

PROFESSIONAL TRAINING REPORT
AT
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By

AHILA R R

(Reg#: 38120007)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119, TAMIL NADU

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI 600 119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AHILA R R (Reg.no.38120007)** who carried out the project entitled **“AI-Virtual Assistant Using Python”** under our supervision from June 2021 to November 2021.

INTERNAL GUIDE

Dr. S. Gowri, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva Voce Examination held on -----

Internal Examiner

External Examiner

DECLARATION

I, **AHILA R R** hereby declare that the project report entitled **“AI-VIRTUAL ASSISTANT USING PYTHON”** done by me under the guidance of **Dr. S. Gowri, M.E., Ph.D.**, at “Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119” is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering/Technology degree in Information Technology.

DATE: 12.11.2021

PLACE: CHENNAI

SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **Sathyabama** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing** and **Dr. R. Subhashini M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Gowri, M.E., Ph.D.**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the **Department of Information Technology** who was helpful in many ways for the completion of the project.

ABSTRACT
AI VIRTUAL ASSISTANT USING PYTHON

My project is **AI VIRTUAL ASSISTANT**. It includes the additional information about how to use the library packages, and how to install the packages in command prompt and the modules. Some files are installed using “python extension files”. Without packages or modules, the program doesn't run in your coding path.

And I used the platform visual studio code which is user friendly, so I prefer Am (both Bluetooth and System Mic) and translates it into computer comprehensible language, providing the user with the appropriate solutions and answers. This tool connects to the internet to deliver results that the user has questioned. The Natural Language Processing algorithm enables computers to communicate with one another using natural human language in a variety of ways.

TABLE OF CONTENTS:

<u>CHAPTER NO:</u>	<u>TITLE</u>	<u>PAGE NO</u>
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF ABBREVIATIONS	iii
1	INTRODUCTION	9
	1.1 ILLUSTRATION	9
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	10
	2.1 AIM	10
	2.2 SCOPE	10
	2.3 AVAILABILITY	10
3	EXPERIMENTAL MATERIALS OR METHODS USED:	11
	3.1 PACKAGES REQUIRED	11
	3.1.1 SPEECH RECOGNITION	11
	3.1.2 PYTTX3	11
	3.1.3 WIKIPEDIA	11
	3.1.4 DATETIME	11
	3.1.5 OS	11
	3.1.6 TIME	11
	3.1.7 WEBBROWSER	11

	3.1.8 JSON	12
	3.1.9 REQUEST	12
	3.1.10 WOLFRAM ALPHA	12
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS:	13
	4.1 COMMAND WORK	13
	• Run and wait	13
	• Wish me command	13
	• Take command	13
	4.2 SYSTEM WORKFLOW	14
		15
5	SUMMARY AND CONCLUSIONS	16
6	REFERENCES	17
	• BOOK REFERENCES	17
	• PAPER REFERENCES	18
7	APPENDIX	19
	A. SOURCE CODE	19-23
	B. SCREENSHOTS	24-32

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
1	WORKFLOW	15

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
AI	Artificial Intelligence
IOT	Internet Of Things
VA	Virtual Assistant
ML using PY	Machine Learning using Python

CHAPTER 1

INTRODUCTION

AI virtual assistant who saves your time by listening to your voice. Which takes your way easier? Most of the VAs like SIRI, OK GOOGLE, CORTANA are the voice assistants developed in the year of 2016. So, I did my new virtual assistant name called "STORM".

1.1 ILLUSTRATION ABOUT VA:

The rise of automation, along with increased computational power, novel application of statistical algorithms, and improved accessibility to data, have resulted in the birth of the personal digital assistant market, popularly represented by Apple's Siri, Microsoft's Cortana, Google's Google Assistant, and Amazon's Alexa. While each assistant may specialize in slightly different tasks, they all seek to make the user's life easier through verbal interactions, so you don't have to search out a keyboard to find answers to questions like "What's the weather today?" or "Where is Switzerland?".

Despite the inherent "cool" factor that comes with using a digital assistant, you may find that the digital assistants don't cater to your specific needs. Fortunately, it's relatively easy to build your own.

I thought it would be cool to create a personal assistant in Python. If you are into movies, you may have heard of Jarvis, an A.I. based character in the Iron Man films. In this tutorial we will create a robot.

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM:

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases.

The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “What can I do for you?” and then responds to verbal input.

2.2 SCOPE:

The implemented voice assistant can perform the following task it can open YouTube, wolfram alpha, Google chrome and stack Facebook. Predict current time, current date, search Wikipedia to abstract required data, play song. The following queries of the voice assistant can be manipulated as per the users need.

2.3 APPLICABILITY:

The mass number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user's life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous.

CHAPTER 3

EXPERIMENTAL MATERIALS OR METHODS USED:

3.1 PACKAGES REQUIRED:

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

3.1.1 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.

3.1.2 PYTTX3: is a text to speech conversion library in python. This package supports text to speech engines on Mac OS x, Windows and on Linux.

3.1.3 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 required from Wikipedia.

3.1.4 DATE TIME: This is an inbuilt module in python, and it works on date and time.

3.1.5 OS: This module is a standard library in python, and it provides the function to interact with operating system.

There are two types of OS,

- **OS-SYS**
- **OS-WIN**

OS-sys:

The os module contains two sub-modules os. sys (same as sys) and os. path that are dedicated to the system and directories; respectively. Whenever possible, you should use the functions provided by these modules for file, directory, and path manipulations

OS-win:

Hyper-V library for OpenStack projects. This library contains Windows / Hyper-V specific code commonly used in OpenStack projects. The library can be used in any other OpenStack projects where it is needed.

3.1.6 Time: The time module helps us to display time

3.1.7 Web browser: This is an in-built package in python. It extracts data from the web

3.1.8 Json: The json module is used for storing and exchanging data.

3.1.9 Request: The request module is used to send all types of HTTP request. It's accepts URL as parameters and gives access to the given URL'S.

3.1.10 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.

CHAPTER 4

RESULTS AND DISCUSSION PERFORMANCE ANALYSIS

4.1 COMMAND WORK

The speech engine or voice engine:

The pytsx3 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

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

4.1.1 RUN AND WAIT:

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.

4.1.2 WISH ME:

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 human". If the hour is greater than 12 and less than 16, the voice assistant wishes you with the following message "Good Afternoon human". If the hour is greater than 16 and less than 19, the voice assistant wishes you with the following message

“Good Evening human” Else it voices out the message “Good Night human”

4.1.3 TAKE COMMAND:

Define a function `take_command` 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.

STEP :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`.

STEP: 2.

EXTRA FEATURES:

It would be interesting to program your AI assistant to answer the following questions like what it can and who created it, isn't it?

STEP: 3.

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.

STEP :4.

PREDICTING TIME:

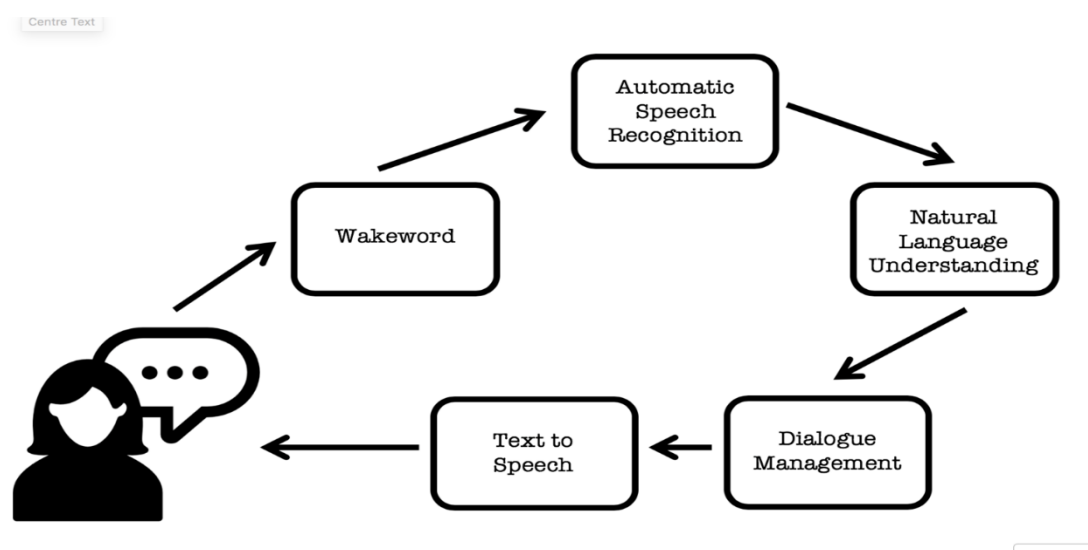
The current time is abstracted from `datetime.now ()` function which displays the hour, minute and second and is stored in a variable name `strTime`.

STEP :5.

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. To access the wolfram alpha API a unique App ID is required which can be generated by the following ways:

4.2 WORKFLOW:



CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 SUMMARY:

When a virtual assistant is not able to answer questions accurately, it's because it lacks the proper context or doesn't understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviours.

They require large amount of information to be fed for it to work efficiently. Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks, and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions to paths, and it should be able to consider user preferences, other active tasks, priorities to recommend a particular plan.

CHAPTER 6

REFERENCES

WEBSITES REFERRED

- www.stackoverflow.com
- www.pythonprogramming.net
- www.codecademy.com
- www.tutorialspoint.com
- www.google.co.in

BOOKS REFERRED

- Python Programming - Kiran Gurbani
- Learning Python - Mark Lutz

YOUTUBE CHANNELS REFERRED

- CS Dojo
- Edureka!
- Documents referred

PAPERS REFFERED

1.Saadman Shahid Chowdury, Atiar Talukdar, Ashik Mahmud and Tanzilur Rahman, Domain specific Intelligent personal assistant with bilingual voice command processing, IEEE, 2018.

2.EV Polyakov, MS Mazhanov, AY Voskov, LS Kachalova, MV and SV Polyakov, "Investigation and development of the intelligent voice assistant for the IOT using machine learning", Moscow workshop on electronic technologies, 2018.

3.Khawir Mahmood, Tausfer Rana and Abdur Rehman Raza, "Singular adaptive multi role intelligent personal assistant (SAM-IPA) for human computer interaction", International conference on open source system and technologies, 2018.

4.Veton Kepuska and Gamal Bohota, "Next generation of virtual assistant (Microsoft Cortana Apple Siri Amazon Alexa and Google Home)", IEEE conference, 2018.

5.Piyush Vashishta, Juginder Pal Singh, Pranav Jain and Jitendra Kumar, "Raspberry PI based voice-operated personal assistant", International Conference on Electronics And Communication and Aerospace Technology ICECA, 2019.

