

In [1]:

```
from tkinter import *
from tkinter import ttk
from googletrans import Translator, LANGUAGES
from PIL import Image, ImageTk
import tkinter as tk
import speech_recognition as sr
```

In [2]:

```
root = tk.Tk()
root.geometry('600x1080')
canvas = tk.Canvas(root, width=600, height=1080)
canvas.pack()

img = Image.open("C:/Users/ibmtr/OneDrive/Desktop/translate.png")
img_re = img.resize((150, 150))
my_img = ImageTk.PhotoImage(img_re)

label = ttk.Label(root, anchor='center', image=my_img)
label.place(x=50, y=50)

image1 = Image.open('C:/Users/ibmtr/OneDrive/Desktop/bg4.webp')
img_re1 = image1.resize((600, 1080), Image.ANTIALIAS)
background_image = ImageTk.PhotoImage(img_re1)

# Add the image to the canvas as a background
canvas.create_image(0, 0, image=background_image, anchor='nw')
```

C:\Users\ibmtr\AppData\Local\Temp\ipykernel\_12500\461559260.py:14: DeprecationWarning: ANTIALIAS is deprecated and will be removed in Pillow 10 (2023-07-01). Use LANCZOS or Resampling.LANCZOS instead.

```
img_re1 = image1.resize((600, 1080), Image.ANTIALIAS)
```

Out[2]:

1

In [3]:

```
root.title("Adro Language Translator")
Label(root, text="ADRO", font="arial 20 bold", bg='white smoke').place(x=340, y=100)
Label(root, text="LANGUAGE TRANSLATOR", font="arial 20 bold", bg='white smoke').place(x=230, y=150)
```

In [4]:

```
Label(root, text="Enter Text", font='arial 13 bold', bg='white smoke').place(x=250, y=250)

Input_text = Text(root, font='arial 10', height=9, wrap=WORD, padx=5, pady=5, width=60)
Input_text.place(x=100, y=300)
```

In [5]:

```
def clear_text():
    Input_text.delete('1.0', END)

clear_button = tk.Button(root, text="Clear", command=clear_text, font = 'arial 11 bold', bg = 'white')
clear_button.place(x=450, y=405)
```

In [6]:

```
# Create an instance of the recognizer
recognizer = sr.Recognizer()

# Define a function to handle the microphone input
def speech_to_text(text=''):
    with sr.Microphone() as source:
        print("Listening...")
        audio = recognizer.listen(source)
        try:
            text = recognizer.recognize_google(audio)
            Input_text.insert(tk.END, text)
        except sr.UnknownValueError:
            Input_text.insert(tk.END, "Could not understand audio")
        except sr.RequestError:
            Input_text.insert(tk.END, "Could not connect to the service")
```

In [7]:

```
# Load microphone icon image
mic_img = Image.open("C:/Users/ibmtr/OneDrive/Desktop/mic.png")
mic_img_re = mic_img.resize((40, 40))
mic_icon = ImageTk.PhotoImage(mic_img_re)

# Create a Label to display the microphone icon
mic_label = ttk.Label(root, image=mic_icon)
mic_label.place(x=400, y=400)

# Call the speech_to_text function when the microphone icon is clicked
mic_label.bind("<Button-1>", lambda event: speech_to_text())
```

Out[7]:

```
'2101478439808<lambda>'
```

In [8]:

```
Output_text = Text(root, font = 'arial 10', height = 9, wrap = WORD, padx=5, pady= 5, width = 60)
Output_text.place(x = 100 , y = 600)
```

In [9]:

```
language = list(LANGUAGES.values())

src_lang = ttk.Combobox(root, values= language, width =25)
src_lang.place(x=20,y=250)
src_lang.set('choose input language')

dest_lang = ttk.Combobox(root, values= language, width =25)
dest_lang.place(x=20,y=550)
dest_lang.set('choose output language')
```

In [10]:

```
def Translate():
    translator = Translator()
    translated=translator.translate(text= Input_text.get(1.0, END) , src = src_lang.get(), de

    Output_text.delete(1.0, END)
    Output_text.insert(END, translated.text)
```

In [11]:

```
= tk.Button(root, text = 'Translate',font = 'arial 14 bold',pady = 5,command = Translate , bg
.place(x = 250, y= 500)
```

In [12]:

```
root.mainloop()
```

In [ ]: