TransLingua:AI-Powered Multi-Language Translator

Team ID : LTVIP2026TMIDS76796

Team Size : 5

Team Leader : Nagisetty Naga Sahana

Team member : P Tejaswini

Team member : Madhu Sreesanareddy

Team member : Ranga Deevana Kumari

Team member : Karimisetty Srikala

College : Santhiram Engineering College

1. INTRODUCTION

1.1 Project Overview

TransLingua is an AI-based multilingual translation system designed to convert text from one language to another with high accuracy and speed. The system uses machine learning and natural language processing techniques to understand linguistic patterns and provide meaningful translations.
The application supports multiple languages and provides a user-friendly interface for real-time translation.

1.2 Purpose

The purpose of this project is to eliminate language barriers in communication by providing an intelligent translation tool that is fast, accurate, and easy to use. The system is intended for students, professionals, and general users who need quick multilingual translation support.

2. IDEATION PHASE

2.1 Problem Statement

Communication across different languages is difficult and time-consuming. Existing translation tools may lack accuracy or accessibility. There is a need for a reliable AI-based translation system that provides real-time and context-aware translation.

2.2 Empathy Map Canvas

- Users need quick and accurate translations.
- Users prefer simple interfaces.
- Users face difficulty understanding foreign language content.
- Users want real-time results without technical complexity.

2.3 Brainstorming

- Use Natural Language Processing for language understanding.
- Implement machine learning-based translation models.
- Provide web-based interface for easy access.
- Support multiple language pairs.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens application
2. User enters text
3. User selects target language
4. System processes input
5. Translated output is displayed

3.2 Solution Requirement

Functional Requirements

- Accept text input
- Detect source language
- Translate to target language
- Display translated output

Non-Functional Requirements
- Fast response time
- High accuracy
- User-friendly interface Scalability

3.3 Data Flow Diagram

Input Text → Preprocessing → Language Detection → Translation Model → Output Display

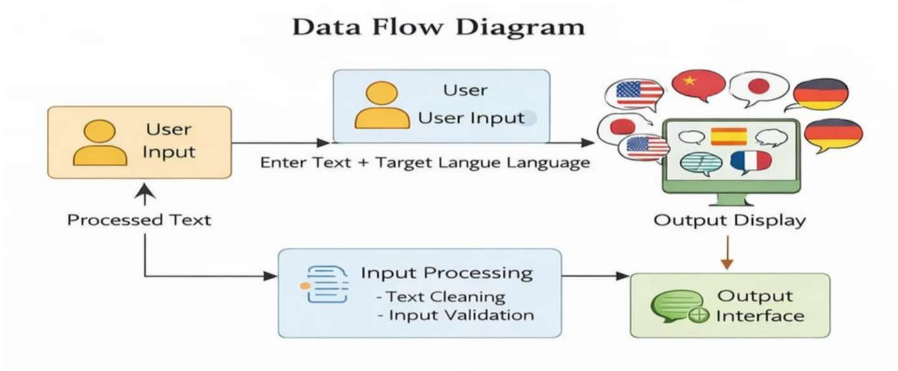


Figure: Data Flow Diagram

3.4 Technology Stack

- Programming Language: Python
- Frontend: HTML, CSS
- Backend: Flask
- Libraries: NLP libraries, Machine Learning models

4. PROJECT DESIGN

4.1 Problem Solution Fit

The system solves communication barriers by providing automated translation using AI, reducing manual effort and improving understanding across languages.