6.Laura BURbach, Patrick Halbach, Nils Plettenberg, Johannes Nakayama, Matrina Ziefle and Andre Calero Valdez, "Ok google Hey Siri Alexa. Acceptance relevant of virtual voice assistants", International communication conference IEEE, 2019.

CHAPTER 7

APPENDIX

A. SOURCE CODE

```
import pyttsx3
import datetime
import speech_recognition as sr
import wikipedia
import wolframalpha
import webbrowser
import os
from selenium import webdriver

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

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 human")
    elif hour >= 12 and hour < 16:
```

```

    speak("Good afternoon human")
elif hour>=16 and hour <19:
    speak("Good Evening human")
else:
    speak("good night human")

speak("I am your Assistant Storm")

```

```

def takeCommand():
    # takes my command from microphone and gives text
    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("user said : ", query)
    except Exception as e:
        print(e)
        speak("Sorry human, can you repeat that again?")
        return "None"
    return query

if __name__ == "__main__":
    wishMe()
    while True:
        speak("How can i help you?")

```

```

query = takeCommand().lower()
if 'wikipedia' in query:
    speak("searching in wikipedia")
    query = query.replace("wikipedia", "")
    results = wikipedia.summary(query, sentences = 2)
    speak("According to wikipedia")
    print(results)
    speak(results)

elif 'how are you' in query:
    speak("yeah i am fine and doing good ")
elif 'call' in query:
    speak("you can call me as Storm")
elif 'what is your name' in query:
    speak("hi buddy, I am your assistant Storm")
elif 'invented' in query:
    speak("the great master VISHNU NAIDU . he didn't invented me.
he gives me a birth...")
elif 'found' in query:
    speak("the great master VISHNU NAIDU . he didn't invented me.
he gives me a birth...")
elif 'facebook' in query:
    webbrowser.open("facebook.com")
    speak("facebook is opened")
elif 'youtube' in query:
    webbrowser.open("youtube.com")
    speak("youtube is opened")
elif 'twitter' in query:
    webbrowser.open("twitter.com")

```

```

    speak("twitter is opened")
elif 'instagram' in query:
    webbrowser.open("instagram.com")
    speak("instagram is opened")
elif 'gmail' in query:
    webbrowser.open("gmail.com")
    speak("gmail is opened")
elif "calculate" in query.lower():
    app_id= "H4G23Y-9RRHKK6W25"
    client = wolframalpha.Client(app_id)
    indx = query.lower().split().index('calculate')
    query = query.split()[indx + 1:]
    res = client.query(' '.join(query))
    answer = next(res.results).text
    speak("The answer is " + answer)
elif 'play music' in query:
    music_dir = 'C:/Users/my pc/Music'
    songs = os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir, songs[1]))
    speak("music is being played")
elif 'time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"the time is {strTime}")
elif 'open code' in query:
    codepath =
"C:\\Users\\human\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
    os.startfile(codepath)

```

elif 'the time' in query:

```
strTime = datetime.datetime.now().strftime("%I:%M:%S")
```

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

elif 'the date' in query:

```
year = int(datetime.datetime.now().year)
```

```
month = int(datetime.datetime.now().month)
```

```
date = int(datetime.datetime.now().day)
```

```
speak("the current Date is")
```

```
speak(date)
```

```
speak(month)
```

```
speak(year)
```

elif "chrome" in query:

```
os.startfile('C:/Program Files  
(x86)/Google/Chrome/Application/chrome.exe')
```

```
speak("chrome is opened")
```

elif "google" in query:

```
os.startfile('C:/Program Files  
(x86)/Google/Chrome/Application/chrome.exe')
```

```
speak("google is opened")
```

