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

## **Project Title:**

**Advancing Nutrition Science through GeminiAI**

## **Team Name:**

NutriAI

## **Team Members:**

* Mohammed Malik
* Gujju Koushik
* Anem Gopi

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

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

### **Key Points:**

1. **Problem Statement:**
   * Many individuals struggle to create healthy and satisfying meal plans that align with their dietary needs and preferences
   * Users need instant access to comprehensive nutritional data on various food items, including macronutrients, micronutrients, and calorie content.
   * There is a lack of AI-driven meal planning tools that factor in allergies, health conditions, activity levels, and taste preferences.
2. **Proposed Solution:**
   * A web-based application using Gemini AI to provide real-time nutritional analysis and personalized meal plans.
   * The app generates week-long meal plans with recipes and grocery lists based on user inputs (dietary restrictions, health conditions, preferences, etc.).
   * Offers food comparisons, dietary insights, and nutritional recommendations for a balanced diet.
3. **Target Users:**
   * **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.
4. **Expected Outcome:**
   * A functional AI-powered nutrition platform offering personalized meal plans, nutritional insights, and recipe recommendations.
   * Improved user awareness and engagement in healthy eating habits through AI-driven dietary analysis.

## **Phase-2: Requirement Analysis**

### **Objective:**

### Define the technical and functional requirements for the **NutriAI** App.

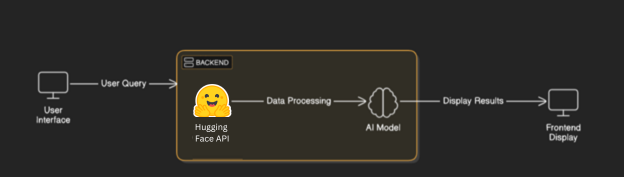
### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **Google Gemini AI API, Hugging Face APIs**
   * Frontend: **Streamlit Web Framework**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Fetch real-time nutritional data using Gemini AI and Edamam/Spoonacular APIs.
   * Display macronutrient, micronutrient, and calorie breakdowns in an intuitive UI.
   * Generate personalized meal plans based on user preferences and dietary needs.
   * Provide recipe suggestions, grocery lists, and portion control recommendations
3. **Constraints & Challenges:**
   * Ensuring real-time updates and accurate nutritional analysis from APIs.
   * Handling API rate limits while optimizing performance.
   * 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:**
   * User enters a nutrition-related query via the UI.
   * Query is processed using Google Gemini AI API.
   * AI model fetches and processes the relevant nutritional data.
   * The frontend displays food details, meal plans, and dietary insights
2. **User Flow:**
   * Step 1: User enters a query (e.g., "Nutritional value of almonds" or "Meal plan for weight loss").
   * Step 2: The backend calls the Gemini AI 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.
3. **UI/UX Considerations:**
   * Clean, intuitive interface for seamless navigation.
   * **Filters for price, mileage, and features**.
   * **Dark & light mode** for better 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 | 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 NutriAI App.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Gradio
   * **Backend:** Google Gemini Flash API and Hugging Face API
   * **Programming Language:** Python
2. **Development Process:**
   * Implement API key authentication and **Hugging Face API integration** for real-time 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**.
3. **Challenges & Fixes:**
   * **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 NutriAI App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "Egg" | The Nutrients present in egg | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | Query "Health Conditions and Diet" | A Meal Planner based on query | ✅ Passed | Tester 2 |
| 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 | Developer |
| 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**