SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603 203,

Kancheepuram District

SCHOOL OF COMPUTING

DEPARTMENT OF NETWORKING & COMMUNICATIONS

Course Project

Voice Assistant

Course Code	**:**	18CSC305J
Course Name	**:**	Artificial Intelligence
Faculty	**:**	Dr.R.Radhika

Team Members:

1. B. Sai Datta Surya -RA1911030010115

2. V. Siva Supradeep -RA1911030010104

3. P. Sai Phanidhar - RA1911030010101

INDEX

No	Particulars
1.	Abstract
2.	Problem Statement
3.	Installing Libraries
4.	Steps in making Voice Assistant
5.	Codes
6.	Screenshot
7.	Proof of GitHub upload
8.	Conclusion
9.	References

ABSTRACT:

Speech probably is the most efficient and natural way to communicate with each other. Thus, being the best way of communication, it could also be a useful interface to communicate with machines and systems like IVR system. The Interactive Voice Response (IVR) system along with the speech recognition technology can play efficient role in providing easy and efficient customer/user service.

This process of recognition is done by breaking down audio into individual sounds, then converting them into a digital format where we will be using Machine learning algorithms ad models to find the word for that sound.

Speech Recognition (SR) is the process of extracting the string of words automatically from the speech signal, by means of an algorithm. It is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine readable format. Speech recognition is a powerful tool of the information exchange using the acoustic signal. Therefore, not surprisingly, the speech signal is for several centuries the subject of research. Speech recognition is a technology that makes a computer able to capture the words spoken by a human with a help of microphone

PROBLEM STATEMENT:

Most of the works done till today on the field of IVR system has been primarily focused on the input mechanisms based on the keyboard or touch pad. In such cases it is tedious to provide the input command every time through typing of texts.

This way of providing input to the computer system may be enhanced if we could provide direct speech input instead of typing.
This enables in fast interaction between the system and user and therefore increases overall satisfaction of the customers. This also increases the speed of access of the information from the system. Furthermore, English language has been widely implemented in IVR systems.

This has created difficulty for people while interacting with the system. Thus by implementing the voice commands it is easier to interact and provide the input to the system.

The major focus of the project being developed is the use of direct voice command for the interactive voice response system without need of typing which then further can be applicable to real world applications like call centers, customer support systems and other several organization inquiry systems.

INSTALLING LIBRARIES:

1. SpeechRecognition

Code: `pip install SpeechRecognition`

SpeechRecognition is a Python speech recognition library that is used to convert our human speech into text.

2. PyAudio

Code: `pip install PyAudio`

SpeechRecognition library doesn't work alone, it uses PyAudio Library. PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. Using PyAudio, you can easily use Python to play and record audio on a variety of platforms.

3. Pywhatkit

Code: `pip install pywhatkit`

pywhatkit library will give power of the internet to our personal assistant Friday for certain functionality for example using the youtube to play songs.

4. Wikipedia

Code: `pip install wikipedia`

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. We want our AI assistant Friday, to be more knowledgeable and informative so we use this approach for it.

STEPS IN MAKING VOICE ASSISTANT:

1. Speech Recognition

2. Activating Personal Assistant

3. Adding Speech for Assistant

4. Making Assistant take play song command

5. Give Assistant Internet Access

6. Adding datetime and Wikipedia access to AI Assistant

7. Adding personal Information about Assistant

SPEECH RECOGNITION:

We start by importing speech_recognition, we do not import the Pyaudio library but speech_recognition uses it for conversion purposes.

The speech_recognition module is used to create a Recognizer() object which takes audio data as input captured by another Microphone() object. We print this to verify if our setup is working properly as expected.

ACTIVATING PERSONAL ASSISTANT:

We have named our personal assistant **Friday**, so here we add a logic to recognize if the speech contains the word "**Friday**" or not by using a simple if condition. If the speech matches "**Friday**" then we just print it. We will put more logic in the upcoming sections.

Adding Speech for Assistant

In the above section, we converted speech to text for our assistant Friday to understand what we say but we also want it to reply back to us. So for this, we have to add text to speech capability to Friday.

We use Pyttsx3 library text to speech conversion, using it as an engine that answers us back or reads the output of our question. Here we initialize pysttsx3 after the listener and we test it by making it read some of the sample text.

Making AI Assistant Take “Play Song” Command

We have come this far by establishing to and fro communication with Friday but how about making him do something for example asking him to play a song.

We encapsulate the logic discussed in the above section in two functions talk() and take_command(). Then we create another function run_Friday() to take voice command using take_command() and reply back using the talk() function.

Now that we want Friday to play the song we want it to play the song when we say “FRIDAY Play <song name> song” for example, “FRIDAY Play Despacito song”.

We extract the song name from the voice command and print it to test if it is working properly. In the next section, we will add code to actually make Friday play the song from the internet.

Giving AI Assistant Internet Access to Play Song

Making Friday play song requires the internet where we have many sources for the song and in our case we choose YouTube. For this, we use pywhatkit library’s playonyt() function where we pass the song name and it opens YouTube with that song.

Adding DateTime and Wikipedia access to AI Assistant

We will now add more capabilities into our Friday such as telling time or date and having the information about anything from Wikipedia. For this, we use Python built-in library datetime and external library wikipedia.

We add these two functionalities in our run_Friday() function by using elif logic. For Wikipedia we are using the command as “tell me about <something>” and it returns the summary from the Wikipedia page.

Adding personal Information about Assistant

Our Friday is ready to do some task which we assign to him but what Friday knows about itself? In case if we ask him about himself. it should be able to answer it.

So we add this information in the run_Friday() function using multiple elif blocks about question and answer. You can add more items if you wish.

CODE:

```
import speech_recognition as sr

import pyttsx3

import pywhatkit

import datetime

import wikipedia

import pyjokes

import sys

import webbrowser

import warnings

warnings.filterwarnings("ignore")

from datetime import date

engine=pyttsx3.init()

engine.setProperty("rate", 150)

voices = engine.getProperty("voices")

engine.setProperty('voice', voices [1].id)

recognizer=sr.Recognizer()

def engine_talk(text):

    engine.say(text)

    engine.runAndWait()

def run_jarvis():

    with sr.Microphone() as source:

        recognizer.adjust_for_ambient_noise(source,duration=1)
```

```
        print('\n')
        print("Start Speaking!!")
        engine_talk('listening.. ')
        recordedaudio=recognizer.listen(source)
    try:

command=recognizer.recognize_google(recordedaudio,language='en
-in')
        command = command.lower()
        if 'jarvis' in command :
            command = command.replace('Friday', '')
            print('you said'+command)
        else :
            print('you said : '+command)
        if 'hi' in command :
            print('hello how can i help you ??')
            engine_talk('hello, how can i help you ??')
        elif 'who are you' in command :
            print('I am friday a k a your virtual assistant
master')
            engine_talk('I am friday a k a your virtual
assistant master. how can i help you ??')
        elif 'what is your name' in command :
            print('I am friday a k a your virtual assistant
master')
            engine_talk('I am friday a k a your virtual
assistant master')
        elif 'spider-man' in command:
```

```
            print('great responsibility comes with great
power')

            engine_talk('great responsibility comes with great
power')

        elif 'what can you do' in command :

            print('''i can play songs on youtube , tell you a
joke, search on wikipedia, tell date and time,find your
location, locate area on map,

                open different websites like instagram,
youtube,gmail, git hub, stack overflow and searches on
google.How may i help you ??''')

            engine_talk('''i can play songs on youtube , tell
you a joke, search on wikipedia, tell date and time,find your
location, locate area on map,

                        open different websites like insta
gram, youtube,gmail, git hub, stack overflow and searches on
google. How may i help you ??''')

        elif 'play' in command:

            song = command.replace('play', '')

            print('Playing' +song)

            engine_talk('Playing' +song)

            pywhatkit.playonyt(song)

        elif 'date and time' in command :

            today = date.today()

            time = datetime.datetime.now().strftime('%I:%M %p')

            # Textual month, day and year

            d2 = today.strftime("%B %d, %Y")

            print("Today's Date is ", d2, 'Current time is',
time)

            engine_talk('Today is : '+ d2)

            engine_talk('and current time is '+ time)
```

```
        elif 'time and date' in command :
            today = date.today()
            time = datetime.datetime.now().strftime('%I:%M %p')
            # Textual month, day and year
            d2 = today.strftime("%B %d, %Y")
            print("Today's Date is ", d2, 'Current time is',
time)
            engine_talk( 'Current time is '+ time)
            engine_talk('and Today is : '+ d2)

        elif 'time' in command:
            time = datetime.datetime.now().strftime('%I:%M %p')
            print('The current time is' +time)
            engine_talk('The current time is')
            engine_talk(time)
        elif 'date' in command:
            today = date.today()
            print("Today's date:", today)
            # Textual month, day and year
            d2 = today.strftime("%B %d, %Y")
            print("Today's Date is ", d2)
            engine_talk('The todays date is')
            engine_talk(d2)
        elif 'tell me about' in command:
            name = command.replace('tell me about' , '')
            info = wikipedia.summary(name, 1)
```

```
            print(info)
            engine_talk(info)
        elif 'wikipedia' in command:
            name = command.replace('wikipedia' , '')
            info = wikipedia.summary(name, 1)
            print(info)
            engine_talk(info)
        elif 'what is' in command:
            name = command.replace('what is ' , '')
            info = wikipedia.summary(name, 1)
            print(info)
            engine_talk(info)
        elif 'who is ' in command:
            name = command.replace('who is' , '')
            info = wikipedia.summary(name, 1)
            print(info)
            engine_talk(info)
        elif 'what is ' in command :
            search =
'https://www.google.com/search?q='+command
            print(' Here is what i found on the internet..')
            engine_talk('searching... Here is what i found on
the internet..')
            webbrowser.open(search)
        elif 'joke' in command:
            _joke = pyjokes.get_joke()
```

```
            print(_joke)

            engine_talk(_joke)

        elif 'search' in command :

            search =
'https://www.google.com/search?q='+command

            engine_talk('searching... ')

            webbrowser.open(search)

        elif "my location" in command:

            url =
"https://www.google.com/maps/search/Where+am+I+?/"

            webbrowser.get().open(url)

            engine_talk("You must be somewhere near here, as
per Google maps")

        elif 'locate ' in command :

            engine_talk('locating ...')

            loc = command.replace('locate', '')

            if 'on map' in loc :

                loc= loc.replace('on map',' ')

            url = 'https://google.nl/maps/place/'+loc+'/&amp;'

            webbrowser.get().open(url)

            print('Here is the location of '+loc)

            engine_talk('Here is the location of '+loc)

        elif 'on map' in command :

            engine_talk('locating ...')

            loc = command.split(" ")

            print(loc[1])
```

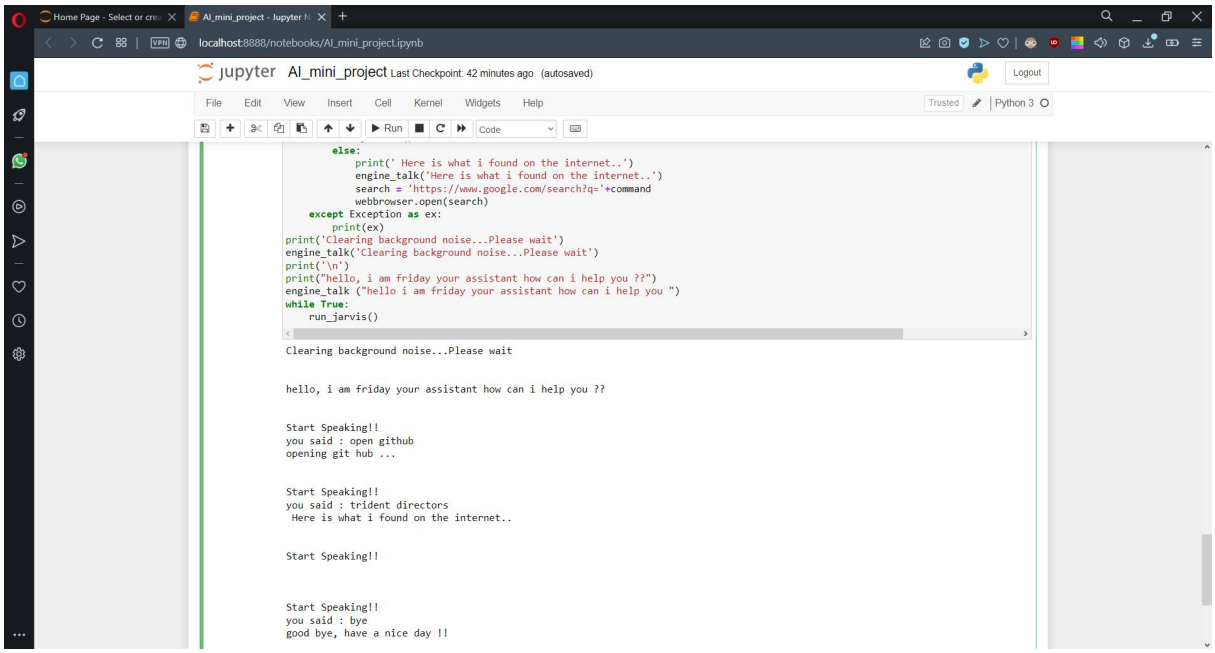
```
            url = 'https://google.nl/maps/place/'+loc[1]
+'/&amp;'
            webbrowser.get().open(url)
            print('Here is the location of '+loc[1])
            engine_talk('Here is the location of '+loc[1])

        elif 'location of' in command :
            engine_talk('locating ...')
            loc = command.replace('find location of', '')
            url = 'https://google.nl/maps/place/'+loc+'/&amp;'
            webbrowser.get().open(url)
            print('Here is the location of '+loc)
            engine_talk('Here is the location of '+loc)
        elif 'where is ' in command :
            engine_talk('locating ...')
            loc = command.replace('where is', '')
            url = 'https://google.nl/maps/place/'+loc+'/&amp;'
            webbrowser.get().open(url)
            print('Here is the location of '+loc)
            engine_talk('Here is the location of '+loc)
        elif 'bootcamps' in command :
            search =
'http://tathastu.twowaits.in/index.html#courses'
            engine_talk('opening boot camps')
            webbrowser.open(search)
        elif 'boot camps' in command :
```

```
            search =
'http://tathastu.twowaits.in/index.html#courses'

            engine_talk('opening boot camps')

            webbrowser.open(search)

        elif 'python bootcamp' in command :

            search =
'http://tathastu.twowaits.in/kickstart_python.html'

            engine_talk('showing pythonboot camp')

            webbrowser.open(search)

        elif 'data science bootcamp' in command :

            search =
'http://tathastu.twowaits.in/kickstart_data_science.html'

            engine_talk('showing data science and ml bootcamp')

            webbrowser.open(search)

        elif 'open google' in command :

            print('opening google ...')

            engine_talk('opening google..')

            webbrowser.open_new('https://www.google.co.in/')

        elif 'gmail' in command :

            print('opening gmail ...')

            engine_talk('opening gmail..')

            webbrowser.open_new('https://mail.google.com/')

        elif 'open youtube' in command :

            print('opening you tube ...')

            engine_talk('opening you tube..')

            webbrowser.open_new('https://www.youtube.com/')

        elif 'open instagram' in command :
```

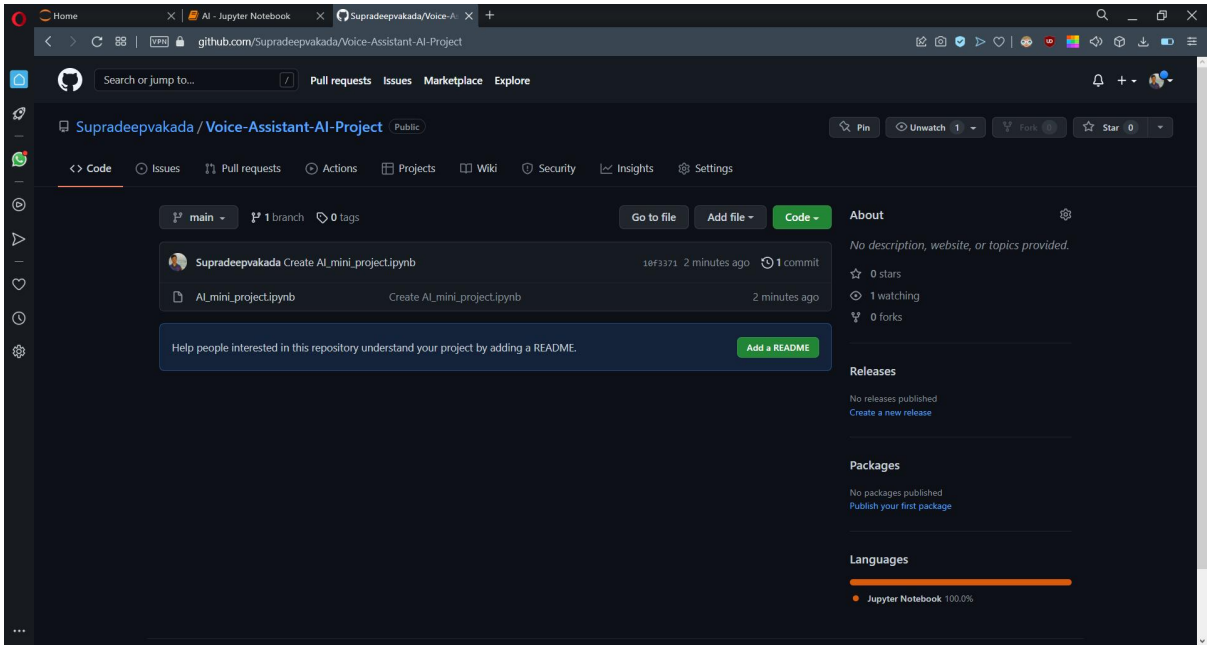
```
        print('opening instagram ...')
        engine_talk('opening insta gram...')
        webbrowser.open_new('https://www.instagram.com/')
    elif 'open stack overflow' in command :
        print('opening stackoverflow ...')
        engine_talk('opening stack overflow...')
        webbrowser.open_new('https://stackoverflow.com/')
    elif 'open github' in command :
        print('opening git hub ...')
        engine_talk('opening git hub...')
        webbrowser.open_new('https://github.com/')
    elif 'bye' in command:
        print('good bye, have a nice day !!')
        engine_talk('good bye, have a nice day !!')
        sys.exit()
    elif 'thank you' in command :
        print("your welcome")
        engine_talk('your welcome')
    elif 'stop' in command:
        print('good bye, have a nice day !!')
        engine_talk('good bye, have a nice day !!')
        sys.exit()
    elif 'bye' in command:
        print('good bye, have a nice day !!')
        engine_talk('good bye, have a nice day !!')
```

```
            sys.exit()

        else:

            print(' Here is what i found on the internet..')

            engine_talk('Here is what i found on the
internet..')

            search =
'https://www.google.com/search?q='+command

            webbrowser.open(search)

    except Exception as ex:

        print(ex)

print('Clearing background noise...Please wait')

engine_talk('Clearing background noise...Please wait')

print('\n')

print("hello, i am friday your assistant how can i help
you ??")

engine_talk ("hello i am friday your assistant how can i help
you ")

while True:

    run_jarvis()
```

SCREENSHOT:



PROOF OF GITHUB UPLOAD:



Repository link:

https://github.com/Supradeepvakada/Voice-Assistant-AI-Project

CONCLUSION:

Hope you liked our project where we created a personal voice assistant that can understand voice command using speech recognition in Python. We just showed you a prototype, but you can add as many creative features and functionalities as you require.