Project 1:

import streamlit as st  
from deep\_translator import GoogleTranslator  
from gtts import gTTS  
import pyperclip  
import base64  
import os  
  
# ------------------------------  
# Supported Languages  
# ------------------------------  
LANGUAGES = {  
'English': 'en', 'Urdu': 'ur', 'French': 'fr',  
'Spanish': 'es', 'German': 'de', 'Arabic': 'ar',  
'Hindi': 'hi', 'Chinese': 'zh-CN'  
}  
  
# ------------------------------  
# Streamlit App Layout  
# ------------------------------  
st.set\_page\_config(page\_title="AI Language Translator", page\_icon="", layout="centered")  
st.title(" AI LANGUAGE TRANSLATOR WITH TTS")  
  
# ------------------------------  
# Initialize Session State  
# ------------------------------  
if "input\_text" not in st.session\_state:  
st.session\_state.input\_text = ""  
if "translated\_text" not in st.session\_state:  
st.session\_state.translated\_text = ""  
if "selected\_lang" not in st.session\_state:  
st.session\_state.selected\_lang = "Urdu"  
  
# ------------------------------  
# Input Section  
# ------------------------------  
st.session\_state.input\_text = st.text\_area(" Enter Text:", value=st.session\_state.input\_text, height=150)  
st.session\_state.selected\_lang = st.selectbox(" Select Target Language:", list(LANGUAGES.keys()),  
index=list(LANGUAGES.keys()).index(st.session\_state.selected\_lang))  
  
translated\_text\_placeholder = st.empty()  
status\_placeholder = st.empty()  
  
# ------------------------------  
# Translate Function  
# ------------------------------  
def translate\_text():  
try:  
target\_lang = LANGUAGES.get(st.session\_state.selected\_lang, 'en')  
translated = GoogleTranslator(source='auto', target=target\_lang).translate(st.session\_state.input\_text)  
st.session\_state.translated\_text = translated  
except Exception as e:  
st.error(f" Translation Error: {str(e)}")  
st.session\_state.translated\_text = ""  
  
# ------------------------------  
# Text-to-Speech Function  
# ------------------------------  
def speak\_text():  
try:  
target\_lang = LANGUAGES.get(st.session\_state.selected\_lang, 'en')  
tts = gTTS(text=st.session\_state.translated\_text, lang=target\_lang)  
tts.save("translated.mp3")  
with open("translated.mp3", "rb") as f:  
audio\_data = f.read()  
b64 = base64.b64encode(audio\_data).decode()  
md = f"""  
<audio controls autoplay>  
<source src="data:audio/mp3;base64,{b64}" type="audio/mp3">  
</audio>  
"""  
st.markdown(md, unsafe\_allow\_html=True)  
except Exception as e:  
st.error(f" TTS Error: {str(e)}")  
  
# ------------------------------  
# Buttons Layout  
# ------------------------------  
col1, col2, col3, col4 = st.columns(4)  
  
with col1:  
if st.button(" Translate"):  
if st.session\_state.input\_text.strip():  
translate\_text()  
translated\_text\_placeholder.text\_area(" Translated Text:",  
value=st.session\_state.translated\_text,  
height=150, disabled=True)  
status\_placeholder.success(" Translation complete.")  
else:  
status\_placeholder.warning(" Please enter text first.")  
  
with col2:  
if st.button(" Speak"):  
if st.session\_state.translated\_text:  
speak\_text()  
[status\_placeholder.info](http://status_placeholder.info/)(" Speaking translation...")  
else:  
status\_placeholder.warning(" No translated text available.")  
  
with col3:  
if st.button(" Copy"):  
if st.session\_state.translated\_text:  
pyperclip.copy(st.session\_state.translated\_text)  
status\_placeholder.success(" Text copied to clipboard!")  
else:  
status\_placeholder.warning(" No text to copy.")  
  
with col4:  
if st.button(" Clear"):  
st.session\_state.input\_text = ""  
st.session\_state.translated\_text = ""  
translated\_text\_placeholder.empty()  
[status\_placeholder.info](http://status_placeholder.info/)(" Field

Project 2:

from tkinter import \*  
from tkinter import ttk, messagebox  
from deep\_translator import GoogleTranslator  
from gtts import gTTS  
import os  
import pyperclip # for copying text  
  
# Supported languages  
LANGUAGES = {  
'English': 'en', 'Urdu': 'ur', 'French': 'fr',  
'Spanish': 'es', 'German': 'de', 'Arabic': 'ar',  
'Hindi': 'hi', 'Chinese': 'zh-CN'  
}  
  