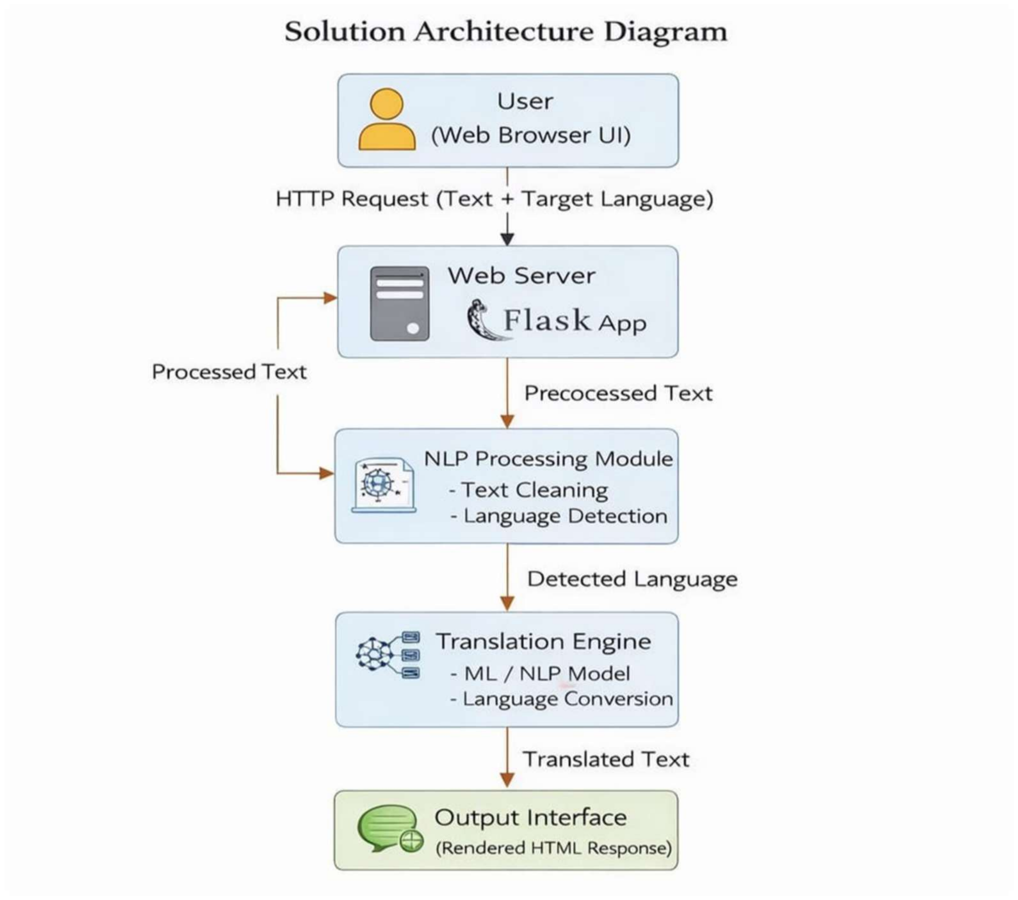
4.2 Proposed Solution

Develop a web-based translator that processes input text using machine learning models and outputs translated text in the selected language.

4.3 Solution Architecture

User Interface → Web Server → NLP Processing Module → Translation Engine → Output

Interface



5. PROJECT PLANNING s SCHEDULING

5.1 Project Planning

- Requirement Analysis
- Model Development
- System Implementation
- Testing
- Deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

