```
# Importing necessary modules required
from playsound import playsound
import speech_recognition as sr
from googletrans import Translator
from gtts import gTTS
#from function_sound_file import function_sound
import os
flag = 0

# A tuple containing all the language and
# codes of the language will be detcted
dic = ('afrikaans', 'af', 'albanian', 'sq',
       'amharic', 'am', 'arabic', 'ar',
       'armenian', 'hy', 'azerbaijani', 'az',
       'basque', 'eu', 'belarusian', 'be',
       'bengali', 'bn', 'bosnian', 'bs', 'bulgarian',
       'bg', 'catalan', 'ca', 'cebuano',
       'ceb', 'chichewa', 'ny', 'chinese (simplified)',
       'zh-cn', 'chinese (traditional)',
       'zh-tw', 'corsican', 'co', 'croatian', 'hr',
       'czech', 'cs', 'danish', 'da', 'dutch',
       'nl', 'english', 'en', 'esperanto', 'eo',
       'estonian', 'et', 'filipino', 'tl', 'finnish',
       'fi', 'french', 'fr', 'frisian', 'fy', 'galician',
       'gl', 'georgian', 'ka', 'german',
       'de', 'greek', 'el', 'gujarati', 'gu',
       'haitian creole', 'ht', 'hausa', 'ha',
       'hawaiian', 'haw', 'hebrew', 'he', 'hindi',
       'hi', 'hmong', 'hmn', 'hungarian',
       'hu', 'icelandic', 'is', 'igbo', 'ig', 'indonesian',
       'id', 'irish', 'ga', 'italian',
       'it', 'japanese', 'ja', 'javanese', 'jw',
       'kannada', 'kn', 'kazakh', 'kk', 'khmer',
       'km', 'korean', 'ko', 'kurdish (kurmanji)',
       'ku', 'kyrgyz', 'ky', 'lao', 'lo',
       'latin', 'la', 'latvian', 'lv', 'lithuanian',
       'lt', 'luxembourgish', 'lb',
       'macedonian', 'mk', 'malagasy', 'mg', 'malay',
       'ms', 'malayalam', 'ml', 'maltese',
       'mt', 'maori', 'mi', 'marathi', 'mr', 'mongolian',
       'mn', 'myanmar (burmese)', 'my',
       'nepali', 'ne', 'norwegian', 'no', 'odia', 'or',
       'pashto', 'ps', 'persian', 'fa',
       'polish', 'pl', 'portuguese', 'pt', 'punjabi',
       'pa', 'romanian', 'ro', 'russian',
       'ru', 'samoan', 'sm', 'scots gaelic', 'gd',
       'serbian', 'sr', 'sesotho', 'st',
       'shona', 'sn', 'sindhi', 'sd', 'sinhala', 'si',
       'slovak', 'sk', 'slovenian', 'sl',
       'somali', 'so', 'spanish', 'es', 'sundanese',
       'su', 'swahili', 'sw', 'swedish',
       'sv', 'tajik', 'tg', 'tamil', 'ta', 'telugu',
       'te', 'thai', 'th', 'turkish',
       'tr', 'ukrainian', 'uk', 'urdu', 'ur', 'uyghur',
       'ug', 'uzbek', 'uz',
       'vietnamese', 'vi', 'welsh', 'cy', 'xhosa', 'xh',
       'yiddish', 'yi', 'yoruba',
```

```
        'yo', 'zulu', 'zu')


# Capture Voice
# takes command through microphone
def takecommand():
        r = sr.Recognizer()
        with sr.Microphone() as source:
                print("listening.....")
                r.pause_threshold = 1
                audio = r.listen(source)

        try:
                print("Recognizing.....")
                query = r.recognize_google(audio, language='en-in')
                print(f"The User said {query}\n")
        except Exception as e:
                print("say that again please.....")
                return "None"
        return query


# Input from user
# Make input to lowercase
query = takecommand()
while (query == "None"):
        query = takecommand()


def destination_language():
        print("Enter the language in which you\
        want to convert : Ex. Hindi , English , etc.")
        print()

        # Input destination language in
        # which the user wants to translate
        to_lang = takecommand()
        while (to_lang == "None"):
                to_lang = takecommand()
        to_lang = to_lang.lower()
        return to_lang

to_lang = destination_language()

# Mapping it with the code
while (to_lang not in dic):
        print("Language in which you are trying\
        to convert is currently not available ,\
        please input some other language")
        print()
        to_lang = destination_language()

to_lang = dic[dic.index(to_lang)+1]


# invoking Translator
translator = Translator()
```

```
# Translating from src to dest
text_to_translate = translator.translate(query, dest=to_lang)

text = text_to_translate.text

# Using Google-Text-to-Speech ie, gTTS() method
# to speak the translated text into the
# destination language which is stored in to_lang.
# Also, we have given 3rd argument as False because
# by default it speaks very slowly
speak = gTTS(text=text, lang=to_lang, slow=False)

# Using save() method to save the translated
# speech in capture_voice.mp3
#from function_sound_file import function_sound

#function_sound()
speak.save("captured_voice.mp3")

# Using OS module to run the translated voice.
playsound('captured_voice.mp3')
#os.rmdir('captured_voice.mp3')
os.remove('captured_voice.mp3')

# Printing Output
print(text)
```