```
import streamlit as st
from mtranslate import translate
import pandas as pd
from gtts import gTTS
import base64
import os

# read language dataset
df = pd.read_csv(r"C:/Users/Hanshu/Desktop/DATA_SCIENCE/excel data_ML/language.csv")
df.dropna(inplace=True)
lang = df['name'].to_list()
langlist=tuple(lang)
langcode = df['iso'].to_list()

# create dictionary of language and 2 letter langcode
lang_array = {lang[i]: langcode[i] for i in range(len(langcode))}

# layout
st.title("Language-Translation")
inputtext = st.text_area("Hi Please Enter text here to Translate",height=100)

choice = st.sidebar.radio('SELECT LANGUAGE', langlist)

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"
}


# 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:
        data = f.read()
    bin_str = base64.b64encode(data).decode()
    href = f'<a href="data:application/octet-stream;base64,{bin_str}"
download="{os.path.basename(bin_file)}">Download {file_label}</a>'
```

```
    return href


C1,C2 = st.columns([4,3])


#I/O
if len(inputtext) > 0 :
    try:
        output = translate(inputtext, lang_array[choice])
        with C1:
            st.text_area("TRANSLATED TEXT",output,height=200)
        # if speech support is available will render autio file
        if choice in speech_langs.values():
            with C2:
                aud_file = gTTS(text=output, lang=lang_array[choice], 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_allow_html=True)
    except Exception as e:
        st.error(e)
```