    time_frame_type = st.selectbox("Select Time Frame Type", ["Days", "Weeks",
"Months"])
    if time_frame_type == "Days":
        time_frame = st.number_input("Time Frame (in days)", min_value=1,
max_value=365, value=7)
    elif time_frame_type == "Weeks":
        time_frame = st.number_input("Time Frame (in weeks)", min_value=1,
max_value=52, value=1) * 7
    else:
        time_frame = st.number_input("Time Frame (in months)", min_value=1,
max_value=12, value=1) * 30

    calorie_goal = st.number_input("Daily Calorie Goal (kcal)", min_value=1000,
max_value=5000, value=1800)

    recent_workout_data = st.text_area("Enter recent workout data (e.g., duration, type,
intensity):", "")

    hydration_goal = st.slider("Daily Hydration Goal (L)", min_value=1.0, max_value=5.0,
value=2.5)

    tracking_period = st.selectbox("Choose Tracking Period", ["Daily", "Weekly"])  #
Ensure this is defined before being used in prompt

    st.sidebar.header("Daily Motivational Tip")
    motivational_tips = [
        "Stay consistent! Small progress adds up.",
        "Remember to rest and recover.",
        "Hydrate! Water is key to wellness.",
        "Take it one day at a time, you've got this!",
        "Celebrate small victories!",
    ]
    st.sidebar.write(motivational_tips[datetime.datetime.now().day %
len(motivational_tips)])

    if st.button("Generate Diet and Fitness Plan"):
        prompt = f"""
        You are a nutrition and fitness expert. Based on the following information, create a
detailed personalized diet and fitness plan:
        - Age: {age}
        - Gender: {gender}
        - Weight: {weight} kg
        - Height: {height} cm
        - Body Fat Percentage: {body_fat_percentage}%
        - Sleep: {sleep_hours} hours per night
        - Stress Level: {stress_level}
        - Goal: {goal}
        - Activity Level: {activity_level}
        - Dietary Preferences: {dietary_preferences}
        - Dietary Restrictions: {", ".join(dietary_restrictions)}
        - Time Frame: {time_frame} days
        - Daily Calorie Goal: {calorie_goal} kcal
        - Protein Goal: {protein_goal} g
        - Carbohydrate Goal: {carb_goal} g
        - Fat Goal: {fat_goal} g
        - Recent Workout Data: {recent_workout_data}
        - Daily Hydration Goal: {hydration_goal} L

        For each day of the time frame, provide:
        1. Detailed dietary recommendations, considering recent workout data and dietary
restrictions.
        2. Breakdown of meals: breakfast, lunch, dinner, and snacks, with specific foods
listed.
        3. Calorie and nutrient breakdown for each meal (e.g., protein, carbs, fats).
        4. Hydration goals and reminders.
        5. A summary of daily and weekly intake to help track calories and nutrient balance,
including insights for {tracking_period} tracking.
        6. Provide tips on food choices and alternatives to reach the calorie goal.
        7. Suggest food alternatives based on available ingredients and preferences.
        8. Include a brief motivational tip for each day to help the user stay on track.
        """
        with st.spinner("Generating your personalized diet and fitness plan..."):
            try:
                # Use Groq client to generate the plan
                messages = [{"role": "user", "content": prompt}]
                completion = client.chat.completions.create(
                    model="llama3-70b-8192"
,
                    messages=messages,
                    temperature=1,
                    max_tokens=1024
                )
                plan = completion.choices[0].message.content
                st.success("Diet and fitness plan generated successfully!")
                st.write(plan)
            except Exception as e:
                st.error(f"An error occurred: {e}")

    if tracking_period == "Weekly":
        st.header("Progress Overview")
        days = [f"Day {i}" for i in range(1, time_frame + 1)]
        calorie_intake = [calorie_goal + (i % 5 * 10 - 20) for i in range(time_frame)]
        fig, ax = plt.subplots(figsize=(12, 6))
        ax.plot(days, calorie_intake, label="Calorie Intake", color='blue', marker='o')
        ax.set_xlabel("Days")
        ax.set_ylabel("Calories (kcal)")
        ax.set_title("Weekly Calorie Intake")
        ax.legend()
        st.pyplot(fig)
```

4.2 IMPLEMENTATION

Create a directory folder as following and copy relevant pieces of code into the required parts: -

```
NutriGuide/
│
├── app.py                  # Main backend server using Groq or FastAPI/Flask
├── requirements.txt             # Python dependencies (if any)
│
├── models/                 # Folder to keep your trained ML models
│   ├── lstm_model.pkl           # Trained LSTM model for long-term price prediction
│   ├── linear_regression.pkl     # Trained Linear Regression model
│   └── voting_regressor.pkl      # Trained Voting Regressor ensemble model
│
├── static/                # Static files for frontend (CSS, JS, images)
│   ├── css/
│   │   └── style.css           # Custom styles for webpage
│   └── js/
│       └── main.js            # Frontend scripts (if any)
│
├── templates/              # HTML templates for your webpage
│   ├── index.html            # Homepage with input forms and prediction display
│   ├── about.html            # About or info page (optional)
│   └── results.html          # Page to display prediction results
│
├── groq_backend/              # Backend logic folder using Groq
│   ├── predict.py            # Code to load models and run predictions
│   └── utils.py            # Utility functions (data preprocessing, etc.)
│
└── README.md                 # Project description and instructions
```

Installing Python Packages

Open your terminal and run the following command to install all required packages:

```
pip install groq numpy pandas scikit-learn tensorflow flask
```

Note:

groq for backend model serving

tensorflow for LSTM model handling

flask to serve the webpage and backend APIs

Add any other dependencies you use in your code here.

