# Virtual Assistant Using Python

Have you ever wondered how cool it would be to have your own A.I. assistant? Imagine how easier it would be to send emails without typing a single word, doing Wikipedia searches without opening web browsers, and performing many other daily tasks like playing music with the help of a single voice command.

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. etc. Now in our voice assistance system, it can be used  to play music and videos,send emails,open any applications, show time and as search tool. This project works on voice input and give output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

Virtual assistants are software programs that help you ease your day to day tasks, such as showing weather reports, creating remainders, making shopping lists etc. They can take commands via text (online chatbots) or by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana.

This system is designed to be used efficiently on desktops. Personal assistants software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user.

## Overview of  Program

### ❖ What can this A.I. Assistant do for you?

✓It can send emails on your behalf.

✓It can play music for you.

✓It can do Wikipedia searches for you.

✓It is capable of opening websites like Google, YouTube, ChatGPT etc., in a web browser.

✓It is capable of opening your code editor or IDE with a single voice command.

### ❖ Related Work

Each company developer of the intelligent assistant applies his own specific methods and approaches for development, which in turn affects the final product. One assistant can synthesize speech more qualitatively, another can more accurately and without additional explanations and corrections perform tasks, others can perform a narrower range of tasks, but most accurately and as the user wants. Obviously, there is no universal assistant who would perform all tasks equally well..

### ❖ Proposed Plan Of Work

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

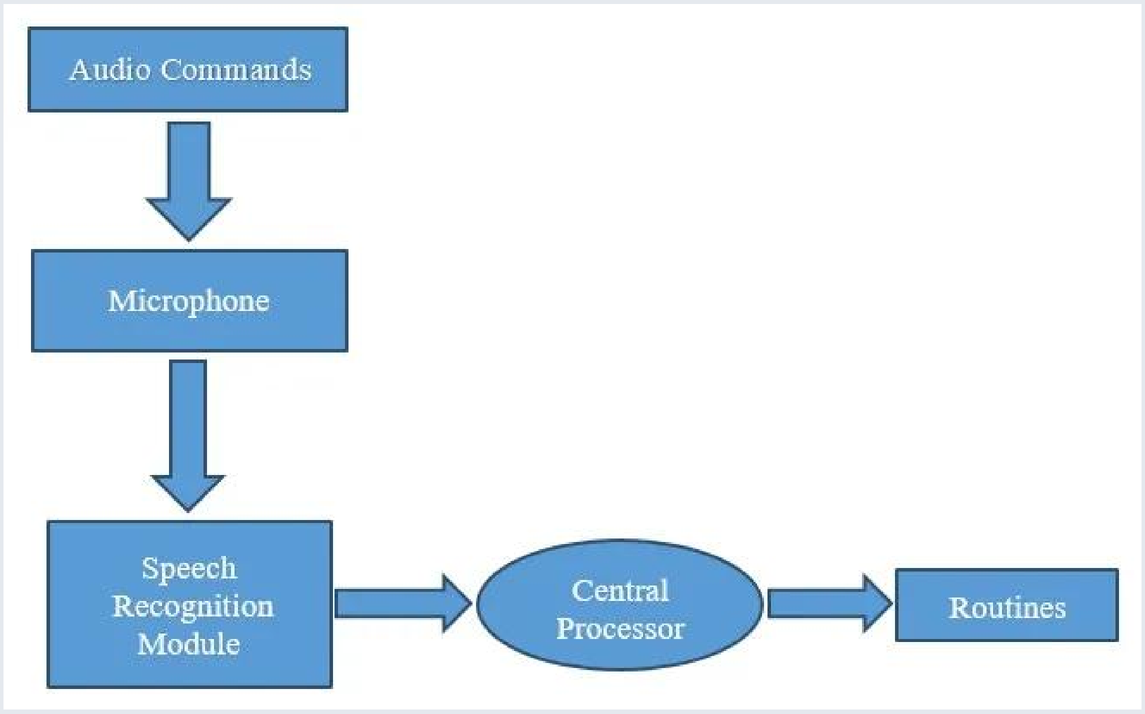

***Fig.2.*** *Basic Workflow*

Fig.2 shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

But, the complexities don't stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

# Code Implementation

## ❖ Starting VS Code

I am going to use the VS Code IDE in this project. Feel free to use any other IDE you are comfortable with. Start a new project and make a file called “virtual assistance.py”.

## ❖ Defining Speak Function

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

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.

After successfully installing pyttsx3, import this module into your program.

## ❖ Writing Our speak() Function :

We made a function called speak() at the starting of this tutorial. 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.
```

## ❖ Defining Wish me Function :

Now, we will make a wishme() function that will make our J.A.R.V.I.S. 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:

```
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")
```

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.

## ❖ 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.

```
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)
```

We have successfully created our takeCommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

```
  try:
        print("Recognizing...")
       query = r.recognize_google(audio, language='en-in')  #Using google for
voice recognition.

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

```
  except Exception as e:
        # print(e)
       print("Say that again please...")   #Say that again will be printed in case of
improper voice
        return "None"
  return query
```

## ◆ <u>Coding logic of Jarvis</u>

Now, we will develop logic for different commands such as Wikipedia searches, playing music, etc.

