import operator  
import sys  
import time  
from email import encoders  
from turtle import title  
from turtledemo.chaos import f  
  
import PyPDF2  
import pyautogui  
import pyjokes as pyjokes  
import pyttsx3  
import requests  
import self as self  
import speech\_recognition as sr  
import datetime  
import os  
import cv2  
import random  
from requests import get  
import wikipedia  
import webbrowser  
import pywhatkit as kit  
import smtplib  
import pyjokes  
import time  
from email.mime.multipart import MIMEMultipart  
from email.mime.text import MIMEText  
from email.mime.base import MIMEBase  
import instaloader  
from bs4 import BeautifulSoup  
from pywikihow import search\_wikihow  
import psutil  
import speedtest  
  
  
  
  
  
  
  
#voice  
engine = pyttsx3.init('sapi5')  
voices = engine.getProperty('voices')  
engine.setProperty('voices', voices[0].id)  
engine.setProperty('rate', 200)  
  
#text to speech  
  
def speak(audio):  
 engine.say(audio)  
 print(audio)  
 engine.runAndWait()  
  
#to convert voice into text  
def takecommand():  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 print("listening...")  
 r.pause\_threshold = 1  
 audio = r.listen(source,timeout=1,phrase\_time\_limit=5)  
  
 try:  
 print("recognizing...")  
 query = r.recognize\_google(audio, language='en-in')  
 print(f"user said: {query}")  
  
 except Exception as e:  
 #speak("say that again please...")  
 return "none"  
 query = query.lower()  
 return query  
  
#to wish  
def wish():  
 hour = int(datetime.datetime.now().hour)  
  
 tt = time.strftime("%I:%M %p")  
  
 if hour>=5 and hour<12:  
 speak(f"good morning sir, its {tt}")  
 elif hour==12:  
 speak(f"good noon sir, its {tt}")  
 elif hour>12 and hour<18:  
 speak(f"good afternoon sir,its {tt}")  
 elif hour>=18 and hour<21:  
 speak(f"good evening sir, its {tt}")  
 else:  
 speak(f"good night sir, its {tt}")  
 speak("i am jarvis. please tell me how can i help you")  
  
  
#for news updates  
def news():  
 main\_url = 'http://newsapi.org/v2/top-headlines?sources=techcrunch&apiKey=fdd4431a92394ce89e3fbd5ed27afe4e'  
 main\_page = requests.get(main\_url).json()  
 #print(main page)  
 articles = main\_page["articles"]  
 #print(articles)  
 head = []  
 day = ["first", "second", "third", "fourth", "fifth", "sixth", "seventh"]  
 for ar in articles:  
 head.append(ar["title"])  
  
 for i in range (len(day)):  
 speak(f"today's {day[i]} news is: {head[i]}")  
  
#to read pdf  
def pdf\_reader():  
 book = open('JARVIS Project Report.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)  
  
  
  
#Executing Tasks through voice commands  
def TaskExecution():  
 wish()  
 while True:  
 query = takecommand()  
  
  
 #logic building for task  
  
 #open notepad  
 if "open notepad" in query:  
 apath = "C:\\Windows\\notepad.exe"  
 os.startfile(apath)  
  
 #open dev c++  
 elif "open c program" in query:  
 bpath = "C:\\Program Files (x86)\\Dev-Cpp\\devcpp.exe"  
 os.startfile(bpath)  
  
 #open cmd  
 elif "open command prompt" in query:  
 os.system("start cmd")  
  
 #open camera  
 elif "open camera" in query:  
 cap = cv2.VideoCapture(0)  
 while True:  
 ret, img = cap.read()  
 cv2.imshow('webcam', img)  
 k = cv2.waitKey(50)  
 if k == 27:  
 break;  
 cap.release()  
 cv2.destroyAllWindows()  
 break  
  
 #play music  
 elif "play music" in query:  
 music\_dir = "D:\\Music"  
 songs = os.listdir(music\_dir)  
 rd = random.choice(songs)  
 os.startfile(os.path.join(music\_dir, rd))  
  
 #to adjust volume  
 elif "volume up" in query:  
 pyautogui.press("volumeup")  
  
 elif "volume down" in query:  
 pyautogui.press("volumedown")  
  
 elif "volume mute" in query or "mute" in query:  
 pyautogui.press("volumemute")  
  
 #to know ip address  
 elif "ip address" in query:  
 ip = get('https://api.ipify.org').text  
 speak(f"your ip address is {ip}")  
  
