1. INTRODUCTION

1.1 Project Overview

- TransLingua is a cutting-edge web application designed to harness the power of advanced AI to provide seamless language translation services. Built using Streamlit and Google's Generative AI, TransLingua offers an intuitive and user-friendly interface for translating text between multiple languages.
- By simply inputting text and selecting the desired source and target languages, users can instantly receive accurate translations powered by the latest AI models. This tool is ideal for anyone needing reliable and fast translations, whether for personal, educational, or professional purposes.

Scenario 1: Global Business Expansion

A company planning to expand its market reach to non-English speaking regions needs to translate business documents, marketing materials, and customer communications into multiple languages.

TransLingua enables the company to quickly translate promotional content and technical documents, ensuring consistency and accuracy across different languages. This helps the company effectively communicate with a broader audience and tailor its messaging to local markets.

Scenario 2: Academic Research and Collaboration

Researchers and academics collaborating on international projects often need to translate research papers, academic articles, and communication between team members who speak different languages.

TransLingua assists in translating complex academic texts and facilitating cross-border collaborations by providing accurate translations of scholarly documents.

Scenario 3: Travel and Tourism Assistance

Travelers visiting foreign countries often face language barriers when trying to communicate with locals or understand travel-related information. TransLingua serves as a handy tool for translating signs, menus, and other text encountered during travel.

This enhances the travel experience by making it easier for tourists to navigate unfamiliar environments and interact with local residents.

2.IDEATION PHASE

2.1 Project Flow

Date	31 January, 2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	4 Marks

- Users input the text they want to translate, select the source language, and choose the target language using the Streamlit UI.
- The input text and language selections are sent to the translation backend, which utilizes an AI-driven translation model to process the request.
- The AI model translates the text from the source language to the target language, providing an accurate and contextually relevant translation.
- The translated text is formatted and refined by the AI to ensure clarity and coherence in the target language.
- The translated text is sent back to the frontend of the Streamlit app for display to the user.
- Users can review the translated text, make additional modifications if needed, and use or save the translated content for their purposes.

To accomplish this, we have to complete all the activities listed below,

Initialize Gemini Pro LLM:

Generate Gemini Pro API

Initialize the pre-trained model

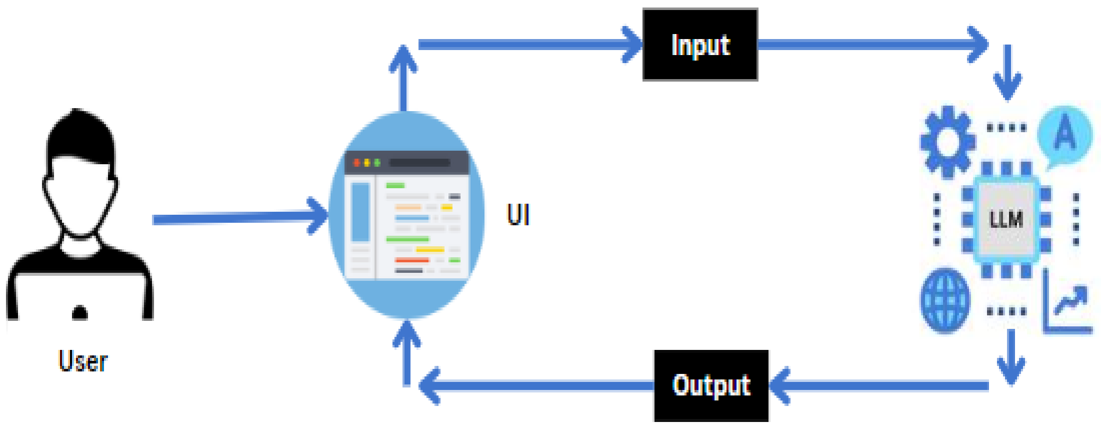
Interfacing with Pre-trained Model

Travel itinerary Generation

Model Deployment

Deploy the application using Streamlit

Architecture:



Prior Knowledge

You must have prior knowledge of the following topics to complete this project.

LLM & Gemini Pro:

A large language model is a type of artificial intelligence algorithm that applies neural network techniques with lots of parameters to process and understand human languages or text using self-supervised learning techniques.

Tasks like text generation, machine translation, summary writing, image generation from texts, machine coding, chat-bots, or Conversational AI are applications of the Large Languag.e Model. Examples of such LLM models are Chat GPT by open AI, BERT (Bidirectional Encoder Representations from Transformers) by Google, etc.

https://www.geeksforgeeks.org/large-language-model-llm/

https://cloud.google.com/vertex-ai/docs/generative-ai/learn-resources

Streamlit:

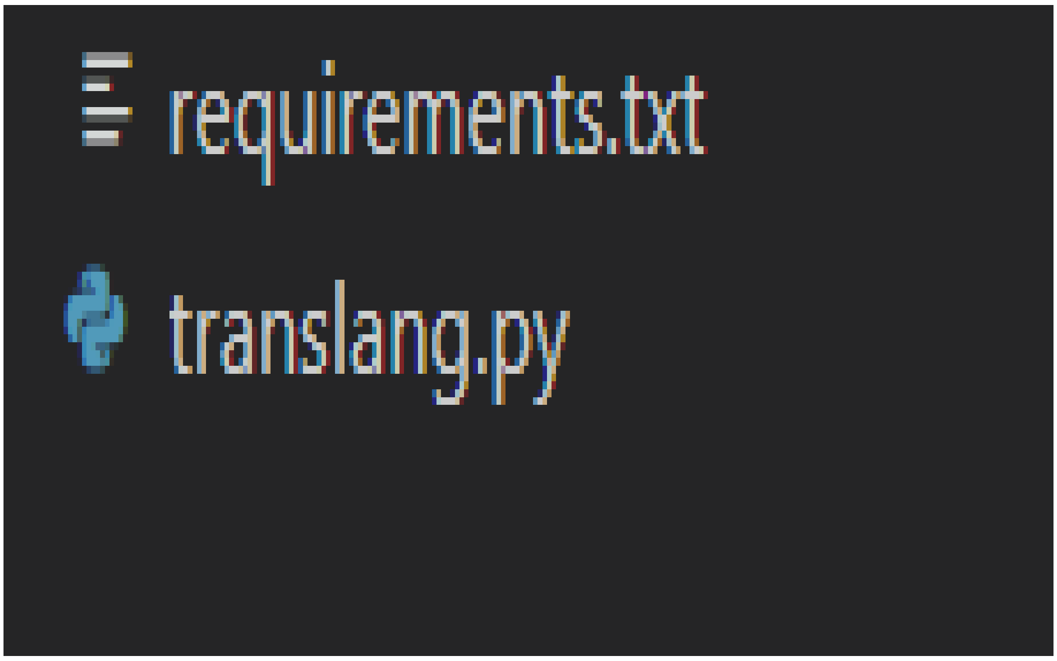
Basic knowledge of building interactive web applications using Streamlit.

Understanding of Streamlit's UI components and how to integrate them with backend logic.

https://www.datacamp.com/tutorial/streamlit

Project Structure:

Create the Project folder which contains application file as shown below



```
requirements.txt
translang.py
```

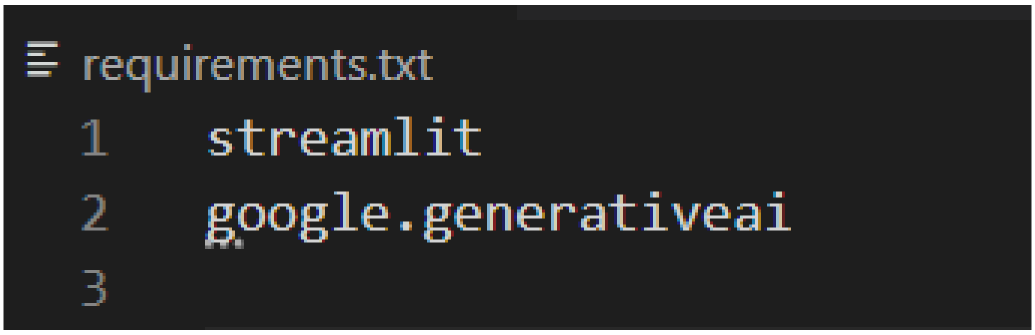
3.REQUIREMENT ANALYSIS

3.1 Requirements Specification

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.

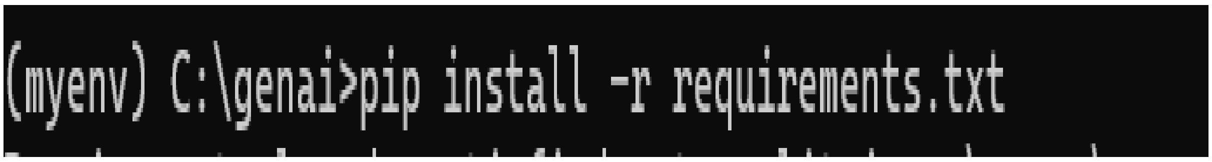
Date	5 February, 2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	4 Marks

Create a requirements.txt file to list the required libraries



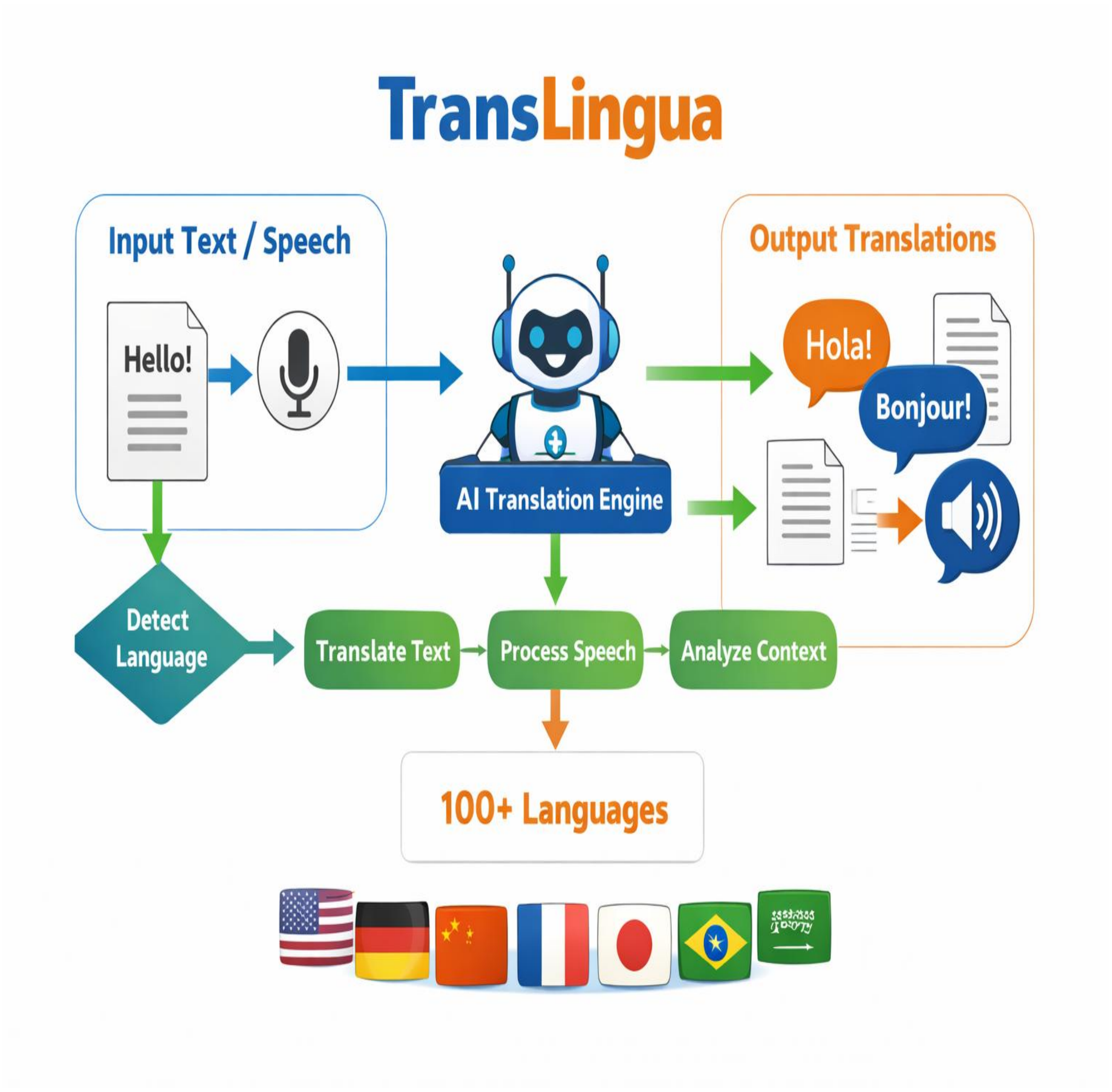
```
requirements.txt
streamlit
google.generativeai
```