- **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 **:**

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()
     # 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)
```

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.

- **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 statement.

**Code:**

```
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 "J.A.R.V.I.S., open youtube." So, open youtube will be in the user's query, and the elif condition will be true.

- **Defining Task 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.

- **Defining Task 4: To play music**

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

```
 elif 'play music' in query:
        music_dir = "C:\\Users\\Deepak\\Music\\Music"
        songs = os.listdir(music_dir)
        #print(songs)
        value=random.randint(0,50)
        os.startfile(os.path.join(music_dir, songs[value]))
```

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, J.A.R.V.I.S. will play any random song from the song directory.

- **Defining Task 5: To play Videos**

```
    elif 'play video' in query:
        video_dir = "D:\\Videos\\#Songs Videos"
        videos = os.listdir(video_dir)
        value=random.randint(0,100)
        os.startfile(os.path.join(video_dir, videos[value]))
```

We are playing videos from our files by applying the same logic that we used to play music.

- **Defining Task 6: To know the current time**

```
elif 'the time' in query:
        strTime = datetime.datetime.now().strftime("%I:%M:%S %p")
        print(strTime)
        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.

- **Defining Task 7: To open the VS Code Program**

```
    elif 'open code' in query:
        codePath = "C:\\Users\\deepak\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
        os.startfile(codePath)
```

To open the VS Code or any other application, we need the code path of the application.

➢ **Steps to get the code path of the application:**

- ✓ **Step 1:** Open the file location.
- ✓ **Step 2:** Right-click on the application and click on properties.
- ✓ **Step 3:** Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, I am saving the target into a variable called codePath, and then we are using the os module to open the application.

- **Defining Task 8: To open Google Chrome**

```
elif 'open chrome' in query:
    codePath = "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe"
    os.startfile(codePath)
```

We are opening Google Chrome application by applying the same logic that we used to open Vs Code.

- **Defining Task 9: To send Email**

To send an email, we need to import a module called smtplib.

❖ **Defining Send email function :**

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

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

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

```python
def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('my_username@gmail.com', 'my password')
    server.sendmail('my_username@gmail.com', to, content)
    server.close()
```

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

```
elif 'email to deepak' in query:
    try:
        speak("What should I say?")
        content = takeCommand()
        to = "sender_username@gmail.com"
        sendEmail(to, content)
        #print("Email has been sent:",content)
        speak("Email has been sent!")
    except Exception as e:
        print("\nSORRY,MAIL NOT SENT")
        speak("Sorry deepak bhai. I am not able to send this email")
```

● **Defining Task 10: To Exit**

```
elif 'exit' in query:
       speak(“Yes sir”)
       exit()
```

This command will simply take you out of the program.

## ❖ Recapitulate

- ✓ First of all, we have created a wishme() function that gives the greeting functionality according to our A.I system time.
- ✓ After wishme() function, we have created a takeCommand() function, which helps our A.I to take command from the user. This function is also responsible for returning the user's query in a string format.
- ✓ We developed the code logic for opening different websites like google, youtube, and stack overflow.
- ✓ Developed code logic for opening VS Code or any other application.
- ✓ At last, we added functionality to send emails.

## ➢ Is it an A.I.?

Many people will argue that the virtual assistant that we have created is not an A.I, but it is the output of a bunch of the statement. But, if we look at the fundamental level, the sole purpose of A.I develop machines that can perform human tasks with the same effectiveness or even more effectively than humans.

It is a fact that our virtual assistant is not a very good example of A.I., but it is an A.I.!

## Execution of Program

```
Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Deepak> & C:/Users/Deepak/AppData/Local/Microsoft/WindowsApps/python3.
9.exe "c:/Users/Deepak/Documents/Python Report/virtual assistant.py"

Listening...

Recognizing...

User said: play music

Listening...

Recognizing...

User said: Amitabh Bachchan Wikipedia

Amitabh Bachchan (pronounced [əmɪˈt̪ɑːbʱ ˈbətːʃən]; born Amitabh Srivastava; 11 October

1942) is an Indian actor, film producer, television host, occasional playback singer and forme
r politician known for his work in Hindi cinema. He is regarded as one of the most influential
 actors in the history of Indian cinema.

Listening...

Recognizing...

User said: what's the time

04:01:33 PM

Listening...

Recognizing...

User said: open YouTube

Listening...

Recognizing...

User said: exit

PS C:\Users\Deepak>
```

## Conclusion

Python is a real-world programming language used in the commercial world. Quicker to learn, contains many libraries. Python trainers few in number. It is the fastest-growing programming language. By choosing python as a career, you can also be a part of the trending technologies. Without python language, we can't deliver exceptional results.

I believe the trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language:

- ✓ It is Free (as in both cost and source code).
- ✓ It is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated;
- ✓ It is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills;
- ✓ It is a real-world programming language that can be and is used in academia and the commercial world;
- ✓ It appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied;
- ✓ and most importantly, its clean syntax offers increased understanding and enjoyment for students