

---

# Hackathon Project Phases Template

**Project Title:**

TransLingua-Infinity

**Team Name:**

AI-Architects

**Team Members:**

- Bhavya sri Gaddam
  - Anvitha Gadari
  - B Meghana
  - Sravani Sunchu
- 

## Phase-1: Brainstorming & Ideation

**Objective:**

To develop an AI-powered language translator tool using streamlit and google translator API to translate from text-to-text, text-to-voice, voice-to-text, voice-to-voice

**Key Points:****1. Problem Statement:**

- Language barriers create significant communication challenges in a globalized world, affecting personal interactions, education and business.
- Users need a guided environment in order to translate their native language into professionally/required language.

## 2. Proposed Solution:

- o An AI-powered application using **Google trans** to provide **real-time vehicle specifications, reviews, and comparisons.**
- o The app offers **maintenance tips** and **eco-friendly vehicle insights** based on user preferences.

## 3. Target Users:

- o **Non-regional people** to communicate with other..
- o **Business** growth and development in a new region.
- o **Educational purposes.**

## 4. Expected Outcome:

- o A functional **AI-powered translator application** that provides user-friendly translation of data.
- 

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the TransLingua - Infinity.

## Key Points:

### 1. Technical Requirements:

- o Programming Language: **Python**
- o Backend: **Google Trans-Translation, gttS, speech recognition, pydub, os module**
- o Frontend: **Streamlit Web Framework**
- o Database: **Not required initially (API-based queries)**

### 2. Functional Requirements:

- o Allow users to input text and translate it between selected language pairs.
- o Enable users to write, save, and view audio files.
- o Allow users to choose from predefined language pairs for translation.
- o Offer a user-friendly and responsive interface to interact with the features.

### 3. Constraints & Challenges:

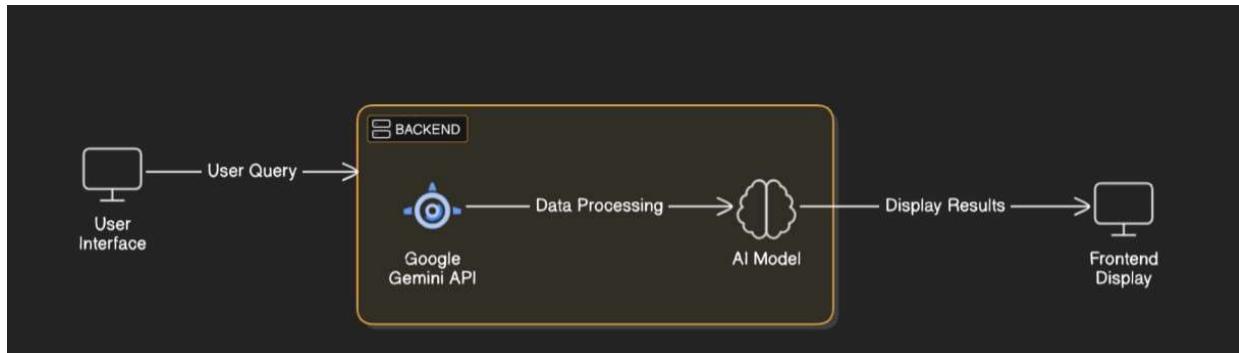
- o Translation models may be slow or resource-heavy on devices with limited processing power.
- o Limited flexibility due to predefined language pairs.
- o Proper validation of user input for accurate translations.

---

## Phase-3: Project Design

### Objective:

Develop the architecture and user flow of the application.



### Key Points:

#### 1. System Architecture:

- o User selects a feature from the Streamlit sidebar and provides input.
- o For translation, the input text is processed using the TransLingua model.
- o AI model fetches and processes the data.
- o The frontend displays **translated text and audio**.

#### 2. User Flow:

- o Step 1: User enters a text to translate.
- o Step 2: The backend **calls the Google trans API** to translate input data.
- o Step 3: The application processes the data and **displays results**.

#### 3. UI/UX Considerations:

- o **Minimalist, user-friendly interface** for seamless navigation.
  - o **Users can select the required language to translate**.
-

# 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	<span style="color:red;">●</span> High	6 hours (Day 1)	End of Day 1	Member 1 & 2	Google trans, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	<span style="color:yellow;">●</span> Medium	2 hours (Day 1)	End of Day 1	Member 3&4	API response format finalized	Basic UI with input fields
Sprint 2	API integration	<span style="color:red;">●</span> High	2 hours (Day 2)	Mid-Day 2	Entire Team	API response, UI elements ready	Translation and audio download
Sprint 2	Error Handling & Debugging	<span style="color:red;">●</span> High	1 hours (Day 2)	Mid-Day 2	Member 1&4	UI inputs	Improved

## Sprint Planning with Priorities

### Sprint 1 – Setup & Integration (Day 1)

- (● High Priority) Set up the environment & install dependencies.
- (● High Priority) Integrate Google trans.
- (● Medium Priority) Build a basic UI with input fields.

### Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement translation & Audio Download.

---

## Phase-5: Project Development

### Objective:

Implement core features of the TransLingua-Infinity.

### Key Points:

#### 1. Technology Stack Used:

- o **Frontend:** Streamlit
- o **Backend:** Google trans
- o **Programming Language:** Python

#### 2. Development Process:

- o Implement **Google trans**.
  - o Develop **user Interface**.
  - o Adding features such as text-to-text, text-to-voice, voice-to-text, voice-to-voice, document summarization.
- 

## Phase-6: Functional & Performance Testing

### Objective:

Ensure that the Translingua-infinity works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Input "How are you doing"	Translated text according to relevant language.	<input checked="" type="checkbox"/> Passed	Tester 1
TC-002	Functional Testing	Speech input	Translated Audio is generated	<input checked="" type="checkbox"/> Passed	Tester 2
TC-003	Performance Testing	Speech to Speech translation	Speech translation	<span style="color: orange;">⚠</span> 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**