If you have a requirements.txt file, you can run:

bash

Copy

Edit

```
pip install -r requirements.txt
```

5. TESTING

5.1 INTRODUCTION TO TESTING

Testing is a critical phase in software development that ensures the application functions correctly, efficiently, and reliably under various conditions. It verifies that the system meets its intended specifications and provides a seamless user experience by identifying and resolving bugs, performance issues, or inconsistencies early on.

In this NutriGuide project, testing was vital to confirm that personalized nutritional recommendations, food image recognition, and nutrient impact predictions operate accurately and cohesively. The aim was to validate that each component — from user input handling to model predictions and result display — works as expected and delivers meaningful health guidance.

The test cases were designed to verify key functionalities and user workflows, including:

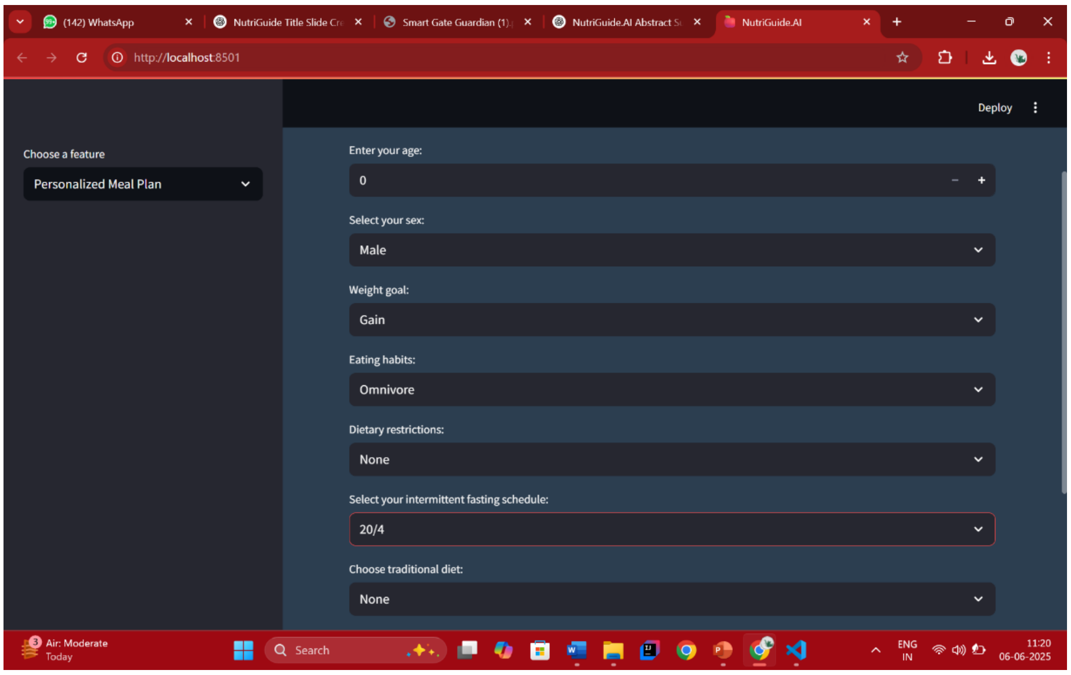
- Correct navigation and transitions across the webpage sections (input forms, recommendation display, and informational pages)
- Accurate generation of personalized nutrition advice based on user data
- Precise identification of food items from images uploaded by users
- Proper prediction of nutrient impact on health metrics
- Robust integration between the frontend and Groq backend for model inference
- Handling of edge cases such as incomplete inputs or unsupported food images
- Through systematic testing, the project ensures a reliable and user-friendly nutritional platform that supports effective, personalized dietary decision-making.

5.2 TEST CASES:

Table 5.1 Test Cases of Rhythm Restore

Test Case ID	Test case Name	Test Description	Input	Output	Remarks
TC_01	Valid User Input	Test valid input data for nutrition recommendation	Age: 25, Weight: 70kg, Goal: Weight Loss	Personalized meal plan and nutrient goals shown	pass
TC_02	Empty User Input	Test form submission with empty or missing fields	Blank form	Error message: "Please fill all required fields."	pass
TC_03	Invalid Data Type	Submit incorrect data types (e.g., text in weight field)	Weight: "abc"	Error message: "Please enter a valid number for weight."	pass
TC_04	Food Image Upload	Upload a clear food image (e.g., apple) for recognition	Image of an apple	Predicted label: "Apple", Nutrition info displayed	pass
TC_05	Unsupported Food Image	Upload a blurry or unsupported food image	Image of a cartoon burger	Error message: "Unable to recognize food item. Try again."	pass
