**Project Title:**

**TransLingua: AI-powered Multi-language Translator**

**Team Name:**

ALPHA ACHIEVERS

**Team Members:**

* Nikhitha
* Tejaswini
* Jyothi
* Likitha
* Sravani

## Phase-1: Brainstorming & Ideation

**Objective:**

The Multi-Language Translator is used to facilitate seamless communication across different languages by accurately translating text, speech or Multimedia content.

**Key Points:**

1. **Problem Statement:**

* Many users face Misinterpretations in meetings, emails, contract, and customer interaction can lead to lost opportunities, inefficiencies and misunderstandings.
* Existing translation tools often lack industry-specific accuracy, real time collaboration and seamless integration with business platform.

1. **Proposed Solution:**

* An AI-Powered, real time MultiLingual Translation system that provides accurate, context awake and secure translations across multiple formats.
* The solution will leverage Advanced AI, Machine Learning and Neural Machine Translation to enhance accuracy and usability.

1. **Target Users:**

* **Business** **Users** for seamless Communication, expand into International Markets and for handling MultiLingual Customer Queries.
* **Student** **Users** for learning new Languages, translating study Materials & to offer MultiLingual courses.
* **Free** **Lancers** to improve work flow efficiency and for seamless Global Collaboration.

1. **Expected Outcome:**

* 1. A functional **AI-powered,** that provides insights based on real-time Multi Language Translation and user queries.

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the Multi Language Translator using AI.

**Key Points:**

1. **Technical Requirements:**

* + Programming Language: **Python**

○ Backend: **Google** **Generative** **AI**

○ Frontend: **Streamlit Web Framework**

○ Database: **Not required initially**

1. **Functional Requirements:**

* + Ability to Translate written & spoken words into different languages.
  + Automatically detect the language being spoken or written.
  + Provide fast, real-time translation for conversations and messages.
  + Enable seamless typing and Auto translation within applications.

1. **Constraints & Challenges:**

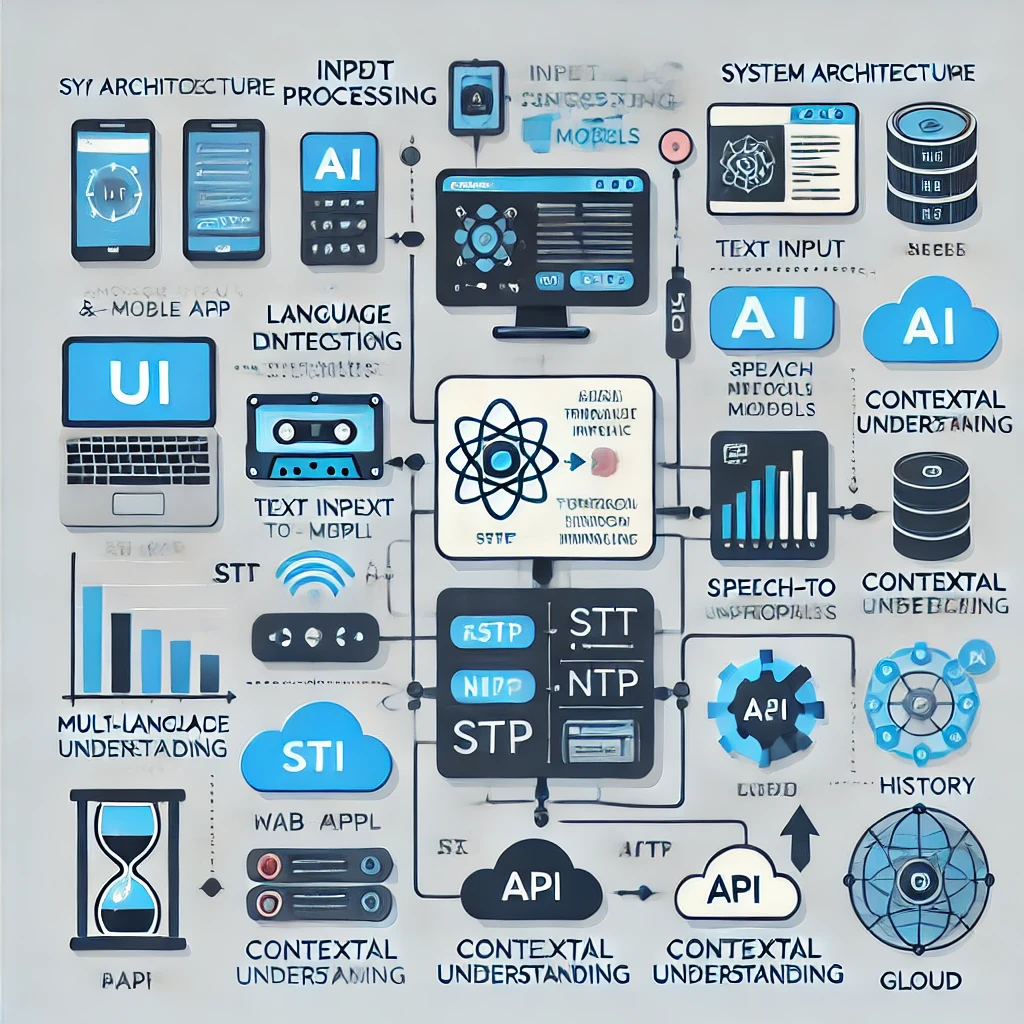
* + Ensuring real-time updates from **Generative AI**.

