23_Voice_Assistant.py

Code –

```
import datetime
import pyttsx3
import speech_recognition
import wikipedia
import webbrowser
import os
import pywhatkit
import pywikihow
import winsound
import win10toast
import pyjokes
import PyDictionary
import googletrans
import smtplib
import validate_email
import pyautogui
import time
import speedtest
import requests
import json
import wolframalpha
from selenium import webdriver
from webdriver_manager.microsoft import EdgeChromiumDriverManager

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
engine.setProperty('rate', 200)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def wishme():
    hour = int(datetime.datetime.now().hour)
    if 0 < hour <= 12:
        print("Good Morning!")
        speak("Good Morning!")
    elif 12 < hour < 17:
        print("Good Afternoon!")
        speak("Good Afternoon!")
    elif 17 < hour < 20:
        print("Good Evening!")
        speak("Good Evening!")
    else:
        print("Good Night!")
        speak("Good Night!")
    print("I am Grey. How may I help you?")
    speak("I am Grey. How may I help you?")


def Command():
    r = speech_recognition.Recognizer()
    with speech_recognition.Microphone() as source:
        print("\nListening...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print("You said:", query, "\n")

    except Exception as e:
        print("Say that again...")
        speak("Say that again...")
        return "none"
    return query


def sendEmail(to, content=""):
    try:
        from1 = "yadav.abhi.1520@gmail.com"
        f = open("psd.txt", "r")
        pwd = f.read()
        f.close()
        server = smtplib.SMTP("smtp.gmail.com", 587)
        server.ehlo()
        server.starttls()
        server.login(from1, pwd)
        server.sendmail(from1, to, content)
        server.close()

    except Exception as e:
        print(e)


def translate1(sentence, language):
    lang1 = 'en'
```

```
if language == 'Afrikaans':
    lang1 = 'af'
elif language == 'Irish':
    lang1 = 'ga'
elif language == 'Albanian':
    lang1 = 'sq'
elif language == 'Italian':
    lang1 = 'it'
elif language == 'Arabic':
    lang1 = 'ar'
elif language == 'Japanese':
    lang1 = 'ja'
elif language == 'Azerbaijani':
    lang1 = 'az'
elif language == 'Kannada':
    lang1 = 'kn'
elif language == 'Basque':
    lang1 = 'eu'
elif language == 'Korean':
    lang1 = 'ko'
elif language == 'Bengali':
    lang1 = 'bn'
elif language == 'Latin':
    lang1 = 'la'
elif language == 'Belarusian':
    lang1 = 'be'
elif language == 'Latvian':
    lang1 = 'lv'
elif language == 'Bulgarian':
    lang1 = 'bg'
elif language == 'Lithuanian':
    lang1 = 'lt'
elif language == 'Catalan':
    lang1 = 'ca'
elif language == 'Macedonian':
    lang1 = 'mk'
elif language == 'Chinese':
    lang1 = 'zh-cn'
elif language == 'Maltese':
    lang1 = 'ms'
elif language == 'Croatian':
    lang1 = 'hr'
elif language == 'Norwegian':
    lang1 = 'no'
elif language == 'Czech':
    lang1 = 'cs'
elif language == 'Persian':
    lang1 = 'fa'
```

```
elif language == 'Danish':
    lang1 = 'da'
elif language == 'Polish':
    lang1 = 'pl'
elif language == 'Dutch':
    lang1 = 'nl'
elif language == 'Portuguese':
    lang1 = 'pt'
elif language == 'English':
    lang1 = 'en'
elif language == 'Romanian':
    lang1 = 'lt'
elif language == 'Esperanto':
    lang1 = 'eo'
elif language == 'Russia':
    lang1 = 'ru'
elif language == 'Estonian':
    lang1 = 'et'
elif language == 'Serbian':
    lang1 = 'sr'
elif language == 'Filipino':
    lang1 = 'tl'
elif language == 'Slovak':
    lang1 = 'sk'
elif language == 'Finnish':
    lang1 = 'fi'
elif language == 'Slovenian':
    lang1 = 'sl'
elif language == 'French':
    lang1 = 'fr'
elif language == 'Spanish':
    lang1 = 'es'
elif language == 'Galician':
    lang1 = 'gl'
elif language == 'Swahili':
    lang1 = 'sw'
elif language == 'Georgian':
    lang1 = 'gl'
elif language == 'Swedish':
    lang1 = 'sv'
elif language == 'German':
    lang1 = 'de'
elif language == 'Tamil':
    lang1 = 'ta'
elif language == 'Greek':
    lang1 = 'el'
elif language == 'Telugu':
    lang1 = 'te'
    elif language == 'Gujarati':
        lang1 = 'gu'
    elif language == 'Thai':
        lang1 = 'th'
    elif language == 'Haitian Creole':
        lang1 = 'ht'
    elif language == 'Turkish':
        lang1 = 'tr'
    elif language == 'Hebrew':
        lang1 = 'iw'
    elif language == 'Ukrainian':
        lang1 = 'hu'
    elif language == 'Hindi':
        lang1 = 'hi'
    elif language == 'Urdu':