# Main window setup  
root = Tk()  
root.title("AI LANGUAGE TRANSLATOR WITH TTS")  
root.geometry("650x550")  
root.config(bg="#e0f7fa")  
  
# Function: Translate Only  
def translate\_text():  
try:  
input\_text = text\_input.get("1.0", END).strip()  
selected\_lang = lang\_box.get()  
target\_lang = LANGUAGES.get(selected\_lang, 'en')  
  
if not input\_text:  
status\_label.config(text=" Please enter text.")  
return  
  
translated = GoogleTranslator(source='auto', target=target\_lang).translate(input\_text)  
text\_output.config(state=NORMAL)  
text\_output.delete("1.0", END)  
text\_output.insert(END, translated)  
text\_output.config(state=DISABLED)  
  
status\_label.config(text=" Translation complete.")  
except Exception as e:  
status\_label.config(text=f" Error: {str(e)}")  
  
# Function: Speak Translated Text  
def speak\_text():  
try:  
translated\_text = text\_output.get("1.0", END).strip()  
selected\_lang = lang\_box.get()  
target\_lang = LANGUAGES.get(selected\_lang, 'en')  
  
if not translated\_text:  
status\_label.config(text=" No translated text available to speak.")  
return  
  
tts = gTTS(text=translated\_text, lang=target\_lang)  
tts.save("translated.mp3")  
if [os.name](http://os.name/) == "nt":  
os.system("start translated.mp3")  
elif [os.name](http://os.name/) == "posix":  
os.system("afplay translated.mp3") # macOS  
else:  
os.system("mpg123 translated.mp3") # Linux  
  
status\_label.config(text=" Speaking translation...")  
except Exception as e:  
status\_label.config(text=f" Error: {str(e)}")  
  
# Function: Copy Translated Text  
def copy\_text():  
try:  
translated\_text = text\_output.get("1.0", END).strip()  
if translated\_text:  
pyperclip.copy(translated\_text)  
messagebox.showinfo("Copied", " Translated text copied to clipboard!")  
else:  
status\_label.config(text=" No text to copy.")  
except Exception as e:  
status\_label.config(text=f" Copy failed: {str(e)}")  
  
# Function: Clear All Fields  
def clear\_fields():  
text\_input.delete("1.0", END)  
text\_output.config(state=NORMAL)  
text\_output.delete("1.0", END)  
text\_output.config(state=DISABLED)  
status\_label.config(text=" Fields cleared.")  
  
# Heading  
Label(root, text=" AI LANGUAGE TRANSLATOR", font=("TIMES NEW ROMAN", 18, "bold"), bg="#e0f7fa").pack(pady=10)  
  
# Input text area  
Label(root, text="Enter Text:", font=("TIMES NEW ROMAN", 14), bg="#e0f7fa").pack()  
text\_input = Text(root, height=4, width=60)  
text\_input.pack(pady=5)  
  
# Language selection dropdown  
Label(root, text="Select Target Language:", font=("TIMES NEW ROMAN", 14), bg="#e0f7fa").pack()  
lang\_box = ttk.Combobox(root, values=list(LANGUAGES.keys()), width=30, state="readonly")  
lang\_box.set("Urdu")  
lang\_box.pack(pady=5)  
  
# Buttons: Translate, Speak, Copy & Clear  
btn\_frame = Frame(root,)  
btn\_frame.pack(pady=10)  
  
Button(btn\_frame, text="Translate", command=translate\_text, bg="green", fg="white", width=15).grid(row=0, column=0, padx=5)  
Button(btn\_frame, text="Speak", command=speak\_text, bg="blue", fg="white", width=15).grid(row=0, column=1, padx=5)  
Button(btn\_frame, text="Copy", command=copy\_text, bg="orange", fg="white", width=10).grid(row=0, column=2, padx=5)  
Button(btn\_frame, text="Clear", command=clear\_fields, bg="red", fg="white", width=10).grid(row=0, column=3, padx=5)  
  
# Output text area (read-only)  
Label(root, text="Translated Text:", font=("TIMES NEW ROMAN", 14), bg="#e0f7fa").pack()  
text\_output = Text(root, height=4, width=60, state=DISABLED)  
text\_output.pack(pady=5)  
  
# Status label  
status\_label = Label(root, text="", bg="#e0f7fa", fg="blue", font=("TIMES NEW ROMAN", 10, "italic"))  
status\_label.pack(pady=5)  
  
# Start GUI loop  
root.mainloop()

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |