**Hackathon Project Phases Template** for the **NutriGen App** project.

Hackathon Project Phases Template

# Project Title:

# Advancing Nutrition Science through Gemini Ai

# 

# Team Name:

# CodeStrom

# Team Members:

* Sowmya Arutla
* Sharanraj Cheduluri
* Bamani Gangothri
* Battula Deepthi Priya

# Phase-1: Brainstorming & Ideation

## Objective:

## Develop an AI-powered nutrition analysis tool using Gemini AI to provide comprehensive food insights and personalized week meal planning.

## Key Points:

1. **Problem Statement:**

* Many individuals struggle to find reliable, up-to-date information about the nutritional value of various food items.
* Users need assistance in understanding macronutrients, micronutrients, and calorie content to make informed dietary choices.
* Creating personalized, balanced, and satisfying meal plans that align with dietary restrictions, allergies, and health conditions is a challenge

**2.Proposed Solution:**

* An AI-powered application using Google Generative AI (Gemini AI) to provide real-time nutritional data, including macronutrients, micronutrients, and calorie content.
* The app generates personalized meal plans based on user preferences, dietary restrictions, and health conditions.
* It offers customized grocery lists and recipe suggestions to ensure nutritional balance, variety, and enjoyment.

1. **Target Users:**

* **Health-conscious individuals** seeking nutritional insights.
* **People with dietary restrictions** needing personalized meal plans.
* **Athletes & fitness** **enthusiasts** optimizing nutrition for performance.

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the NutriGen App.

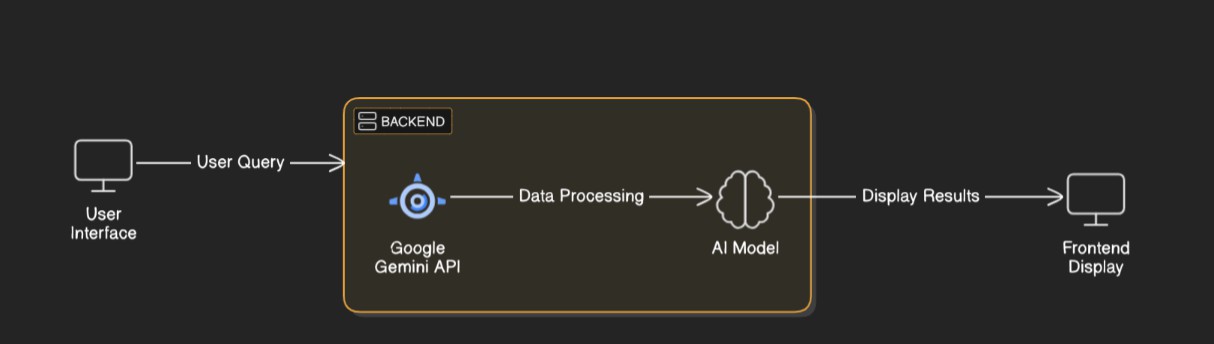
## Key Points:

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **Google Gemini AI**
   * Frontend: **Streamlit Web Framework**
   * Database: **Firebase for storing Authentication)**
2. **Functional Requirements:**
   * Fetch real-time nutritional data for food items using Google Generative AI API.
   * Generate personalized meal plans based on dietary preferences, allergies, and health conditions.
   * Provide smart grocery lists with cost-effective and seasonal food recommendations.
   * Offer AI-driven nutrition coaching with habit tracking and real-time meal feedback.
3. **Constraints & Challenges:**
   * Ensuring accurate and up-to-date nutritional data from AI-generated responses.
   * Handling API rate limits while optimizing query efficiency.
   * Providing a smooth and intuitive UI for users across web and mobile platforms.

# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of the application.



## Key Points:

1. **System Architecture:**

* User enters food-related query via UI.
* Query is processed using Google Gemini API.
* AI model fetches and processes nutritional data.
* The frontend displays detailed nutrition insights, meal plans, and recommendations.

1. **User Flow:**

* Step 1: User enters a query (e.g., "Nutritional value of bananas" or "Low-carb meal plan for a week").
* Step 2: The backend calls the Gemini AI API to retrieve relevant nutritional data.
* Step 3: The app processes the data and presents it in an intuitive, easy-to-read format.

1. **UI/UX Considerations:**

* Minimalist, user-friendly interface for seamless navigation.
* Filters for dietary preferences, allergens, and health goals.
* Dark & light mode for enhanced user experience.

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| Sprint 1 | Environment Setup & API Integration | 🔴 High | 6 hours  (Day 1) | End of Day 1 | Deepthi Priya | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 3 hours  (Day 1) | End of Day 1 | Sharanraj &  Sowmya | API response format finalized | Basic UI with input fields |
| Sprint 2 | Nutritional Data Retrieval & Analysis | 🔴 High | 5 hours  (Day 2) | Mid-Day 2 | Gangothri &  Sharanraj | API response, UI elements ready | Accurate nutrient analysis |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Gangothri &  Sowmya | API logs, UI inputs | Improved API stability |
| Sprint 3 | Meal Plan Generation & Personalization | 🟡  Medium | 2 hours  (Day 2) | Mid-Day 2 | Deepthi Priya | API response, UI layout completed | Personalized meal plans generated |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour  (Day 2) | End of Day 2 | Entire Team | Working prototype | Demo-ready project |

## Sprint Planning with Priorities

**Sprint 1 – Setup & Integration (Day 1)**

**(**🔴 **High Priority)** Set up the **environment** & install dependencies.

**(**🔴 **High Priority)** Integrate **Google Gemini API**.

**(**🟡 **Medium Priority)** Build a **basic UI with input fields**.

## Sprint 2 – Core Features & Debugging (Day 2)

**(**🔴 **High Priority)** Implement **search & comparison functionalities**. **(**🔴 **High Priority)** Debug API issues & handle **errors in queries**.

## Sprint 3 – Testing, Enhancements & Submission (Day 2)

**(**🟡 **Medium Priority)** Test API responses, refine UI, & fix UI bugs.

**(**🟢 **Low Priority)** Final **demo preparation & deployment**.

# Phase-5: Project Development

## Objective:

Implement core features of the NutiGen App.

## Key Points:

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Google Gemini Flash API (Gemini Ai)
   * **Programming Language:** Python
2. **Development Process**

* Implement API key authentication and integrate Google Gemini API for nutrition data retrieval.
* Develop personalized meal planning logic based on dietary preferences, allergies, and health conditions.
* Optimize search queries for accurate, efficient, and relevant nutritional insights.

1. **Challenges & Fixes:**

* Challenge: Delayed API response times.
* Fix: Implement caching to store frequently queried results and reduce API calls.

Challenge: Limited API calls per minute.

* Fix: Optimize queries to fetch only essential data and use request batching.

Challenge: Ensuring accurate personalized meal plans.

* Fix: Incorporate AI-driven recommendation models based on dietary needs and preferences.

Challenge: Handling diverse dietary restrictions and allergies.

* Fix: Allow users to specify multiple constraints and use AI to generate safe,

balanced meal options

# Phase-6: Functional & Performance Testing

## Objective:

Ensure that the NutiGen App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "Nutritional value of an apple" | Apple's macronutrient and micronutrient data should be displayed. | ✅ Passed | Deepthipriya |
| TC-002 | Functional Testing | Query "Meal plan for a diabetic person" | A personalized diabetic-friendly meal plan should be generated. | ✅ Passed | sharanraj |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Sowmya |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect nutrient data in API response | Data accuracy should be improved. | ✅ Fixed | Deepthipriya |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Gangothri |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | App should be accessible online. | 🚀 Deployed | DevOps |

# Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**