        lang1 = 'ur'
    elif language == 'Hungarian':
        lang1 = 'hu'
    elif language == 'Vietnamese':
        lang1 = 'vi'
    elif language == 'Icelandic':
        lang1 = 'is'
    elif language == 'Welsh':
        lang1 = 'cy'
    elif language == 'Indonesian':
        lang1 = 'id'
    elif language == 'Yiddish':
        lang1 = 'yi'

    translator = googletrans.Translator()
    res = translator.translate(sentence, dest=lang1)
    return res.text, res.pronunciation


if __name__ == '__main__':
    #wishme()
    while True:
        query = Command().lower()

        if 'who are you' in query:
            print("I am Your Voice Assistant")
            speak("I am Your Voice Assistant")

        elif 'hello' in query:
            print("Hello, I'm Grey, How may I help you?")
            speak("Hello, I'm Grey, How may I help you?")

        elif 'how are you' in query:
            print("I am fine")
            speak("I am fine")

        elif 'how old are you' in query:
            print("I am 2 days old")
            speak("I am 2 days old")

        elif 'your name' in query:
            print("My name is Grey")
            speak("My name is Grey")

        elif 'how was your day' in query:
            print("Great What about you?")
            speak("Great What about you?")

        elif 'tell me about yourself' in query:
            print("Hello, myself Grey!! I am your voice assistant. I can perform several task. How may I
help you?")
            speak("Hello, myself Grey!! I am your voice assistant. I can perform several task. How may I
help you?")

        elif 'wikipedia' in query:
            print("Searching Wikipedia...")
            speak("Searching Wikipedia...")
            query = query.replace("wikipedia", "").replace("search ", "").replace("it ", "").replace("on", "")
            result = wikipedia.summary(query, sentences=2)
            print("According to Wikipedia...")
            speak("According to Wikipedia...")
            print(result)
            speak(result)

        elif 'meaning' in query:
            word = query.replace("what ", "").replace("is ", "").replace("the ", "").replace("of", "")\
                .replace("meaning", "").strip()
            res = PyDictionary.PyDictionary.meaning(word)
            print(str(res['Noun'][0]).capitalize())
            speak(res['Noun'][0])

        elif 'synonym' in query:
            word = query.replace("what ", "").replace("is ", "").replace("the ", "").replace("of", "")\
                .replace("synonym", "").strip()
            res = PyDictionary.PyDictionary.synonym(word)
            print(str(res[0]).capitalize(), ", ", end="")
            print(str(res[1]).capitalize())
            speak(res[0:2])

        elif 'antonym' in query:
            word = query.replace("what ", "").replace("is ", "").replace("the ", "").replace("of", "")\
                .replace("antonym", "").strip()
            res = PyDictionary.PyDictionary.antonym(word)
            print(str(res[0]).capitalize(), ", ", end="")
            print(str(res[1]).capitalize())
            speak(res[0:2])

        elif 'open google chrome' in query:
            print("Opening chrome")
            speak("Opening chrome")
            pyPath = "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe"
            os.startfile(pyPath)

        elif 'open browser' in query or 'edge' in query or 'microsoft edge' in query:
            print("Opening browser")
            speak("Opening browser")
            pyautogui.hotkey('win', '1')

        elif 'open google' in query:
            print("Opening google")
            speak("Opening google")
            webbrowser.open("www.google.com")

        elif 'open youtube' in query:
            print("Opening youtube")
            speak("Opening youtube")
            webbrowser.open("www.youtube.com")

        elif 'open gmail' in query:
            print("Opening Gmail")
            speak("Opening Gmail")
            webbrowser.open("https://mail.google.com/mail/u/0/?tab=rm&ogbl#inbox")

        elif 'screenshot' in query:
            print("How you want to take screenshot? Full Screen or Specific Screen")
            speak("How you want to take screenshot? Full Screen or Specific Screen")
            ans = Command().lower()
            print("taking screenshot")
            speak("taking screenshot")

            if 'full screen' in ans:
                pyautogui.hotkey('win', 'prtsc')
            else:
                pyautogui.hotkey('win', 'shift', 's')