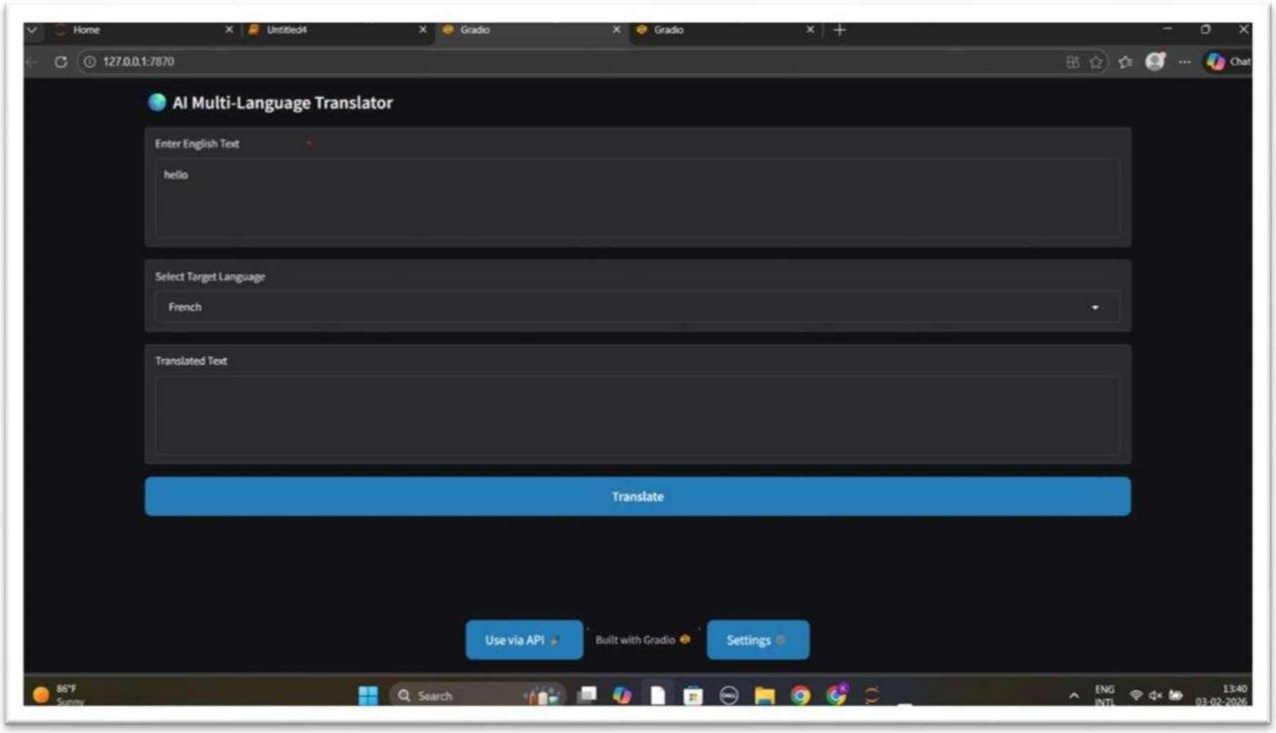
6.1 Performance Testing

- Tested translation accuracy
- Evaluated response time
- Verified multilingual support
- Checked system stability under multiple inputs

