

# Voice Assistant Made Easy

Edara Nithin

<sup>1</sup>Research Scholar & Associate Professor, Computer Science and Engineering, UoM, Mysore, Karnataka, India

<sup>2</sup>Student, Department of Computer Science and Engineering, Karnataka, India

## To Cite this Article

Edara Nithin, "Voice Assistant Made Easy", *International Journal for Modern Trends in Science and Technology*, 6(8S): 102-107, 2020.

## Article Info

Received on 16-July-2020, Revised on 15-August-2020, Accepted on 25-August-2020, Published on 28-August-2020.

## ABSTRACT

*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. To call any technology that makes our lives easier by one name is almost impossible. There are a variety of terms that refer to agents that can perform tasks or services for an individual, and they are almost interchangeable. They differ mainly based on how we interact with the technology, the app, or a combination of both. The voice assistant will be helpful for all of us to make our work more easier by using the assistant which we have specified.*

**KEYWORDS:** Voice Assistant 1, Smart Assistant 2, Digital Assistant 3, A.I.Assistance

## I. INTRODUCTION

Here, I made a voice assistant using python.

- What can we do with this A.I. assistant?
- we can send E-mails using this.
- we can play music using this .
- we can do wikipedia searches using this.
- we can make google searches, youtube searches,etc using this.
- we can open IDE code using single voice command and much more !!!

## II. PROCEDURE FOLLOWED

Let's start building our own voice assistant.

**\*\*1 Starting with Visual Studio Code(VS code)**

I Prefer VS code IDE as I am familiar with That , you can use any platform what you prefer open new file and name it as voice assistant.py

**\*\*2 Defining Speak Function**

The and first and foremost thing for an A.I.

assistant is that it should be able to speak. To make our voice assistant talk, 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 which will help us to convert text to speech. In short, it is a text-to-speech library. It works offline, and it is compatible with Python 2 as well the Python 3.

Installation:

- open command prompt Type the following  
pip install pyttsx3.

- In case you receive such errors:
  - No module named win32com.client.
  - No module named win32.
  - No module named win32api

Then, install pyiwin32:

- open command prompt type the following

pip install pyiwin32. After successfully installing pytttsx3, import this module in your program.

Usage:

```
import pytttsx3
engine = pytttsx3.init('sapi5')
voices= engine.getProperty('voices') #getting
details of current voice engine. setProperty('voice',
voice[0].id) .
```

Here, we can use different type of voices

voice[0] gives male voice ,voice[1] gives female voice

In the Above code , in line no 2 we used 'sapi5'

- **What is sapi5?**
- Speech API developed by Microsoft. Helps in synthesis and recognition of voice.
- **Writing Our speak() Function**
- We made a function called speak() at the starting. Now, we will write our speak() function so that it can 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: Now, we will create a main() function, and inside this main() Function, we will call our speak function.

Code:

```
if __name__=="__main__" :
speak("Welcome")
```

Whatever you will write inside this speak() function will be converted into speech. Congratulations! With this, our voice assistant has its own voice, and it is ready to speak.

### **\*\*3 Defining Wish me Function**

Now, we are going to make a wishme() function, that will make our voice assistant to 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 our wishme() function:

```
defwishme():
hour = int(datetime.datetime.now().hour)
```

Here, we have stored the integer value of the

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

```
defwishMe():
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!")
i=random.randint(1,3)
if i==1:
speak("I am your personal assistant sir. what can i
do for you?")
elif i==2:
speak("Hiithere.I am your personal assistant")
else:
speak("Namaste. order me!")
#u need to import random module for this
```

### **\*\*4 Defining Take command Function**

The next most important thing for our A.I. assistant is that it should be able to 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 be able to 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:

```
pip install speechRecognition
```

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

```
importsr.speechRecognition as sr
```

Let's start coding our takeCommand() function.

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

```
r = sr.Recognizer()
withsr.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") #User query will be printed.
```

except Exception as e:

```
# print(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
```

### **\*\*5 Coding logic of voice assistant**

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

#### **\*\*5.1 Defining Task 1: To search something on Wikipedia**

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

Installing wikipedia module:

```
pip install wikipedia
```

After successfully installing wikipedia module, import the wikipedia module 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 search query of the user 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 help of speak function.

#### **\*\*5.2 Defining Task 2: To open YouTube site in a web-browser**

To open any website, we need to import a module called web browser.

It is an in-built module, and we do not need to install it with 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 the elif loop to check whether the Youtube is in the query of the user or not. Let's suppose, the user gives command as "openyoutube."