 #to search in wikipedia  
 elif "wikipedia" in query:  
 speak("searching wikipedia...")  
 query = query.replace("wikipedia", "")  
 results = wikipedia.summary(query, sentences=2)  
 speak("according to wikipedia")  
 speak(results)  
 # print(results)  
  
 #to open youtube  
 elif "open youtube" in query:  
 webbrowser.open("www.youtube.com")  
  
 #to open facebook  
 elif "open facebook" in query:  
 webbrowser.open("www.facebook.com")  
  
 #to open github  
 elif "open github" in query:  
 webbrowser.open("www.github.com")  
  
 #to search in google  
 elif "search google" in query:  
 speak("sir,what should i search on google")  
 cm = takecommand().lower()  
 webbrowser.open(f"{cm}")  
  
 #send whatsapp message  
 elif "send message" in query:  
 speak("what is the message?")  
 mes = takecommand().lower()  
 kit.sendwhatmsg\_instantly("+918092989706",f"{mes}")  
  
 #to play song on youtube  
 elif "play song on youtube" in query:  
 speak("which song do i play?")  
 pl = takecommand().lower()  
 kit.playonyt(f"{pl}")  
  
 #to close notepad  
 elif "close notepad" in query:  
 speak("okey sir, closing notepad")  
 os.system("taskkill /f /im notepad.exe")  
  
  
 #to find a joke  
 elif "tell me a joke" in query:  
 joke = pyjokes.get\_joke()  
 speak(joke)  
  
 #to shut down the system  
 elif "shut down the system" in query:  
 os.system("shutdown /s /t 5")  
  
 #to restart the system  
 elif "restart the system" in query:  
 os.system("shutdown /r /t 5")  
  
 #to sleep the system  
 elif "sleep the system" in query:  
 os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")  
  
 #to switch window  
 elif "switch the window" in query:  
 pyautogui.keyDown("alt")  
 pyautogui.press("tab")  
 time.sleep(1)  
 pyautogui.keyUp("alt")  
  
 #news  
 elif "tell me news" in query:  
 speak("Please wait sir, fetching the latest news!!!`")  
 news()  
  
 #to send mail  
 elif "send mail" in query:  
  
 speak("sir, what should i say?")  
 query = takecommand().lower()  
 if "send a file" in query:  
 email = 'striversayan7@gmail.com'  
 password = 'Chottu@2002'  
 send\_to\_email = 'sayanhalder2k20@gmail.com'  
 speak("okay sir, what is the subject for this email?")  
 query = takecommand().lower()  
 subject = query  
 speak("and sir, what is the messege for this email?")  
 query2 = takecommand().lower()  
 messege = query2  
 speak("sir, please enter the correct path of the file into the shell")  
 file\_location = input("please enter the path here")  
  
 speak("please wait! i am sending the mail now")  
  
 msg = MIMEMultipart()  
 msg['From'] = email  
 msg['To'] = send\_to\_email  
 msg['Subject'] = subject  
  
 msg.attach(MIMEText(messege, 'plain'))  
  
 #setup the attachment  
 filename = os.path.basename(file\_location)  
 attachment = open(file\_location, "rb")  
 part = MIMEBase('application', 'octet-stream')  
 part.set\_payload(attachment.read())  
 encoders.encode\_base64(part)  
 part.add\_header('Context-Disposition', "attachment; filename- %s" % filename)  
  
 #attach the attachment to the MIMEMultipart object  
 msg.attach(part)  
  
 server = smtplib.SMTP('smtp.gmail.com', 587)  
 server.starttls()  
 server.login(email, password)  
 text = msg.as\_string()  
 server.sendmail(email, send\_to\_email, text)  
 server.quit()  
 speak("email has been sent to sayan")  
  
 else:  
 email = 'striversayan7@gmail.com'  
 password = 'Chottu@2002'  
 send\_to\_email = 'sayanhalder2k20@gmail.com'  
 messege = query  
  
 server = smtplib.SMTP('smtp.gmail.com', 587)  
 server.starttls()  
 server.login(email, password)  
 server.sendmail(email, send\_to\_email, messege)  
 server.quit()  
 speak("email has been sent to sayan")  
  