Install the required libraries.



```
(myenv) C:\genai>pip install -r requirements.txt
```

Flow Diagram:



4.PROJECT DESIGN

4.1 Problem – Solution Fit Template

Date	2 February ,2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	2 Marks

Purpose:

- **Enable Global Communication**

Allow users from different language backgrounds to communicate easily and effectively.

- **Provide Context-Aware Translation**

Use AI-based neural translation models to improve accuracy and understand sentence meaning rather than word-by-word translation.

- **Support Multi-Modal Input**

Accept both text and speech as input.

- **Deliver Multi-Format Output**

Provide translated text, voice output, and downloadable files.

- **Support Multiple Languages**

Offer translation support for 100+ languages.

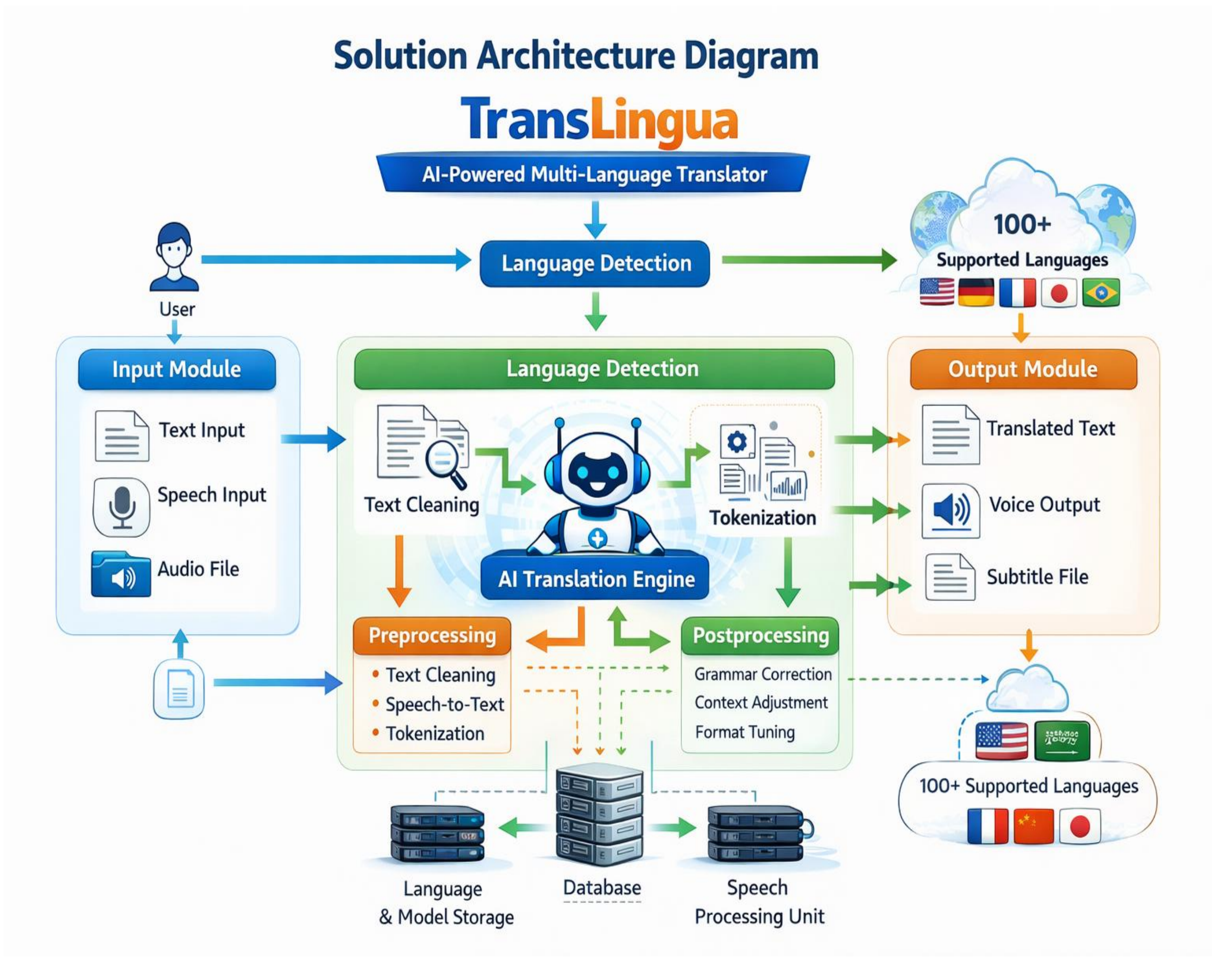
- **Improve Accessibility**

Assist travelers, students, businesses, healthcare professionals, and content creators

4.2 Solution Architecture

Date	5 February, 2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	4 Marks

Solution Architecture Diagram:



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

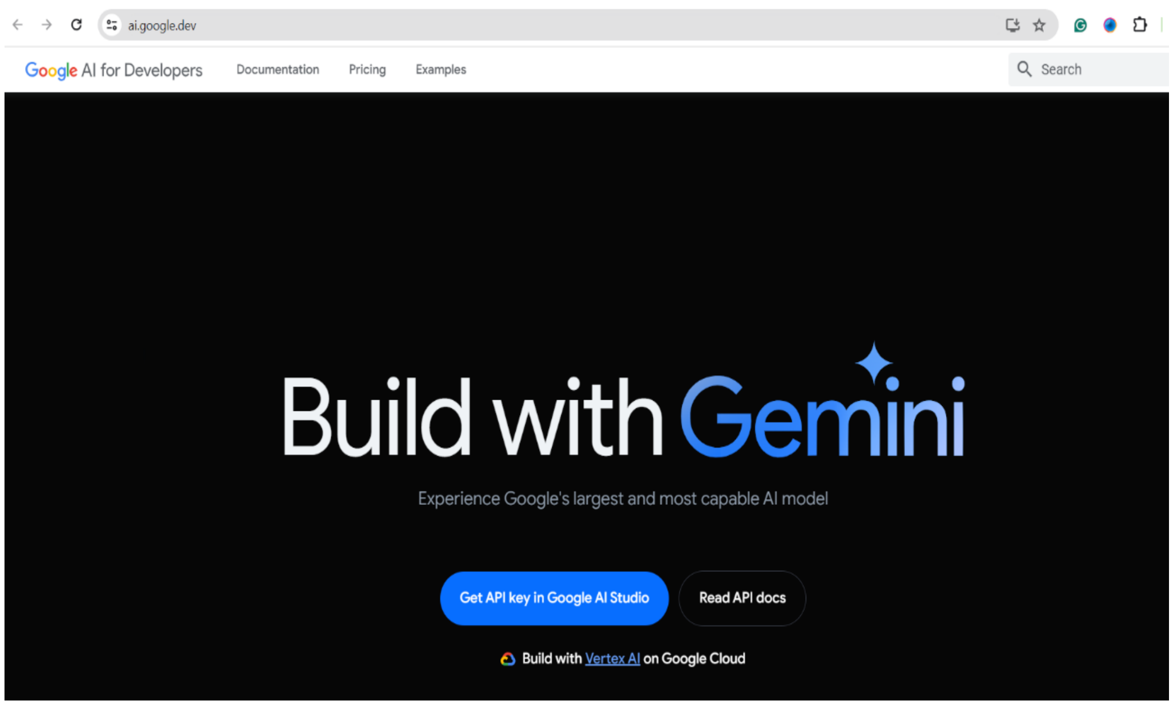
Date	7 February ,2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	5 Marks

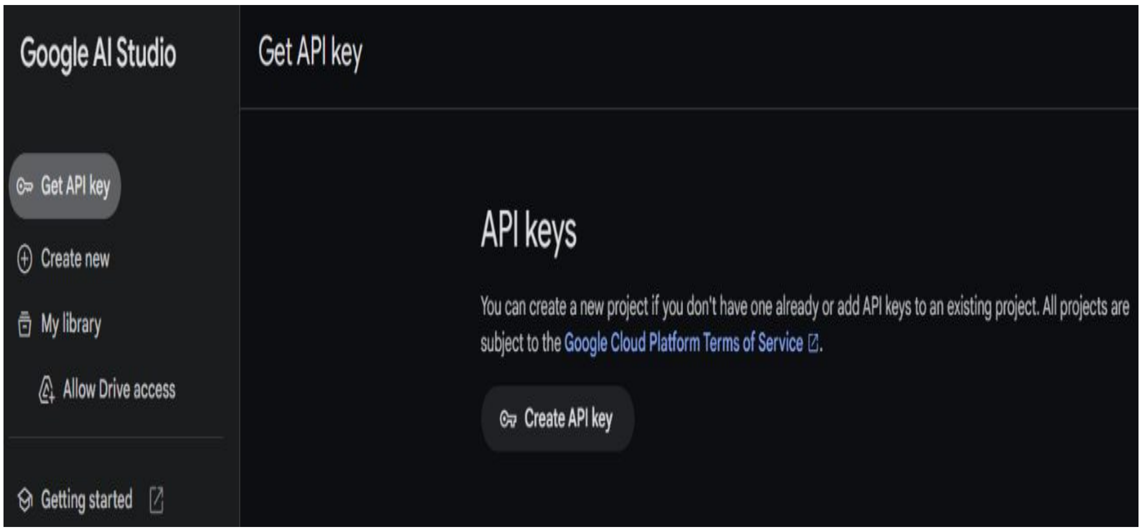
Initializing the Models

For initializing the model we need to generate PALM API.

Generate PALM API

Click on the link (https://developers.generativeai.google/).
Then click on "Get API key in Google AI Studio".
Click on "Get API key" from the right navigation menu.
Now click on "Create API key". (Refer the below images)
Copy the API key.

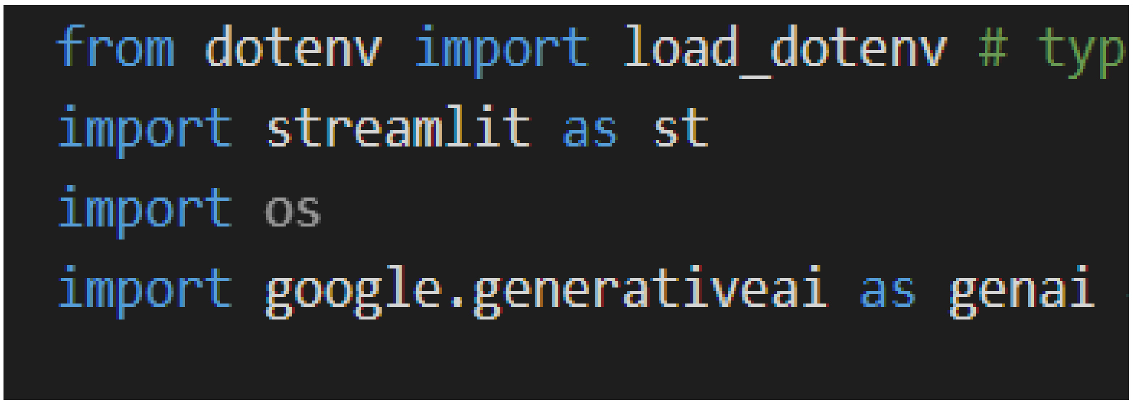

ai.google.dev
Google AI for Developers
Documentation
Pricing
Examples
Search
Build with Gemini
Experience Google's largest and most capable AI model
Get API key in Google AI Studio
Read API docs
Build with Vertex AI on Google Cloud



You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the Google Cloud Platform Terms of Service.