TC_06	Prediction Integration with Groq Backend	Test if prediction is made through Groq API when form is submitted	Valid user profile input	Backend returns prediction result, frontend displays recommendation	pass
TC_07	Navigation Flow	Navigate between Home → Input → Results	Click "Start", fill form, submit	Pages load smoothly with relevant data displayed	pass
TC_08	Nutrition Impact Prediction	Provide macro/micronutrient data and verify predicted outcome	Carbs: 60g, Protein: 40g, Fat: 20g	Output: "Expected weight loss: 0.5 kg/week"	pass
TC_09	Responsive Design Test	Open on mobile/tablet screen	Resize browser or open on phone	Layout adjusts without breaking or misaligned UI	pass
TC_10	API Timeout/Error Handling	Simulate slow or failed backend response	Delayed Groq response or disconnect	Message: "Prediction service unavailable. Please try later."	pass

6. RESULTS



Choose a feature: Personalized Meal Plan

Enter your age: 0

Select your sex: Male

Weight goal: Gain

Eating habits: Omnivore

Dietary restrictions: None

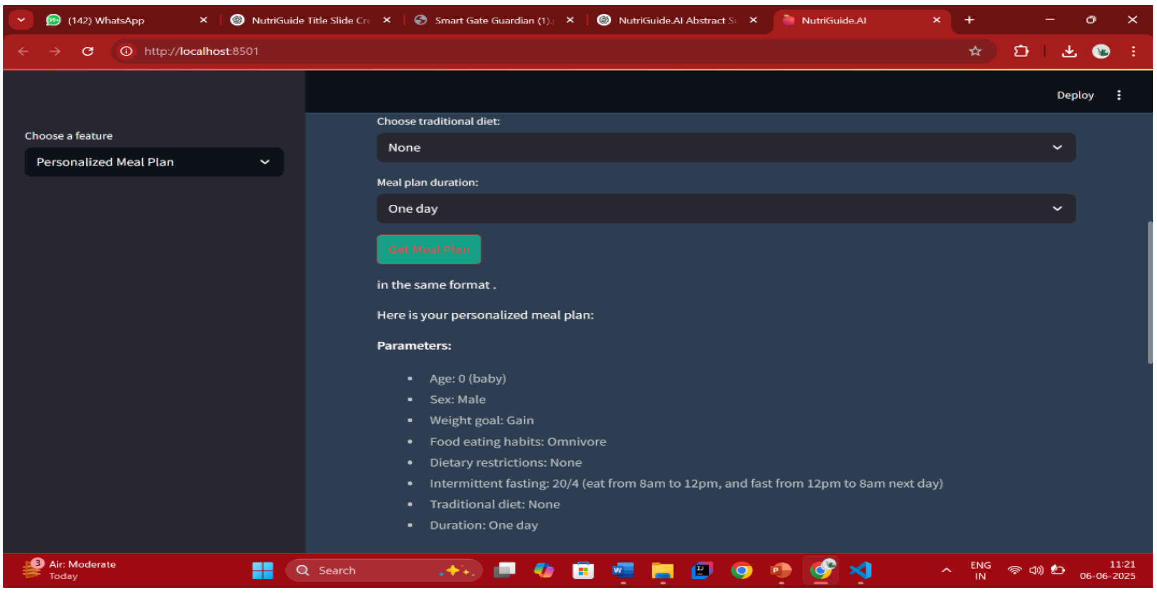
Select your intermittent fasting schedule: 20/4

Choose traditional diet: None

Fig. 6.1: Personalized Meal Plan Input Interface in NutriGuide.AI

Fig. 6.1 shows the personalized meal planning interface of the NutriGuide.AI web application, accessed through the localhost development server. The page allows users to input key health and dietary parameters such as age, sex, weight goal, eating habits, intermittent fasting schedule, and dietary restrictions. Based on these selections, the system generates a tailored meal plan aligned with the user's health objectives.

.



Choose traditional diet: None

Meal plan duration: One day

in the same format .

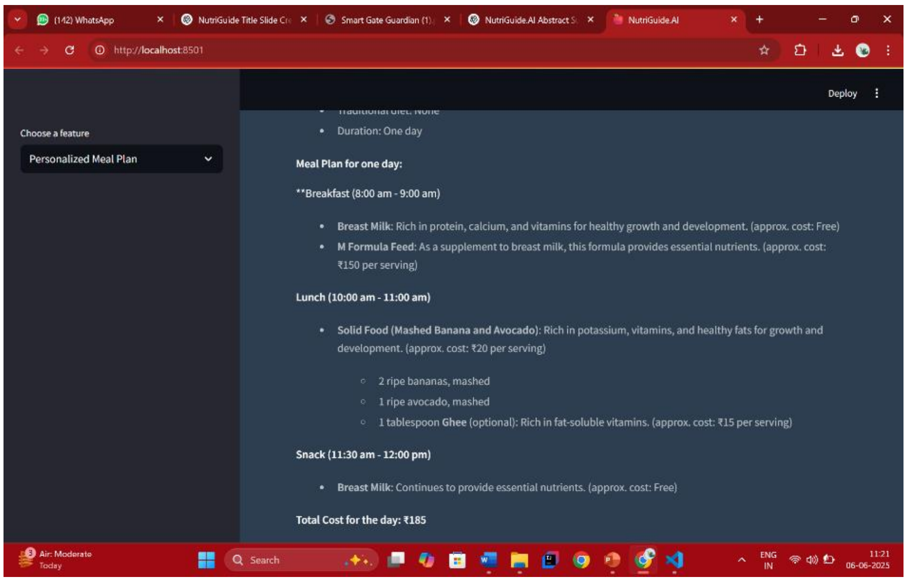
Here is your personalized meal plan:

Parameters:

- Age: 0 (baby)
- Sex: Male
- Weight goal: Gain
- Food eating habits: Omnivore
- Dietary restrictions: None
- Intermittent fasting: 20/4 (eat from 8am to 12pm, and fast from 12pm to 8am next day)
- Traditional diet: None
- Duration: One day

Fig. 6.2: Personalized Meal Plan Output Display in NutriGuide.AI

Fig. 6.2 displays the personalized meal plan results page of the NutriGuide.AI web application. After the user submits their dietary preferences, goals, and schedule, the system processes this data and presents a one-day customized meal plan. The parameters used to generate the plan, such as age, sex, weight goal, eating habits, and intermittent fasting schedule, are clearly listed for user reference.



