New Wikipedia

import pyttsx3

import wikipedia

voice = pyttsx3.init()

In = input("Searching  wikipedia...")

result = wikipedia.summary(In, sentences = 10 )

print(result)

voice.say(result)

voice.runAndWait()

New Language Translator

from playsound import playsound

import speech\_recognition as sr

from googletrans import Translator

from gtts import gTTS

import os

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')

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"user said {query}\n")

    except Exception as e:

        print("say that again please.....")

        return "None"

    return query

query = takecommand()

while (query == "None"):

    query = takecommand()

def destination\_language():

    print("Enter the language in which you want to convert")

    print()

    to\_lang = takecommand()

    while (to\_lang == "None"):

        to\_lang = takecommand()

    to\_lang = to\_lang.lower()

    return to\_lang

to\_lang = destination\_language()

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]

text\_to\_translate = Translator.translate(query, dest=to\_lang)

text = text\_to\_translate.text

speak = gTTS(text=text, lang=to\_lang, slow=False)

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

playsound('captured\_voice.mp3')

os.remove('captured\_voice.mp3')

print(text)

Code With Harry Assistant

import pyttsx3 #pip install pyttsx3

import speech\_recognition as sr #pip install speechRecognition

import datetime

import wikipedia #pip install wikipedia

import webbrowser

import os

import smtplib

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[0].id)

def speak(audio):

    engine.say(audio)

    engine.runAndWait()

def wishMe():

    hour = int(datetime.datetime.now().hour)

    if hour>=0 and hour<12:

        speak("Good Morning!")

    elif hour>=12 and hour<18:

        speak("Good Afternoon!")

    else:

        speak("Good Evening!")

    speak("I am Jarvis Sir. Please tell me how may I help you")

def takeCommand():

    #It takes microphone input from the user and returns string output

    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"User said: {query}\n")

    except Exception as e:

        # print(e)

        print("Say that again please...")

        return "None"

    return query

def sendEmail(to, content):

    server = smtplib.SMTP('smtp.gmail.com', 587)

    server.ehlo()

    server.starttls()

    server.login('youremail@gmail.com', 'your-password')

    server.sendmail('youremail@gmail.com', to, content)

    server.close()

if \_\_name\_\_ == "\_\_main\_\_":

    wishMe()

    while True:

    # if 1:

        query = takeCommand().lower()

        # Logic for executing tasks based on query

        if 'wikipedia' in query:

            speak('Searching Wikipedia...')

            query = query.replace("wikipedia", "")

            results = wikipedia.summary(query, sentences=2)

            speak("According to Wikipedia")

            print(results)

            speak(results)

        elif 'open youtube' in query:

            webbrowser.open("youtube.com")

        elif 'open google' in query:

            webbrowser.open("google.com")

        elif 'open stackoverflow' in query:

            webbrowser.open("stackoverflow.com")

        elif 'play music' in query:

            music\_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'

            songs = os.listdir(music\_dir)

            print(songs)

            os.startfile(os.path.join(music\_dir, songs[0]))

        elif 'the time' in query:

            strTime = datetime.datetime.now().strftime("%H:%M:%S")

            speak(f"Sir, the time is {strTime}")

        elif 'open code' in query:

            codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"

            os.startfile(codePath)

        elif 'email to harry' in query:

            try:

                speak("What should I say?")

                content = takeCommand()

                to = "harryyourEmail@gmail.com"

                sendEmail(to, content)

                speak("Email has been sent!")

            except Exception as e:

                print(e)

                speak("Sorry my friend harry bhai. I am not able to send this email")

# TEXT TO SPEECH

from gtts import gTTS   
import os  
mytext =str(input("Enter string " ))  
language = 'en'  
myobj = gTTS(text=mytext, lang=language, slow=False)  
myobj.save("welcome.mp3")  
os.system("mpg321 welcome.mp3