AI Desktop Voice Assistant

What can this A.I. assistant do for you?

1)It can send emails on your behalf.
2)It can play music for you.
3)It can do Wikipedia searches for you.
4)It is capable of opening websites like Google, Youtube, etc., in a web browser.

Defining Speak Function

The first and foremost thing for an A.I. assistant is that it should be able to speak.we will make a function called speak(). This function will take audio as an argument, and then it will pronounce it.

```
def speak(audio):
    pass      #For now, we will write the conditions later.
```

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak() function we made. We are going to install a module called pyttsx3.

What is pyttsx3?

- A python library that will help us to convert text to speech. In short, it is a text-to-speech library.
- It works offline.

Installation:

```
pip install pyttsx3
```

USAGE:

```
Import pyttx3
engine = pyttsx3.init('sapi5')
voices= engine.getProperty('voices') #getting details of current voice
engine.setProperty('voice', voice[0].id)
```

What is sapi5?

- Microsoft developed speech API.
- Helps in synthesis and recognition of voice.

What is voiceid?

- Voice id helps us to select different voices.
- voice[0].id = Male voice
- voice[1].id = Female voice

1.1 Writing Our speak() Function :

We made a function called speak() at the starting. Now, we will write our speak() function to convert our text to speech.

```
def speak(audio):
        engine.say(audio)
        engine.runAndWait()#Without this command, speech will not be audible to us.
```

Creating Our main() function:

We will create a main() function, and inside this main() Function, we will call our speak function.

```
If name==main():
        speak('hello world')
```

Whatever you will write inside this speak() function will be converted into speech

1.2 Defining Wish me Function :

→ Now, we will make a wishme() function that will make our assistant wish or greet the user according to the time of computer or pc.

→ To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program by:

```
->Import datetime
```

Now, let's start defining the wishme() function:

```
Def wishme():
        hour = int(datetime.datetime.now().hour)
```

Here, we have stored the current hour or time integer value into a variable named hour. Now, we will use this hour value inside an if-else loop.

1.3 Defining Take command Function :

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

Before defining the takeCommand() function, we need to install a module called speechRecognition. Install this module by:

Installation: pip install speechrecognition

Now importing it:

```
Import  speechrecognition as sr
```

Let's start coding the takeCommand() function :

```
def takeCommand():
   #It takes microphone input from the user and returns string output
```

Let's start coding the takeCommand() function :

```
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)
```

To handle the errors we need to use exceptional handling

```
    try:
    print("Recognizing...")
    query = r.recognize_google(audio, language='en-in') #Using google for voice recognition.
    print(f"User said: {query}\n")  #User query will be printed.
  except Exception as e:
    print("Say that again please...")  #Say that again will be printed in case of improper voice
    return "None" #None string will be returned
  return query
```

1.4 Now, we will develop logic for different commands

1.4.1 Defining Task 1: To search something on Wikipedia

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module :

```
Pip install wikipedia
```

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

```
if __name__ == "__main__":
  wishMe()
  while True:
  # if 1:
    query = takeCommand().lower() #Converting user query into lower case

    # Logic for executing tasks based on query
    if 'wikipedia' in query:  #if wikipedia found in the query then this block will be executed
      speak('Searching Wikipedia...')
      query = query.replace("wikipedia", "")
      results = wikipedia.summary(query, sentences=2)
      speak("According to Wikipedia")
      print(results)
```

```
        speak(results)
```

In the above code, we have used an if statement to check whether Wikipedia is in the user's search query or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

1.4.2 Defining Task 2: To open YouTube site in a web-browser

To open any website, we need to import a module called webbrowser. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statemen

```
elif  “open youtube in query”:

        webbrowser.open("youtube.com")
```

Here, we are using an elif loop to check whether Youtube is in the user's query. Let' suppose the user gives a command as , open youtube." So, open youtube will be in the user's query, and the elif condition will be true

1.4.3 To open Google site in a web-browser

```
elif  “open google in query:
        webbrowser.open(“google.com”)
```

We are opening Google in a web-browser by applying the same logic that we used to open youtube.

1.4.4 To play music

To play music, we need to import a module called os. Import this module directly with an import statement.

```
 Import os
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]))
```

In the above code, we first opened our music directory and then listed all the songs present in the directory with the os module's help. With the help of os.startfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, ASSISTENT. will play any random song from the song directory.

1.4.5 To know the current time

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

In the above, code we are using the datetime() function and storing the current or live system time into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into speech.

1.4.6 To send Email

-->To send an email, we need to import a module called smtplib.
What is smtplib?

- Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. An instance method called sendmail is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:
- The sender: Email address of the sender.
- The receiver: T Email of the receiver.
- The *message:* A string message which needs to be sent to one or more than one recipient.

Defining Send email function :

We will create a sendEmail() function, which will help us send emails to one or more than one recipient.

```
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()
```

In the above code, we are using the SMTP module, which we have already discussed above.

Note: Do not forget to *'enable the less secure apps'* feature in your Gmail account. Otherwise, the sendEmail function will not work properly.

Calling sendEmail() function inside the main() function:

```
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")
```

We are using the try and except block to handle any possible error while sending emails.

CONCLUSION:

With this, you have successfully made your very first virtual assistant.

THE END