**Language Translator V.2.1**

**Abstract introduction**

This application aims to bridge language barriers by offering a user-friendly interface for text translation and voice-to-text functionality.

**Existing problem**

Language barriers can hinder communication, limit access to information, and impede cross-cultural understanding.

**Proposed solution**

This translation application addresses these challenges by providing a convenient and accessible tool for language translation and voice input.

**Dependencies:**

* **Python:** The programming language in which the code is written.
* **tkinter:** A Python library for creating graphical user interfaces (GUIs).
* **ttk:** A sub-module of tkinter for themed widgets.
* **googletrans:** A third-party library for Google Translate API integration.
* **gtts:** A third-party library for text-to-speech conversion.
* **os:** A built-in Python module for interacting with the operating system.
* **speech\_recognition:** A third-party library for speech-to-text recognition.

**Hardware Requirements:**

* **Processor:** Any modern processor capable of running Python.
* **RAM:** At least 1 GB of RAM, but more is recommended for better performance.
* **Storage:** A few MBs of storage space for the program files and any temporary audio files.
* **Microphone:** A microphone is required for speech recognition functionality.
* **Speakers or headphones:** Speakers or headphones are required to hear the translated text's audio output.

**Software Requirements:**

* **Operating System:** Windows, macOS, or Linux with Python installed.
* **Python Version:** Python 3.6 or later is recommended.
* **Third-Party Libraries:** The required third-party libraries (googletrans, gtts, speech\_recognition) can be installed using the Python package manager pip.

**Code:**

import tkinter as tk

from tkinter import ttk

from googletrans import Translator

from gtts import gTTS

import os

import speech\_recognition as sr

def translate\_text():

input\_text = input\_text\_area.get("1.0", "end-1c")

target\_language = target\_language\_var.get()

if input\_text:

translator = Translator()

translated\_text = translator.translate(input\_text, dest=target\_language).text

output\_text\_area.delete("1.0", "end")

output\_text\_area.insert("1.0", translated\_text)

text\_to\_speech(translated\_text, target\_language)

def text\_to\_speech(text, lang):

speak = gTTS(text=text, lang=lang, slow=False)

speak.save("translated\_voice.mp3")

os.system("start translated\_voice.mp3")

def capture\_voice():

recognizer = sr.Recognizer()

with sr.Microphone() as source:

print("Speak something...")

audio = recognizer.listen(source)

try:

voice\_text = recognizer.recognize\_google(audio)

input\_text\_area.delete("1.0", "end")

input\_text\_area.insert("1.0", voice\_text)

except sr.UnknownValueError:

print("Could not understand audio")

except sr.RequestError as e:

print("Could not request results; {0}".format(e))

root = tk.Tk()

root.title("Translation App")

inner\_frame = tk.Frame(root)

inner\_frame.pack(padx=10, pady=10)

input\_text\_label = tk.Label(inner\_frame, text="Input Text:")

input\_text\_label.pack(anchor="w", padx=(0,5), pady=(0,5))

input\_text\_area = tk.Text(inner\_frame, width=40, height=5)

input\_text\_area.pack(anchor="w", padx=(0,5))

languages = ['en', 'es', 'fr', 'de'] # Example target languages

target\_language\_var = tk.StringVar()

target\_language\_label = tk.Label(inner\_frame, text="Target Language:")

target\_language\_label.pack(anchor="w", padx=(0,5), pady=(5,5))

target\_language\_dropdown = ttk.Combobox(inner\_frame, textvariable=target\_language\_var, values=languages, width=37)

target\_language\_dropdown.pack(anchor="w", padx=(0,5), pady=(0,5))

target\_language\_dropdown.current(0) # Set default value

submit\_button = tk.Button(inner\_frame, text="Translate", command=translate\_text)

submit\_button.pack(anchor="w", padx=(0,5), pady=(5,0))

voice\_input\_button = tk.Button(inner\_frame, text="Voice Input", command=capture\_voice)

voice\_input\_button.pack(anchor="w", padx=(0,5), pady=(5,0))

output\_text\_label = tk.Label(inner\_frame, text="Translated Text:")

output\_text\_label.pack(anchor="w", padx=(0,5), pady=(10,5))

output\_text\_area = tk.Text(inner\_frame, width=40, height=5)

output\_text\_area.pack(anchor="w", padx=(0,5))

root.mainloop()

**Screenshots:**



