**Hackathon Project Phases Template** for the **Nutrition through GeminiAI App** project.

**Project Title:**

**Advancing Nutrition Science through GeminiAI**

**Team Name:**

(Science Bite Squad)

**Team Members:**

* Yamini Devandla
* Mahendarkar Sahithi
* Machini Manikanta

# Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-powered nutrition assistant using GeminiAI to help users access detailed nutritional information, generate personalized meal plans, and make informed dietary choices.

**Key Points:**

1. **Problem Statement:**

* 1. Nutrition science struggles with complexity and a lack of personalized solutions. It's hard to analyse the huge amount of data we have on food, health, and genetics to determine what truly works for an individual.

○ Users need instant access to comprehensive nutritional data on various food items, including macronutrients, micronutrients, and calorie content.

○ Gemini AI can potentially solve this by speeding up research, personalizing dietary advice, and making nutrition more practical for everyone.

1. **Proposed Solution:**

* 1. A Create a platform where GeminiAI integrates and analyses diverse nutrition data (genomics, diet logs, health records, research papers) due to Faster discovery of links between diet and health, personalized insights

○ The app generates week-long meal plans with recipes and grocery lists based on user inputs (dietary restrictions, health conditions, preferences, etc.)

1. **Target Users:**

* 1. **Health-conscious individuals looking for detailed nutritional information.**

○ Users with specific dietary needs (allergies, weight management, fitness goals).

○ Dietitians and nutritionists seeking AI-assisted meal planning tools.

1. **Expected Outcome:**

* 1. A functional AI-powered nutrition platform offering personalized meal plans, nutritional insights, and recipe recommendations.

○ Gemini AI in nutrition science will lead to quicker research, personalized eating plans, improved public health, better food choices, healthier products, and more effective treatments for nutrition-related ailments. This will allow individuals to make smarter choices to healthier, happy life.

# Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the **Nutrition through GeminiAI** App.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python,**SQL

○ Backend: **Google Gemini AI API, Hugging Face APIs**

○ Frontend: **Streamlit Web Framework**

○ Database: **Not required initially (API-based queries)**

1. **Functional Requirements:**

* 1. GeminiAI should process and analyse vast amounts of nutrition data (e.g., dietary surveys, research papers, food composition databases) quickly.

○ Gemini AI should predict the potential health impacts of different dietary patterns.

○ Provide recipe suggestions, grocery lists, and portion control recommendations

1. **Constraints & Challenges:**

* 1. Access to comprehensive and standardized nutrition data can be limited.

○ GeminiAI can inadvertently learn and amplify existing biases in the data

○ Providing a smooth and engaging UI/UX using Streamlit.

# Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* 1. User enters a nutrition-related query via the UI.

○ Query is processed using Google GeminiAI API.

○ AI model fetches and processes the relevant nutritional data.

○ The frontend displays food details, meal plans, and dietary insights

1. **User Flow:**

* 1. Step 1: User enters a query (e.g., "Nutritional value of almonds" or "Meal plan for weight loss").

○ Step 2: The backend calls the GeminiAI API to retrieve food data or generate a personalized meal plan.

○ Step 3: The app processes the data and **displays results** in an easy-to-read format.

1. **UI/UX Considerations:**

* 1. Prioritize clear and intuitive interfaces.

○ **Filters for price, mileage, and features**.

○ Translate data into clear, practical steps, Ensure the platform is usable by everyone.

# 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 | Koushik | Google API Key,  Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | Medium | 2 hours (Day 1) | End of Day 1 | Malik | API response format finalized | Basic UI with input fields |
| Sprint 2 | Nutrional Data Analysis | High | 3 hours (Day 2) | Mid-Day 2 | Gopi | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | High | 1.5 hours (Day 2) | Mid-Day 2 | Malik and Koushik | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | Medium | 1.5 hours (Day 2) | Mid-Day 2 | Malik and Koushik | API response, UI layout completed | Responsive UI, better user experience |
| 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 Nutrition through GeminiAI App.

**Key Points:**

1. **Technology Stack Used:**

* 1. **Frontend:** Gradio

○ **Backend:** Google Gemini Flash API and Hugging Face API

○ **Programming Language:** Python

1. **Development Process:**

* 1. Implement API key authentication and **Hugging Face API integration** for realtime nutrition and meal planning data.

○ Develop logic for **personalized meal plans** based on user input (dietary restrictions, preferences, etc.).

○ Optimize **search queries** for food nutrition data and meal plan generation for **performance** and **relevance**.

1. **Challenges & Fixes:**

* 1. **Challenge:** Delayed API response times.

**Fix:** Implement **caching** to store frequently queried results.

○ **Challenge:** Limited API calls per minute.

**Fix:** Optimize queries to fetch **only necessary data**.

# Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the Nutrition through GeminiAI App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "chicken" | The Nutrients present in chicken | Passed | Tester 1 |
| TC-002 | Functional | Query "Health Conditions | A Meal Planner based | Passed | Tester 2 |
|  | Testing | and Diet" | on query |  |  |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | Fixed | Develop er |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | Failed - UI broken on mobile | Tester 2 |
| 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**