- Duration: One day

Meal Plan for one day:

**Breakfast (8:00 am - 9:00 am)

- **Breast Milk**: Rich in protein, calcium, and vitamins for healthy growth and development. (approx. cost: Free)
- **M Formula Feed**: As a supplement to breast milk, this formula provides essential nutrients. (approx. cost: ₹150 per serving)

Lunch (10:00 am - 11:00 am)

- **Solid Food (Mashed Banana and Avocado)**: Rich in potassium, vitamins, and healthy fats for growth and development. (approx. cost: ₹20 per serving)
    - 2 ripe bananas, mashed
    - 1 ripe avocado, mashed
    - 1 tablespoon **Ghee** (optional): Rich in fat-soluble vitamins. (approx. cost: ₹15 per serving)

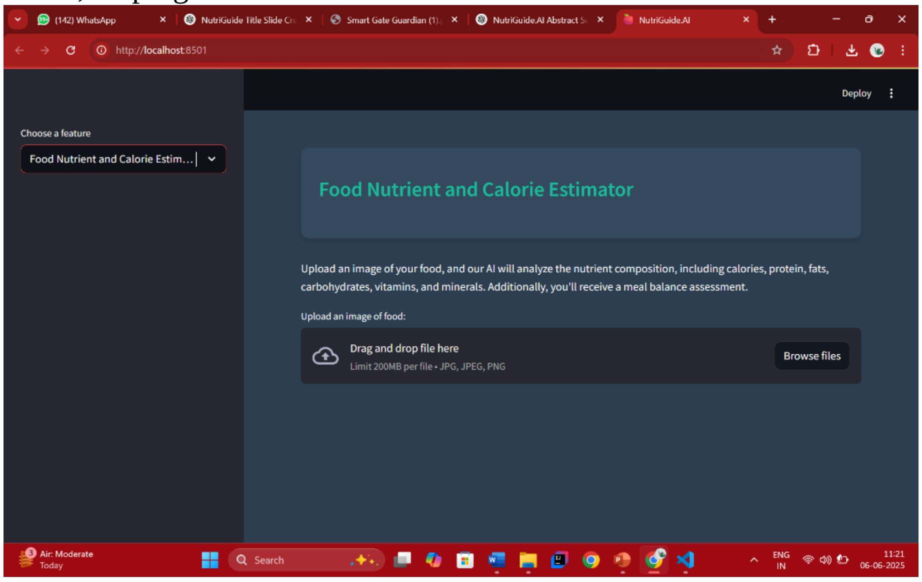
Snack (11:30 am - 12:00 pm)

- **Breast Milk**: Continues to provide essential nutrients. (approx. cost: Free)

Total Cost for the day: ₹185

Fig. 6.3 One-Day Personalized Meal Plan Output in NutriGuide.AI

Fig. 6.3 presents the generated one-day personalized meal plan displayed within the NutriGuide.AI web interface. Based on the user's input parameters, the system outlines meal suggestions with detailed time slots, nutritional explanations, and approximate costs. This structured plan includes breakfast, lunch, and snacks tailored for optimal growth and dietary alignment, helping users make informed and accessible food choices.


Choose a feature
Food Nutrient and Calorie Estim...
Deploy
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files

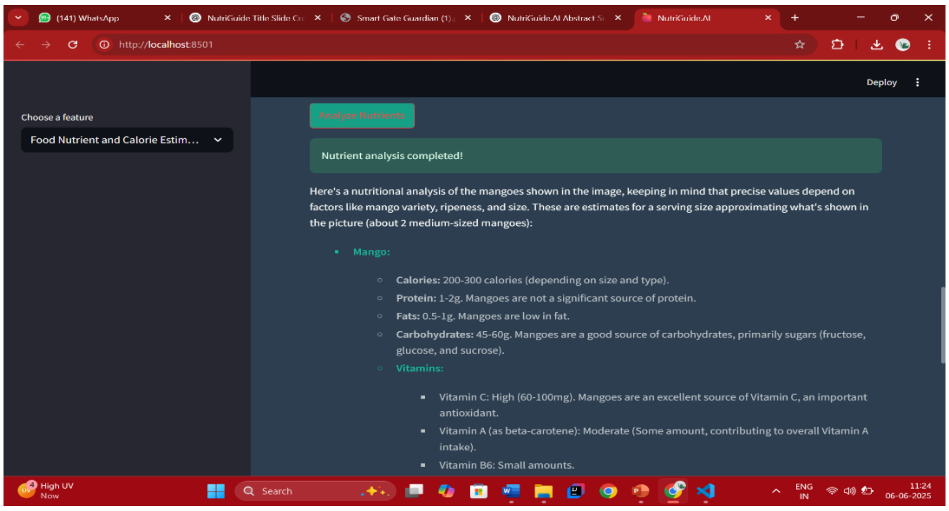
Food Nutrient and Calorie Estimator

Upload an image of your food, and our AI will analyze the nutrient composition, including calories, protein, fats, carbohydrates, vitamins, and minerals. Additionally, you'll receive a meal balance assessment.

Upload an image of food:

Fig. 6.4: Food Nutrient and Calorie Estimator Interface in NutriGuide.AI

Fig. 6.4 illustrates the "Food Nutrient and Calorie Estimator" page of the NutriGuide.AI web application. This feature allows users to upload an image of their meal, which the AI then analyzes to estimate its nutrient composition—including calories, proteins, fats, carbohydrates, and micronutrients. The interface supports image formats like JPG and PNG, and provides a drag-and-drop option for ease of use.