        elif 'search' in query:
            print("Searching")
            speak("Searching")
            query = query.replace("search", "")
```

```
            pywhatkit.search(query)

        elif 'tell me about' in query:
            print("Searching")
            speak("Searching")
            query = query.replace("tell me about ", "")
            result = pywhatkit.info(query, 2)
            speak(result)

        elif 'who is' in query:
            print("Searching")
            speak("Searching")
            query = query.replace("who is ", "")
            result = pywhatkit.info(query, 2)
            speak(result)

        elif'what is' in query:
            print("Searching")
            speak("Searching")
            result = pywhatkit.info(query, 2)
            speak(result)

        elif 'how to reach' in query or 'map' in query:
            location = query.replace("how to reach", "").replace("map", "").replace("in", "")
            driver = webdriver.Edge(EdgeChromiumDriverManager().install())
            driver.get("https://www.google.com/maps/@18.966408,73.0144436,16z")
            time.sleep(2)
            place = driver.find_element_by_class_name("tactile-searchbox-input")
            place.send_keys(location)
            submit = driver.find_element_by_xpath(
                "/html/body/jsl/div[3]/div[9]/div[3]/div[1]/div[1]/div[1]/div[2]/div[1]/button")
            submit.click()

            time.sleep(3)
            direction = driver.find_element_by_xpath(
                "/html/body/jsl/div[3]/div[9]/div[8]/div/div[1]/div/div/div[4]/div[1]/div/button/img")
            direction.click()

            time.sleep(3)
            find = driver.find_element_by_xpath(

"/html/body/jsl/div[3]/div[9]/div[3]/div[1]/div[2]/div/div[3]/div[1]/div[1]/div[2]/div/div/input")
            time.sleep(3)

            search = driver.find_element_by_xpath(

"/html/body/jsl/div[3]/div[9]/div[3]/div[1]/div[2]/div/div[3]/div[1]/div[1]/div[2]/div/div/input")
            search.click()
        elif 'how to' in query:
            print("Searching")
            speak("Searching")
            res = pywikihow.search_wikihow(query, 1)
            assert len(res) == 1
            res[0].print()
            speak(res[0].summary)

        elif 'play' in query or 'music' in query or 'youtube' in query:
            song = query.replace("play", "").replace("music", "").replace(" on ", "").replace("youtube", "")
            print("Playing" + song)
            speak("Playing" + song)
            pywhatkit.playonyt(song)

        elif 'song' in query:
            song = query.replace("play", "")
            song1 = song.replace("song", "")
            print("Playing" + song1)
            speak("Playing" + song1)
            pywhatkit.playonyt(song1)

        elif 'reminder' in query or 'remind me' in query:
            r = query.replace("set a reminder for ", "").replace("remind me to ", "").replace("reminder",
"")\
                .replace("remind", "").replace(" p.m.", "").replace(" a.m.", "").replace("am",
"").replace("pm", "")
            mes = r.split(" at ")
            reminder = mes[0].capitalize()
            t = mes[1]
            time = t.split(":")
            print("Reminder Set")
            speak("Reminder Set")
            #print(time)
            if len(t) == 1:
                h = int(time[0])
                m = 0
            elif len(t) == 2:
                if '12' in query and 'a.m.' in query:
                    h = 0
                    m = 0
                else:
                    h = int(time[0])
                    m = 0
            else:
                if '12' in query and 'a.m.' in query:
                    h = 0
                    m = int(time[1])
                else:
                    h = int(time[0])
                    m = int(time[1])
            #print(h, m)
            if 'p.m.' in query:
                h = h + 12
            while True:
                if h == datetime.datetime.now().hour and m == datetime.datetime.now().minute:
                    print("Reminder")
                    speak("Reminder")
                    notification = win10toast.ToastNotifier()
                    notification.show_toast("Reminder", reminder, duration=3)
                    break

        elif 'alarm' in query or 'wake me' in query:
            t = query.replace("set an alarm for ", "").replace("wake me up at ", "")\
                .replace(" p.m.", "").replace(" a.m.", "").replace("am", "").replace("pm", "")
            time = t.split(":")
