import ipaddress

from multiprocessing.connection import wait

import operator

import pyttsx3

from requests import request

import requests #pip install pyttsx3

import speech\_recognition as sr #pip install speechRecognition

import datetime

import wikipedia #pip install wikipedia

import webbrowser

import os

import smtplib

import pywhatkit as kit

import random

import time

import sys

from sys import exit

import pyjokes

import PyPDF2

import math

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtCore import QTimer, QTime, QDate, Qt

from PyQt5.QtGui import QMovie

from PyQt5.QtCore import \*

from PyQt5.QtGui import \*

from PyQt5.QtWidgets import \*

from PyQt5.uic import loadUiType

from andygui import Ui\_ANDY

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

# print(voices[1].id)

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

def speak(audio):

     engine.say(audio)

     engine.runAndWait()

def sendEmail(to, content):

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

    server.ehlo()

    server.starttls()

    server.login('mohd9182cl@gmail.com', 'king9182@')

    server.sendmail('mohd9182cl@gmail.com', to, content)

    server.close()

def pdf\_reader():

    book = open('C:\\Users\\admin\\Downloads\\ext.pdf','rb')

    pdfReader = PyPDF2.PdfFileReader(book)

    pages = pdfReader.numPages

    speak(f"Total number of pages in this book {pages} ")

    speak("sir please enter the page number i have to read")

    pg = int(input("please enter the page number: "))

    page =pdfReader.getPage(pg)

    text = page.extractText()

    speak(text)

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 Andy Sir. Please tell me how may I help you")

class MainThread(QThread):

    def \_\_init\_\_(self):

        super(MainThread,self).\_\_init\_\_()

    def run(self):

        permission = self.takecommand().lower()

        while True:

            if 'wake up' in permission:

                 self.taskexecution()

            elif 'goodbye' in permission:

                speak("goodbye sir, Thanks for using me  , Have a good Day")

                sys.exit()

    def takecommand(self):

        #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...")

            self.query = r.recognize\_google(audio, language='en-in')

            print(f"User said: {self.query}\n")

        except Exception as e:

            # print(e)

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

            return "None"

        return self.query

    def taskexecution(self):

        wishMe()

        while True:

       # if 1:

            self.query = self.takecommand().lower()

            # Logic for executing tasks based on self.query

            if 'wikipedia' in self.query:

                speak('Searching Wikipedia...')

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

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

                speak("According to Wikipedia")

                print(results)

                speak(results)

            #-----opening youtube

            elif 'open youtube' in self.query:

                webbrowser.open("youtube.com")

            #----opening google

            elif 'open google' in self.query:

                webbrowser.open("google.com")

            #----opening stackoverflow

            elif 'open stackoverflow' in self.query:

                webbrowser.open("stackoverflow.com")

           #-----playing offline music

            elif 'play music' in self.query:

                music\_dir = 'D:\\mp3'

                songs = os.listdir(music\_dir)

                print(songs)

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

            #-------time query

            elif 'the time' in self.query:

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

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

            #-------------opening vscode

            elif 'open code' in self.query:

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

                os.startfile(codePath)

            #-------open notebook

            elif 'open notepad' in self.query:

                npath="C:\\Windows\\system32\\notepad.exe"

                os.startfile(npath)

            #------open cmd

            elif 'open command prompt' in self.query:

                print("Opening command prompt")

                cpath="C:\\Windows\\system32\\cmd.exe"

                os.startfile(cpath)

            elif ' bilal ansari' in self.query or "prabhat rawat" in self.query:

                speak("Bilal Ansari and Prabhat rawat are my creators ! they developed me in year 2022 for their  University project.")

            #-----email

            elif 'send email' in self.query:

                try:

                    speak("What should I say?")

                    content = self.takecommand()

                    speak("send to whom please type the mail id below")

                    mailid=input()

                    to=mailid.lower()

                    sendEmail(to, content)

                    speak("Email has been sent!")

                except Exception as e:

                    print(e)

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

            #-------whatsapp message

            elif 'send message' in self.query:

                kit.sendwhatmsg("+911212121212", "this is testing protocol",9,35)

            #---------song on youtube

            elif 'play song on youtube' in self.query:

                speak("which song")

                print("Which song")

                self.query = self.takecommand().lower()

                kit.playonyt(self.query)

            #-------search on google

            elif 'search on google' in self.query:

                print("sir, what should I search on google?")

                speak("sir, what should I search on google?")