elif 'bye' in query:

```
speak("see you soon human")
```

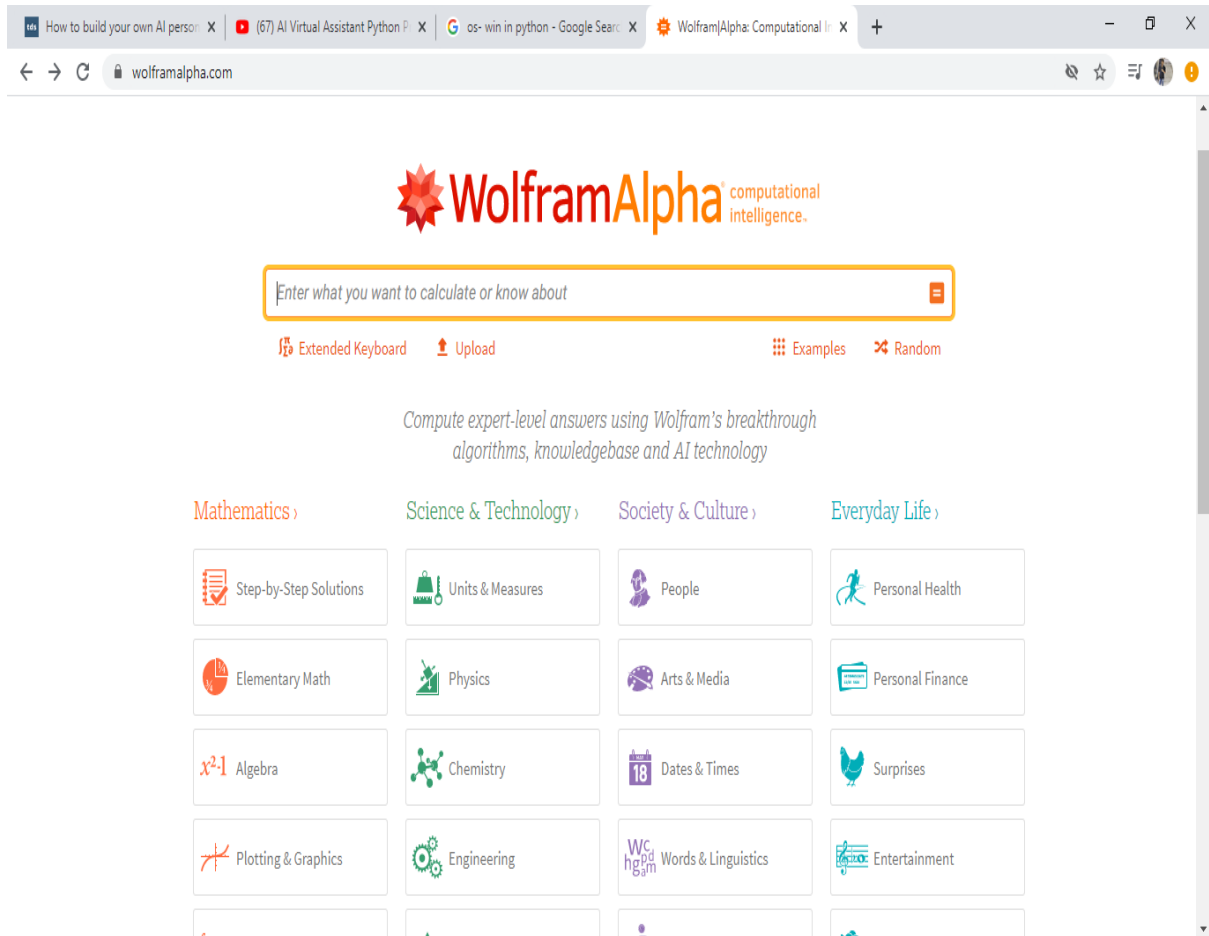
```
exit()
```

else :

```
webbrowser.open(query)
```

B. SCREENSHOTS:

Login to the official page of wolfram alpha and create an account if you do not possess one.



- Sign in using your wolfram ID



Enter Your Password

sinv.js2001@gmail.com

Password

.....

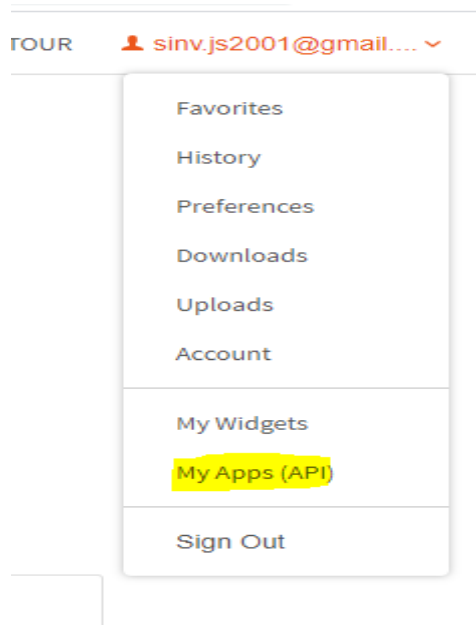
☒ Remember me

Sign in

Forgot your password?

Sign in to a different account.

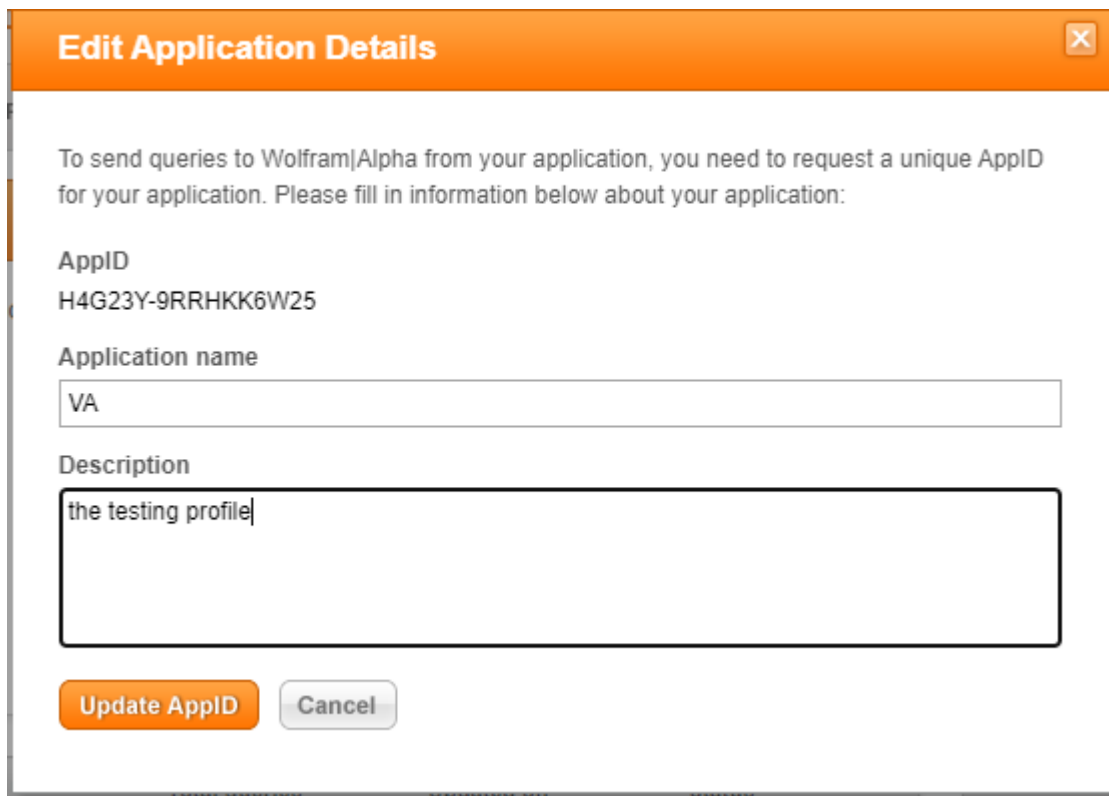
- Now you will view the homepage of the website.
- Head to the account section in the top right corner where you see your email. In the drop-down menu, select the My Apps (API) option.