Initialize the pre-trained model

Import necessary files



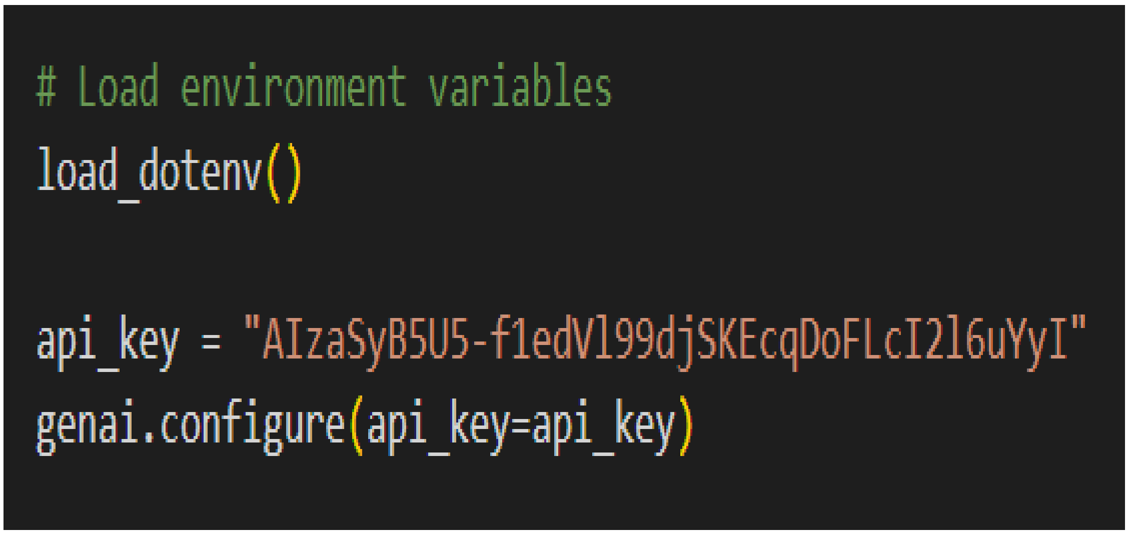
```
from dotenv import load_dotenv # typ
import streamlit as st
import os
import google.generativeai as genai
```

Streamlit, a popular Python library, is imported as st, enabling the creation of user interfaces directly within the Python script.

Google Generative AI (genai): Imported to interact with the Gemini Pro model.

The `load_dotenv` function loads key-value pairs from a `.env` file into environment variables.

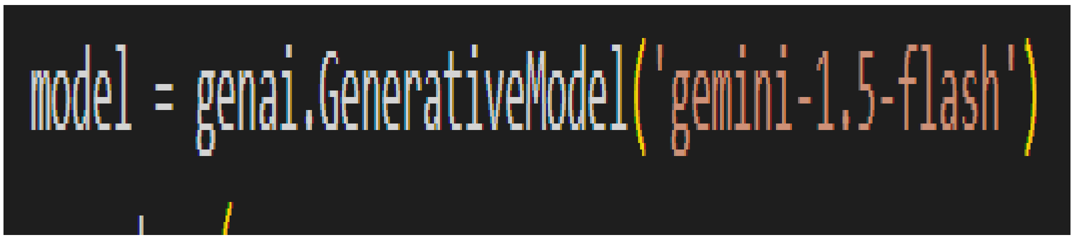
The `os` module is then used to access and manage these variables.



```python
# Load environment variables
load_dotenv()

api_key = "AIzaSyB5U5-f1edVl99djSKEcqDoFLcI2l6uYyI"
genai.configure(api_key=api_key)
```

Configuring the API key: The configure function is used to set up or configure the Google API with an API key. The provided API key, in this case, is " AIzaSyB5U5-f1edVl99djSKEcqDoFLcXXXXXX".

Define the model to be used



```python
model = genai.GenerativeModel('gemini-1.5-flash')
```

Created an instance of GenerativeModel with the model_name set to "gemini-1.5-flash".

6. FUNCTIONAL AND PERFORMANCE TESTING

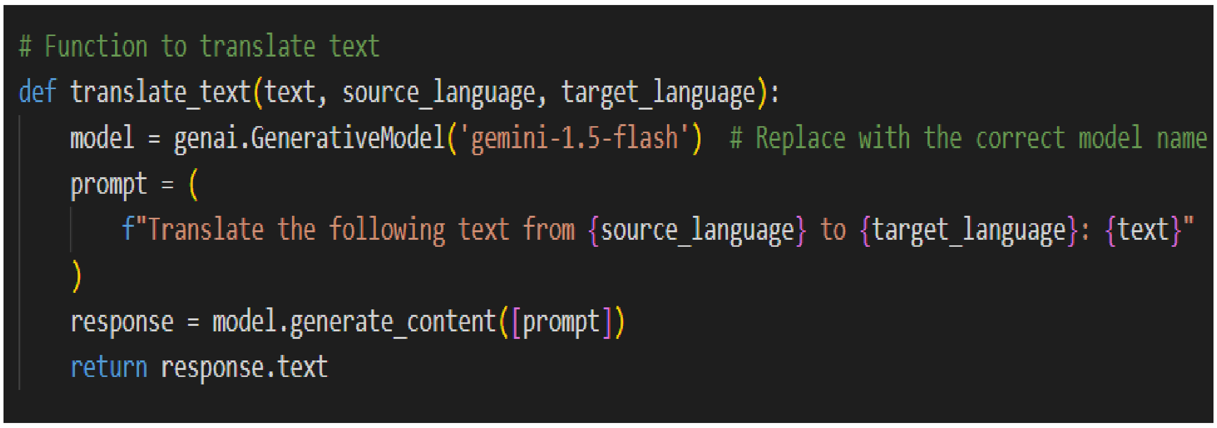
6.1 Model Performance Test

Date	9 February ,2026
Team ID	LTVIP2026TMIDS66199
Project Name	TransLingua: AI-Powered Multi-Language Translator
Maximum Marks	10 Marks

Interfacing with Pre-trained Model

In this milestone, we will build a prompt template to generate feedback based on the project details entered by the user.