                cm=self.takecommand().lower()

                webbrowser.open(f"{cm}")

            #to close application

            elif 'close notepad' in self.query:

                speak("ok Sir,closing notepad")

                os.system("taskkill /f /im notepad.exe")

            elif 'close command prompt' in self.query:

                speak("ok Sir,closing command prompt")

                os.system("taskkill /f /im cmd.exe")

            #----to sel alarm

            elif 'set alarm' in self.query:

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

                if nn==22:

                 music\_dir = 'D:\\mp3'

                 songs = os.listdir(music\_dir)

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

           #----joke

            elif 'joke' in self.query:

                joke = pyjokes.get\_joke()

                print(joke)

                speak(joke)

            #--------to shutdown the system

            elif 'shutdown the system' in self.query:

                os.system("shutdown /s /t 5")

            #--------to get current location

            elif 'where am i' in self.query  or "where we are" in self.query:

                try:

                    speak("wait sir, let me check")

                    ipadd = requests.get('https://api.ipify.org').text

                    print(ipadd)

                    url = 'https://get.geojs.io/v1/ip/geo/'+ipadd+'.json'

                    geo\_requests= requests.get(url)

                    geo\_data = geo\_requests.json()

                    city = geo\_data['city']

                    country = geo\_data['country']

                    speak(f"sir, i am not sure, but i think we are in {city} city of {country} country")

                except Exception as e:

                    speak("Sorry sir , Due to network issue i am not able to find where we are")

                    pass

            #---------Normal conversation

            elif 'hello andy' in self.query:

                speak("hello sir, May I help you with something")

            #---------to read pdf document

            elif 'read pdf' in self.query:

                pdf\_reader()

            #-------TO perform calculations

            elif "do some calculations" in self.query or "can you calculate" in self.query:

                r = sr.Recognizer()

                with sr.Microphone() as source:

                    speak("Say what do you want to calculate, example : 3 plus 3")

                    print("listening....")

                    r.adjust\_for\_ambient\_noise(source)

                    audio = r.listen(source)

                my\_string = r.recognize\_google(audio)

                print(my\_string)

                def get\_operator\_fn(op):

                    return {

                        '+' : operator.add,

                        '-' : operator.sub,

                        'x' : operator.mul,

                        'X' : operator.mul,

                        'divided' : operator.\_\_truediv\_\_,

                    }[op]

                def eval\_binary\_expr(op1, oper, op2):

                    op1,op2 = int(op1), int(op2)

                    return get\_operator\_fn(oper)(op1, op2)

                speak("Your result is")

                speak(eval\_binary\_expr(\*(my\_string.split())))

                print(eval\_binary\_expr(\*(my\_string.split())))

            #-------Normal greetings

            elif 'hello andy' in self.query:

                print("Hello sir, may I help you with something")

                speak("Hello sir, may I help you with something")

            elif 'how are you' in self.query:

                print("I am good sir, what about you")

                speak("I am good sir, what about you")

            elif'i am good' in self.query:

                print("it is good to hear that from you ,may i help you with something")

                speak("it is good to hear that from you ,may i help you with something")

            elif'fine' in self.query:

                print("it is good to hear that from you ,may i help you with something")

                speak("it is good to hear that from you,  may i help you with something")

            elif 'go to sleep' in self.query:

                print("ok sir i am going to sleep you can call me again if you need any help")

                speak("ok sir i am going to sleep you can call me again if you need any help")

                self.run()

            #time.sleep(5)

            #speak("Sir, do you have any other work")

startExecution = MainThread()

class Main(QMainWindow):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.ui= Ui\_ANDY()

        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect(self.startTask)

        self.ui.pushButton\_2.clicked.connect(self.close)

    def startTask(self):

        self.ui.movie = QtGui.QMovie("../../../Downloads/sndy12.gif")

        self.ui.label.setMovie(self.ui.movie)

        self.ui.movie.start()

        timer= QTimer(self)

        timer.timeout.connect(self.showTime)

        timer.start(1000)

        startExecution.start()

    def showTime(self):

        current\_time = QTime.currentTime()

        current\_date = QDate.currentDate()

        label\_time = current\_time.toString('hh:mm:ss')

        label\_date = current\_date.toString(Qt.ISODate)

        self.ui.textBrowser.setText(label\_date)

        self.ui.textBrowser\_2.setText(label\_time)

app = QApplication(sys.argv)

andy = Main()

andy.show()

exit(app.exec\_())