7. RESULTS

7.1 Output Screenshots

- Text input interface
- Language selection module
- Translated output display



AI Multi-Language Translator

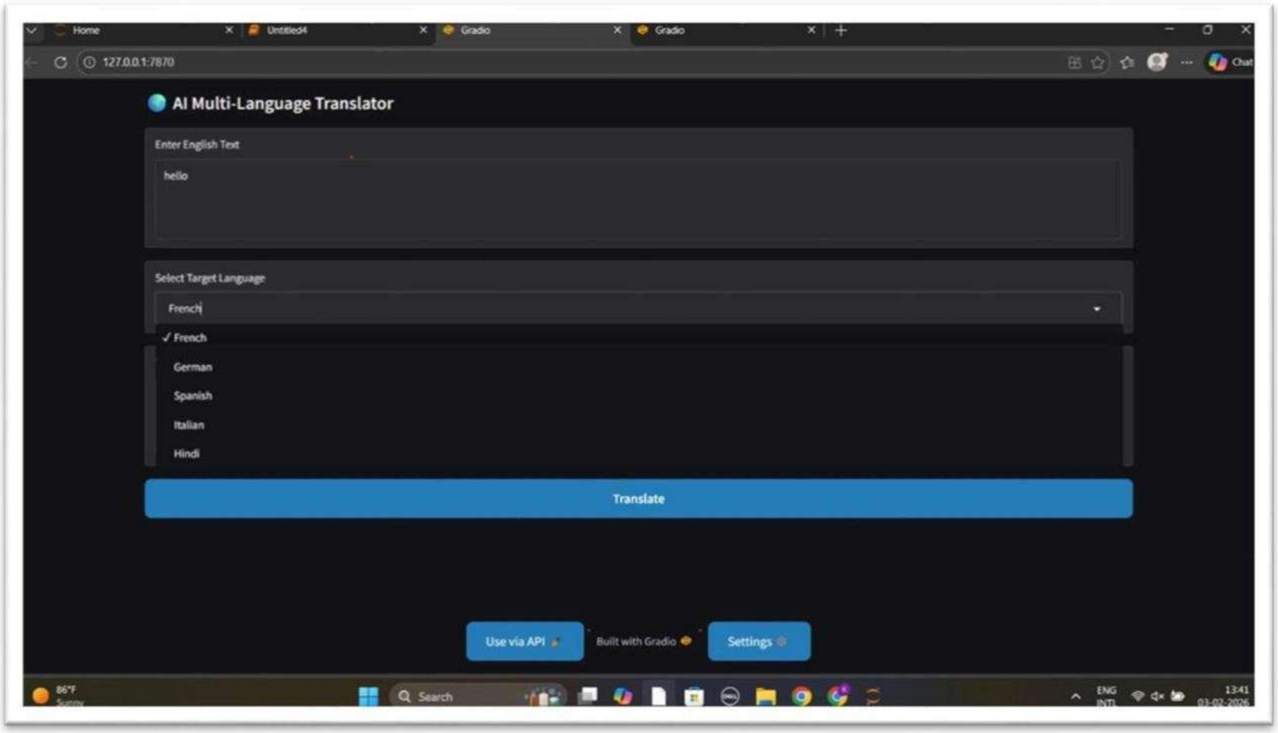
Enter English Text

hello

Select Target Language

French

Translated Text

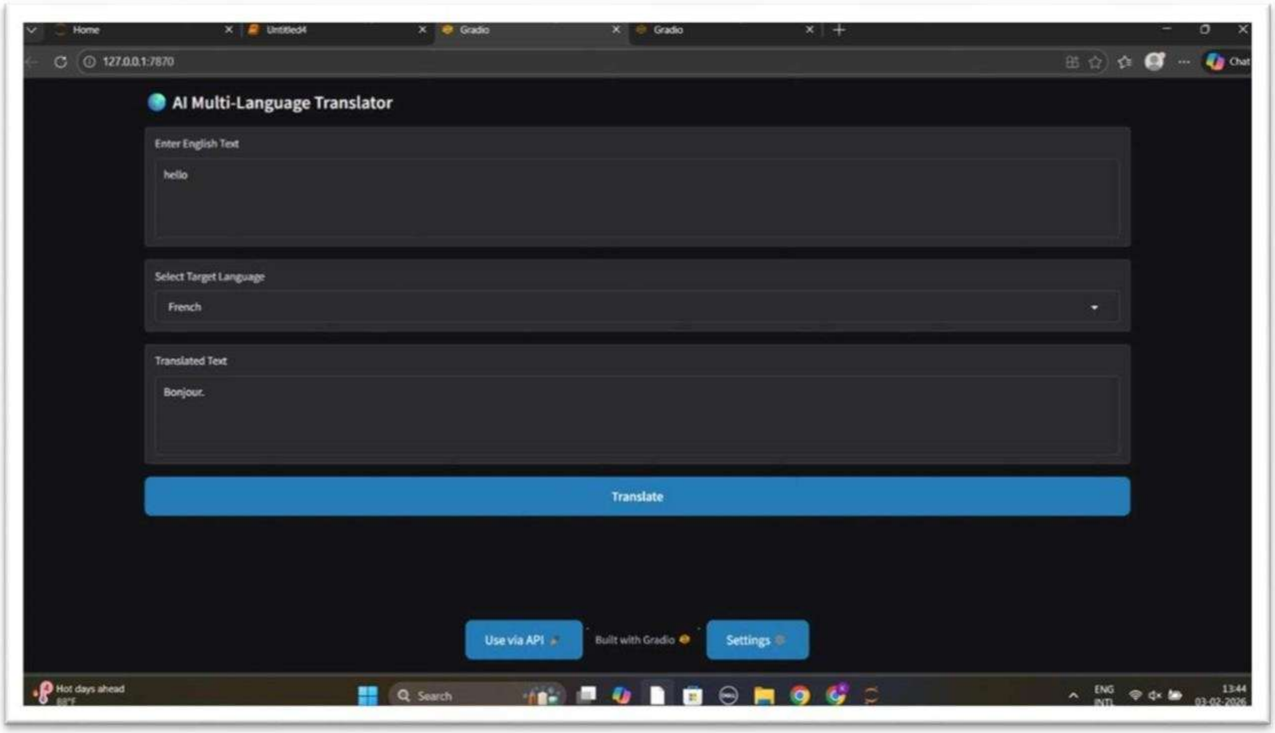

AI Multi-Language Translator

Enter English Text

hello

Select Target Language

French


AI Multi-Language Translator

Enter English Text

hello

Select Target Language

French

Translated Text

Bonjour.

8. ADVANTAGES s DISADVANTAGES

Advantages

- Fast translation

- User-friendly interface
- Supports multiple languages
- AI-based accuracy

Disadvantages

- Requires internet connection
- Accuracy depends on training data
- Limited support for rare languages

6. CONCLUSION

TransLingua successfully demonstrates the implementation of an AI-powered multilingual translation system. The project highlights the use of machine learning and NLP techniques to overcome language barriers and improve communication efficiency.

10. FUTURE SCOPE

- Voice-to-text translation□
- Mobile application version□
- Offline translation support□
- Improved contextual understanding□
- Integration with chat applications□

11. APPENDIX

Source Code

- Streamlit

```
!pip install gradio transformers
```

- app.py import gradio as gr from transformers import MarianMTModel, MarianTokenizer

```
# Language to model mapping language_model_map
= {
   "French": "Helsinki-NLP/opus-mt-en-fr",
   "German": "Helsinki-NLP/opus-mt-en-de",
    "Spanish": "Helsinki-NLP/opus-mt-en-es",
    "Italian": "Helsinki-NLP/opus-mt-en-it",
    "Hindi": "Helsinki-NLP/opus-mt-en-hi"
}

def translate_text(text, target_language):