Activity 1: Create a function to generate travel guide



```python
# Function to translate text
def translate_text(text, source_language, target_language):
    model = genai.GenerativeModel('gemini-1.5-flash')  # Replace with the correct model name
    prompt = (
        f"Translate the following text from {source_language} to {target_language}: {text}"
    )
    response = model.generate_content([prompt])
    return response.text
```

def translate_text(text, source_language, target_language):: Defines a function translate_text that takes three parameters: text (the text to be translated), source_language (the language of the input text), and target_language (the language to which the text should be translated).

model = genai.GenerativeModel('gemini-1.5-flash'): Initializes the generative model with the specified model name ('gemini-1.5-flash'). Replace with the appropriate model name if different.

prompt = (f"Translate the following text from {source_language} to {target_language}: {text}"): Constructs a prompt string that instructs the model to translate the provided text from the source language to the target language.

response = model.generate_content([prompt]): Sends the constructed prompt to the model to generate the translated content. The generate_content method processes the prompt and returns the model's response.

return response.text: Returns the translated text from the model's response.

7. RESULTS

7.1 Output Screenshots

In this milestone, we are deploying the created model using streamlit. Model deployment using Streamlit involves creating a user-friendly web interface, enabling users to interact with the model through a browser.

Streamlit provides easy-to-use tools for developing and deploying data-driven applications, allowing for seamless integration of models into web-based applications.

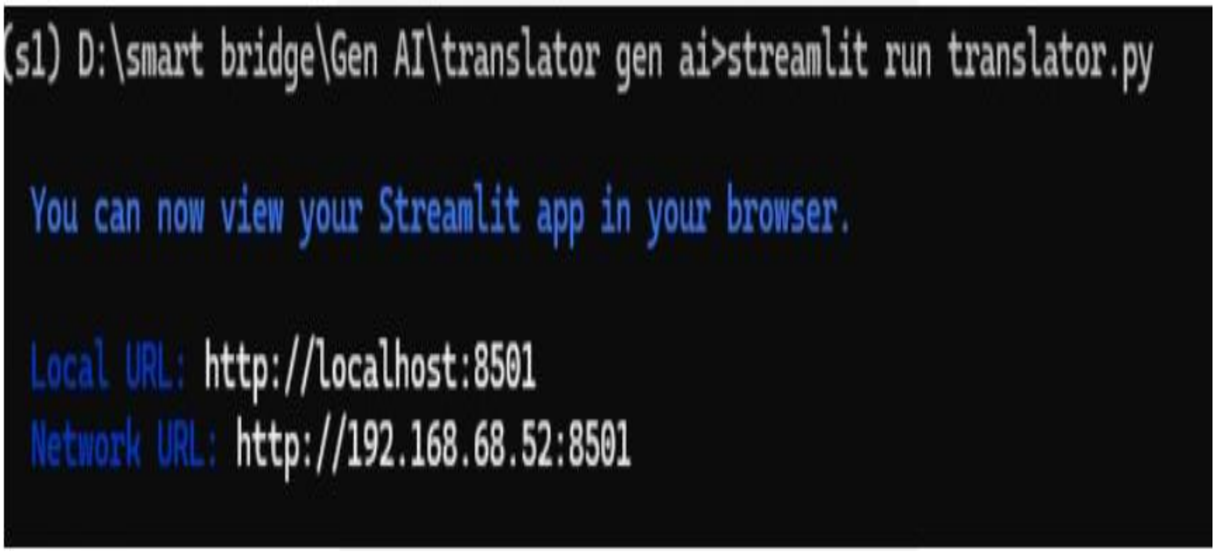
Running the web application

Open the anaconda prompt from the start menu

Navigate to the folder where your Python script is.