○ 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:** 
   1. User enters the required text in input language.

○ Query is processed using **Google Generative AI**.

○ AI model fetches and processes the data.

* The text displays in required output language.

1. **User Flow:**

* 1. Step 1: User enters a query.

○ Step 2: The backend **calls the Generative AI** to retrieve text data.

○ Step 3: The app processes the data and **displays results** in an easy-to-read format.

1. **UI/UX Considerations:**

* 1. **Minimalist, user-friendly interface** for seamless translation.
* Real-Time MultiLingual Translation.

○ **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 | 🔴 High | 6 hours  (Day 1) | End of Day  1 | Nikhitha | Google Generative AI,  Python. | API connection established & working | |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | Tejaswini | Python, sreamlit setup | Basic UI with input fields | |
| Sprint 2 | Multi Language  Translator | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Jyothi | Prompt Engineering. | Search functionality with filters | |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Sravani | Input Validation | |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Likitha | AI 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 Generative AI**.

**(**🟡 **Medium Priority)** Build a **basic UI with input fields**.

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

**(**🔴 **High Priority)** Implement **Multi Language Translator using AI.**

**(**🔴 **High Priority)** Debug AI issues & handle **errors in queries**. **Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(**🟡 **Medium Priority)** Test AI responses, refine UI, & fix UI bugs. **(**🟢 **Low Priority)** Final **demo preparation & deployment**.

## Phase-5: Project Development

**Objective:**

Implement core features of the Multi Language Translator.

**Key Points:**

1. **Technology Stack Used:**

* 1. **Frontend:** Streamlit

○ **Backend:** Google Generative AI

○ **Programming Language:** Python

1. **Development Process:**

* Using AI Multi Language Translator is Developed.
* Integration of Multi Lingual Translation system.
* Promoting the share of translator on social Media Platforms.
* Monitoring the translator performance.

1. **Challenges & Fixes:**

* 1. **Challenge:** Delayed AI response times.

**Fix:** Implement **caching** to store frequently queried results.

○ **Challenge:** Limited AI calls per minute.

**Fix:** Optimize queries to fetch **only necessary data**.

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the Multi Language Translator works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Query “generate a text on AI advancements” | AI related text should be generated correctly. | ✅ Passed | Tejaswini |
| TC-002 | Functional  Testing | Query "translating a text” | A well-structured text should be created. | ✅ Passed | Nikhitha |
| TC-003 | Performance  Testing | AI response time under  5seconds” | AI should return results quickly. | ⚠ Needs Optimization | Jyothi |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect AI responses. | Data accuracy should be improved. | ✅ Fixed | Likitha |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Sravani |
| TC-006 | Deployment  Testing | Host the app using  Streamlit Sharing | App should be accessible online. | 🚀 Deployed | team |