            print("Alarm set")
            speak("Alarm set")
            #print(t, len(t))
            #print(time)
            if len(t) == 1:
                h = int(time[0])
                m = 0
            elif len(t) == 2:
                if '12' in query and 'a.m.' in query:
                    h = 0
                    m = 0
                else:
                    h = int(time[0])
                    m = 0
            else:
                if '12' in query and 'a.m.' in query:
                    h = 0
                    m = int(time[1])
                else:
                    h = int(time[0])
                    m = int(time[1])
            #print(h, m)
            if 'p.m.' in query:
                h = h + 12
            while True:
                if h == datetime.datetime.now().hour and m == datetime.datetime.now().minute:
                    print("Time to wake up")
                    speak("Time to wake up")
                    notification = win10toast.ToastNotifier()
                    notification.show_toast("Alarm", "Time to wake up", duration=3)
```

```
                    winsound.Beep(2000, 2000)
                    break

        elif 'joke' in query:
            joke = pyjokes.get_joke()
            print(joke)
            speak(joke)
            print("Ha Ha Ha. I hope you find it funny")
            speak("Ha Ha Ha. I hope you find it funny")

        elif 'open calendar' in query or 'show calendar' in query:
            print("Opening calendar")
            speak("Opening calendar")
            os.startfile("C:\\Users\\yadav\\Downloads\\Calendar - Shortcut")

        elif 'open notepad' in query:
            print("Opening notepad")
            speak("Opening notepad")
            os.startfile("C:\\ProgramData\\Microsoft\\Windows\\Start
Menu\\Programs\\Accessories\\Notepad")

        elif 'open calculator' in query:
            print("Opening calculator")
            speak("Opening calculator")
            cal_directory = "C:\\Windows\\System32\\calc.exe"
            os.startfile(cal_directory)

        elif 'open camera' in query:
            print("Opening camera")
            speak("Opening camera")
            os.startfile("microsoft.windows.camera:")

        elif 'open this pc' in query:
            print("Opening This PC")
            speak("Opening This PC")
            pyautogui.hotkey('win', 'e')

        elif 'file explorer' in query:
            print("Opening File Explorer")
            speak("Opening File Explorer")
            pyautogui.hotkey('win', 'e')

        elif 'open mail' in query or 'outlook' in query:
            print("Opening mail")
            speak("Opening mail")
            os.startfile("C:\\Users\\yadav\\Downloads\\Mail - Shortcut")

        elif 'cmd' in query or 'command prompt' in query:
```

```
    print("Opening Command Prompt")
    speak("Opening Command Prompt")
    os.startfile("C:\\WINDOWS\\system32\\cmd")

elif "setting" in query:
    print("Opening Settings")
    speak("Opening settings")
    pyautogui.hotkey('win', 'i')

elif 'the time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    print(strTime)
    speak(f"The time is {strTime}")

elif 'date' in query or "today's date" in query:
    strDate = datetime.date.today().strftime("%d/%m/%Y")
    print(strDate)
    speak(f"Today's date is {strDate}")

elif 'the day' in query:
    day = datetime.datetime.today().strftime("%A")
    print(f"Today is {day}")
    speak(f"Today is {day}")

elif 'send an email' in query or "send mail" in query or "send email" in query or "email" in query:
    try:
        print("To whom do i send? Please provide Email Id.")
        speak("To whom do i send? Please provide Email Id.")
        people = input("Email ID:")
        if validate_email.validate_email(people):
            print("Email ID is valid")
            print("What should I send?")
            speak("What should I send?")
            print("You want to speak or write the mail? ")
            speak("You want to speak or write the mail? ")
            while True:
                sw = Command()
                if sw != "none":
                    if "speak" in sw:
                        content = Command()
                        if content != "none":
                            break
                    elif "write" in sw or "right" in sw:
                        content = input("Message:")
                        break
                    break
            print("Are you sure?")
            speak("Are you sure?")