    model_name = language_model_map[target_language]
tokenizer = MarianTokenizer.from_pretrained(model_name)
    model = MarianMTModel.from_pretrained(model_name)

    inputs = tokenizer(text, return_tensors="pt", padding=True)
translated = model.generate(**inputs)

    translated_text = tokenizer.decode(
        translated[0], skip_special_tokens=True
    )
    return translated_text


# Custom CSS (as shown in PDF)
custom_css = """ body {     background-
color: #f5f7fa !important;
} button {     background-color: #1f77b4
!important;
    color: white !important;     border-
radius: 8px !important;     padding: 12px
24px !important;     font-size: 16px
!important;
    transition: all 0.3s ease-in-out !important;
}  button:hover  {        background-color:
#ff7f0e !important;
    transform: scale(1.05);
}
"""

# Gradio UI with
gr.Blocks(css=custom_css) as demo:
    gr.Markdown("##        AI Multi-Language Translator")

    with                  gr.Row():
input_text     =     gr.Textbox(
lines=4,
            label="Enter English Text",
            placeholder="Type something in English..."
        )

    with gr.Row():
        target_lang = gr.Dropdown(
list(language_model_map.keys()),            label="Select
Target Language",            value="French"
        )

    with gr.Row():
output_text = gr.Textbox(
lines=4,
            label="Translated Text"
        )

    translate_btn = gr.Button("Translate")

    translate_btn.click(
fn=translate_text,
inputs=[input_text, target_lang],
outputs=output_text
    )

demo.launch()
```

Dataset Link

Language dataset used for training and evaluation.

GitHub C Project Demo Link

https://github.com/Deevana06/TransLingua