Hackathon Project Phases Template for the Flavour Fusion **App** project.

Hackathon Project Phases Template

Project Title:

Flavour Fusion App

Team Name:

Ruchi Sangam

Team Members:

- Surigi Prasanna Goud
- Veligandla Lakshmi Praveena
- Pratiksha Shinde
- Ramagiri Rushi Kumar
- Nampally Manish Kumar

Phase-1: Brainstorming & Ideation

Objective:

To leverage AI technology in flavour fusion recipe blogging by generating innovative, personalized, and culturally diverse recipes that cater to user preferences, promote culinary creativity, and enhance the overall cooking experience.

Key Points:

1. **Problem Statement:**

    - With the growing demand for diverse and personalized culinary experiences, there is a need for innovative platforms that generate unique flavour fusion recipes. However, manually curating fusion recipes that balance taste, cultural authenticity, and dietary preferences is time-consuming and complex. The challenge lies in developing an AI-driven recipe blogging platform that can

intelligently generate, customize, and recommend fusion recipes while maintaining flavour harmony, nutritional value, and user engagement.

2. **Proposed Solution:**
    - To address the challenge of creating personalized and innovative flavour fusion recipes, the proposed solution is to develop an AI-driven recipe blogging platform that leverages machine learning and data analytics to generate, customize, and recommend fusion recipes.
3. **Target Users:**
    - Individuals seeking personalized and innovative recipes for daily cooking
    - Influencers and vloggers generating food-related content for social media
    - Researchers studying AI applications in food innovation and sustainability.
4. **Expected Outcome:**
    - A functional AI-powered FlavourFusion app that provides insights based on real-time data and user queries.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the FlavourFusion.

Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: **Python**
    - Frontend: html , CSS
    - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
    - Secure login with two-factor authentication.
    - Allow bloggers to create, edit and delete blogs posts.

.

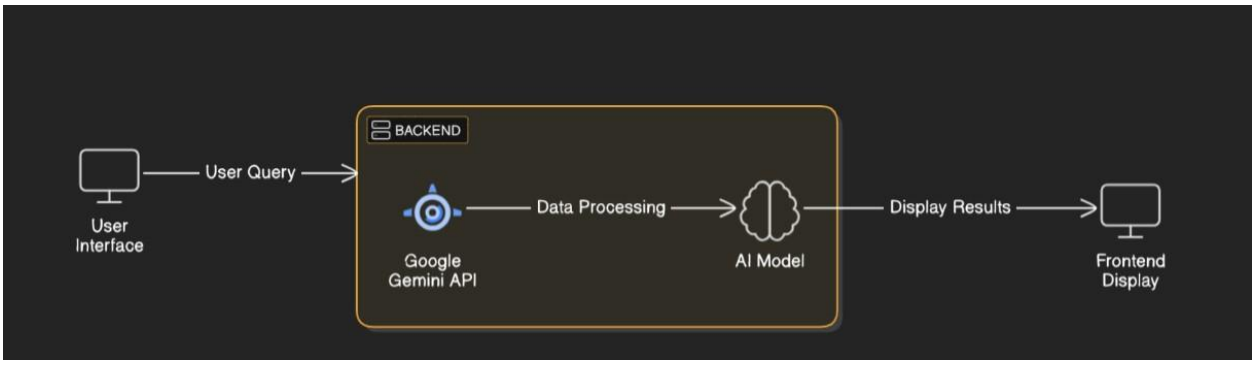
3. **Constraints & Challenges:**

    - Real-time AI processing speed for large datasets or media content.
    - Ensuring AI-generated recipes are unique, high-quality and not plagiarized.
    - 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:**

    - User enters vehicle-related query via UI.
    - AI model fetches and processes the data.
    - The frontend displays **vehicle details, reviews, and comparisons**.

2. **User Flow:**

    - Step 1: User enters a query (e.g., "Italian Dish").
    - Step 2: The backend **calls the python** to retrieve vehicle data.
    - Step 3: The app processes the data and **displays results** in an easy-to-read format.

3. **UI/UX Considerations:**

    - **Minimalist, user-friendly 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	Prasanna	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	Prasanna and Manish	API response format finalized	Basic UI with input fields
Sprint 2	Recipe search	High	3 hours (Day 2)	Mid-Day 2	Rishi and Praveena	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	Prasanna and Manish	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	Prasanna, Rishi, Manish, Pratiksha, Praveena	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 Flavour Fusion App.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** html, CSS
    - **Backend:** python
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Gemini API integration**.
    - Develop **recipes**.
    - 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 FlavourFusion App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "continental food "	Displays ingredients, steps-by-step instructions, nutrients and tips.	☑ Passed	Prasanna 1
TC-002	Performance Testing	API response time under 500ms	API should return results quickly.	☑ Passed	Prasanna
TC-003	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	☑ Fixed	Prasanna
TC-004	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Test
TC-005	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**