Task 1

Source code of language translator

# Importing necessary modules required

import speech\_recognition as spr

from googletrans import Translator

from gtts import gTTS

import os

# Creating Recogniser() class object

recog1 = spr.Recognizer()

# Creating microphone instance

mc = spr.Microphone()

# Capture Voice

with mc as source:

print("Speak 'hello' to initiate the Translation !")

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

recog1.adjust\_for\_ambient\_noise(source, duration=0.2)

audio = recog1.listen(source)

MyText = recog1.recognize\_google(audio)

MyText = MyText.lower()

# Here initialising the recorder with

# hello, whatever after that hello it

# will recognise it.

if 'hello' in MyText:

# Translator method for translation

translator = Translator()

# short form of english in which

# you will speak

from\_lang = 'en'

# In which we want to convert, short

# form of hindi

to\_lang = 'hi'

with mc as source:

print("Speak a stentence...")

recog1.adjust\_for\_ambient\_noise(source, duration=0.2)

# Storing the speech into audio variable

audio = recog1.listen(source)

# Using recognize.google() method to

# convert audio into text

get\_sentence = recog1.recognize\_google(audio)

# Using try and except block to improve

# its efficiency.

try:

# Printing Speech which need to

# be translated.

print("Phase to be Translated :"+ get\_sentence)

# Using translate() method which requires

# three arguments, 1st the sentence which

# needs to be translated 2nd source language

# and 3rd to which we need to translate in

text\_to\_translate = translator.translate(get\_sentence,

src= from\_lang,

dest= to\_lang)

# Storing the translated text in text

# variable

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

speak.save("captured\_voice.mp3")

# Using OS module to run the translated voice.

os.system("start captured\_voice.mp3")

# Here we are using except block for UnknownValue

# and Request Error and printing the same to

# provide better service to the user.

except spr.UnknownValueError:

print("Unable to Understand the Input")

except spr.RequestError as e:

print("Unable to provide Required Output".format(e))