- You will see this following window, now click Get APP_ID button



- Now you will get the following dialog box, give a suitable name and description and click the App ID button, an App ID will be generated, and this is an unique ID. Using the App Id use can access the Wolfram alpha API.



Edit Application Details

To send queries to Wolfram|Alpha from your application, you need to request a unique AppID for your application. Please fill in information below about your application:

AppID
H4G23Y-9RRHKK6W25

Application name
VA

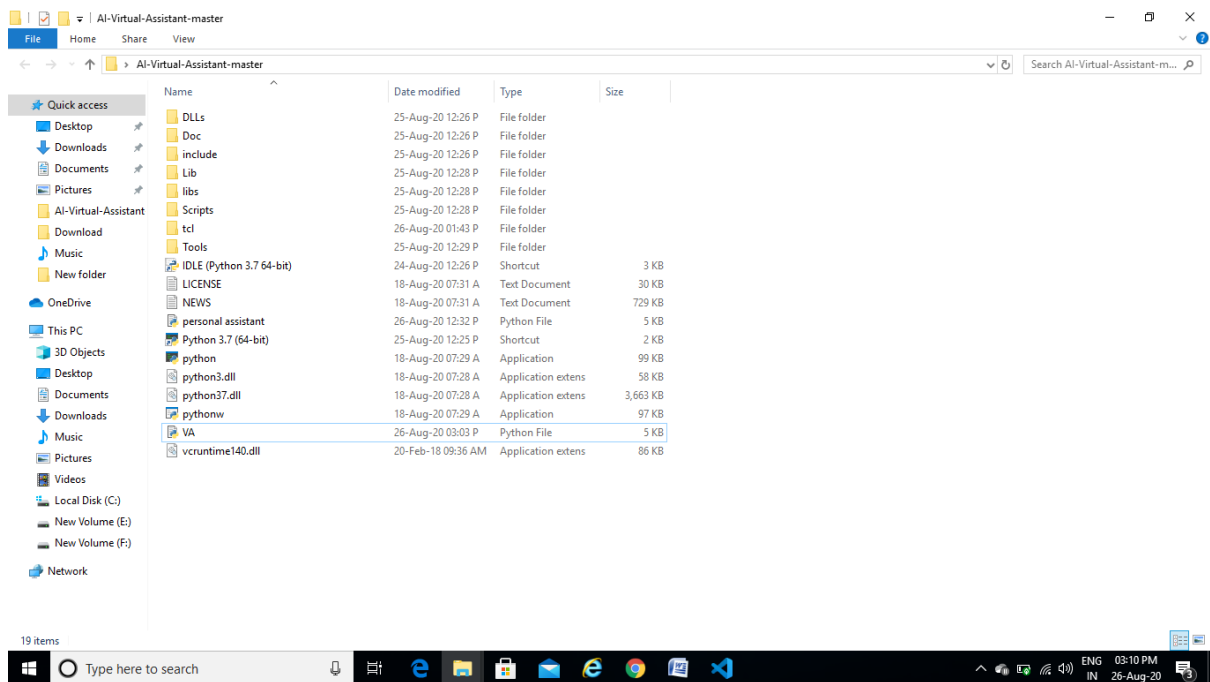
Description
the testing profile

Update AppID Cancel

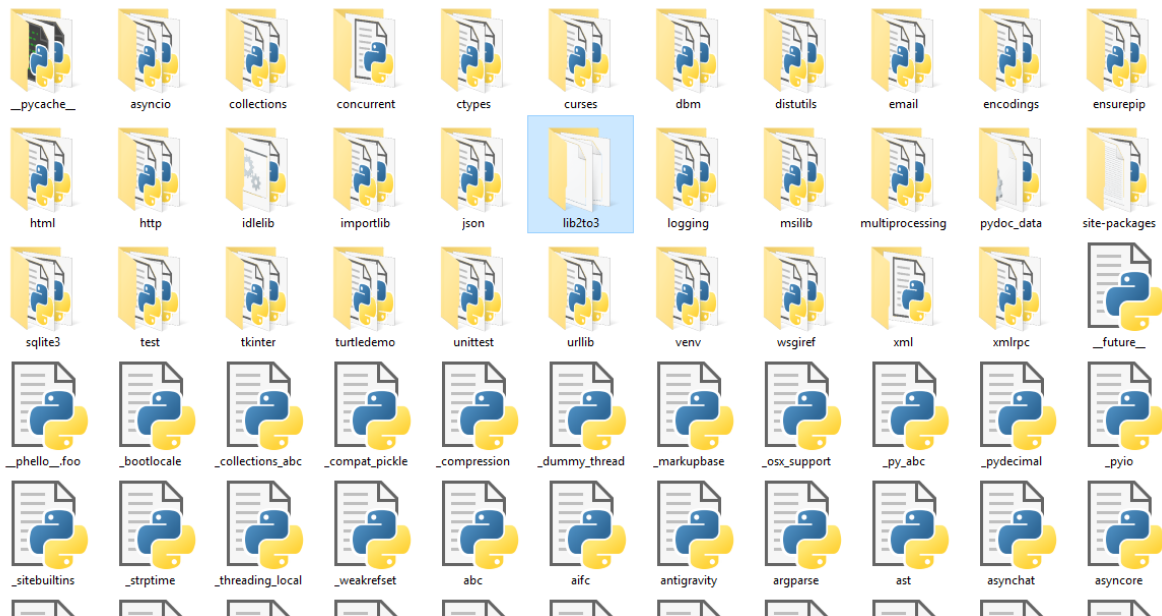
- To get out from assistant:

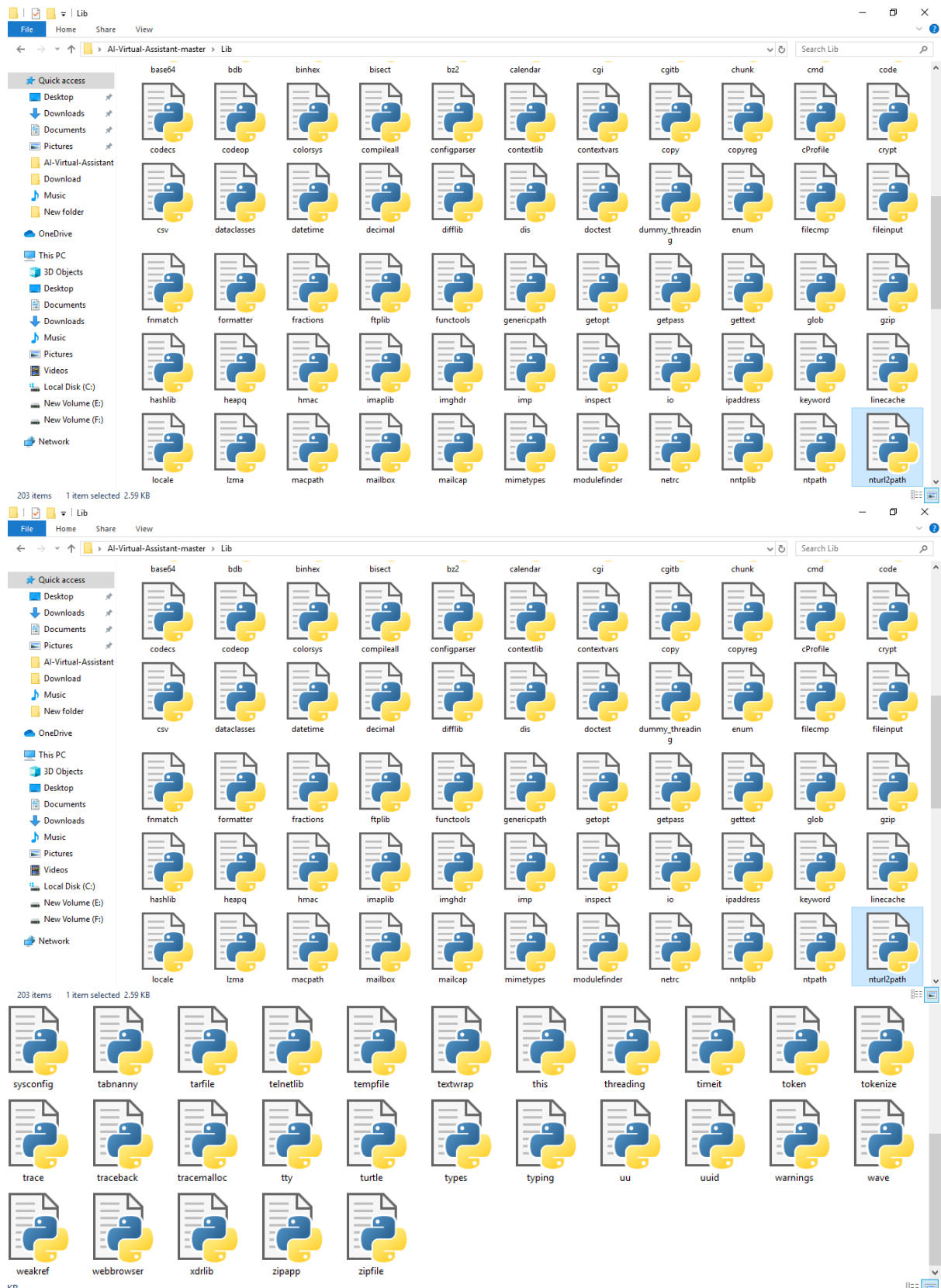
The word **“BYE”** here is used to process the system function to log off or to turn off your assistant. This invokes your AI assistant to automatically turn off your VA.

