Hackathon Project Phases

Project Title:

Translingua:AI-Powered Multi-Language Translator

Team Name:

AI Avengers

Team Members:

- Shaik Nazma(Team lead)
- Shaik Reshma
- Singareddy Rajya Lakshmi
- Sura Jyothika Mythreyi
- Vemula Krishna Priyanka

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered multi-language translator to break down language barriers and foster seamless global communication.

Key Points:

1. **Problem Statement:**
    - In an increasingly globalized world, effective communication across language barriers is crucial for businesses, education, travel, and social interaction. So Translingua is a cutting edge web application designed to harness the power of advanced AI to provide seamless language translation services.

2. **Proposed Solution:**
    - An AI-powered application using **Gemini Flash** to translate any text into multiple languages.
    - The app enables a person to quickly translate content and documents, ensuring consistency and accuracy across different languages.
3. **Target Users:**
    - **Bussinessmen** needing cross language collaboration.
    - **Travellers** looking to communicate with locals/
    - **Healthcare professionals** needing to translate medical documents.
4. **Expected Outcome:**
    - A functional **AI-powered multi language translator information app** that provide fast translations of text between multiple languages.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Translingua App.

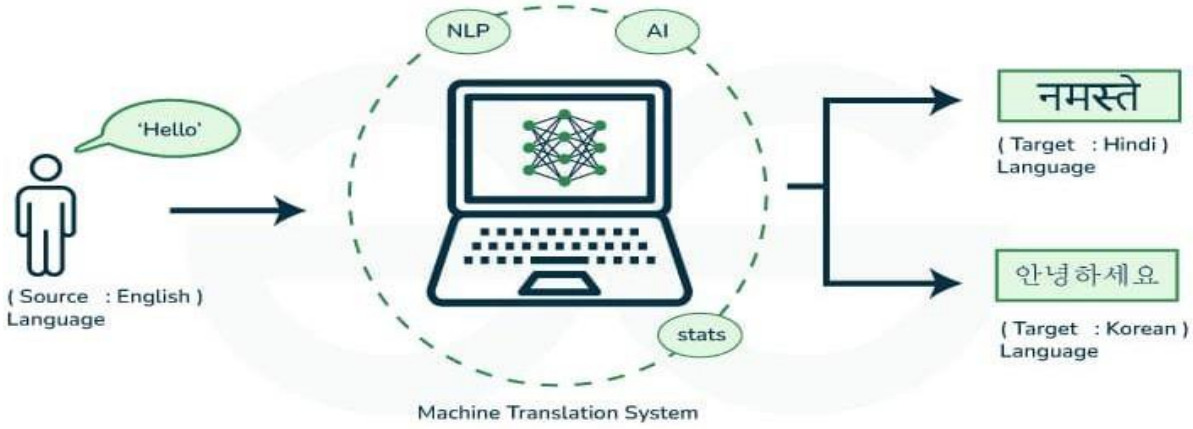
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: **Google Gemini Flash API**
    - Frontend: **Streamlit Web Framework**
    - Database: **Required initially (API-based queries)**
2. **Functional Requirements:**
    - Ability to **translate text into many languages** using Gemini Flash API.
    - Display **specifications, reviews, and comparisons** in an intuitive UI.
    - Access to **diverse and representative** datasets.
    - Allow users to **continuously engage with native speakers** to ensure cultural and contextual relevance.
3. **Constraints & Challenges:**
    - Ensuring real-time updates from **Gemini API**.
    - Handling **large language models** for translation.
    - 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 text related query via UI.
    - Query is processed using **Google Gemini API**.
    - AI model fetches, processes and translates the data.
    - The frontend displays **translated text.**
2. **User Flow:**
    - Step 1: User enters a query (e.g., "Translate Hello to hindi language").
    - Step 2: The backend **calls the Gemini Flash API** to translate hello to hindi.
    - Step 3: The app translates the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
    - **Flexible layouts** to accommodate varying text lengths.
    - **Support for right to left languages**.
    - **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	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 4	API response format finalized	Basic UI with input fields
Sprint 2	Language Search & Translation	□ High	1 hour (Day 2)	Mid-Day 2	Member 2&3	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	□ High	1.5 hours (Day 2)	Mid-Day 2	Member 1&5	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	□ Medium	1.5 hours (Day 2)	Mid-Day 2	Member 3&4	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 **language search and translation. (□**

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 Translingua 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 **Gemini API integration**.
    - Collection of **data,preparation,model selection,training.**
    - Optimize **search queries for performance and relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** Delayed API response times.
      Fix: Implement **caching** to store frequently queried results.
    - **Challenge:** Struggling to translate expressions that carry cultural meaning
      Fix: Enriching training **data**.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Translate bomb to spanish	bomba	✓ Passed	Tester 1
TC-002	Functional Testing	Translate bomb to dutch	bom	✓ 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 API responses.	Data accuracy should be improved.	✅ 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

---

Final Submission

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