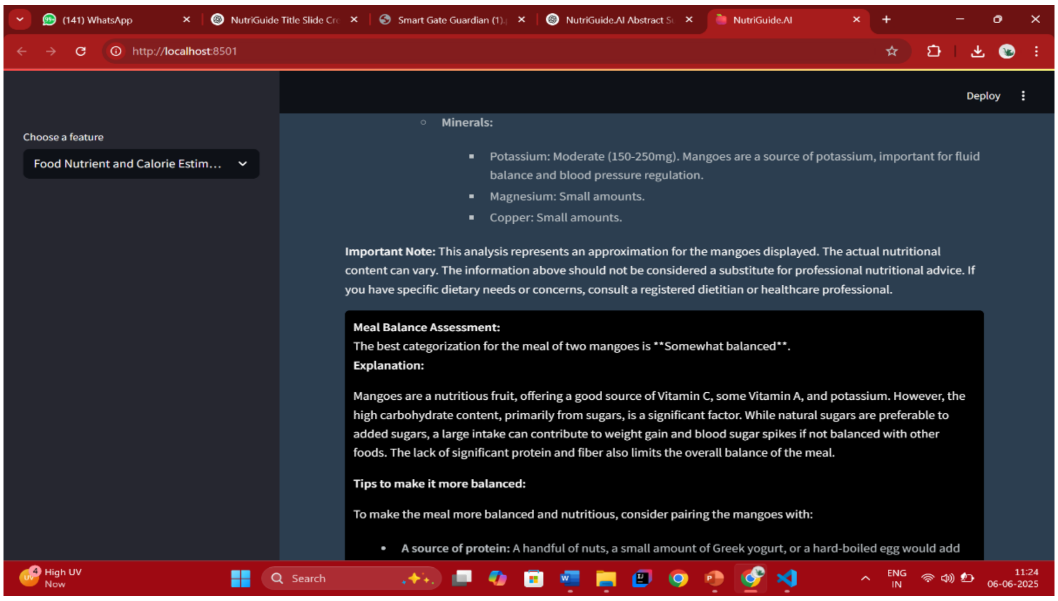
Nutrient analysis completed!

Here's a nutritional analysis of the mangoes shown in the image, keeping in mind that precise values depend on factors like mango variety, ripeness, and size. These are estimates for a serving size approximating what's shown in the picture (about 2 medium-sized mangoes):

- **Mango:**
  - **Calories:** 200-300 calories (depending on size and type).
  - **Protein:** 1-2g. Mangoes are not a significant source of protein.
  - **Fats:** 0.5-1g. Mangoes are low in fat.
  - **Carbohydrates:** 45-60g. Mangoes are a good source of carbohydrates, primarily sugars (fructose, glucose, and sucrose).
  - **Vitamins:**
    - Vitamin C: High (60-100mg). Mangoes are an excellent source of Vitamin C, an important antioxidant.
    - Vitamin A (as beta-carotene): Moderate (Some amount, contributing to overall Vitamin A intake).
    - Vitamin B6: Small amounts.

Fig. 6.5 : Nutrient Analysis Output of Food Image in NutriGuide.AI

Fig. 6.5 shows the output screen of the “Food Nutrient and Calorie Estimator” feature in the NutriGuide.AI application. After analyzing the uploaded food image, the system provides a detailed nutritional breakdown—in this case, for mangoes—including calorie range, macronutrients (proteins, fats, carbohydrates), and key vitamins. The AI-generated estimates are contextualized based on the food’s type, ripeness, and quantity.



- **Minerals:**
  - Potassium: Moderate (150-250mg). Mangoes are a source of potassium, important for fluid balance and blood pressure regulation.
  - Magnesium: Small amounts.
  - Copper: Small amounts.

Important Note: This analysis represents an approximation for the mangoes displayed. The actual nutritional content can vary. The information above should not be considered a substitute for professional nutritional advice. If you have specific dietary needs or concerns, consult a registered dietitian or healthcare professional.

Meal Balance Assessment:
The best categorization for the meal of two mangoes is **Somewhat balanced**.
Explanation:

Mangoes are a nutritious fruit, offering a good source of Vitamin C, some Vitamin A, and potassium. However, the high carbohydrate content, primarily from sugars, is a significant factor. While natural sugars are preferable to added sugars, a large intake can contribute to weight gain and blood sugar spikes if not balanced with other foods. The lack of significant protein and fiber also limits the overall balance of the meal.

