Hackathon Project Phases Template

Project Title:

TransLingua: AI-Based Multi-Language Translator

Team Name:

The Polyglot Pirates

Team Members:

- A. Sai Eshwari
- B. Jeevana
- T. Lakshmi

Phase-1: Brainstorming & Ideation

Objective: Develop an AI-based multi-language translation application that provides seamless real-time translations across various languages with high accuracy, leveraging Generative AI (GenAI).

Key Points:

1. Problem Statement:
   - Existing translation tools often lack contextual accuracy and fluency.
   - Many languages, especially regional ones, have poor AI-based translation support.
   - Real-time, natural, and AI-enhanced translation is needed for various users.

2. Proposed Solution:
   - AI-powered application for text, voice, and document translation.
   - Uses NLP and deep learning models to improve accuracy.
   - Provides real-time, voice-assisted translation capabilities.

3. Target Users:
   - Students, researchers, businesses, and travelers.

4. Expected Outcome:
   - A functional AI-driven multi-language translation tool ensuring high accuracy.

Phase-2: Requirement Analysis

Objective: Define the technical and functional requirements for TransLingua.

Key Points:

1. Technical Requirements:
   - Programming Language: Python, JavaScript
   - Frontend: React.js (Web), Flutter (Mobile)

- Backend: FastAPI / Flask (Python)
- AI Models: OpenAI GPT, Google T5, MarianMT
- Database: Firebase / PostgreSQL
- APIs: OpenAI, Google Translate API, DeepL API

2. Functional Requirements:
- Text and speech translation for multiple languages.
- Context-aware and real-time translation.
- UI/UX-friendly multilingual chatbot.

3. Constraints & Challenges:
- Achieving near-human translation accuracy.
- Reducing latency for real-time translation.
- Handling regional and low-resource languages.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of TransLingua.

Key Points:

1. System Architecture:
- User inputs text or speech.
- Backend processes input using NLP models.
- AI model translates input into the target language.
- Translated text/audio is displayed or played back.

2. User Flow:
- Step 1: User selects input and target language.
- Step 2: Inputs text or voice for translation.
- Step 3: AI model processes and translates the content.
- Step 4: Output is displayed or spoken in the selected language.

3.UI/UX Considerations:

- Minimalist, user-friendly interface for seamless navigation.
- Filters for language selection and translation options.
- Dark & light mode for better user experience.

Phase-4: Project Planning (Agile Methodologies)

Sprint	Task	Priority	Duration	Deadline	Assigned To	Expected Outcome
Sprint 1	Environment Setup & API Integration	High	6 hours	Day 1	Sai Eshwari	API integration ready
Sprint 1	Frontend UI Development	Medium	3 hours	Day 1	Jeevana	UI prototype ready
Sprint 2	Translation Logic Implementation	High	4 hours	Day 2	Lakshmi	Translation working
Sprint 2	Error Handling & Debugging	High	2 hours	Day 2	Sai Eshwari	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	2 hours	Day 2	Jeevana	Responsive UI
Sprint 3	Final Presentation & Deployment	Low	1 hour	Day 2	Entire Team	Ready for demo

Phase-5: Project Development

Objective: Implement core features of the TransLingua application.

Key Points:

Technology Stack Used:

- **Frontend:** React.js (Web), Flutter (Mobile)
- **Backend:** FastAPI / Flask (Python)
- **Programming Language:** Python, JavaScript

Development Process:

- Implement API key authentication and integration.

- Develop translation logic using AI models.
- Optimize query handling for better performance.

Challenges & Fixes:

- **Challenge:** Delayed API response times.
  - **Fix:** Implement caching to store frequently queried results.
- **Challenge:** Limited API calls per minute.
  - **Fix:** Optimise quries to fetch only neccesary data.

Phase-6: Functional & Performance Testing

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Translate 'Hello' to Spanish	Output: 'Hola'	Passed	Sai Eshwari
TC-002	Performance Testing	API response time under 500ms	Fast translation	Needs Optimization	Lakshmi
TC-003	Bug Fixes	Fix incorrect sentence structures	Fluent translations	Fixed	Developer
TC-004	UI Testing	Ensure responsiveness on mobile	UI adjusts properly	Failed	Tester 2
TC-005	Deployment Testing	Deploy on VS Code environment	Successfully runs locally	Deployed	DevOps

Final Submission :

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