

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

Hackathon Project Phases Template

Project Title:

Advancing Nutrition Science through GeminiAI

Team Name:

LUMINEX

Team Members:

- Nikhat Fatima
 - G. Naveena
 - K. Thripada
-

Phase-1: Brainstorming & Ideation

Objective:

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

Key Points:

1. Problem Statement:

- Users often find it difficult to obtain reliable and detailed nutritional information on various food items, making informed dietary choices challenging.
- Designing personalized meal plans that accommodate dietary restrictions, allergies, and health conditions requires significant effort and expertise.

2. Proposed Solution:

- Develop an AI-powered nutrition assistant using GeminiAI to deliver accurate and instant nutritional insights, including macronutrients, micronutrients, and calorie content.
- Create a smart meal planner that generates personalized meal plans based on user dietary restrictions, allergies, and health conditions.

3. Target Users:

- Health-conscious individuals needing accurate nutrition information.
- People with dietary restrictions, allergies, or health conditions.
- Fitness enthusiasts seeking AI-driven meal planning.

4. Expected Outcome:

- An **AI-driven nutrition assistant** that empowers users with precise food insights and personalized meal plans for healthier living.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Advancing Nutrition Science through GeminiAI.

Key Points:

1. Technical Requirements:

- Programming Language: Python
- Backend: Google Gemini Flash API
- Frontend: Streamlit Web Framework
- Database: Not required initially (API-based queries)
- AI Integration: GeminiAI for nutritional analysis and meal planning
- User Input System: Form-based input for dietary needs and health conditions
- Real-Time Processing: Instant food nutrition breakdown and meal plan generation

2. Functional Requirements:

- The system should provide instant nutritional analysis of food items using Gemini AI.
- Users should be able to input dietary restrictions, allergies, and health conditions.
- The AI should generate personalized meal plans based on user inputs.
- The platform should process API queries in real time without needing a database.

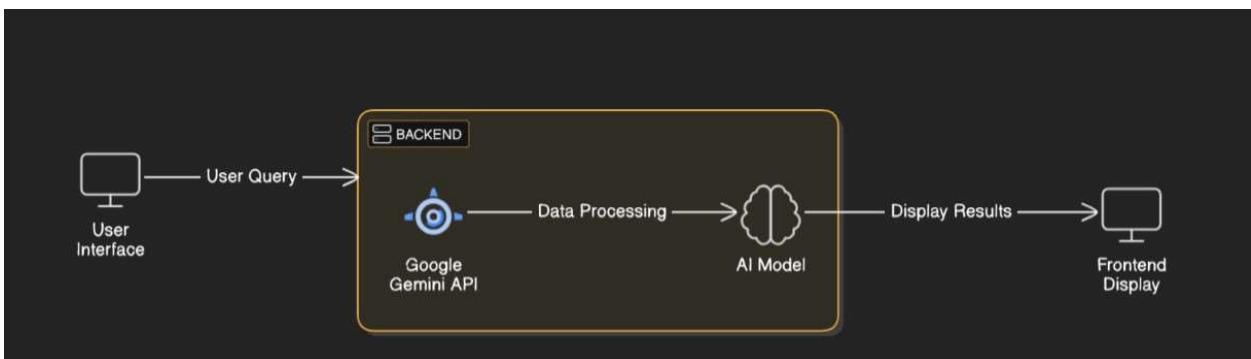
3. Constraints & Challenges:

- Ensuring the accuracy and reliability of nutritional data from AI-generated insights.
- Handling diverse dietary needs and health conditions while maintaining meal variety.
- Managing real-time processing efficiently without a dedicated database.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. System Architecture:

- User Interface (UI): Built with Streamlit for user input and result display.
- Backend Processing: Google Gemini Flash API handles nutritional analysis and meal planning.
- API Integration: Queries external food databases for real-time nutritional data.
- Response Handling: Processes user input, generates meal plans, and delivers results instantly.

2. User Flow:

- User Input: Users enter dietary restrictions, allergies, and health conditions.
- Data Processing: The system sends input to GeminiAI via API for analysis.
- AI Analysis: GeminiAI processes the data and generates nutritional insights and meal plans.
- Results Display: The personalized meal plan and food insights are shown to the user.

3. UI/UX Considerations:

- **Simple & Intuitive UI:** Streamlined design for easy navigation and quick inputs.
 - **Clear Data Presentation:** Well-structured nutritional insights and meal plans for better readability.
 - **Accessibility & Responsiveness:** Optimized for different devices with user-friendly interactions.
-

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 | Member 1 | 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 | Nutritional Analysis & Meal Plan Generator | ● High | 3 hours (Day 2) | Mid-Day 2 | Member 1& 2 | API response, UI elements ready | AI-driven nutritional breakdown & meal plan generation |
| 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 | Member 2& 3 | 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** for **nutritional analysis**.
- ( **Medium Priority**) Build a **basic UI** with **input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

( **High Priority**) Implement AI-driven nutritional breakdown and meal plan.

( **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 **Advancing Nutrition Science through GeminiAI**.

Key Points:

1. Technology Stack Used:

- **Programming Language:** Python
- **Frameworks & APIs:** Streamlit for frontend, Google Gemini Flash API for backend processing

- **Data Handling:** API-based queries (no database required initially)

2. Development Process:

- Phase 1: Define requirements, design UI, and set up GeminiAI integration.
- Phase 2: Develop core features, including user input handling and meal plan generation.
- Phase 3: Test, optimize performance, and deploy the application for user access.

3. Challenges & Fixes:

- **Challenge:** Ensuring accurate nutritional data from AI-generated insights.
Fix: Validate results using trusted external food databases.
- **Challenge:** Handling diverse dietary needs and allergies efficiently.
Fix: Implement a flexible input system with AI-driven customization.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Advancing Nutrition Science through GeminiAI webapplication works as expected.

| Test Case ID | Category | Test Scenario | Expected Outcome | Status | Tester |
|--------------|--------------------------|---|--|--|-----------|
| TC-001 | Functional Testing | Query "Nutritional value of almonds" | Detailed nutritional breakdown should be displayed. | <input checked="" type="checkbox"/> Passed | Tester 1 |
| TC-002 | Functional Testing | Generate a meal plan for a diabetic user | AI should provide a balanced diabetes-friendly plan. | <input checked="" type="checkbox"/> 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 micronutrient values in responses | Data accuracy should be improved. | <input checked="" type="checkbox"/> 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 |
|--------|--------------------|--------------------------------------|----------------------------------|--|--------|