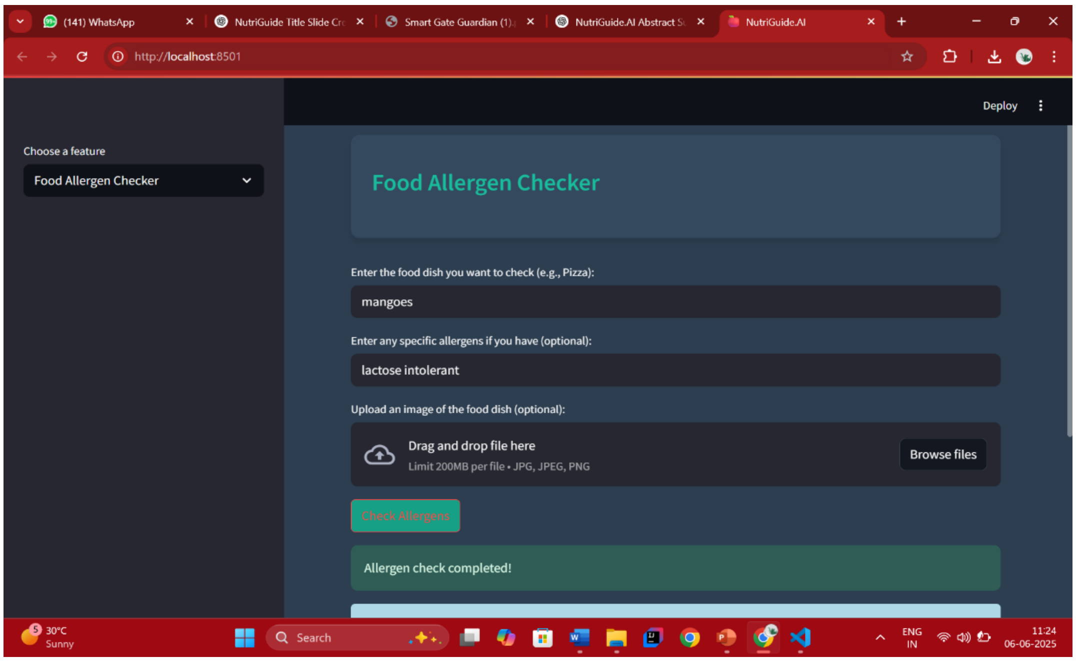
Tips to make it more balanced:

To make the meal more balanced and nutritious, consider pairing the mangoes with:

- **A source of protein:** A handful of nuts, a small amount of Greek yogurt, or a hard-boiled egg would add

Fig. 6.6: Meal Balance Assessment for Mangoes in NutriGuide.AI

Fig. 6.6 presents the detailed output of the AI-generated meal balance assessment in the NutriGuide.AI platform. Following the nutrient analysis, the system evaluates the overall balance of the meal—in this case, two mangoes—and provides a categorization (e.g., “Somewhat balanced”) along with an explanation. It also offers practical tips to enhance the nutritional value by suggesting complementary foods such as nuts, yogurt, or eggs, helping users improve meal composition effectively.



Choose a feature

Food Allergen Checker

Food Allergen Checker

Enter the food dish you want to check (e.g., Pizza):

mangoes

Enter any specific allergens if you have (optional):

lactose intolerant

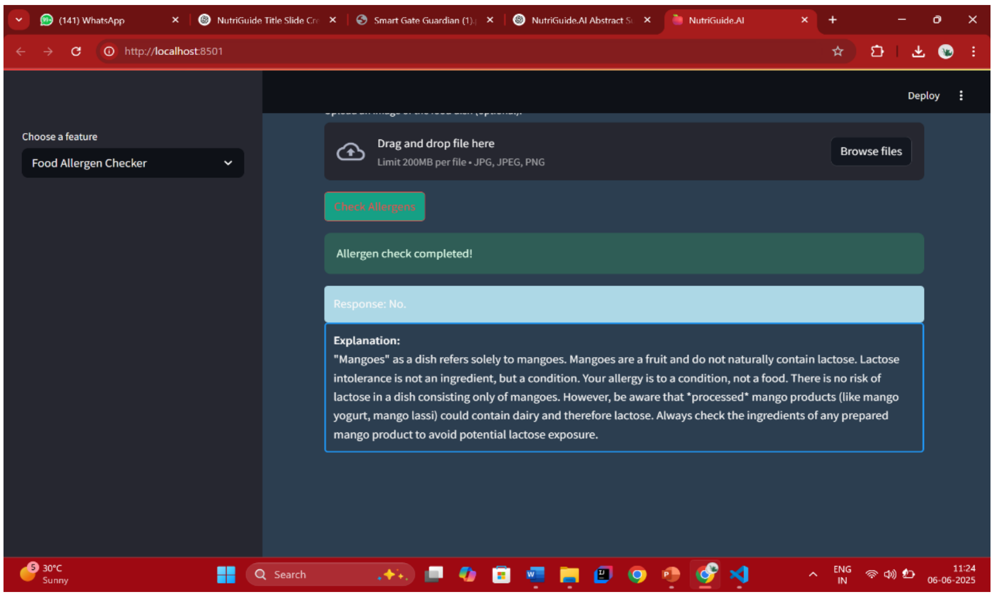
Upload an image of the food dish (optional):

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

Allergen check completed!

Fig. 6.7: Food Allergen Checker Interface in NutriGuide.AI

Fig. 6.7 displays the "Food Allergen Checker" feature of the NutriGuide.AI web application. Users can input a dish name and specify known allergies (e.g., lactose intolerance) to check for potential allergens. Optionally, users may upload an image of the food dish for enhanced analysis. Upon submission, the system verifies the food against allergen risk and provides real-time feedback, supporting safer dietary decisions



Choose a feature

Food Allergen Checker

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

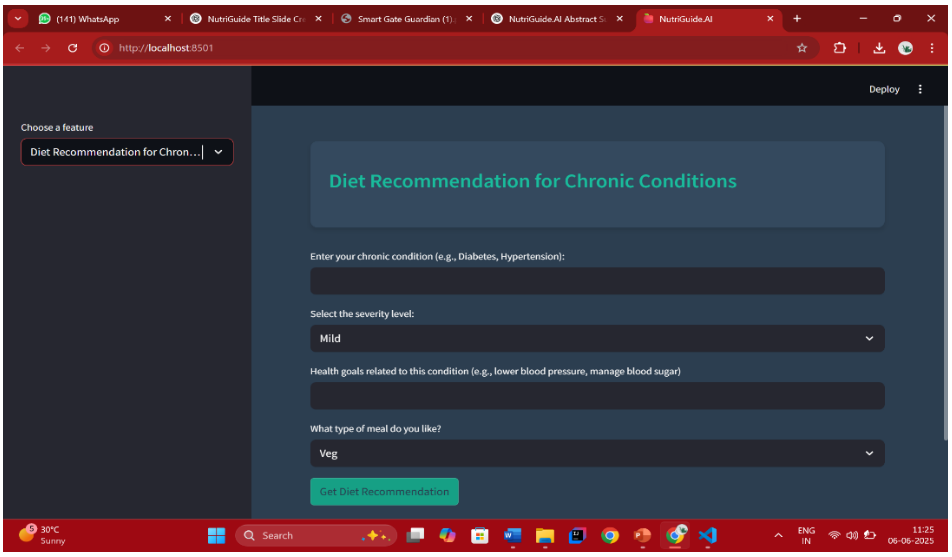
Allergen check completed!