 #to check temperature  
 elif "temperature" in query:  
 search = "temperature in kolkata"  
 url = f"https://www.google.com/search?q={search}"  
 r = requests.get(url)  
 data = BeautifulSoup(r.text,"html.parser")  
 temp = data.find("div",class\_="BNeawe").text  
 speak(f"current {search} is {temp}")  
  
 #to activate how to do mode  
 elif "activate how to do mode" in query:  
 speak("how to do mode is activated please tell me what do you want to know?")  
 how = takecommand().lower()  
 try:  
 if "exit" in how or "close" in how:  
 speak("okay sir, how to do mode is closed")  
 break  
 else:  
 max\_results = 1  
 how\_to = search\_wikihow(how, max\_results)  
 assert len(how\_to) == 1  
 how\_to[0].print()  
 speak(how\_to[0].summary)  
 except Exception as e:  
 speak("sorry sir, i am not able to find this!")  
  
 #to check battery of system  
 elif "how much power" in query or "battery" in query:  
 battery = psutil.sensors\_battery()  
 percentage = battery.percent  
 speak(f"sir our system have {percentage} percent battery")  
 if percentage>=65:  
 speak("we have enough power to continue our work")  
 elif percentage>=30 and percentage<65:  
 speak("we should connect our system to charging point")  
 elif percentage>=15 and percentage<30:  
 speak("we don't have enough power, please connect to charging")  
 elif percentage<15:  
 speak("we have very low power, connect to charger or i am going to sleep very soon!")  
  
 #to check internet speed  
 elif "internet speed" in query:  
 speak("i am calculating sir, please wait for sometime")  
 st = speedtest.Speedtest()  
 dl = st.download()  
 up = st.upload()  
 dl2 = ((dl/8)/1024)  
 up2 = ((up/8)/1024)  
 speak(f"sir we have {dl2} KB per second download speed and {up2} KB per second upload speed")  
  
  
  
 #to find address  
 elif "where i am" in query or "where we are" in query:  
 speak("wait sir, let me check")  
 try:  
 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()  
 #print(geo\_data)  
 city = geo\_data['city']  
 #state = geo\_data['state']  
 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 our location")  
 pass  
  
  
 #to check instagram profile  
 elif "instagram profile" in query:  
 speak("sir please enter the user name correctly")  
 name = input("enter username here:")  
 webbrowser.open(f"www.instagram.com/{name}")  
 speak(f"sir here is the profile of the user {name}")  
 time.sleep(5)  
 speak("sir, would you like to download profile picture of this account?")  
 condition = takecommand().lower()  
 if "yes" in condition:  
 mod = instaloader.Instaloader()  
 mod.download\_profile(name, profile\_pic\_only=True)  
 speak("i am done sir, profile picture is saved in our main folder. now i am ready for next command")  
 else:  
 pass  
  
 #to take screenshot  
 elif "take screenshot" in query:  
 speak("sir, please tell me the name for this screenshot file")  
 name = takecommand().lower()  
 speak("please sir hold the screen for few seconds, i am taking screenshot")  
 time.sleep(3)  
 img = pyautogui.screenshot()  
 img.save(f"{name}.png")  
 speak("i am done sir, the screenshot is saved in our main folder. now i am ready for the next command")  
  
  
 #to read PDF file  
 elif "read pdf" in query:  
 pdf\_reader()  
  
 #to do calculation  
 elif "do some calculation" in query or "can you calculate" in query:  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 speak("say what 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, #plus  
 '-': operator.sub, #minus  
 'x': operator.mul, #multiplied by  
 'divided': operator.\_\_truediv\_\_, #divided  
 }[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())))  
  
 #conversations  
 elif "hello" in query or "hey" in query:  
 speak("hello sir, may i help you with something?")  
  
 elif "how are you" in query:  
 speak("i am fine sir, what about you?")  
  
 elif "good" in query or "fine" in query:  
 speak("that's great to hear from you!")  
  
 elif "thank you" in query or "thanks" in query:  
 speak("it's my pleasure sir.")  
  
 elif "you can sleep" in query or "sleep now" in query:  
 speak("ok sir, i am going to sleep you can call me anytime")  
 break  
  
 elif "goodbye" in query or "that's all" in query:  
 speak("thanks for using me sir, have a good day")  
 sys.exit()  
 break  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 permission = takecommand().lower()  
 if "wake up" in permission or "jarvis" in permission:  
 TaskExecution()  
  
 elif "goodbye" in permission or "that's all" in permission:  
 speak("thanks for using me sir, have a good day")  
 sys.exit()