```

```
                    ans = Command().lower()
                    while True:
                        if 'yes' in ans:
                            sendEmail(people, content)
                            print("Email has been sent")
                            speak("Email has been sent")
                            break
                        elif 'no' in ans:
                            print("Email discarded")
                            speak("Email discarded")
                            break
                        else:
                            print("Couldn't understand")
                            speak("Couldn't understand")
                            break
                else:
                    print("Email ID is invalid")

            except Exception as e:
                print(e)

        elif 'send a message' in query or 'message' in query:
            print("Whom do you want to send message?")
            speak("Whom do you want to send message? Enter Number")
            num = input("Enter Number: ")
            print("What message you want to send?")
            speak("What message you want to send?")
            msg = input("Enter Message: ")
            now = datetime.datetime.now()

            h = int(now.strftime("%H"))
            m = int(now.strftime("%M"))
            pywhatkit.sendwhatmsg(num, msg, h, m+1)

        elif 'news' in query or 'headlines' in query:
            print("Getting Headlines:")
            speak("Getting Headlines:")
            r = requests.get("https://newsapi.org/v2/top-
headlines?country=in&apiKey=7f7c801a2dc94113a53b4aca28a4103c")
            news = json.loads(r.content)
            for i in range(10):
                print("News", i + 1, ":", news['articles'][i]['title'])
                #print(news['articles'][i]['content'])
            for i in range(2):
                speak("News" + str(i + 1) + ":" + news['articles'][i]['title'])

        elif 'speed test' in query:
            print("Testing please wait....")
```

```
      speak("Testing please wait....")
      speed = speedtest.Speedtest()
      download = str(round(speed.download()/(1025*1025), 2))
      upload = str(round(speed.upload()/(1025*1025), 2))
      print("Download speed is", download, "Mbps")
      speak("Download speed is " + download + "Mbps")
      print("Upload speed is ", upload, "Mbps")
      speak("Upload speed is " + upload + "Mbps")

    elif 'translate' in query or 'translation' in query or ' in ' in query:
      sentence = query.replace("translate", "").replace("this", "").replace("translation", "")
      sentence1 = sentence.split(" ")
      #print(sentence1)
      index = sentence1.index('in') + 1
      sentence = sentence.replace(" in", "").replace(sentence1[index], "").strip()
      lang = sentence1[index].capitalize()
      #print(lang)
      text, pronunciation = translate1(sentence, lang)
      print(text)
      speak(pronunciation)

    elif 'call' in query:
      print("Whom do you wanna call?")
      speak("Whom do you wanna call?")
      num = input("Enter Phone Number: ")
      time.sleep(1)
      pyautogui.press('win')
      time.sleep(2)
      pyautogui.typewrite('Your Phone')
      pyautogui.press('enter')
      time.sleep(3)
      pyautogui.typewrite(num)
      pyautogui.press('enter')

    elif 'weather' in query:
      def weather_data(query):
        res = requests.get('http://api.openweathermap.org/data/2.5/weather?' + query +
'&APPID=b35975e18dc93725acb092f7272cc6b8&units=metric')
        return res.json()

      def print_weather(result, city):
        # print(result)
        print("{}'s Temperature is {}°C ".format(city, result['main']['temp']))
        speak("{}'s temperature is {}°Celsius ".format(city, result['main']['temp']))
        print("Wind speed is {} m/s".format(result['wind']['speed']))
        speak("Wind speed: {} meter per second".format(result['wind']['speed']))
        print("Weather is {}".format(result['weather'][0]['main']))
        speak("Weather is {}".format(result['weather'][0]['main']))
```

```
                print("Humidity is {}".format(result['main']['humidity']))
                speak("Humidity is {}".format(result['main']['humidity']))
                print("Description is {}".format(result['weather'][0]['description']))
                speak("Description is {}".format(result['weather'][0]['description']))

            def main():
                print("Which city do you want?")
                speak("Which city do you want?")
                while True:
                    city = Command()
                    if city != "none":
                        break

                try:
                    query = 'q=' + city
                    w_data = weather_data(query)
                    print_weather(w_data, city)
                    print()
                except:
                    print('City name not found...')
                    speak('City name not found...')
            if __name__ == '__main__':
                main()

        elif 'bye' in query or 'exit' in query or 'goodbye' in query:
            print("Bye Bye!! Have a nice day")
            speak("bye bye!! Have a nice day")
            break

        elif query != 'none':
            try:
                client = wolframalpha.Client('EGU8HX-YQ32EJWTV4')
                q = query.replace("plus", "+").replace("minus", "-").replace("into", "*").replace("divided
by", "/")
                res = client.query(q)
                output = next(res.results).text
                print(f'Answer is {output}')
                speak(f'Answer is {output}')
            except Exception as e:
                print("Sorry I can't do that")
                speak("Sorry I can't do that")
```