Response: No.

Explanation:
"Mangoes" as a dish refers solely to mangoes. Mangoes are a fruit and do not naturally contain lactose. Lactose intolerance is not an ingredient, but a condition. Your allergy is to a condition, not a food. There is no risk of lactose in a dish consisting only of mangoes. However, be aware that *processed* mango products (like mango yogurt, mango lassi) could contain dairy and therefore lactose. Always check the ingredients of any prepared mango product to avoid potential lactose exposure.

Fig. 6.8: Allergen Check Result and Explanation

Fig. 6.8 shows the result screen of the "Food Allergen Checker" in the NutriGuide.AI platform. Based on the user's input (dish: mangoes; allergy: lactose intolerance), the system confirms that the food does not contain the specified allergen. An explanatory note clarifies that mangoes naturally do not contain lactose and advises caution only with processed mango-based products that may include dairy ingredients.


Choose a feature
Diet Recommendation for Chron...

Diet Recommendation for Chronic Conditions

Enter your chronic condition (e.g., Diabetes, Hypertension):

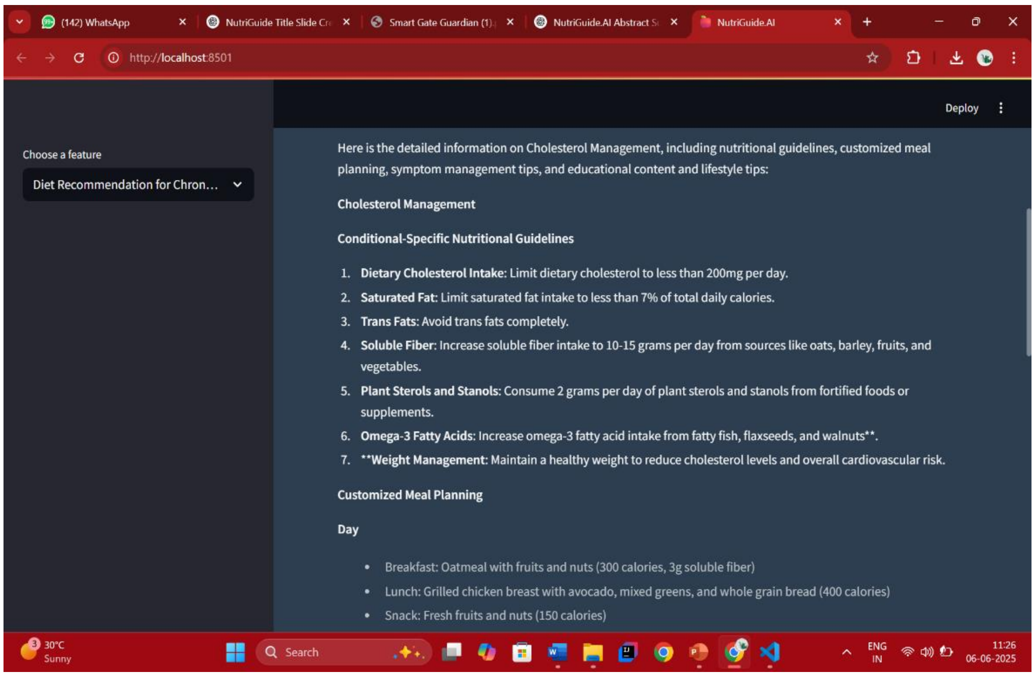
Select the severity level: Mild

Health goals related to this condition (e.g., lower blood pressure, manage blood sugar)

What type of meal do you like? Veg

Fig. 6.9: Diet Recommendation for Chronic Conditions – Input Interface

Fig. 6.9 illustrates the "Diet Recommendation for Chronic Conditions" interface within the NutriGuide.AI application. Users can input specific chronic conditions such as diabetes or hypertension, select the severity level, define related health goals, and choose their dietary preference (e.g., veg or non-veg). The system uses this information to generate tailored diet recommendations aligned with medical and nutritional guidelines



Here is the detailed information on Cholesterol Management, including nutritional guidelines, customized meal planning, symptom management tips, and educational content and lifestyle tips:

Cholesterol Management

Conditional-Specific Nutritional Guidelines

1. **Dietary Cholesterol Intake:** Limit dietary cholesterol to less than 200mg per day.
2. **Saturated Fat:** Limit saturated fat intake to less than 7% of total daily calories.
3. **Trans Fats:** Avoid trans fats completely.
4. **Soluble Fiber:** Increase soluble fiber intake to 10-15 grams per day from sources like oats, barley, fruits, and vegetables.
5. **Plant Sterols and Stanols:** Consume 2 grams per day of plant sterols and stanols from fortified foods or supplements.
6. **Omega-3 Fatty Acids:** Increase omega-3 fatty acid intake from fatty fish, flaxseeds, and walnuts**.
7. ****Weight Management:** Maintain a healthy weight to reduce cholesterol levels and overall cardiovascular risk.

