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

Hackathon Project Phases Template

# Project Title:

Flavor Fusion: AI-Driven Recipe Blogging

# Team Name:

Byte Bites

# Team Members:

* Thoshini
* Bhavigna
* Dharani
* Bhavitha

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an AI-powered recipe blog expert tool using Gemini Flash to help users create customized recipe blogs and entertain users with programmer jokes while generating content.

## Key Points:

1. **Problem Statement:**
   * The app allows users to input a topic and specify the desired word count for their recipe blog.
   * Using the specified parameters, the AI generates detailed and engaging recipe content.
   * Additionally, the app includes a fun feature that tells a programmer jokes to entertain users while the AI generates the content.
2. **Proposed Solution:**
   * An AI-powered application using **Gemini Flash** that **delivers a detailed and creative recipe with the specified word count given by the user.**
   * The app offers **jokes while generating the content**
3. **Target Users:**
   * **Food Bloggers**: Individuals who regularly create and share recipes and culinary content on their blogs.
   * **Home Cooks**: Enthusiastic cooks looking for inspiration and unique recipes to try at home.
   * **Recipe Developers**: Professionals or hobbyists who develop new recipes and need fresh ideas and inspiration.
4. **Expected Outcome:**
   * A functional **AI-powered recipe generation app** that provides insights based on user queries.

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the Flavor Fusion App.

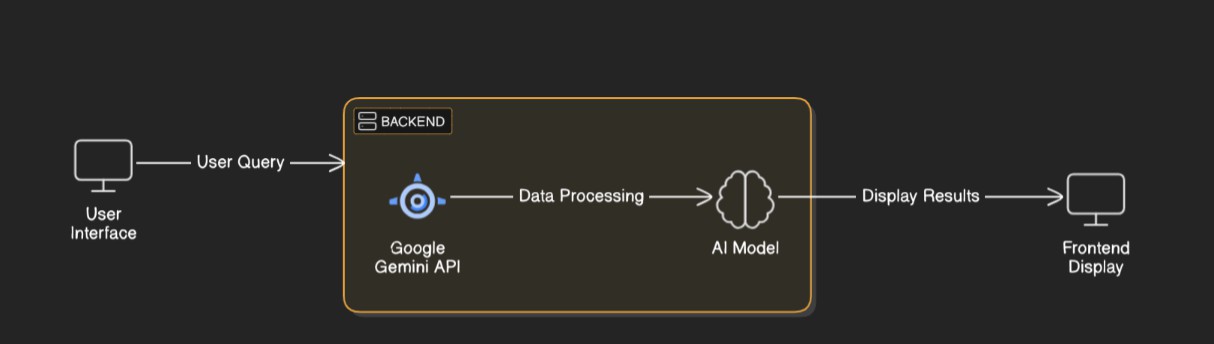
## Key Points:

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **Google Gemini Flash API**
   * Frontend: **Streamlit Web Framework, Base64**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Ability to **fetch recipe details** using Gemini Flash API.
   * Display a **list of ingredients and instructions** in an intuitive UI.
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:**
   * User enters recipe-related query via UI.
   * Query is processed using **Google Gemini API**.
   * AI model fetches and processes the data.
   * The front end displays the jokes, a **list of ingredients, and instructions.**
2. **User Flow:**
   * Step 1: User enters a query (e.g., "Vegan Chocolate Cake 1200").
   * Step 2: The backend **calls the Gemini Flash API** to retrieve recipe 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.

# 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 | Recipe Generation Logic Implementation | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1& 2 | API response, UI elements ready | Working content Generation module |
| Sprint 2 | Error Handling, Performance Optimization 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**.

**(**🟡 **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 Flavor Fusion 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 **Google** **Gemini AI integration**.
   * Develop **recipe generation logic & formatting**.
   * Optimize the **UI and ensure responsive design**.
3. **Challenges & Fixes:**
   * **Challenge:** Slow API response times.

**Fix:** Implement caching for frequently queried topics.

* + **Challenge:** Inconsistent blog.

**Fix:** Use text processing techniques for structuring content.

# Phase-6: Functional & Performance Testing

## Objective:

Ensure that the Flavor Fusion App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "Vegan Chocolate Cake 1200” | Well-structed recipe generated. | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | The joke feature  is triggered. | Joke is displayed during content generation. | ✅ Passed | Tester 2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |

# Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**