FILES AND PLATFORM:

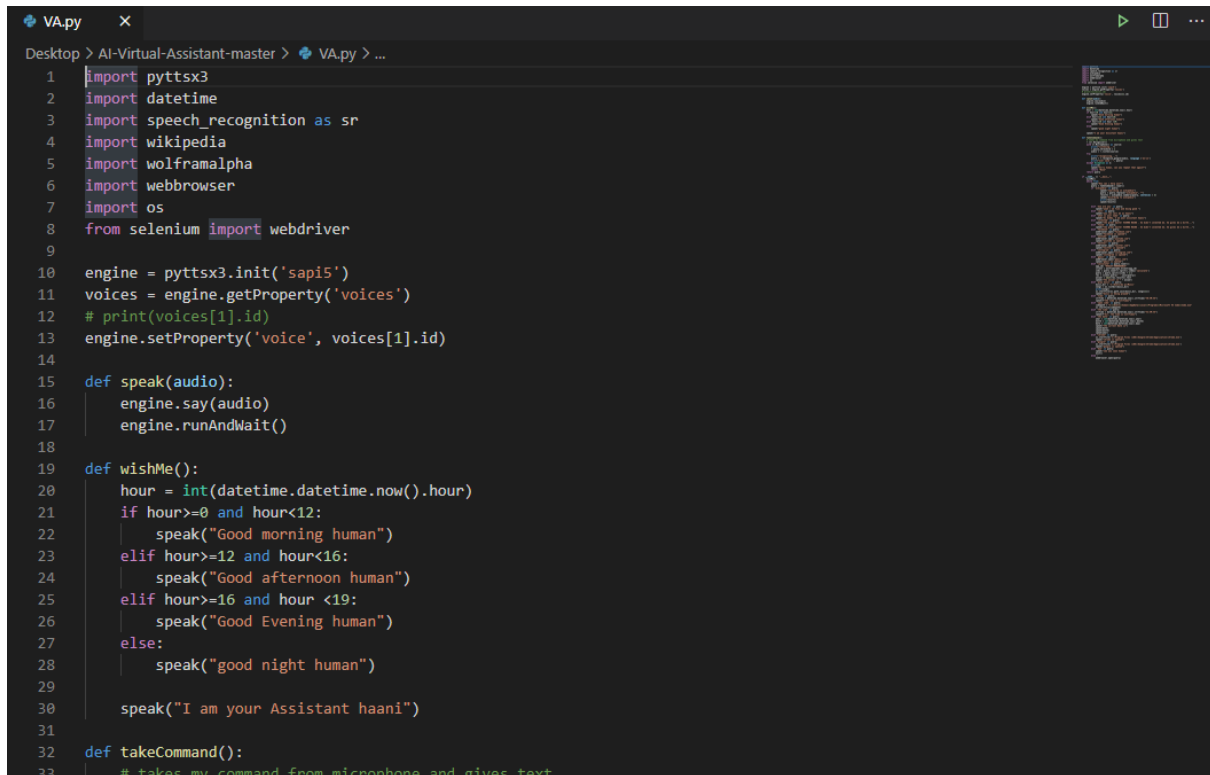


LIBRARIES -> (PACKAGES)





CODING SCREENSHOT:



The screenshot shows a code editor window titled 'VA.py' with a dark theme. The code is a Python script for a virtual assistant named 'haani'. It includes imports for pyttsx3, datetime, speech_recognition, wikipedia, wolframalpha, webbrowser, and os. It also imports webdriver from selenium. The script initializes a text-to-speech engine using pyttsx3 and sets the voice to 'sapi5'. It defines three functions: 'speak(audio)' which uses the engine to speak the provided audio; 'wishMe()' which checks the current time and speaks appropriate greetings or a self-introduction; and 'takeCommand()' which is a placeholder for taking commands from the microphone. The code is as follows:

```
1 import pyttsx3
2 import datetime
3 import speech_recognition as sr
4 import wikipedia
5 import wolframalpha
6 import webbrowser
7 import os
8 from selenium import webdriver
9
10 engine = pyttsx3.init('sapi5')
11 voices = engine.getProperty('voices')
12 # print(voices[1].id)
13 engine.setProperty('voice', voices[1].id)
14
15 def speak(audio):
16     engine.say(audio)
17     engine.runAndWait()
18
19 def wishMe():
20     hour = int(datetime.datetime.now().hour)
21     if hour>=0 and hour<12:
22         speak("Good morning human")
23     elif hour>=12 and hour<16:
24         speak("Good afternoon human")
25     elif hour>=16 and hour <19:
26         speak("Good Evening human")
27     else:
28         speak("good night human")
29
30     speak("I am your Assistant haani")
31
32 def takeCommand():
33     # takes my command from microphone and gives text
```

