

# Hackathon Project Phases Template

## Project Title:

TRANSLINGUA Using Streamlit

## Team Name:

MARVELS

## Team Members:

- N.Samskruthi
  - K.Rishika
  - K.Hethoshna
- 

## Phase-1: Brainstorming & Ideation

### Objective:

The project aims to create an interactive language learning platform that combines multilingual conversations, translation tools, educational games, and social features. Users can track progress, take and share notes, and engage in a community of learners. The platform will include various monetization strategies like subscriptions and in-app purchases. The goal is to enhance language learning with a fun, personalized, and social experience.

### Key Points:

#### 1. Problem Statement:

The problem is that traditional language learning methods often lack interactivity, personalization, and social engagement, making it difficult for users to stay motivated and effectively practice new languages. Existing platforms may not fully integrate real-

time communication, translation tools, interactive games, and social features to create a holistic learning experience.

There is a need for a comprehensive, user-friendly solution that combines these elements to support diverse language learners in an engaging, collaborative, and motivating environment.

## 2. **Proposed Solution:**

The proposed solution is to develop an all-in-one language learning platform that integrates multilingual conversations, real-time translation, interactive games, and social features. This platform will allow users to practice languages through guided conversations, play educational games, take and share notes, and interact with a community of learners. By personalizing the learning experience and fostering social engagement, the platform aims to keep users motivated and provide a holistic, effective way to learn languages.

## 3. **Target Users:**

- **Students** who need to learn a new language for academic purposes or personal enrichment.
- **Professionals** seeking to learn a new language for career advancement or travel.
- **Language enthusiasts** who enjoy learning multiple languages and want to connect with others in a community setting.
- **Teachers and tutors** looking for tools to assist in language instruction and create a more interactive learning experience.
- **Social media-savvy users** who enjoy engaging with a community for language practice and sharing progress.

## 4. **Expected Outcome:**

- The expected outcome of the project is a significant improvement in users' language skills through interactive and engaging features. Users will stay motivated and consistent with their learning, leading to higher retention rates. The platform will also foster a thriving community, encouraging social interaction and collaboration among language learners.
-

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the Translingua Language Learning App.

## Key Points:

### 1. Technical Requirements:

- Programming Language: **Python**
- Backend: **Google Gemini Flash API**
- Frontend: **Streamlit Web Framework**
- Database: **Not required initially (API-based queries)**

### 2. Functional Requirements:

- **Multilingual Conversations:** Enable real-time multilingual communication using NLP models.
- **Translation Feature:** Integrate machine translation tools for users to translate between selected languages.
- **Interactive Language Learning Games:** Offer a variety of quizzes, puzzles, and interactive stories for language practice.
- **Note-Taking and Sharing:** Allow users to take, organize, and share language-learning notes.
- **Social Community Features:** Create a community space for users to share progress, ask questions, and connect with others.
- **Personalization:** Users can set language goals, select learning preferences, and receive tailored recommendations.

### 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 the language which needs to be translated query via UI.
- Query is processed using **Google Gemini API**.
- AI model fetches and processes the data.
- The frontend displays **translated text in preferred language**

### 2. User Flow:

- Step 1: User enters a query (e.g. My name is Sada)
- Step 2: The backend **calls the Gemini Flash API** to retrieve the translated language data.
- Step 3: The app processes the data and **displays results** in an easy-to-read format.

### 3. UI/UX Considerations:

- Clean, minimalist, and intuitive interface.
  - Dark and light modes for enhanced readability and user experience.
  - Dynamic content and language support based on user prefer
-

## 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	<input type="checkbox"/> High	6 hours (Day 1)	End of Day 1	Member 3	Google Translate API key, Python setup	API connection established & working
Sprint 1	Frontend UI Development	<input type="checkbox"/> Medium	3 hours (Day 1)	End of Day 1	Member 1	API response format finalized	Basic UI with input fields
Sprint 2	Multilingual Conversation Feature	<input type="checkbox"/> High	3 hours (Day 2)	Mid-Day 2	Member 1	API response, UI elements ready	Real-time conversation feature working
Sprint 2	Error Handling & Debugging	<input type="checkbox"/> High	1.5 hours (Day 2)	Mid-Day 2	Member 3	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	<input type="checkbox"/> Medium	1.5 hours (Day 2)	Mid-Day 2	Member 2	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	<input type="checkbox"/> 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 AutoSage App.

### Key Points:

#### 1. Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** Google Gemini Flash API
- **Programming Language:** Python

#### 2. Development Process:

- ☐ **Implement API Key Authentication & Translation API Integration:** Set up secure API authentication for language translation and real-time updates.
  - **Develop Multilingual Conversation Logic:** Enable real-time communication in different languages using NLP models.
  - **Create Language Learning Games:** Design and integrate interactive games that help users practice language skills in a fun and engaging way.
  - **Implement Note-Taking & Sharing Features:** Allow users to take notes, organize them, and share them with others.
  - **Optimize UI for Performance & Usability:** Ensure the user interface is smooth, responsive, and intuitive.

#### 3 Challenges & Fixes:

**Challenge:** Slow API response times for translation and NLP processing.

**Fix:** Implement caching for frequently accessed translation data and pre-process common queries to reduce latency.

**Challenge:** Handling large data sets for game progress, notes, and social interactions.

**Fix:** Use efficient database queries and optimize data storage to reduce load times and ensure quick retrieval.

---

## 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	Query "Translate 'hello' from English to Spanish"	Correct translation should be displayed.	<input type="checkbox"/> Passed	Tester 1
TC-002	Functional Testing	Test "Start new game with basic vocabulary"	Game should launch with vocabulary challenges.	<input type="checkbox"/> Passed	Tester 2
TC-003	Performance Testing	Check app loading time under 3 seconds	App should load in less than 3 seconds.	<input type="checkbox"/> Passed	Tester 3
TC-004	Bug Fixes & Improvements	Fixed issue where game progress wasn't saved	Progress should be saved correctly after each game.	<input type="checkbox"/> Fixed	Developer
TC-005	Final Validation	Ensure UI responsiveness on both desktop and mobile	UI should be functional across both devices.	<input type="checkbox"/> Fixed	Tester 2
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	<input type="checkbox"/> Deployed	DevOps

---

# **Final Submission**

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**
3. **Presentation**