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

# **Hackathon Project Phases Template**

## **Project Title:**

### Advancing Nutrition Science through Gemini AI

## **Team Name:**

The SmartBites

## **Team Members:**

* Aishwarya Reddy Kasireddy
* Jhoshna Pabbathi
* Akanksha Katherapaka

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

### **Objective:**

### Develop an AI-powered nutrition expert tool using Google Generative AI to help users analyze dietary needs and explore healthy food choices with nutritional count and ingredient list.

### **Key Points:**

1. **Problem Statement:**
   * Many individuals struggle to find reliable, up-to-date nutritional information about the foods they eat.
   * Users also need guidance on portion control, balanced diets, and understanding the nutritional value of different food items to make healthier choices.
2. **Proposed Solution:**
   * A web-based application powered by Google Generative AI to provide real-time nutritional analysis of various food items.
   * The app will deliver instant, detailed information on macronutrients (protein, fat, carbohydrates), micronutrients (vitamins, minerals), and calorie content**.**
3. **Target Users:**
   * Individuals looking to track their food intake and make healthier eating choices.
   * Fitness enthusiasts seeking to monitor their macronutrient intake and align it with their goals.
4. **Expected Outcome:**
   * A user-friendly, functional web-based application that delivers comprehensive, accurate, and real-time nutritional information on various foods based on the given input.

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

### **Objective:**

Define the technical and functional requirements for the AutoSage App.

### **Key Points:**

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:**
   * Ability to **fetch vehicle details** using Gemini Flash API.
   * Display nutritional information in an intuitive UI
   * Provide personalized dietary recommendations based on user preferences
   * Suggest eco-friendly and sustainable food choices
3. **Constraints & Challenges:**
   * Ensuring real-time updates from **Gemini API**.
   * Handling **API rate limits** and optimizing API calls.
   * Providing a **smooth UI experience** with Streamlit.

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

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * The user enters a food-related query or dietary preference via the UI
   * Query is processed using **Google Gemini API**.
   * AI model fetches and processes the data.
   * The frontend displays nutritional data with personalized dietary recommendations..
2. **User Flow:**
   * Step 1: User enters a query (e.g., "Low-carb meal plans for weight loss").
   * Step 2: The backend **calls the Gemini Flash API** to retrieve nutritional data for the generated diet plan.
   * Step 3: The app processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
   * The UI will feature a clean, straightforward design focused on user ease of navigation. This will ensure that users can easily find and understand the nutritional information they need.



## 

## **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 | Aishwarya | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Jhoshna | API response format finalized | Basic UI with input fields |
| Sprint 2 | Food Search & Nutritional information | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Akanksha | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Aishwarya & Akanksha | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Jhoshna | 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 **Food Search & Nutritional Information**.  
 **(🔴 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 AutoSage App.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Google Gemini Flash API
   * **Programming Language:** Python
2. **Development Process:**
   * Implement **API key authentication** and **Gemini API integration**.
   * Develop **vehicle comparison and maintenance tips logic**.
   * Optimize **search queries 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 Nutrition Science through GeminiAI web based application works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Generate meal plan for "Balanced" diet preference | Meal plan should be generated with 7 days of meals (Breakfast, Lunch, Dinner) with nutritional details | ✅ Passed | Aishwarya |
| TC-002 | Functional Testing | Generate meal plan for "Vegan" diet preference | Meal plan should be generated for Vegan diet, with detailed meals & nutrition | ✅ Passed | akanksha |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Jhoshna |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | ✅ Fixed | Akanksha |
| TC-005 | Final Validation | Ensure all diet types are available in the dropdown | The dropdown should list all the correct diet options: "Balanced", "Vegetarian", "Vegan", "Keto", "High Protein | ✅ Passed | Aishwarya |
| 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**