**Hackathon Project Phases Template** for the **AutoSage App** project.

# **Hackathon Project Phases Template**

## **Project Title:**

**AI Powered Multi Language Translator**

## **Team Name:**

Code Crashers

## **Team Members:**

* Araveti Asha Bharathi
* Challa Vyshnavi

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

Develop an **AI-powered multi-language translator** using **Hugging Face MarianMT** to provide **fast, accurate, and context-aware translations** for text and speech across multiple languages.

**1.Problem Statement:**

* Many users struggle to find accurate and real-time translations that preserve meaning and context.
* Traditional translation tools often fail to capture nuances and cultural differences.
* Users require an easy-to-use speech and text translation system for seamless communication.

**2.Proposed Solution:**

* An AI-powered translation application using Hugging Face MarianMT for precise and context-aware translations.
* Supports text and speech translation, allowing real-time multilingual communication.
* Provides offline translation capabilities by leveraging pre-trained models.

**3.Target Users:**  Travelers & Tourists who need instant translations on the go

* Students & Language Learners for better understanding and learning of new languages.
* Businesses & Professionals communicating across different languages.

**Expected Outcome:**

* + A **functional AI-powered translation app** that offers **fast, reliable, and accurate translations**, making global communication effortless.

## **Phase-2: Requirement Analysis**

### **Objective:**

* Define the **technical and functional requirements** for the **TransLingua AI-Powered Multi-Language Translator**.

### **Key Points:**

* **Technical Requirements:**
  + Programming Language: **Python**
  + Backend: **Hugging Face MarianMT Models** (for AI-powered translation)
  + Frontend: **Streamlit Web Framework**(for interactive UI)
  + Database: **Not required initially (API-based queries)**

**Functionalities:**

* + Ability to translate text using Hugging Face MarianMT models.
  + Display translated text in a user-friendly UI.
  + Provide speech-to-text and text-to-speech functionality for seamless communication.
  + Support multiple language translations with accurate and context-aware outputs.
  + Enable users to select source and target languages before translation.
* **Constraints & Challenges:**
  + Limited API Calls: Free-tier models may have restrictions on the number of requests.
  + Processing Time: Real-time translation may experience slight delays, especially for longer texts.
  + Accuracy: Machine translations may not always be 100% accurate for complex sentences.
  + Limited Language Pairs: MarianMT supports specific language pairs, which may limit certain translations.
  + Dependency on Internet: The app requires an internet connection to fetch translations via Hugging Face models.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * User enters the text and the language of input tex and also the language of output.
   * Query is processed using **HuggingFace MarianMT.**
   * AI model fetches and processes the data.
   * The frontend displays **the translated text and also a voiced output**.
2. **User Flow:**
   * Step 1: User Inputs Text, Voice, or Document.
   * Step 2:The Backend Processes the Query using **HuggingFace MarianMT**.
   * Step 3: The app processes the data and **displays translated output.**
3. **UI/UX Considerations:**
   * **Minimalist, user-friendly interface** for seamless navigation.
   * Thematic Customization.
   * **Responsive & Adaptive Design**
   * **Enhanced Interaction Features**.

## 

## **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 | Vyshnavi | Python, Streamlit, Hugging Face MarianMT API | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Asha | API response format finalized | Basic UI with input fields |
| Sprint 2 | Text & Speech Translation Implementation | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Asha | API response, UI elements ready | Functionality for text and speech translation |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Vyshnavi | API logs, UI inputs | Improved API stability |
| Sprint 3 | UI & Performance Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Vyshnavi | 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 **HuggingFace MarianMT 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:** HuggingFace MarianMT
   * **Programming Language:** Python
2. **Development Process:**
   * Implement API key authentication and Hugging Face MarianMT integration.
   * Develop text, speech, and document translation functionalities.
   * Optimize UI/UX for a seamless and interactive experience
   * Enable real-time language detection and text-to-speech output.
   * Ensure compatibility with multiple file formats (TXT, DOCX, PDF).
   * Enhance performance by reducing response latency.
3. **Challenges & Fixes:**
   * **Challenge:**Slow translation processing for large text inputs.  
      **Fix:** Implement batch processing and optimize API calls.
   * **Challenge:**Unsupported languages for text-to-speech conversion.  
      **Fix:** Provide alternative language options with fallback to English
   * **Challenge:**Difficulty in speech recognition for certain accents.
   * **Fix:**Use a robust speech-to-text model and allow manual text input as an alternative.
   * **Challenge:**UI responsiveness issues on mobile devices.
   * **Fix:**Implement a responsive design with CSS media queries.

## **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 "Hello" from English to Spanish | Output should be "Hola" | ✅ Passed | Testerv |
| TC-002 | Functional Testing | Query "Motorcycle maintenance tips for winter" | Seasonal tips should be provided. | ✅ Passed | Tester |
| TC-003 | Tester | Translation response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Tester |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect translations for certain phrases | 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 |
| TC-006 | Deployment Testing | Deploy app on a cloud server | 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 and Presentation**