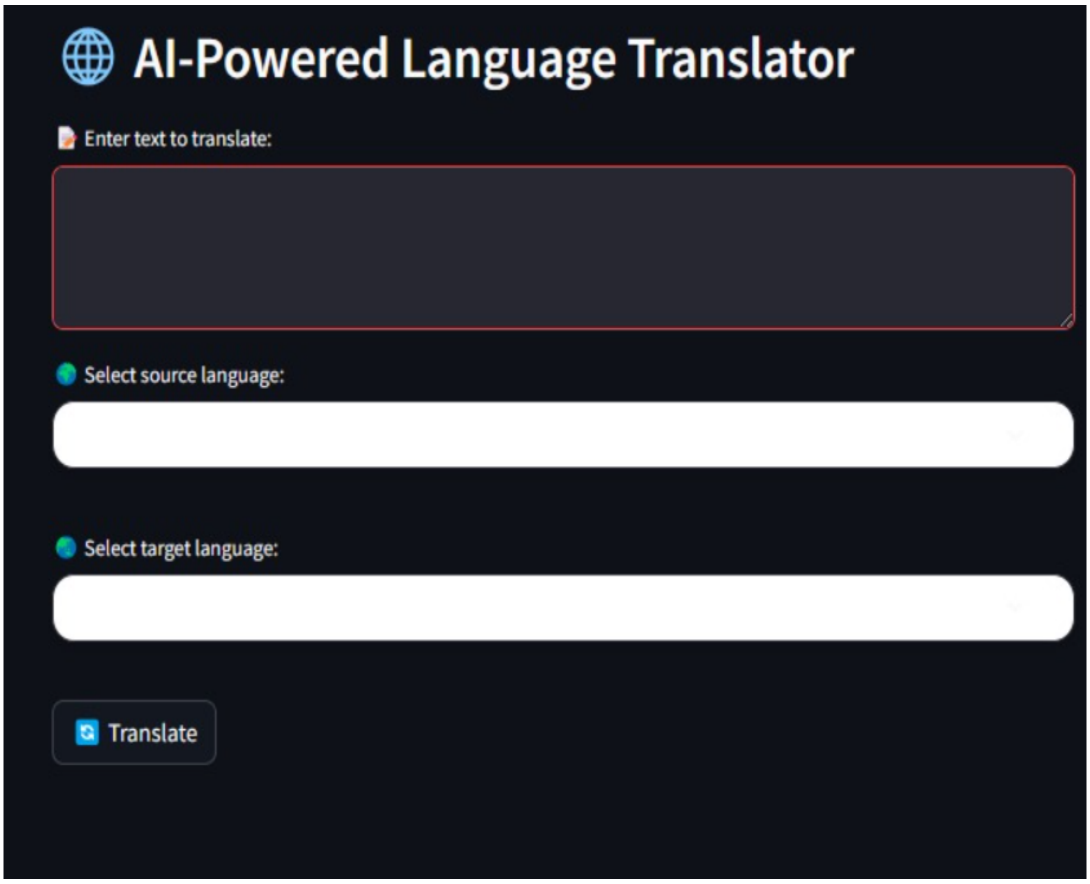
Now type "streamlit run app.py" command

Navigate to the localhost where you can view your web page

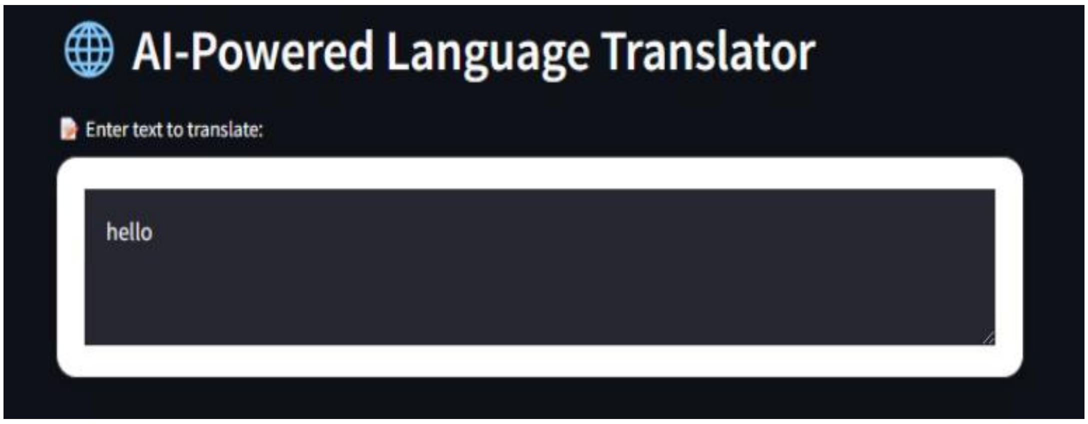


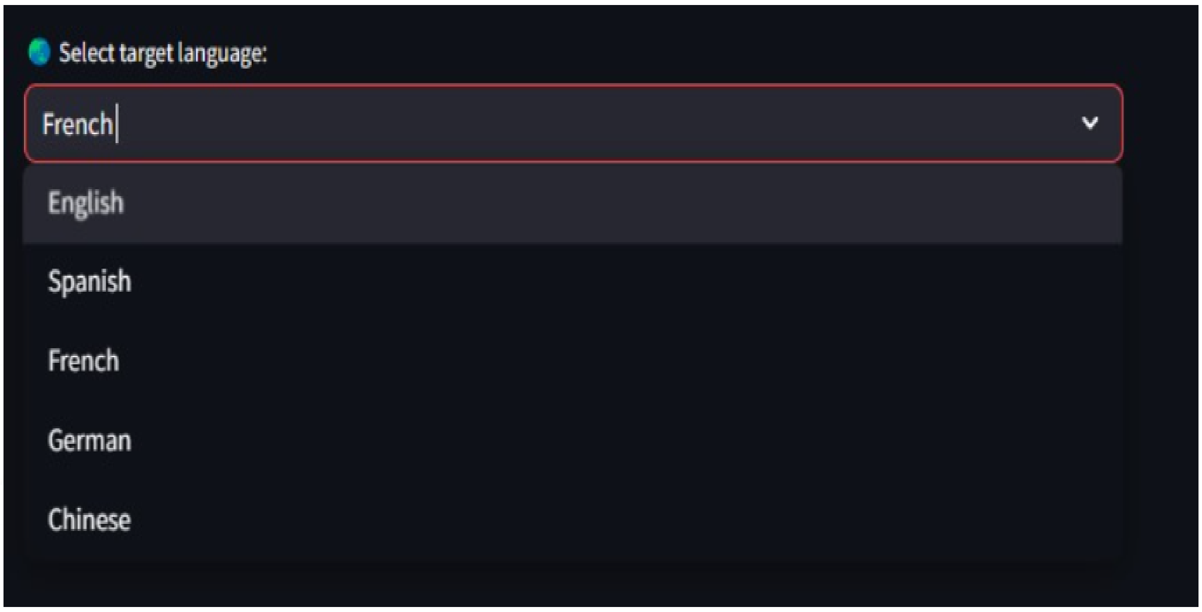
```
(s1) D:\smart bridge\Gen AI\translator gen ai>streamlit run translator.py

 You can now view your Streamlit app in your browser.

 Local URL: http://localhost:8501
 Network URL: http://192.168.68.52:8501
```

Now, the application will open in the web browser,



After giving the input:





Select target language:

