Ai Virtual Assistant Using Python

AI Virtual Assistant using Python
VOICE RECOGNITION
Presented By : Shubham Singh Chouhan

Virtual assistant or Voice Assistant is an awesome thing. If you want your machine to run on your command like Jarvis did for Tony. Yes it is possible. It is possible using Python. Python offers a good major library so that we can use it for making a virtual assistant. Windows has Sapi5 and Linux has Espeak which can help us in having the voice from our machine. It is a weak A.I.

An AI personal assistant is a piece of software that understands verbal or written commands and completes task assigned by the client. It is an example of weak AI that is it can only execute and perform quest designed by the user.

Virtual Assistant also known as voice recognition, personal voice assistant, Jarvis, etc. But all the working progresses are same.

A **voice assistant** is a digital assistant that uses **voice recognition**, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user.
Based on specific commands, sometimes called intents, spoken by the user, voice assistants can return relevant information by listening for specific keywords and filtering out the ambient noise.

Skills

The implemented voice assistant can perform the following task it can open YouTube, Gmail, Google chrome and stack overflow. Predict current time, take a photo, search Wikipedia to abstract required data, predict weather in different cities, get top headline news from Times of India and can answer computational and geographical questions too.

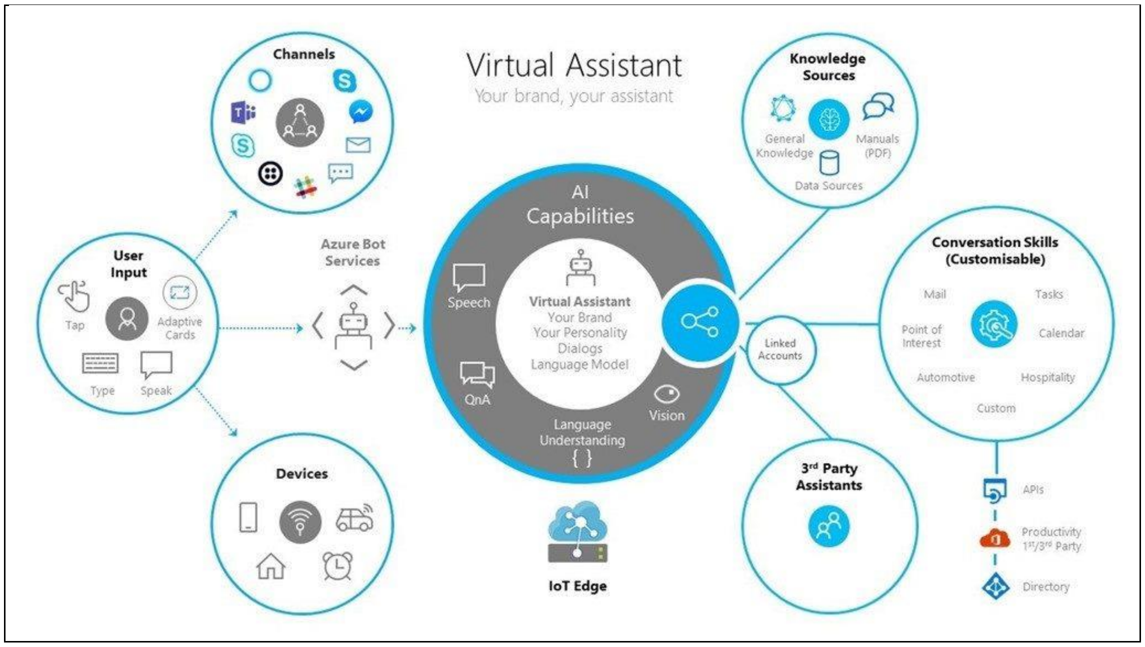
Aim to build Virtual Assistant

A virtual assistant is an independent contractor who provides administrative services to clients while operating outside of the client's office. A virtual assistant typically operates from a home office but can access the necessary planning documents, such as shared calendars, remotely.

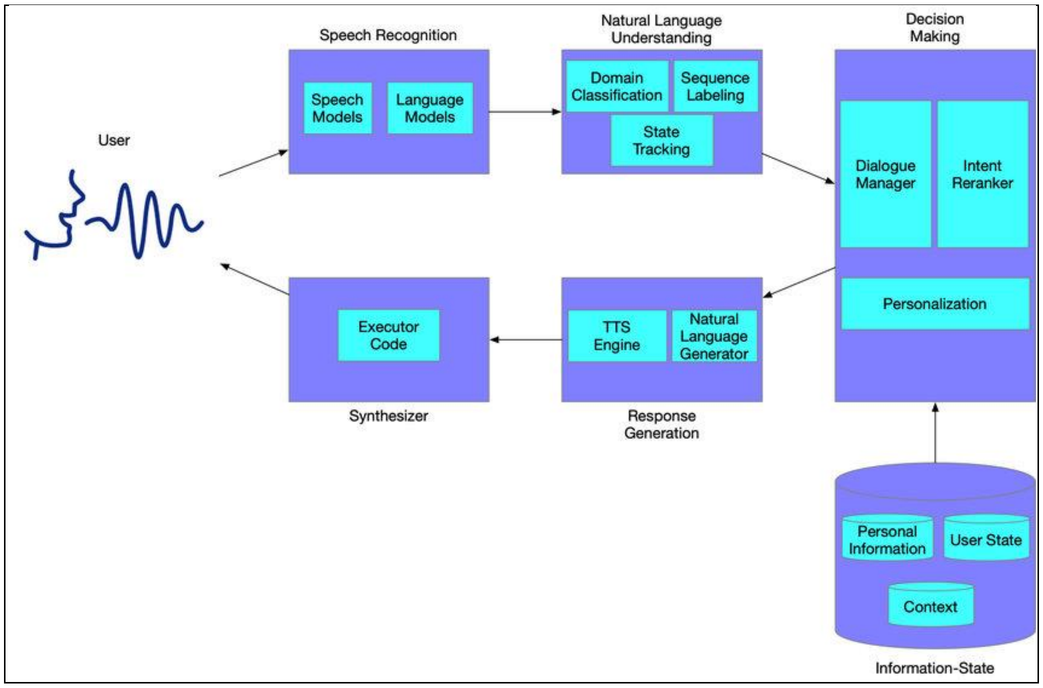
Voice assistants can make calls, send text messages, look things up online, provide directions, open apps, set appointments on our calendars, and initiate or complete many other tasks. With the addition of separate apps on the phone, our voice can be a type of remote control for our lives.

Working Structure of Virtual Assistant

Voice assistants are programs on digital devices that listen and respond to verbal commands. A user can say, “What's the weather?” and the voice assistant will answer with the weather report for that day and location. They could say, “Tell me a story,” and the assistant will jump into a tale.



Components of Virtual Assistant

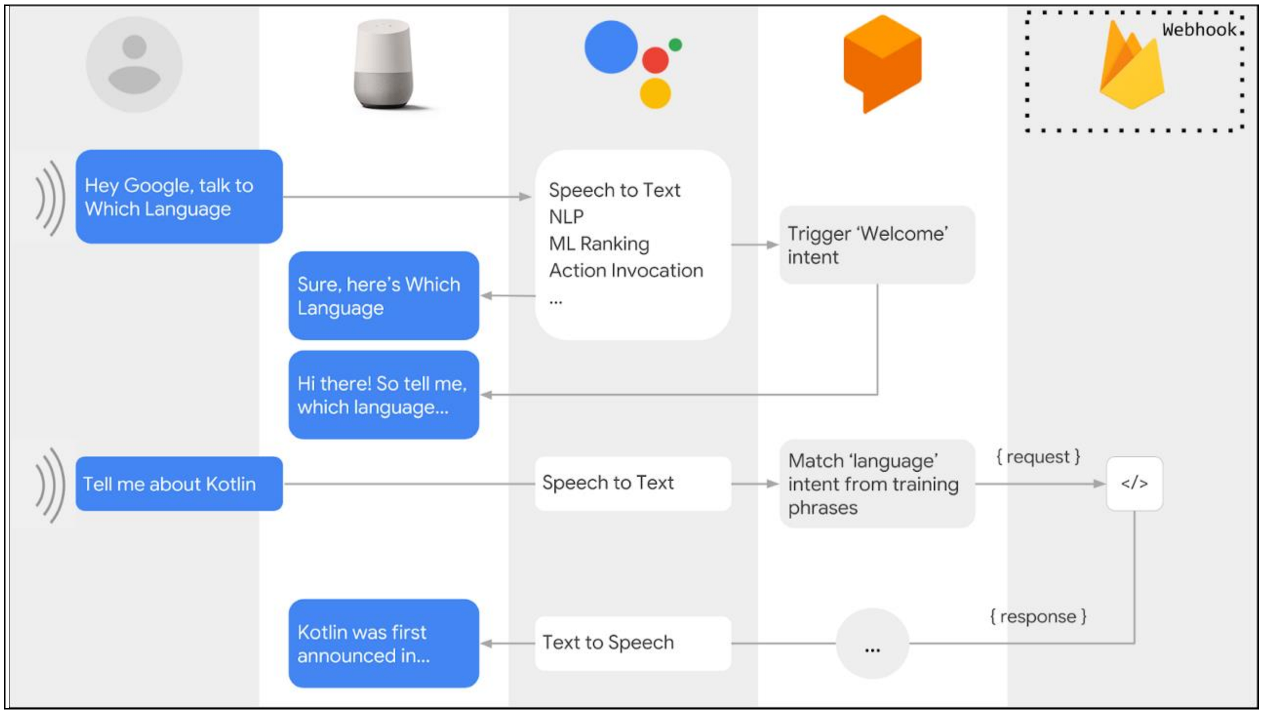


Real Life Examples of Virtual Assistant

❖ Google Assistant

Google Assistant offers voice commands, voice searching, and voice-activated device control, letting you complete a number of tasks after you've said the "OK Google" or "Hey Google" wake words. It is designed to give you conversational interactions. Google Assistant will: Control your devices and your smart home.

Google Assistant does not store the users' data without their permission. To store the audio, the user can go to Voice & Audio Activity (VAA) and turn on this feature. Audio files are sent to cloud and used by Google to improve the performance of Google Assistant, but only if the VAA feature is turned on.

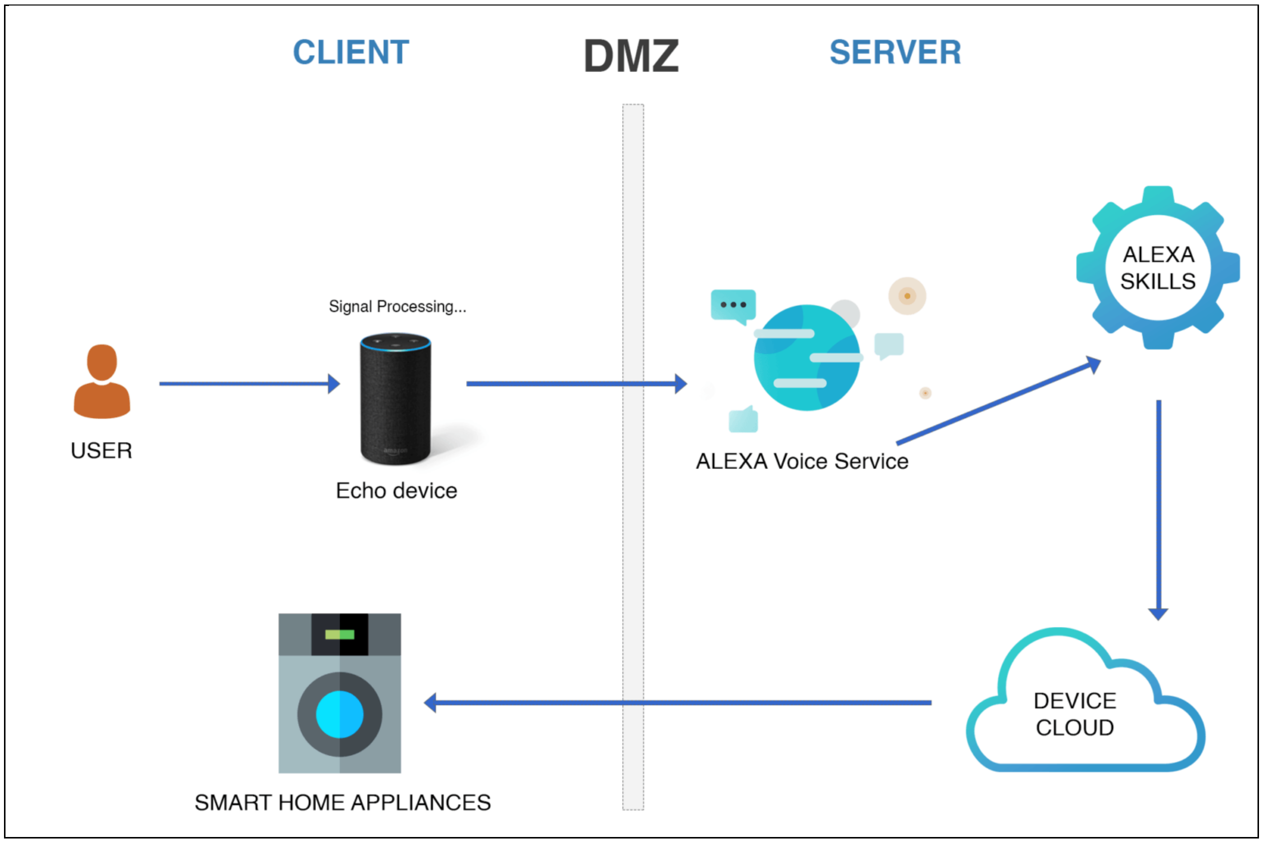

Webhook
Hey Google, talk to Which Language
Speech to Text
NLP
ML Ranking
Action Invocation
...
Trigger 'Welcome' intent
Sure, here's Which Language
Hi there! So tell me, which language...
Tell me about Kotlin
Speech to Text
Match 'language' intent from training phrases
{ request }
</>
{ response }
Kotlin was first announced in...
Text to Speech
...

❖ Amazon's Alexa

Amazon's Alexa works if it is connected to Wi-Fi or has a stable internet connection. Alexa can access everything from Google if connected to the internet. Without Wi-Fi, Alexa cannot connect to various applications and features.

you can link your Android smartphone to Alexa and use voice commands to trigger phone calls, text messages, and music playback on your connected device.

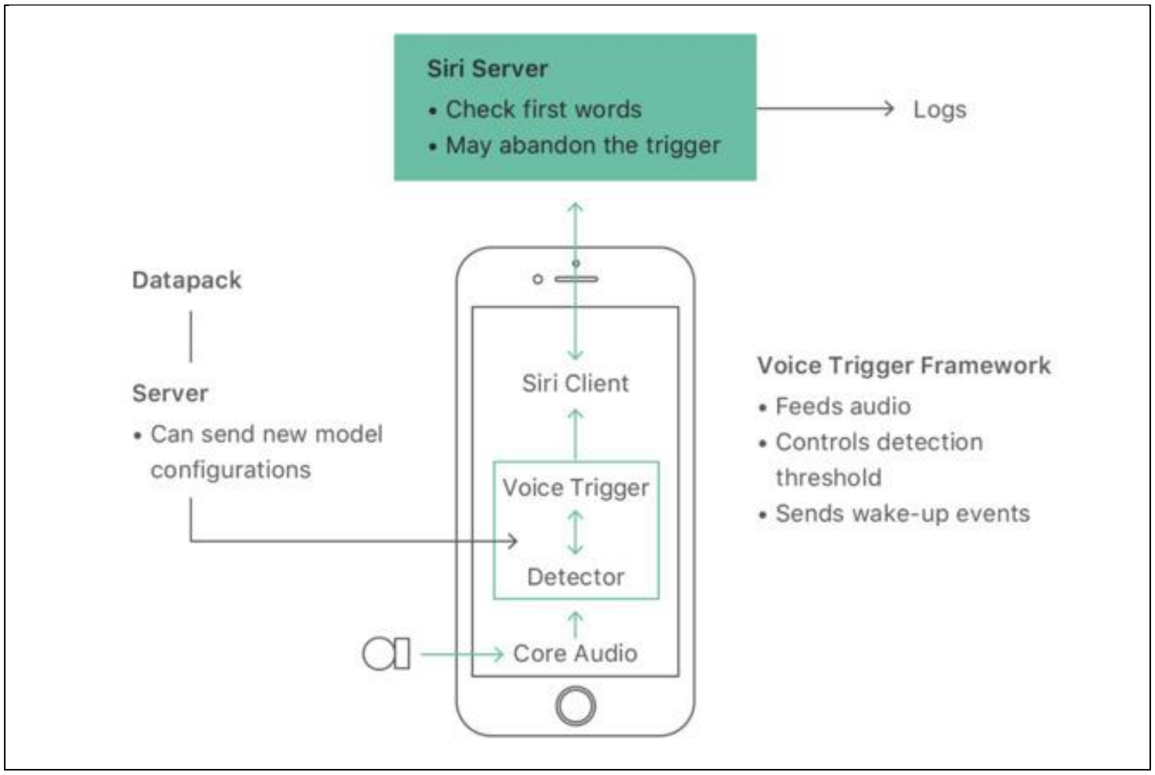
Amazon's Virtual Assistant Alexa only listens to conversations when its wake word (like Alexa, Amazon, Echo) is used. It starts recording the conversation after the call of a wake word. It stops listening after 8 seconds of silence. It sends the recorded conversation to the cloud. It is possible to delete the recording from the cloud by visiting 'Alexa Privacy' in 'Alexa'. There is a feature to stop Alexa from listening to your conversations using 'mute' feature of Alexa. After muting the device, it cannot listen even if the wake words (like Alexa) were used.



❖ **Apple's Siri**

Upon receiving your request, Siri records the frequencies and sound waves from your voice and translates them into a code. Siri then breaks down the code to identify particular patterns, phrases, and keywords.

Apple does not record audio to improve Siri, it uses transcripts instead. It only sends data which is important for analysis, for instance, if the user asks Siri to read their message, it won't send the message to the cloud, the machine will directly read the message without server's interference. Users can opt-out anytime if they don't want Siri to send the transcripts in the cloud.



<u>**Technology Used**</u>

- **Python**
- **Python Modules / Packages**
- **Python Libraries**
- **Speech Recognition**

MODULES NEEDED

To build a personal voice assistant it’s necessary to install the following packages in your system using the pip command.

1) **Speech recognition** — Speech recognition is an important feature used in house automation and in artificial intelligence devices. The main function of this library is it tries to understand whatever the humans speak and converts the speech to text.

2) **pyttsx3** — pyttxs3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.

3) **wikipedia** — Wikipedia is a multilingual online encyclopedia used by many people from academic community ranging from freshmen to students to professors who wants to gain information over a particular topic. This package in python extracts data’s required from Wikipedia.

4) **ecapture** — This module is used to capture images from your camera

5) **datetime** — This is an inbuilt module in python and it works on date and time

6) **os** — This module is a standard library in python and it provides the function to interact with operating system

7) **time** — The time module helps us to display time

8) **Web browser** — This is an in-built package in python. It extracts data from the web

9) **Subprocess** — This is a standard library use to process various system commands like to log off or to restart your PC.

10) **Json**- The json module is used for storing and exchanging data.

11) **request**- The request module is used to send all types of HTTP request. Its accepts URL as parameters and gives access to the given URL'S.

12) **wolfram alpha** — Wolfram Alpha is an API which can compute expert-level answers using Wolfram's algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.

IMPLEMENTATION

Import the following libraries:

```
import speech_recognition as sr
import pyttsx3
import datetime
import wikipedia
import webbrowser
import os
import time
import subprocess
from ecapture import ecapture as ec
import wolframalpha
import json
import requests
```

```
# other libraries
import city
import ctypes
import pyjokes
import smtplib
from email.message import EmailMessage
```

Setting up the speech engine:

The pyttsx3 module is stored in a variable name engine.

Sapi5 is a Microsoft Text to speech engine used for voice recognition.

The voice Id can be set as either 0 or 1,

0 indicates Male voice |&| 1 indicates Female voice

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
```

Now define a function **speak** which converts text to speech. The speak function takes the text as its argument, further initialize the engine.

runAndWait: This function Blocks while processing all currently queued commands. It Invokes callbacks for engine notifications appropriately and returns back when all commands queued before this call are emptied from the queue.

```
def speak(text):
    engine.say(text)
    engine.runAndWait()
```

Initiate a function to greet the user:

Define a function **wishMe** for the AI assistant to greet the user.

The **now().hour** function abstract's the hour from the current time.

If the hour is greater than zero and less than 12, the voice assistant wishes you with the message “Good Morning”.

If the hour is greater than 12 and less than 18, the voice assistant wishes you with the following message “Good Afternoon”.

Else it voices out the message “Good evening”

```
def wishMe():
    hour = datetime.datetime.now().hour
    if hour>=0 and hour<12:

        print('Hello, Good Morning')
        speak('Hello, Good Morning')

    elif hour>=12 and hour<18:

        print('Hello, Good Afternoon')
        speak('Hello, Good Afternoon')
    else:

        print('Hello, Good Evening')
        speak('Hello, Good Evening')

# wishMe()
```

Setting up the command function for your AI assistant :

Define a function **takecommand** for the AI assistant to understand and to accept human language. The microphone captures the human speech and the recognizer recognizes the speech to give a response.

The exception handling is used to handle the exception during the run time error and,the **recognize_google** function uses google audio to recognize speech.

```
def takecommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening....")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognising...")
        statement = r.recognize_google(audio, 
language='en-in')
        print(f"User said: {statement}\n")

    except Exception as e:
        print('say that again please...')
        speak('say that again please...')
        return "None"
    return statement

print("Loading your AI persoanl assistant My-
Assistant001")
speak("Loading your AI persoanl assistant My-
Assistant001")
wishMe()
```

The Main function:

The main function starts from here, the commands given by the humans is stored in the variable **statement**.

```
if __name__ == '__main__':

    while True:
        print("Tell me how can I help you sir?")
        speak("Tell me how can I help you sir?")
        statement = takecommand().lower()

        if statement == 0:
            continue
```

If the following trigger words are there in the statement given by the users it invokes the virtual assistant to speak the below following commands.

```
# ok bye assistant
elif "good bye" in statement or "ok bye" in statement or
"stop" in statement:

    print("Your personal assistant My-Assistant001 is
shutting down, Good bye")
    speak("Your personal assistant My-Assistant001 is
shutting down, Good bye")

    break
```

Operations:

1. Fetching data from Wikipedia:

The following commands helps to extract information from wikipedia. The **wikipedia.summary()** function takes two arguments, the statement given by the user and how many sentences from wikipedia is needed to be extracted is stored in a variable **result.**

```
# 1. Fatching data from Wikipedia
if 'wikipedia' in statement:
   speak('Searching Wikipedia...')
   statement = statement.replace("wikipedia", "")
   results = wikipedia.summary(statement, sentences=3)
   speak("According to Wikipedia")
```

```
    print(results)
    speak(results)
```

2. Accessing the Web Browsers — Google chrome , G-Mail and YouTube:

The web browser extracts data from web. The **open_new_tab** function accepts **URL** as a parameter that needs to be accessed.

The **Python time sleep function** is used to add delay in the execution of a program. We can use this function to halt the execution of the program for given **time** in seconds.

```
# 2. Accessing the Web Browsers — Google chrome , G-Mail, Facebook and
YouTube

elif 'open youtube' in statement:
    webbrowser.open_new_tab("https://www.youtube.com")
    speak("youtube is open now")
    print("youtube is open now")
    time.sleep(5)

elif 'open google' in statement:
    webbrowser.open_new_tab("https://www.google.com")
    speak("Google chrome is open now")
    print("Google chrome is open now")
    time.sleep(5)

elif 'open gmail' in statement:
    webbrowser.open_new_tab("gmail.com")
    speak("Google Mail open now")
    print("Google Mail open now")
    time.sleep(5)

elif 'open facebook' in statement:
    webbrowser.open_new_tab("facebook.com")
    speak('facebook is open now')
    print('facebook is open now')
    time.sleep(5)
```

3 -Predicting time:

The current time is abstracted from **datetime.now()** function which displays the hour, minute and second and is stored in a variable name **strTime.**

```
# 3. Predicting time
elif 'time' in statement:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"the time is {strTime}")
    print(f"the time is {strTime}")
```

4 -To fetch latest news:

If the user wants to know the latest news , The voice assistant is programmed to fetch top headline news from Time of India by using the web browser function.

```
# 4. To fetch latest news
elif 'news' in statement:
    news =
webbrowser.open_new_tab("https://timesofindia.indiatimes.com/home/headlin
es")
    speak('Here are some headlines from the Times of India,Happy reading')
    time.sleep(6)
```

5 -Capturing photo:

The **ec.capture()** function is used to capture images from your camera. It accepts 3 parameter.

Camera index — The first connected webcam will be indicated as index 0 and the next webcam will be indicated as index 1.

Window name — It can be a variable or a string. If you don't wish to see the window, type as False.

Save name — A name can be given to the image and if you don't want to save the image, type as false.

```
# 5. Capturing photo

elif "camera" in statement or "take a photo" in statement:
    ec.capture(0, "jarvis camera", "img.jpg")
```

6 -Searching data from web:

From the **web browser** you can **search** required data by passing the user statement (command) to the **open_new_tab()** function.

User: *please search images of computer*

The Voice assistant opens the google window & fetches computer images from web.

```
# 6. Searching data from web

elif 'search' in statement:
    statement = statement.replace("search", "")
    webbrowser.open_new_tab(statement)
    time.sleep(5)
```

7- Setting your AI assistant to answer geographical and computational questions:

Here we can use a third party API called **Wolfram alpha API** to answer computational and geographical questions. It is made possible by the Wolfram Language. The **client** is an instance (class) created for wolfram alpha. The **res** variable stores the response given by the wolfram alpha.

```
# 7. Setting your AI assistant to answer geographical and computational questions

elif 'ask' in statement or 'find' in statement:
   speak('I can answer to computational and geographical questions  and what
question do you want to ask now')
   print('I can answer to computational and geographical questions  and what
question do you want to ask now')
   question = takecommand()
   app_id = "app id" # wolframalpha account unique app id paste here.
   client = wolframalpha.Client('R2K75H-7ELALHR35X')
   res = client.query(question)
   answer = next(res.results).text
   print(answer)
   speak(answer)
```

User: *What is the capital of California?*

My-Assistant001: *Sacramento, United States of America*

8- To forecast weather:

The **city_name variable** takes the command given by the human using the **takeCommand()** function.

The **get** method of **request** module returns a **response** object. And the **json** methods of response object converts json format data into python format.

The variable **X** contains list of nested dictionaries which checks whether the value of ‘COD’ is 404 or not that is if the city is found or not.

The values such as temperature and humidity is stored in the main key of variable **Y**.

```
# 8. To forecast weather

elif "weather" in statement:
    api_key = "Apply your unique id"
    base_url = "https://api.openweathermap.org/data/2.5/weather?"
    speak("what is the city name")
    city_name = takecommand()
    complete_url = base_url+"appid="+api_key+"&q="+city_name
    response = requests.get(complete_url)
    x = response.json()
    if x["cod"]!="404":
        y = x['main']
        current_temperature = y["temp"]
        current_humidiy = y["humidity"]
        z = x["weather"]
        weather_description = z[0]["description"]
        speak(" Temperature in kelvin unit is " +
              str(current_temperature) +
              "\n humidity in percentage is " +
              str(current_humidiy) +
              "\n description  " +
              str(weather_description))
        print(" Temperature in kelvin unit = " +
              str(current_temperature) +
              "\n humidity (in percentage) = " +
              str(current_humidiy) +
              "\n description = " +
              str(weather_description))
```

User: *I want to get the weather data*

My-Assistant001: *What is the city name?*

User: *Delhi*

My-Assistant001**:** *Temperature in kelvin unit is 304.09 , Humidity in percentage is 64 and Description is light rain.*

9 – Calculate mathematical problems

```
# 9. Calculator
elif 'calculate' in statement.lower():
   app_id = "K9PU3W-439VPGYUQU"
   clint = wolframalpha.Client(app_id)
   indx = statement.lower().split().index('calculate')
   query = statement.split()[indx + 1:]
   res = clint.query(' '.join(query))
   answer = next(res.results).txt
   print('the answer is' + answer)
   speak('the answer is' + answer)
```

10- Write & Show notes:
With this using, you can wtite notes and show notes also.

```
# 10. Write & Show notes
elif "write a note" in statement:
    speak("What should i write, sir")
    note = takecommand()
    file = open('data.txt', 'w')
    speak("Sir, Should i include date and time")
    snfm = takecommand()
    if 'yes' in snfm or 'sure' in snfm:
        strTime = datetime.datetime.now().strftime("%H:%M:%S")
        file.write(strTime)
        file.write(" :- ")
        file.write(note)
    else:
        file.write(note)

elif "show notes" in statement:
    speak("Showing Notes")
    file = open("data.txt", "r")
    print(file.read(6))
    speak(file.read(6))
```

11- Extra features:

AI assistant to answer the following questions like what it can and who created it etc.

```
# 11. basic daily routine questions

elif 'how are you' in statement:
   speak("I am fine. Thank You")
   print("I am fine. Thank You")
   speak("How are you sir?")

elif 'good' in statement or 'fine' in statement:
   speak("its good to know that you are fine")
   print("its good to know that you are fine")

elif 'who are you' in statement or "define yourself" in statement:
   print('''Hello, I am your personal assistant. My name is My-Assistant001. My
technology is a
   based on artificial intelligence. And my latest version is 0.0.1 . I am here to
make your life easier.
   You can command me to perform various tasks such as calculating sums,
opening applications and search
    anything in a second's etcetra.''')

   speak('''Hello, I am your personal assistant. My name is My-Assistant001. My
technology is a
   based on artificial intelligence. And my latest version is 0.0.1 . I am here to
make your life easier.
   You can command me to perform various tasks such as calculating sums,
opening applications and search
    anything in a second's etcetra.''')


elif "who made you" in statement or "who created you" in statement:
   print("I have been created by The Computer Science Engineer - Mister
Shubham Singh Chouhan.")
   speak("I have been created by The Computer Science Engineer - Mister
Shubham Singh Chouhan.")
```

12- Basic Operation related to the system like shut down, restart, lock windows etc.

```
# 12. basic operations on system
elif 'lock windows' in statement:
   speak("locking windows")
   ctypes.windll.user32.lockWorkStation()

elif 'shutdown windows' in statement:
   speak("shutting down windows")
   subprocess.call('shutdown / p /f')

elif 'hibernate windows' in statement:
   speak("hibernating windows")
   subprocess.call('shutdown / h')

elif 'restart' in statement:
   subprocess.call(["shutdown", "/r"])
```

13- Some other operations perform like play music, send mail, & listen jokes etc.

```
# 13. Some Another operations like play music, send mail, & listen jokes

elif 'play music' in statement or "play song" in statement:
   speak("Here you go with music")
   print("Here you go with music")
   # music_dir = "G:\\Song"
   music_dir = "E:\Music"
   songs = os.listdir(music_dir)
   print(songs)
   random = os.startfile(os.path.join(music_dir, songs[0]))


elif 'send mail to friend' in statement:
   try:
      msg = EmailMessage()
      msg['Subject'] = "Check out new mail"
```

```
        msg['From'] = 'sender mail'
        msg['To'] = 'receiver mail'
        msg.set_content('Hello, My dear friend! How are you? ')
    except Exception as e:
        print(e)
        speak("sorry sir, i am not able to send this email")


elif 'jokes' in statement:
    joke = speak(pyjokes.get_joke())
    print(joke)
```

14- Open an application MS Word , & many more.

```
#14. Open any application
elif "open sublime" in statement or "open sublime text" in statement:
    speak("Here, you go to sublime text")
    print("Here, you go to sublime text")
    codepath = "path here"
    os.startfile(codepath)

elif "open word" in statement or "open msword" in statement:
    speak("Opening Microsoft Word")
    print("Opening Microsoft Word")
    codepath = "C:/ProgramData/Microsoft/Windows/Start
Menu/Programs/Microsoft Office/Microsoft Office Word 2007"
    os.startfile(codepath)

elif "open excel" in statement or "open msexcel" in statement:
    speak("Opening Microsoft Excel")
    print("Opening Microsoft Excel")
    codepath = "C:/ProgramData/Microsoft/Windows/Start
Menu/Programs/Microsoft Office/Microsoft Office Excel 2007"
    os.startfile(codepath)

# else:
#     speak("Application not available")
#     print("Application not available")
#     return
```

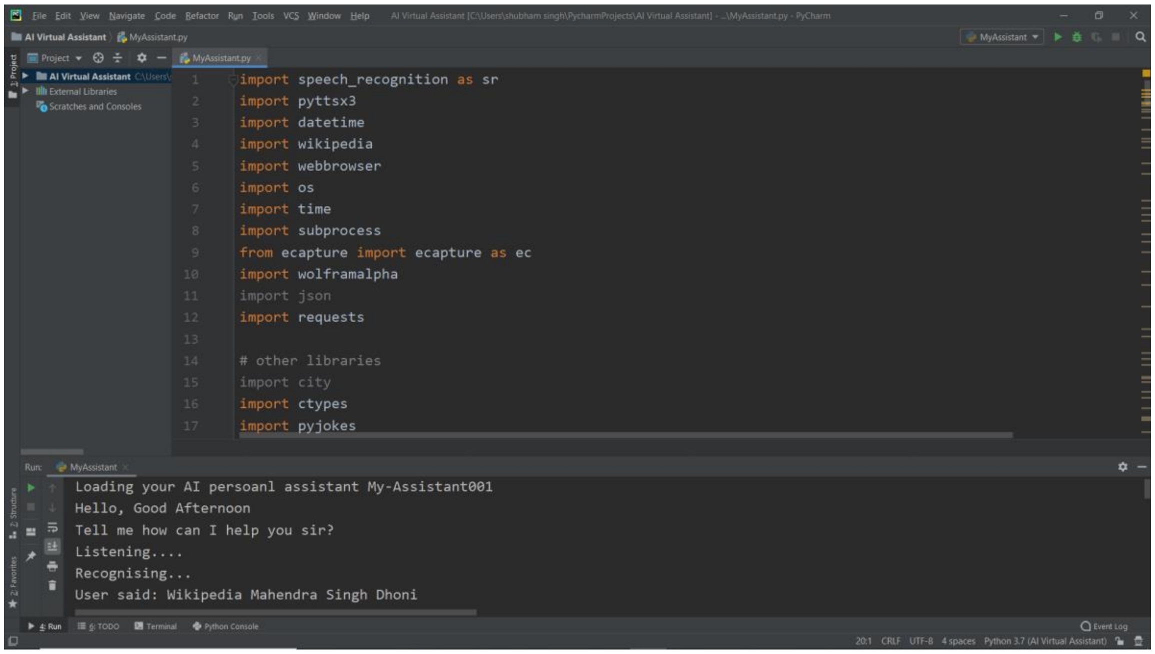
15. To log off your PC:

The **subprocess.call()** function here is used to process the system function to log off or to turn off your PC. This invokes your AI assistant to automatically turn off your PC.

```
   # 15. To log off your PC
   elif "log off" in statement or "sign out" in statement:
        speak("Ok , your pc will log off in 10 sec make sure you exit from all
applications")
        subprocess.call(["shutdown", "/l"])

time.sleep(3)
```

Screenshot's of Testing

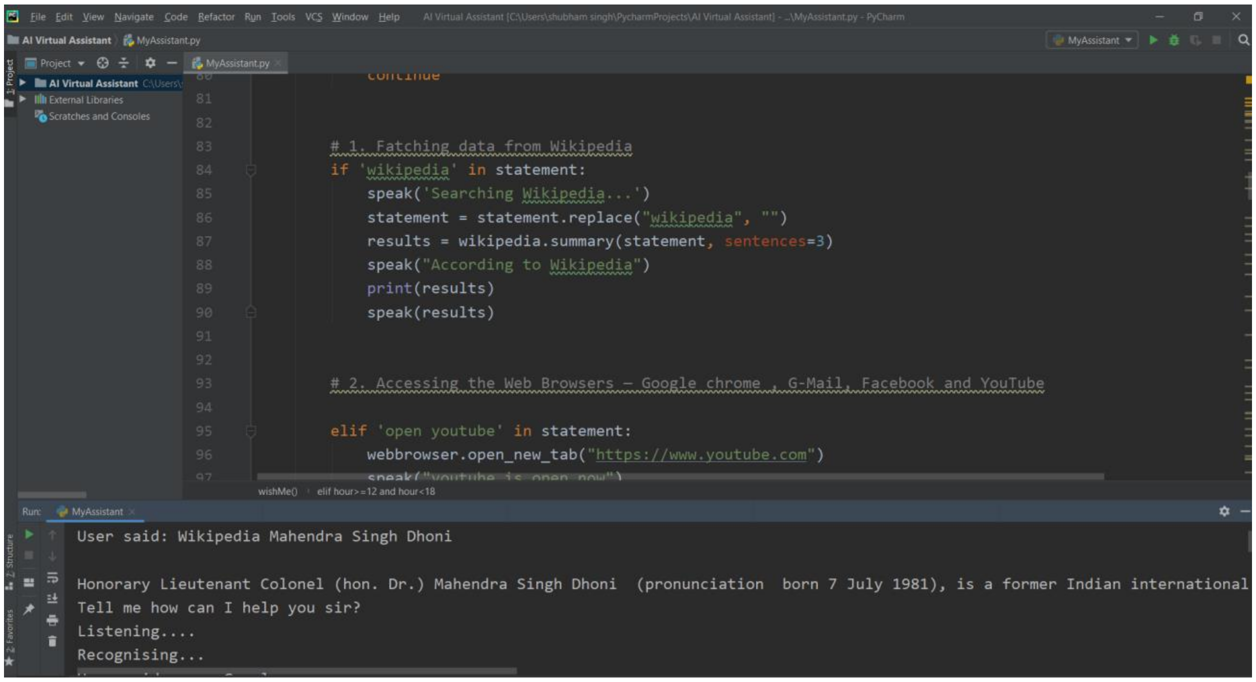
Formal talks:



```
import speech_recognition as sr
import pyttsx3
import datetime
import wikipedia
import webbrowser
import os
import time
import subprocess
from ecapture import ecapture as ec
import wolframalpha
import json
import requests

# other libraries
import city
import ctypes
import pyjokes
```

```
Loading your AI persoanl assistant My-Assistant001
Hello, Good Afternoon
Tell me how can I help you sir?
Listening....
Recognising...
User said: Wikipedia Mahendra Singh Dhoni
```

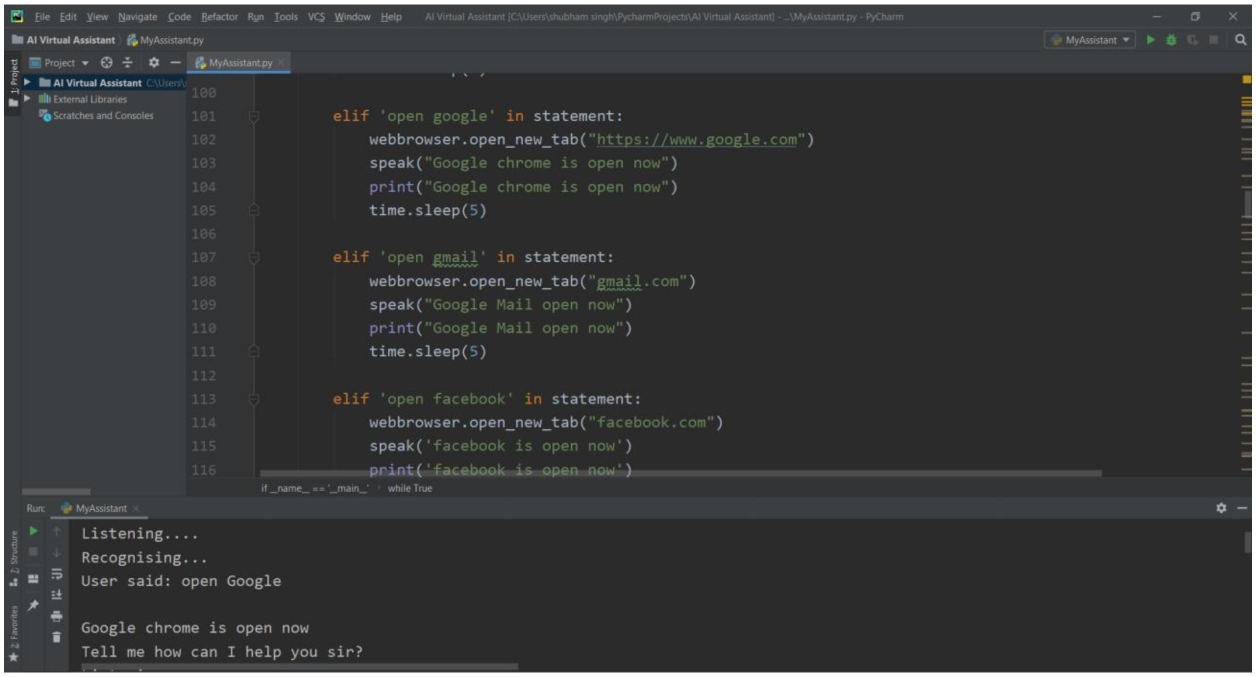
Search Wikipedia:



```
            continue


        # 1. Fatching data from Wikipedia
        if 'wikipedia' in statement:
            speak('Searching Wikipedia...')
            statement = statement.replace("wikipedia", "")
            results = wikipedia.summary(statement, sentences=3)
            speak("According to Wikipedia")
            print(results)
            speak(results)


        # 2. Accessing the Web Browsers — Google chrome , G-Mail, Facebook and YouTube

        elif 'open youtube' in statement:
            webbrowser.open_new_tab("https://www.youtube.com")
            speak("youtube is open now")
```

```
User said: Wikipedia Mahendra Singh Dhoni

Honorary Lieutenant Colonel (hon. Dr.) Mahendra Singh Dhoni  (pronunciation  born 7 July 1981), is a former Indian international
Tell me how can I help you sir?
Listening....
Recognising...
```

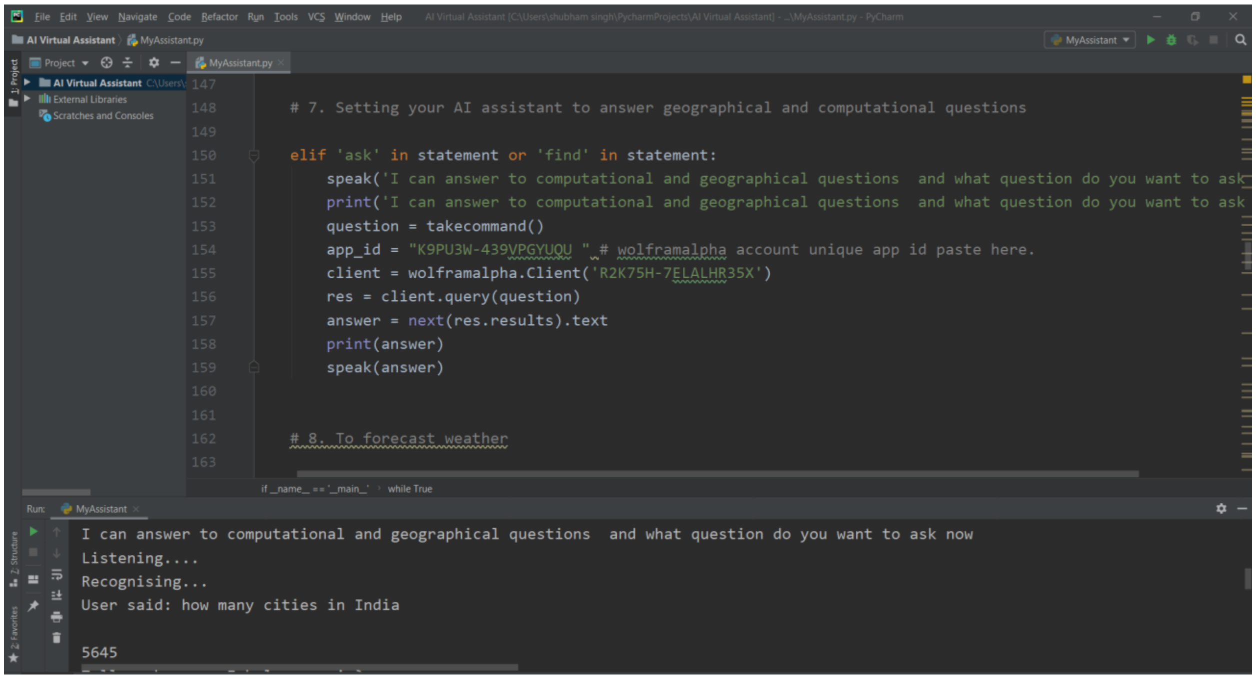
Open Google:



```
        elif 'open google' in statement:
            webbrowser.open_new_tab("https://www.google.com")
            speak("Google chrome is open now")
            print("Google chrome is open now")
            time.sleep(5)

        elif 'open gmail' in statement:
            webbrowser.open_new_tab("gmail.com")
            speak("Google Mail open now")
            print("Google Mail open now")
            time.sleep(5)

        elif 'open facebook' in statement:
            webbrowser.open_new_tab("facebook.com")
            speak('facebook is open now')
            print('facebook is open now')
```

```
Listening....
Recognising...
User said: open Google

Google chrome is open now
Tell me how can I help you sir?
```

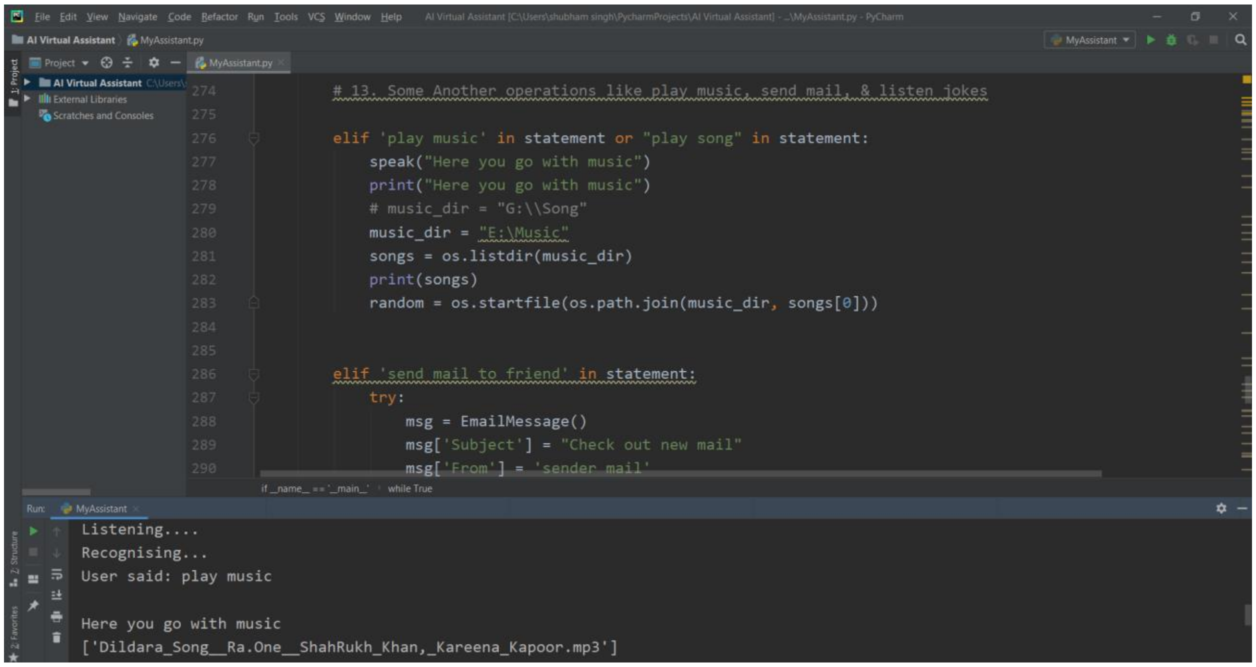
Ask computational or geographical questions:



```
# 7. Setting your AI assistant to answer geographical and computational questions

elif 'ask' in statement or 'find' in statement:
    speak('I can answer to computational and geographical questions  and what question do you want to ask
    print('I can answer to computational and geographical questions  and what question do you want to ask
    question = takecommand()
    app_id = "K9PU3W-439VPGYUQU " # wolframalpha account unique app id paste here.
    client = wolframalpha.Client('R2K75H-7ELALHR35X')
    res = client.query(question)
    answer = next(res.results).text
    print(answer)
    speak(answer)


# 8. To forecast weather
```

```
I can answer to computational and geographical questions  and what question do you want to ask now
Listening....
Recognising...
User said: how many cities in India

5645
```

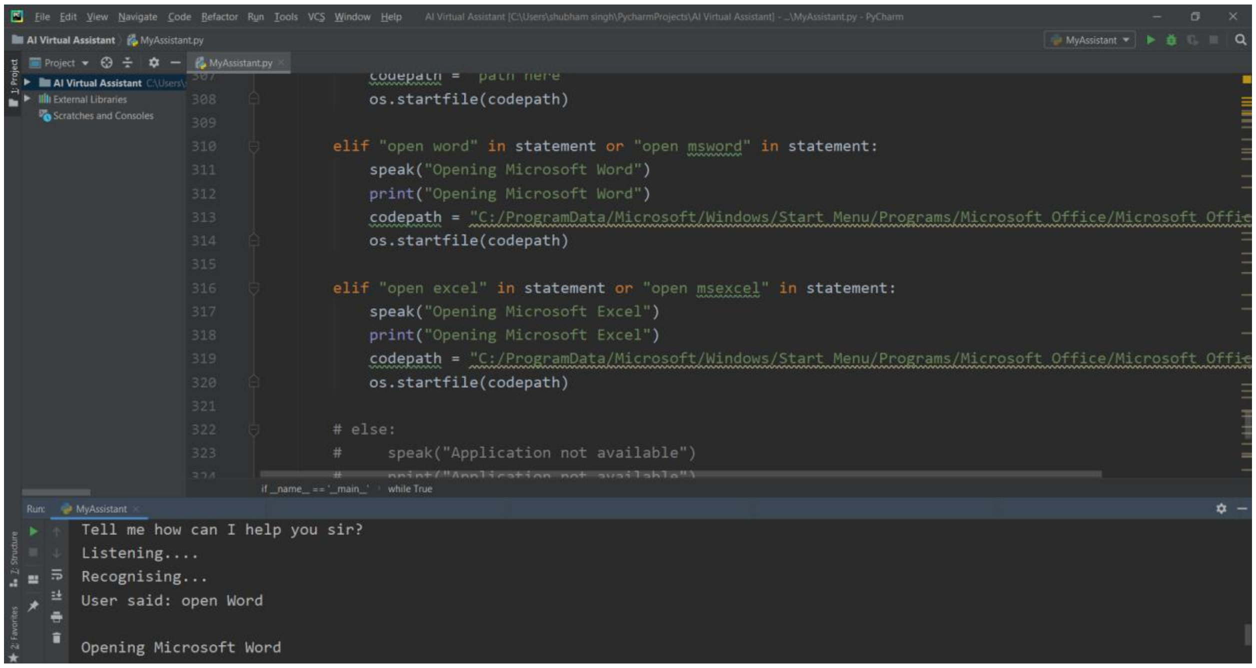
Play Music:



```
    # 13. Some Another operations like play music, send mail, & listen jokes

    elif 'play music' in statement or "play song" in statement:
        speak("Here you go with music")
        print("Here you go with music")
        # music_dir = "G:\\Song"
        music_dir = "E:\Music"
        songs = os.listdir(music_dir)
        print(songs)
        random = os.startfile(os.path.join(music_dir, songs[0]))


    elif 'send mail to friend' in statement:
        try:
            msg = EmailMessage()
            msg['Subject'] = "Check out new mail"
            msg['From'] = 'sender mail'
```

```
Listening....
Recognising...
User said: play music

Here you go with music
['Dildara_Song__Ra.One__ShahRukh_Khan,_Kareena_Kapoor.mp3']
```

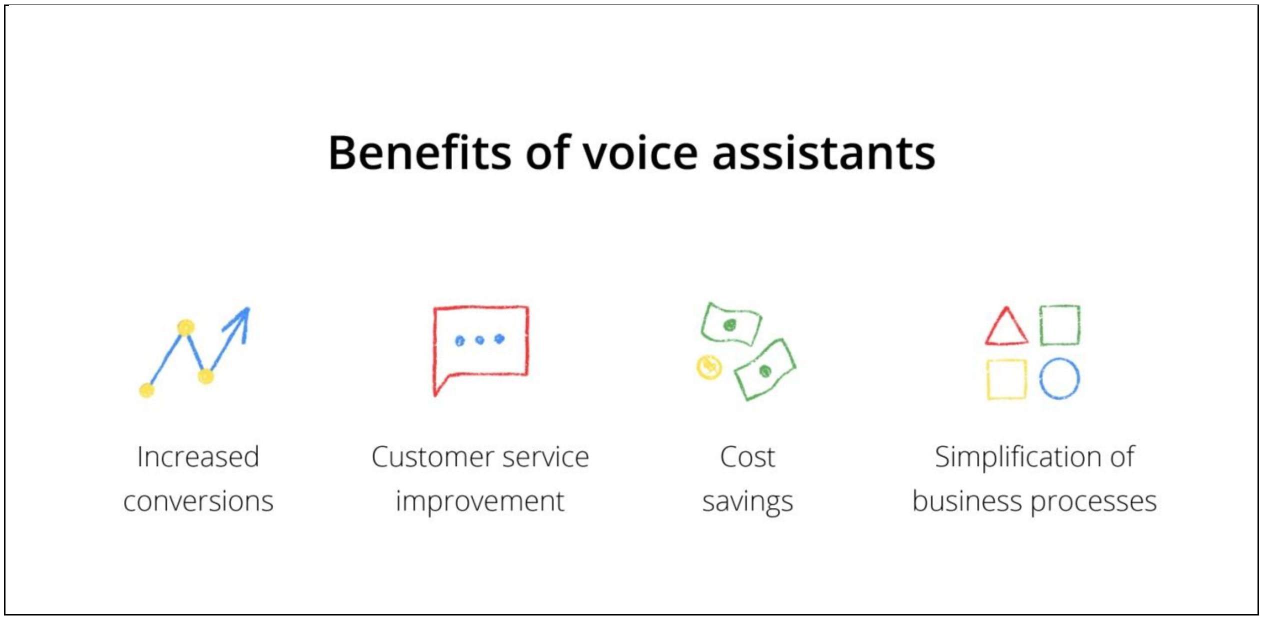
Open Application:



```
            codepath = "path here"
            os.startfile(codepath)

        elif "open word" in statement or "open msword" in statement:
            speak("Opening Microsoft Word")
            print("Opening Microsoft Word")
            codepath = "C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Microsoft Office/Microsoft Offic
            os.startfile(codepath)

        elif "open excel" in statement or "open msexcel" in statement:
            speak("Opening Microsoft Excel")
            print("Opening Microsoft Excel")
            codepath = "C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Microsoft Office/Microsoft Offic
            os.startfile(codepath)

        # else:
        #     speak("Application not available")
        #     print("Application not available")
```

```
Tell me how can I help you sir?
Listening....
Recognising...
User said: open Word

Opening Microsoft Word
```

Advantage’s of Virtual Voice Assistant:


Benefits of voice assistants
Increased conversions
Customer service improvement
Cost savings
Simplification of business processes

Project URL:

- https://github.com/shubhamsc11/AI-Virtual-Assistant

Reference's URL:

- https://www.geeksforgeeks.org/build-a-virtual-assistant-using-python/
- https://www.geeksforgeeks.org/voice-assistant-using-python/
- https://www.geeksforgeeks.org/personal-voice-assistant-in-python/
- https://realpython.com/
- https://pypi.org/

Thank You