

# NutriAI - Project Development Phase

## 1. Problem Identification

The objective of this project is to develop an AI-powered web application that provides detailed nutritional information for various food items. Many users lack quick access to reliable nutritional data. NutriAI solves this problem using Generative AI to provide instant, structured, and informative outputs.

---

## 2. Requirement Analysis

Functional Requirements: - User should be able to input food items. - System should generate nutritional information. - Display macronutrients and micronutrients. Non-Functional Requirements: - Fast response time - Secure API integration - User-friendly interface

---

## 3. System Design

Architecture Used: - Frontend: Streamlit - Backend Logic: Python - AI Model: Gemini 2.5 Flash - Integration: LangChain + Google Generative AI API Flow: User Input → Prompt Template → Gemini Model → AI Response → Display Output

---

## 4. Implementation Phase

Technologies Used: - Python 3.13 - Streamlit - LangChain - Google Generative AI Key Modules: - PromptTemplate for structured prompting - Gemini model integration - Streamlit UI form handling

---

## 5. Testing Phase

Testing was performed using various food inputs such as: - Apple - Mango - Grapes The application successfully generated nutritional details including calories, protein, fats, carbohydrates, vitamins,

and minerals.

---

## **6. Deployment Phase**

The project can be deployed using: - Streamlit Cloud - Localhost deployment - GitHub version control Environment variables were used to securely manage API keys.

---

## **7. Conclusion**

NutriAI demonstrates the integration of Generative AI with web applications to solve real-world problems. The project enhances understanding of AI APIs, prompt engineering, and full-stack AI integration.

---