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

# **Hackathon Project Phases Template**

## **Project Title:**

**Advancing Nutrition Science through GeminiAI**

## **Team Name:**

Code Cortex

## **Team Members:**

* Chava Abhilasha
* Mankidi Vineela Krishna Sri
* Chukkabotla Rithisha
* Bhuma Poojitha

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

### **Objective:**

The objective of advancing nutrition science through **Gemini AI** is to provide **accurate, personalized dietary recommendations** and **meal analyses**, empowering users to make healthier food choices using AI technology.

### **Key Points:**

1. **Problem Statement:**Current nutrition science lacks efficient, scalable tools for delivering **personalized and real-time dietary recommendations**. This makes it difficult for individuals to make informed food choices that align with their unique **health conditions, allergies, activity levels, and taste preferences**. Traditional approaches are slow, generalized, and not adaptable to real-time nutritional needs.
2. **Proposed Solution:**
   * An **AI-powered application** using **Gemini AI** to generate **real-time personalized meal plans** based on user profiles.
   * The app will analyze **dietary restrictions, allergies, activity levels, and taste preferences** to recommend **nutritious meal plans** with detailed recipes and grocery lists.
   * AI-driven **nutritional insights** will help users make healthier dietary choices and improve **public health outcomes**.
3. **Target Users:**
   * Nutrition researchers
   * Healthcare professionals (doctors, dietitians, nurses)
   * Food and beverage companies
   * Health and wellness startups
   * Public health officials
4. **Expected Outcome:**
   * A Faster Nutrition Research – AI-driven data analysis and insights
   * Personalized Diet Plans – Tailored recommendations for individuals.
   * Improved Public Health – Better disease prevention through AI-backed nutrition.
   * Simplified Knowledge Sharing – Easy access to scientific nutrition insights.
   * Accelerated Innovation – AI-powered food and supplement development.

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

### **Objective:**

### Develop an AI-powered nutrition system leveraging Gemini AI for personalized dietary recommendations and real-time health insights.

#### **1. Technical Requirements:**

* **Programming Language:** Python
* **Backend:** Google Gemini Flash API
* **Frontend:** Streamlit Web Framework
* **Database:** Not required initially (API-based queries)

**2.Functional Requirements:**

* Generate personalized meal plans based on user inputs.
* Provide detailed recipes and grocery lists.
* AI-powered nutritional analysis for meal suggestions.
* Adaptive dietary recommendations based on real-time user health data**.**

**3.Constraints & Challenges:**

* + Ensuring **real-time updates** from Gemini API.
  + Handling **API rate limits** and optimizing API calls.
  + Providing a **seamless user experience** with Streamlit.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * User **inputs dietary preferences** via UI.
   * **Gemini AI** processes the request and **generates personalized meal plans**.
   * The frontend **displays the results** with **recipes, grocery lists, and nutritional insights**..
2. **User Flow:**

 **Step 1:** User enters dietary preferences, health conditions, allergies, and activity   
 levels.

 **Step 2:** Backend **queries Gemini API** for AI-generated meal plans.

 **Step 3:** AI processes data and returns **tailored meal plans with detailed**

**Insights.**

**3**.**UI/UX Considerations:**

* + Minimalist, user-friendly interface for easy navigation.
  + Filters for meal preferences (vegetarian, keto, low-carb, etc.).
  + 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 | Shanawaz | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Member 2 | API response format finalized | Basic UI with input fields |
| Sprint 2 | Vehicle Search & Comparison | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | anwar | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Member 1&4 | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | mohammad | 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 Python, Gradio, and Gemini API environment  
 **(🔴 High Priority)** Integrate **Google Gemini API** for processing meal plan queries.  
 **(🟡 Medium Priority)** Develop a **basic UI with Gradio** for user input (e.g., dietary restrictions, allergies, and health conditions).

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

**(🔴 High Priority)** Implement **meal plan generation logic** based on user inputs  
 **(🔴 High Priority)** Optimize API queries for **efficient responses.  
(🔴 High Priority)** Debug issues like **API rate limits and response formatting**.

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

**(🟡 Medium Priority)** Test API responses, refine UI, & layout issues.  
 **(🟢 Low Priority)** Final **presentation, documentation, and deployment.**

## **Phase-5: Project Development**

### **Objective:**

Implement core features of the AutoSage App.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Google Gemini Flash API
   * **Programming Language:** Python

* **Development Process:**
  + Implement **API key authentication** and **Gemini API integration**.
  + Develop **meal plan generation and nutritional analysis logic**Optimize **search queries for faster response times**.

1. **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 AutoSage App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "7-day keto meal plan" | AI-generated keto meal plan | ✅ Passed | shanwaz |
| TC-002 | Functional Testing | Query "Best meals for diabetics" | AI suggests diabetes-friendly meals | ✅ Passed | anwar |
| 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 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**