Customized Meal Planning

Day

- Breakfast: Oatmeal with fruits and nuts (300 calories, 3g soluble fiber)
- Lunch: Grilled chicken breast with avocado, mixed greens, and whole grain bread (400 calories)
- Snack: Fresh fruits and nuts (150 calories)

Fig. 6.10: Cholesterol Management – Diet Recommendation Output

Fig. 6.10 shows the output of the "Diet Recommendation for Chronic Conditions" feature, where the NutriGuide.AI system provides a detailed plan for cholesterol management. It includes specific nutritional guidelines such as limiting saturated fats and increasing soluble fiber intake, along with a customized daily meal plan. The advice aligns with standard medical recommendations, helping users manage their condition effectively through informed dietary choices.

7. CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The NutriGuide project successfully showcases how artificial intelligence can empower individuals to make informed and personalized dietary decisions using a simple and accessible web platform. By analyzing real-time user input and integrating computer vision for food recognition, the system delivers tailored nutrition recommendations, aiding users in achieving their health goals—be it weight loss, muscle gain, or managing chronic conditions.

The project's core strength lies in its backend integration with Groq, enabling fast, efficient predictions without the need for extensive infrastructure. The interface is designed to be lightweight and intuitive, allowing users to upload food images, input dietary data, and instantly receive meaningful feedback. Furthermore, the absence of complex installations or databases makes NutriGuide deployable in low-resource settings or by users with minimal technical skills.

Overall, NutriGuide is a highly accessible, AI-driven solution that bridges the gap between personalized nutrition and everyday decision-making.

7.2 FUTURE ENHANCEMENTS

While the current version of NutriGuide performs well, the following enhancements could further improve its usability, intelligence, and reach:

- **Daily Habit Learning**: Train models to learn from recurring dietary patterns and suggest smarter long-term plans based on trends in eating behavior.
- **Mobile-First Experience**: Build an Android/iOS version of NutriGuide for easier use, push notifications, and offline access.
- **Deeper Model Integration**: Introduce advanced deep learning models such as LSTM or Transformers to better analyze user input sequences and predict long-term health outcomes.
- **Enhanced Food Detection**: Improve the image classification system to support a wider variety of food items and cuisines, increasing the system's accuracy and applicability.

- **Third-Party API Integration**: Integrate APIs from trusted nutrition databases like USDA or Edamam to fetch detailed nutritional data automatically for recognized food items.
- **Community Support Features**: Add social features such as progress sharing, user challenges, and discussion forums to foster user engagement and motivation.
- **Personal Progress Tracker**: Develop an interactive dashboard that visualizes user history, calorie trends, and nutrition balance for better self-monitoring and goal setting.

8. REFERENCES

[1] K. Azzimani, H. Bihri, A. Dahmi, S. Azzouzi and M. E. H. Charaf, "An AI Based Approach for Personalized Nutrition and Food Menu Planning," 2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Fez, Morocco, 2022, pp. 1-5, doi: 10.1109/ICECOCS55148.2022.9983099.

[2] Sarani Rad, Fatemeh, Maryam Amiri, and Juan Li. "Optimizing Nutritional Decisions: A Particle Swarm Optimization–Simulated Annealing-Enhanced Analytic Hierarchy Process Approach for Personalized Meal Planning." Nutrients 16.18 (2024): 3117.

[3] Rafael D. Gutiérrez, Miguel García-Torres, Pedro M. Sánchez, Antonio Berlanga, ChatDiet: Empowering personalized nutrition-oriented food recommender systems, Smart Health, Volume 34, 2024, 100821, ISSN 2352-6483, https://doi.org/10.1016/j.smhl.2024.100821

[4] Richa Arora, Divya Sharma, and Pooja Nair, An AI Based Approach for Personalized Nutrition and Food Menu Planning, 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), IEEE, 2022, pp. 1197–1201, https://doi.org/10.1109/ICSCDS53736.2022.9760970

[5] E. Kaldirim, T. Ucar, and M. Ince, A Deep Learning Based Food Image Recognition System for Calorie Estimation, 2020 28th Signal Processing and Communications Applications Conference (SIU), IEEE, 2020, pp. 1–4, https://doi.org/10.1109/SIU49456.2020.9302277

[6] F. Farahani, M. Firouzi, A. Sarrafzadeh, "Towards Personalized Nutrition Recommendation Using Federated Learning: Opportunities and Challenges," IEEE Internet of Things Journal, vol. 10, no. 6, pp. 4956–4968, March 2023, doi: 10.1109/JIOT.2022.3220000

[7] S. Chun, J. Kim, D. Oh, "FoodAI: An Intelligent Food Recognition System Based on Deep Convolutional Neural Networks," Sensors, vol. 21, no. 10, 2021, 3363, doi: 10.3390/s21103363

[8] M. Jayaraman, R. Rath, and V. S. Hegde, "DietLens: Mobile Application for Nutritional Analysis of Food using Machine Learning," 2022 IEEE World AI IoT Congress (AIIoT), pp. 118–123, doi: 10.1109/AIIoT54504.2022.9880987
[9] M. Elbadawi, A. Abuarqoub, and Y. Jararweh, "Nutrition-Aware IoT-Enabled Food Recommendation System," Journal of Ambient Intelligence and Humanized Computing, vol. 12, 2021, pp. 9535–9550, doi: 10.1007/s12652-020-02591-2
[10] H. Mousavi, N. Shirmohammadi, "AI-Powered Nutrition Assistant: Combining NLP and Recommendation Systems for Personalized Dietary Advice," Procedia Computer Science, vol. 198, 2022, pp. 191–198, doi: 10.1016/j.procs.2021.12.220