```python
# import all the libraries for requirments
import streamlit as st # for design frontend
#Translating text between a wide range of languages.
#Detecting the language of input text (in some versions).
#Easy integration into Python projects for seamless language translation functionality.
from mtranslate import translate
import pandas as pd # for dataframe
import os
from gtts import gTTS # google text to speach
import base64 # help to convert the binary data to text format data

# Read the language data set
data=pd.read_csv(r"C:\Users\sunil\Desktop\DK\NIT\NIT- Data Science and AI Class\4. November\8th - N

data.dropna(inplace=True) # if any null value found the drop
lang = data['name'].to_list() # take all language name to a list
langlist=tuple(lang) # late this list to tuple if any duplicates the list have it remove which is t
langcode = data['iso'].to_list() # the short form take to a list

# map languange with their sort form
lang_array = {lang[i]: langcode[i] for i in range(len(langcode))}


# frontend  layout
st.title("Language Translation App") # App heading

# app description
st.write(""" ### Language Interpretation App

This app provides real-time language translation to help users communicate across different languag
It detects the input language and translates text into the selected target language, making it idea
multilingual communication, cross-cultural interaction, and understanding content in various langua
# input area for language
inputtext = st.text_area("Write Your Text",height=100)

options =list(langlist)
# creat a side bar area where all language contain with radio button to translaate the text that yo
selected_option = st.selectbox("Select Language to Translate :", options)
#choice = st.sidebar.radio('CHOSE LANGUAGE',options) # here we pass the tuple of language that we c

# creat a dictionary whichhold all the languages
speech_langs = {
    "af": "Afrikaans",
    "ar": "Arabic",
    "bg": "Bulgarian",
    "bn": "Bengali",
    "bs": "Bosnian",
    "ca": "Catalan",
    "cs": "Czech",
    "cy": "Welsh",
    "da": "Danish",
    "de": "German",
    "el": "Greek",
    "en": "English",
    "eo": "Esperanto",
    "es": "Spanish",
```

```
    "et": "Estonian",
    "fi": "Finnish",
    "fr": "French",
    "gu": "Gujarati",
    "od" : "odia",
    "hi": "Hindi",
    "hr": "Croatian",
    "hu": "Hungarian",
    "hy": "Armenian",
    "id": "Indonesian",
    "is": "Icelandic",
    "it": "Italian",
    "ja": "Japanese",
    "jw": "Javanese",
    "km": "Khmer",
    "kn": "Kannada",
    "ko": "Korean",
    "la": "Latin",
    "lv": "Latvian",
    "mk": "Macedonian",
    "ml": "Malayalam",
    "mr": "Marathi",
    "my": "Myanmar (Burmese)",
    "ne": "Nepali",
    "nl": "Dutch",
    "no": "Norwegian",
    "pl": "Polish",
    "pt": "Portuguese",
    "ro": "Romanian",
    "ru": "Russian",
    "si": "Sinhala",
    "sk": "Slovak",
    "sq": "Albanian",
    "sr": "Serbian",
    "su": "Sundanese",
    "sv": "Swedish",
    "sw": "Swahili",
    "ta": "Tamil",
    "te": "Telugu",
    "th": "Thai",
    "tl": "Filipino",
    "tr": "Turkish",
    "uk": "Ukrainian",
    "ur": "Urdu",
    "vi": "Vietnamese",
    "zh-CN": "Chinese"
}

if st.button("Convert"):
    # function to decode audio file for download
    def get_binary_file_downloader_html(bin_file, file_label='File'):
        with open(bin_file, 'rb') as f:
            df = f.read()
        bin_str = base64.b64encode(df).decode()
        href = f'<a href="df:application/octet-stream;base64,{bin_str}" download="{os.path.basename
        return href
```

```
    c1,c2 = st.columns([4,3])


    # I/O
    if len(inputtext) > 0 :
        try:
            output = translate(inputtext,lang_array[selected_option])
            with c1:
                st.text_area("TRANSLATED TEXT",output,height=200)
            # if speech support is available will render autio file
            if selected_option in speech_langs.values():
                with c2:
                    aud_file = gTTS(text=output, lang=lang_array[selected_option], slow=False)
                    aud_file.save("lang.mp3")
                    audio_file_read = open('lang.mp3', 'rb')
                    audio_bytes = audio_file_read.read()
                    bin_str = base64.b64encode(audio_bytes).decode()
                    st.audio(audio_bytes, format='audio/mp3')
                    st.markdown(get_binary_file_downloader_html("lang.mp3", 'Audio File'), unsafe_a
        except Exception as e:
            st.error(e)


# Add copyright text at the bottom using Markdown
st.markdown("""
    ---
    &copy; 2024 Creted By Darshanikanta. No copy right
""")
```