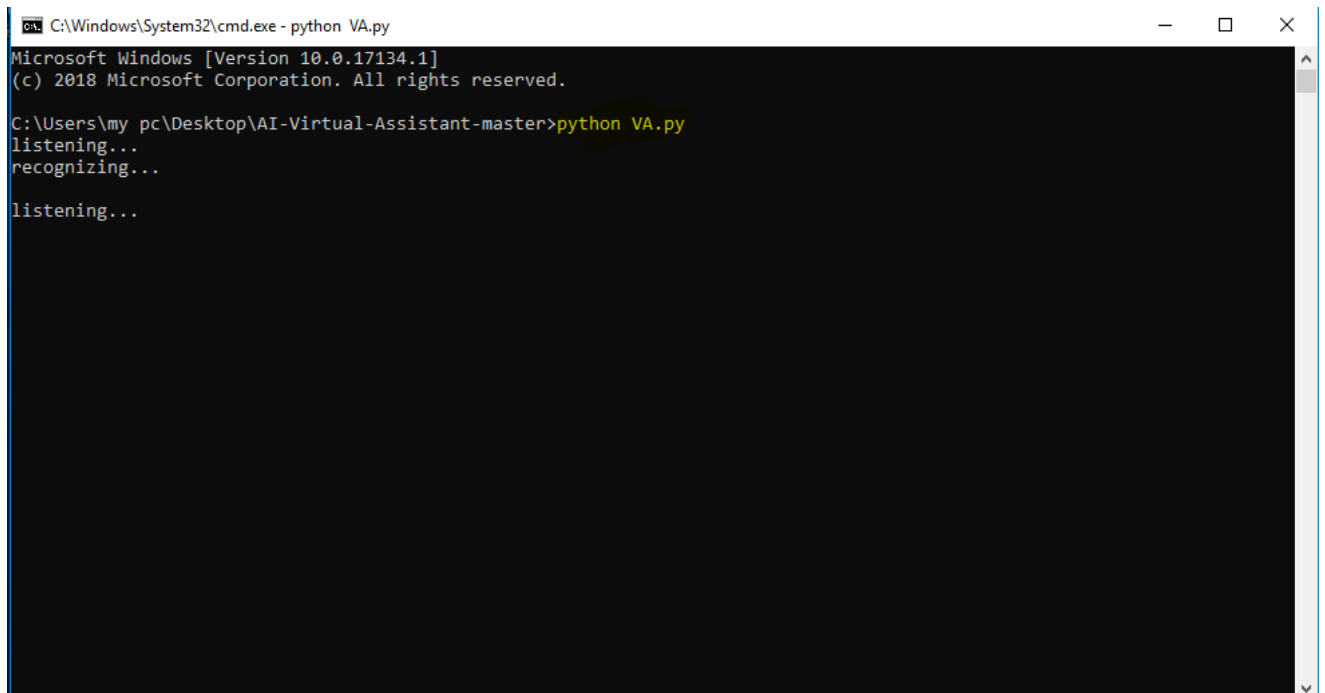
```
VA.py x
Desktop > AI-Virtual-Assistant-master > VA.py > takeCommand

32 def takeCommand():
33     # takes my command from microphone and gives text
34     r = sr.Recognizer()
35     with sr.Microphone() as source:
36         print("listening...")
37         r.pause_threshold = 1
38         audio = r.listen(source)
39     try:
40         print("recognizing...")
41         query = r.recognize_google(audio, language='en-in')
42         print("user said : ", query)
43     except Exception as e:
44         print(e)
45         speak("Sorry human, can you repeat that again?")
46         return "None"
47     return query
48
49 if __name__ == "__main__":
50     wishMe()
51     while True:
52         speak("How can i help you?")
53         query = takeCommand().lower()
54         if 'wikipedia' in query:
55             speak("searching in wikipedia")
56             query = query.replace("wikipedia", "")
57             results = wikipedia.summary(query, sentences = 2)
58             speak("According to wikipedia")
59             print(results)
60             speak(results)
61
62         elif 'how are you' in query:
63             speak("yeah i am fine and doing good ")
64         elif 'call' in query:
```

```
VA.py x
Desktop > AI-Virtual-Assistant-master > VA.py > ...

94         speak("The answer is " + answer)
95     elif 'play music' in query:
96         music_dir = 'C:/Users/my pc/Music'
97         songs = os.listdir(music_dir)
98         print(songs)
99         os.startfile(os.path.join(music_dir, songs[1]))
100         speak("music is being played")
101     elif 'time' in query:
102         strTime = datetime.datetime.now().strftime("%H:%M:%S")
103         speak(f"the time is {strTime}")
104     elif 'open code' in query:
105         codepath = "C:/Users/human/AppData/Local/Programs/Microsoft VS Code/Code.exe"
106         os.startfile(codepath)
107     elif 'the time' in query:
108         strTime = datetime.datetime.now().strftime("%I:%M:%S")
109         speak(f"Sir, the time is {strTime}")
110     elif 'the date' in query:
111         year = int(datetime.datetime.now().year)
112         month = int(datetime.datetime.now().month)
113         date = int(datetime.datetime.now().day)
114         speak("the current Date is")
115         speak(date)
116         speak(month)
117         speak(year)
118     elif "chrome" in query:
119         os.startfile('C:/Program Files (x86)/Google/Chrome/Application/chrome.exe')
120         speak("chrome is opened")
121     elif "google" in query:
122         os.startfile('C:/Program Files (x86)/Google/Chrome/Application/chrome.exe')
123         speak("google is opened")
124     elif 'bye' in query:
125         speak("see you soon human")
126         exit()
```

AND THE WAY TO EXECUTE THE CODE:



```
C:\Windows\System32\cmd.exe - python VA.py
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\my pc\Desktop\AI-Virtual-Assistant-master>python VA.py
listening...
recognizing...
listening...
```

(Or)

PY LAUNCHER:



```
C:\Windows\py.exe
listening...
```

THANK YOU 😊