French

English

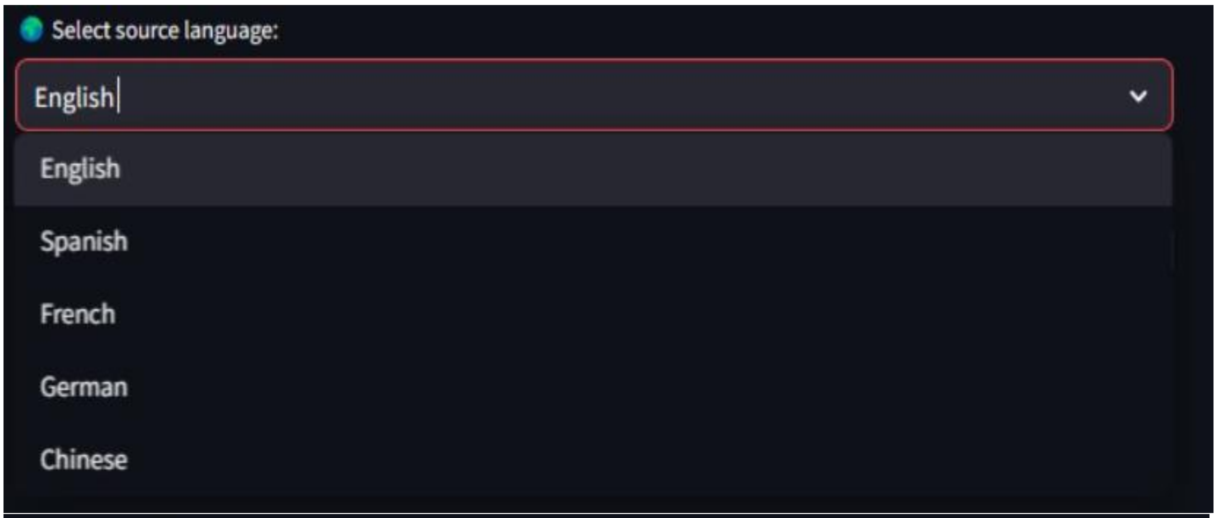
Spanish

French

German

Chinese

The Output generated:



Select source language:

English

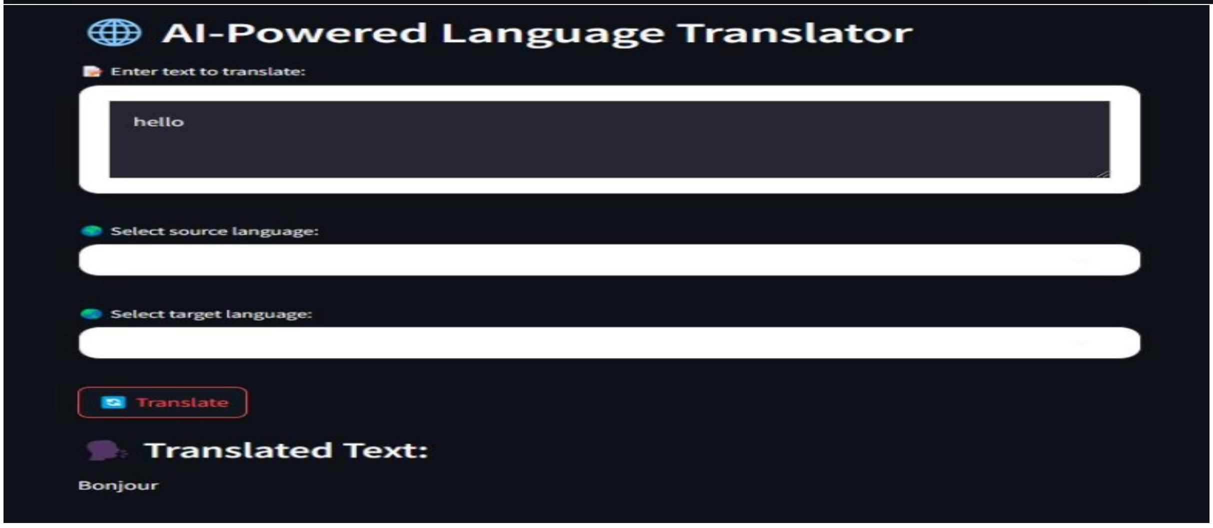
English

Spanish

French

German

Chinese



AI-Powered Language Translator

Enter text to translate:

hello

Select source language:

Select target language:

Translate

Translated Text:

Bonjour

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages:

- Breaks Language Barriers

Enables communication between people who speak different languages.

- Real-Time Translation

Provides instant translation for text and speech.

- Context-Aware Accuracy

Uses AI-based neural models for better contextual understanding.

- Multi-Modal Support

Supports:

Text-to-Text

Speech-to-Text

Text-to-Speech

Speech-to-Speech

- Supports 100+ Languages

Wide global usability.

- Improves Accessibility

Helpful for:

Travelers

Students

Businesses

Healthcare providers

- Scalable Architecture

Can be expanded to include more languages and features.

8.2 Disadvantages

- Internet Dependency

Requires stable internet connection (if cloud-based).

- Translation Errors

May produce incorrect translations for:

  Slang

  Idioms

  Cultural expressions

- High Development Cost

AI model training requires:

  Large datasets

  Powerful hardware

- Data Privacy Concerns

User data must be securely handled.

- Performance Issues

May slow down with heavy traffic or large input files.

- Accent & Noise Sensitivity

Speech recognition may fail with:

  Strong accents

  Background noise

9. CONCLUSION

The Language Translator project is a user-friendly web application that leverages advanced AI technology to facilitate seamless language translation. By integrating Streamlit with a robust translation API, users can effortlessly translate text between different languages by specifying the source and target languages.

The application provides real-time input validation and instant translation results, ensuring an efficient and interactive experience. This project demonstrates the practical application of AI in bridging language barriers, offering an accessible and effective tool for accurate and personalized translation needs.

10. FUTURE SCOPE

The TransLingua system can be further enhanced and expanded with advanced features to improve usability, accuracy, and scalability.

- **Offline Translation Mode**

Develop an offline version using lightweight AI models so users can translate without an internet connection.

- **Real-Time Conversation Mode**

Enable live two-way conversation translation for meetings, travel, and customer support.

- **Mobile Application Development**

Create Android and iOS apps for better accessibility and portability.

- **Advanced Speech Recognition**

Improve accent recognition and background noise filtering using advanced speech AI models

- **Domain-Specific Translation**

Customize translation models for:

Healthcare

Legal documents

Technical content

Education

- **Chat & Video Call Integration**

Integrate with platforms like:

Zoom

Google Meet

WhatsApp

Microsoft Teams

11. APPENDIX

Source Code:

All codes are submitted in Git-Hub Repository.

Git-Hub Repository Link:

https://github.com/swathikareddy9885/TransLingua-AI-Powered-Multi-Language-Translator-main.git

Project Demo Link:

https://drive.google.com/file/d/1uM7tHIVAE69U9Yjaj3K8yvqqc2YArneD/view?usp=drivesdk