So, open youtube will be in the user's query, and

the elif condition will be true.

#### **\*\*5.3 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 while opening Youtube.

#### **\*\*5.4 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 = 'D:\\telugu\\music\\Folder3'
```

```
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 help of the os module.

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, It will play any random song from the song directory.

Here, you use your music stored location. Above is just for example

#### **\*\*5.5 Defining Task 5: To know the current time**

```
elif 'the time' in query:
```

```
strTime =
```

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

```
speak("Now, the time is {strTime}")
```

In the above, code we are using datetime() function and storing the current or live of the system 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 the speech.

#### **\*\*5.6 Defining Task 6: To open the VS Code**

code:

```
elif 'open code' in query:
```

```
codePath=
```

```
"C:\\Users\\Nithin\\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.

you are supposed to give your application location, above is mine

Steps of getting 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.

**\*\*5.7 Defining Task 7: 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 to 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:

- i. The sender: Email address of the sender.
- ii. The receiver: Email of the receiver.
- iii. The message: A string message which needs to be sent to one or more than one recipient.

**\*\*5.7.1 Defining Send email function :**

Now, we will create a sendEmail() function, which will help us to send emails to one or more than one recipients.

```
defsendEmail(to, content):  
server = smtplib.SMTP('smtp.gmail.com', 587)  
server.ehlo()  
server.starttls()  
server.login('youremail@gmail.com','yourpasswor')  
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 = "yourEmail@gmail.com"  
sendEmail(to, content)  
speak("Email has been sent!")  
except Exception as e:  
print(e)  
speak("Sorry. Failed to send message.try again")
```

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

**\*\*6 some stuff**

your A.I must know some basics like its inventor or its name etc for that we can add this

```
elif 'how are you' in query:  
speak("I am fine, Thank you")  
speak("How are you, Sir")  
elif 'fine' in query or "good" in query:  
speak("It's good to know that your fine")  
elif 'change my name to' in query:  
query = query.replace("change my name to", "")  
assname = query  
elif 'change name' in query:  
speak("What would you like to call me, Sir ")  
assname = takeCommand()  
speak("Thanks for naming me")  
elif 'what's your name' in query or 'What is your name' in query:  
speak("My friends call me")  
speak(assname)  
print("My friends call me", assname)
```

```
elif 'exit' in query:  
speak("Thanks for giving me your time")  
exit()  
elif "who made you" in query or "who created you" in query:  
speak("I was designed by C I E T in Lam")
```

**\*\*7 Recapitulate**

What have we done so far?

At First, we have created a wishme() function to wish us according to the system time to our A.I.

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.

**\*\*8 Is it an A.I.?**

A lot of people will argue that the virtual assistant that we have created is not an A.I, but it is the output of the bunch of the statement. But, if we look at the very basic level, the sole purpose of A.I is to 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. !

### III. MATH

```
importos  
import pytttsx3
```

```

import smtplib
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import random
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
#to change voice change the value from 0 to 1
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
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!")
    i = random.randint(1,3)
    if i == 1:
        speak("I am your personal assistant sir. what can i
do for you?")
    elif i == 2:
        speak("Hi there. I am your personal assistant")
    else:
        speak("Namaste sir. order me!")
def takeCommand():
    #It takes input
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")
    except Exception as e:
        # print(e)
        print("Say that again please...")
        return "None"
    return query
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()

#use your mail id and password
if __name__ == "__main__":
    wishMe()
    while True:
        # if 1:
        query = takeCommand().lower()
        # using if and elif so that you can add extra what
        you want 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)
        elif 'open youtube' in query:
            webbrowser.open("youtube.com")
        elif 'open google' in query:
            webbrowser.open("google.com")
        elif 'open facebook' in query:
            webbrowser.open("facebook.com")
        elif 'play music' in query:
            music_dir = 'D:\\User1\\songs\\Favorite Songs2'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[0]))
        elif 'the time' in query:
            strTime =
datetime.datetime.now().strftime("%H:%M:%S")
            speak("Now, the time is {strTime}")
        elif 'how are you' in query:
            speak("I am fine, Thank you")
            speak("How are you, Sir")
        elif 'fine' in query or "good" in query:
            speak("It's good to know that your fine")
        elif 'change my name to' in query:
            query = query.replace("change my name to", "")
            assname = query
        elif 'open code' in query:
            codePath =
"C:\\Users\\Nithin\\AppData\\Local\\Programs
\\Microsoft VS Code\\Code.exe"
            os.startfile(codePath)
        elif 'who made you' in query or 'who created you' in
query:
            speak("I was designed by C I E T in Lam")
            exit()
        elif 'send mail' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "yourEmail@gmail.com"
                sendEmail(to, content)
                speak("Email has been sent!")
            except Exception as e:

```

