Hackathon Project Phases Template for the MULTI-LANGUAGE TRANSLATOR project.

Hackathon Project Phases Template

Project Title:

AI-powered multi-language translator

Team Name:

TEAM “AI-MERS”

Team Members:

PRAMOD MAHAJAN

PRAVEEN

ROHAN VARMA

VISHWA SAHITH

SOWMITHA

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered multi language translator using Google to help users compare and analyze about multi language translator,supporting text and voice search.

Key Points:

1. **Problem Statement**:
    - In an increasingly globlazied world,seamless communication across multiple languages is Essential

- The project aims to users with a seamless,real-time translation experience,and breaking language barriers in global communication.

2. **Proposed Solution**:

- An AI-powered application using Google to provide multiple languages translation.
- The web offers maintenance tips and better language communication based on user preferences.

3. **Target Users**:

    - Web users looking for specifications and comparisons
    - Web users need seasonal maintenance tips.

4. **Expected Outcome**:

- A functional AI-powered multi language translator that provides insights based on real-time data and user queries.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the multi language translator web.

Key Points:

1. **Technical Requirements**:

    - Programming Language: Python
    - Backend: Google cloud translator
    - Frontend: Streamlit
    - Database: Not required initially

2. **Functional Requirements**:

    - Ability to language translator using Google.
    - Provide real-time language tips based on types of languages .

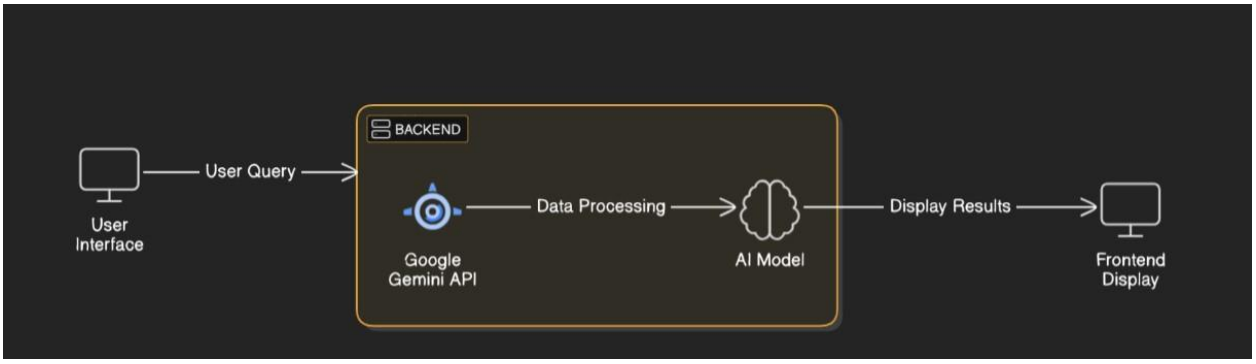
2. **Constraints** & **Challenges**:

    - Ensuring real-time updates from Gemini AI and ChatGPT.
    - Handling API rate limits and optimizing API calls.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture**:
    - User enters language related query via UI.
    - Query is processed using Google Gemini API.
    - AI model fetches and processes the data.
2. **User Flow:**
    - Step 1: User enters a query (e.g., " ‘hello’ translate into japanese").
    - Step 2: The backend **calls the Gemini AI** to retrieve language data.
    - Step 3: The web processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
    - **UI interface handles user input and output,** for seamless navigation.
    - **Filters,types of languages selection**
    - **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	PRAMOD	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	ROHAN	API response format finalized	Basic UI with input fields
Sprint 2	Gathering data	High	3 hours (Day 2)	Mid-Day 2	PRAVEEN	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	ROHAN	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	VISHWA SAHITH	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	Low	1 hour (Day 2)	End of Day 2	SOWMITHA	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 multi language translator.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Google Gemini API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Google API integration**.
    - Develop **language skills and communication skills.**
    - 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 **AI-Powered multi-language translator** works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester

TC-001	Functional Testing	Query “ most translated language”	Relevant language should be displayed	✅ Passed	Praveen
TC-002	Functional Testing	Query “translate ‘hello’ in to Spanish”.	Seasonal tips should be provided.	✅ Passed	Vishwa sahith
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	❌ Needs Optimization	Pramod
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	rohan
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	sowmitha
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	✅ Deployed	praveen

Final Submission

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