SmartChef - AI Recipe Application

Project Title:

SmartChef App

Team Name:

Innovate crew

Team Members:

- K. Medha Reddy
- K. Prashanthi
- M. Srilatha
- M. Jahnavi

Phase-1: Brainstorming & Ideation

Objective:

To provide AI-powered recipe recommendations and cooking assistance based on user preferences. It enhances the cooking experience by generating unique recipes and optimizing ingredients efficiently.

Key Points:

1. **Problem Statement:**
    - Many people struggle with finding new and personalized recipes that match their available ingredients.
    - Traditional recipe blogs lack AI-driven customization, making meal planning and cooking less efficient and time-consuming.

2. **Proposed Solution:**
    - Develop an AI-powered recipe app that generates personalized recipes based on user preferences, and available ingredients.
    - Integrate smart meal planning, ingredient optimization, and step-by-step cooking assistance to enhance the user experience.
3. **Target Users:**
    - **Home Cooks** – Easy, personalized, and AI-generated recipes.
    - **Busy Professionals –** Require quick meal planning with available ingredients.
4. **Expected Outcome:**
    - An **AI-powered recipe app** that provides personalized recipes, smart meal planning, and efficient ingredient optimization for diverse user needs.

Phase-2: Requirement Analysis

Objective:

To define the technical and functional requirements for the SmartChef app.

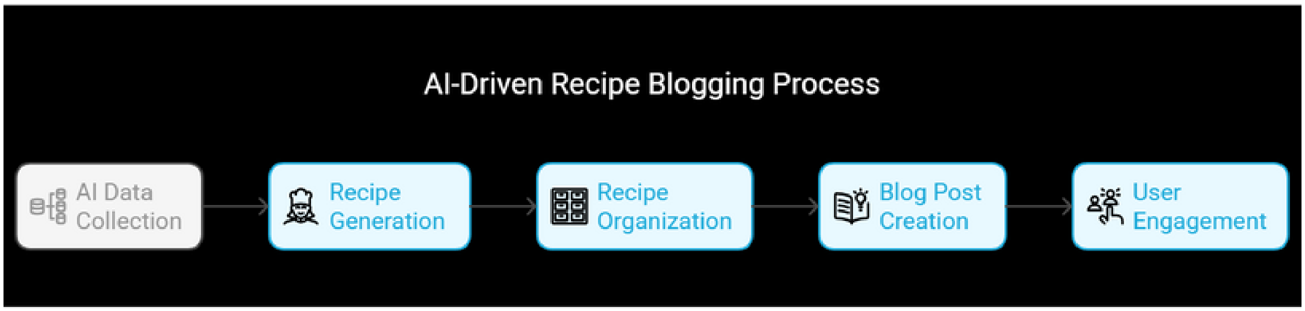
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Frontend: **HTML, CSS, Java Script**
    - Database: The **MealDB API**
2. **Functional Requirements:**
    - Display AI-generated recipes, reviews, and comparisons.
    - Provide real-time cooking tips and ingredient substitutions.
    - Allow users to search recipes based on available ingredients.
3. **Constraints & Challenges:**
    - Ensuring accurate and diverse recipe generation to meet different ingredients.
    - Managing API response time for a smooth and fast user experience.
    - Handling user customization requests while keeping AI suggestions relevant.

Phase-3: Project Design

Objective:

To develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   ○ User Input Handling: The user enters a query related to recipes.
   ○ AI Processing: The application processes the input and fetches relevant recipe data.
   ○ Data Retrieval: Fetches recipe details from an API or predefined data without a structured database.

2. **User Flow:**
   ○ Step 1: User inputs ingredients or recipe preferences (e.g., "pasta recipes").
   ○ Step 2: AI processes the input and generates structured recipes using collected data.
   ○ Step 3: The system organizes and displays recipes in a blog-friendly format for users to explore.

3. **UI/UX Considerations:**
   ○ Clean, visually appealing interface.
   ○ Easy navigation between recipes and blog posts.
   ○ Engaging features like ratings, comments, and sharing options.

Phase-4: Project Planning (Agile Methodologies)

Objective:

To break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	API Integration & Setup	🔴 High	4 hours (Day 1)	Mid - Day 1	Member 1	Google API Key, Python, Streamlit setup API	API successfully fetching recipes
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	Mid-Day 1	Member 2	response format finalized	Basic UI with search input
Sprint 2	Search Functionality & Display	🔴 High	3 hours (Day 1)	End of day 1	Member 1& 2	API response, UI elements ready	Display recipes based on inputs
Sprint 2	Error Handling & Debugging	🔴 High	2 hours (Day 1)	Mid-Day1.5	Member 1&4	API logs, UI inputs	Stable API responses
Sprint 3	Testing & UI Enhancements	🟡 Medium	2 hours (Day 2)	Mid-Day1.5	Member 2&3	API response, UI layout completed	Smooth user experience across devices
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Fully working app deployed online

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(🔴 **High Priority)** Set up the **environment** & install dependencies.
(🔴 **High Priority)** Integrate **MealDB API** for fetching recipes.
(🟡 **Medium Priority)** Build a **basic UI with search input fields.**

Sprint 2 – Core Features & Debugging (Day 1)

(🔴 **High Priority)** Implement **search functionality** to fetch recipes based on ingredients.
(🔴 **High Priority)** Debug API issues & handle **errors in queries**.

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

(🟡 **Medium Priority)** Test API responses, refine UI for better usability.
(🟢 **Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

To implement core features of the SmartChef app.

Key Points:

1. **Technology Stack Used:**
   - ○ **Frontend:** HTML, CSS, Java Script
   - ○ **Database:** The MealDB API
   - ○ **Programming Language:** Python
2. **Development Process:**
   - ○ Implement API authentication and integrate MealDB API for recipe data generation.
   - ○ Develop logic for ingredient-based, dietary, and seasonal recipe recommendations.
   - ○ Optimize search queries to improve response time and result accuracy.
3. **Challenges & Fixes:**
   - ○ **Challenge:** Slow API response.
     Fix: Implement **caching** to store frequently requested recipes.
   - ○ **Challenge:** API rate limits.
     Fix: Optimize queries to reduce redundant API calls.

Phase-6: Functional & Performance Testing

Objective:

To ensure that the SmartChef App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Tomato"	Display recipes using tomato.	✅ Passed	Tester 1
TC-002	Functional Testing	Query "Chicken"	Show chicken-based recipes.	✅ Passed	Tester 2

TC-003	Performance Testing	API response time under 500ms	Recipes should be displayed quickly.	⚠️ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Recipes should match user input accurately.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work smoothly 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. **GitHub/Code Repository Link**
3. **Presentation**