```

print(e)
speak("Sorry. something went wrong, try again")
speak("My friends call me")
speak(assname)
print("My friends call me", assname)
elif 'exit' in query:
speak("Thanks for giving me your time")
exit()
elif 'who made you' in query or 'who created you' in
query:
speak("I was designed by C I E T in Lam")
elif 'open code' in query:
codePath =
"C:\\Users\\Nithin\\AppData\\Local\\Programs
\\Microsoft VS Code\\Code.exe"
os.startfile(codePath)
elif 'send mail' in query:
try:
speak("What should I say?")
content = takeCommand()
to = "yourEmail@gmail.com"
sendEmail(to, content)
speak("Email has been sent!")
except Exception as e:
print(e)
speak("Sorry. something went wrong, try again")

```

#### IV. RESULTS

```

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Nithi\Desktop\Project\voice assistant\final\final.py ====
Good Evening!
User : Hello
I am your personal assistant sir. what can i do for you?
Listening...

```

```

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Nithi\Desktop\Project\voice assistant\final\final.py ====
Good Evening!

```

```

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Nithi\Desktop\Project\voice assistant\final\final.py ====
Good Evening!
User : Hello
Hii there.I am your personal assistant
user: search google for chalapathi institute of technology
Search ResultsChalapathi Institute of Engineering and Technology www.chalapathe
ngg.ac.in AICTE Sponsored National E-Conference on Recent Advances in Computer S
cience and Engineering 2020 (NCRACSE-2020), on 28th & 29th August 2020

```

```

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Nithi\Desktop\Project\voice assistant\final\final.py ====
Good Evening!
User : Hello
I am your personal assistant sir. what can i do for you?
User: Open Chrome
opening chrome
Listening...

```

```

import speech_recognition as sr
import wikipedia
import webbrowser
import os
import sys

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

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

def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as mic:
        r.pause_threshold = 1
        audio = r.listen(mic)

    try:
        query = r.recognize_google(audio)
    except Exception as e:
        print(e)
        speak("Sorry. something went wrong, try again")
        return ""

    return query

def sendEmail(to, content):
    import smtplib
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login('yourEmail@gmail.com', 'yourPassword')
    text = content
    server.sendmail('yourEmail@gmail.com', to, text)
    server.close()

def openCode():
    codePath = "C:\\Users\\Nithin\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
    os.startfile(codePath)

def openChrome():
    webbrowser.open("http://www.google.com")

def searchGoogle(query):
    results = wikipedia.summary(query, sentences=3)
    speak(results)

def main():
    speak("Good Evening!")
    while True:
        query = takeCommand().lower()
        if 'hello' in query:
            speak("Hello")
            speak("I am your personal assistant sir. what can i do for you?")
        elif 'open code' in query:
            openCode()
        elif 'open chrome' in query:
            openChrome()
        elif 'send mail' in query:
            speak("What should I say?")
            content = takeCommand()
            to = "yourEmail@gmail.com"
            sendEmail(to, content)
            speak("Email has been sent!")
        elif 'search google for' in query:
            query = query.replace('search google for', '')
            searchGoogle(query)
        else:
            speak("Sorry. something went wrong, try again")

if __name__ == '__main__':
    main()

```

#### V. CONCLUSION

With this, we have successfully made our first virtual assistant. Explore and try to add other functionalities to voice assistant. We have designed a voice assistant which is easier 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. The voice assistant will be helpful for all of us to make our work more easier by using the assistant which we have designed.

#### REFERENCES

- [1] [https://en.wikipedia.org/wiki/Virtual\\_assistant](https://en.wikipedia.org/wiki/Virtual_assistant)
- [2] Personal Assistant with Voice Recognition Intelligence, 2017, IJERT, ISSN 0974-3154 Volume 10, Number 1.
- [3] Aarthi Easwara Moorthy, Kim-Phuong L. Vu, Voice Activated Personal Assistant: Acceptability of Use in the Public Space, International Conference on Human Interface and the Management of Information, HIMI 2014: Human Interface and the Management of Information, Information and Knowledge in Applications and Services pp 324-334
- [4] <https://voicebot.ai/2017/07/14/timeline-voice-assistant-s-short-history-voice-revolution>
- [5] <https://www.smartsheet.com/voice-assistants-artificial-intelligence>