"A.I. VOICE ASSISTANT USING PYTHON"

MAJOR PROJECT

Submitted in Partial Fulfilment of

The Requirements for The Award of The Degree

Of

BACHELOR OF TECHNOLOGY

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted BY

SOUVIK SAHA (Roll : 27000317019 , Reg. No. : 172700110100)

SABYASACHI DHAR (Roll : 27000317064, Reg. No: 172700110220)

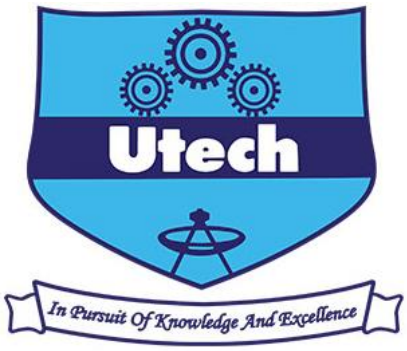
INDRANI MAJUMDER (Roll : 27000317049 , Reg. No. : 172700110070)

ROUNAK CHOWDHURY (Roll : 27000317035 ,Reg. No. :172700110089)

Under the guidance of

BIPA DUTTA

(Assistant Professor of ECE Dept.)



MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

BRAINWARE GROUP OF INSTITUTIONS – S.D.E.T.,

398, RAMKRISHNAPUR ROAD, BARASAT (NEAR JAGADIGHATA MARKET) KOLKATA 700124

“A.I. VOICE ASSISTANT USING PYTHON”

MAJOR PROJECT

Submitted in Partial Fulfilment of

The Requirements for The Award of The Degree

Of

BACHELOR OF TECHNOLOGY

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted BY

SOUVIK SAHA (Roll : 27000317019 , Reg. No. : 172700110100)

SABYASACHI DHAR (Roll : 27000317064, Reg. No: 172700110220)

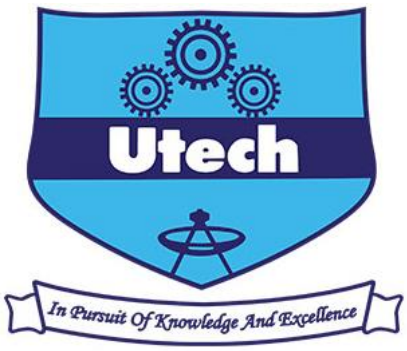
INDRANI MAJUMDER (Roll : 27000317049 , Reg. No. : 172700110070)

ROUNAK CHOWDHURY (Roll : 27000317035 ,Reg. No. :172700110089)

Under the guidance of

BIPA DUTTA

(Assistant Professor of ECE Dept.)



MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

BRAINWARE GROUP OF INSTITUTIONS – S.D.E.T.,

398, RAMKRISHNAPUR ROAD, BARASAT (NEAR JAGADIGHATA MARKET) KOLKATA 700124

CERTIFICATE

It is certified that the work contained in the project report titled "**A.I. VOICE ASSISTANT USING PYTHON",** by Souvik Saha, has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor(s)

Bipa Dutta

Electronics & Communication Engineering

B.G.I. – S.D.E.T., KOLKATA -125

March, 2021

Co-guide: Rajkumar Mandal

Arnab Das

Signature of HOD/TIC

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in this Major Project entitled, **"A.I. VOICE ASSISTANT USING PYTON"** submitted to **MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, Kolkata - 700064** in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING,** is an authentic record of my own work carried out from DECEMBER, 2020 to MARCH, 2021 under the supervision of **Mr./Mrs. BIPA DUTTA, Assistant Professor of E.C.E. Dept., BRAINWARE GROUP OF INSTITUTIONS – S.D.E.T., Kolkata - 700125.**

The matter embodied in this project report has not been submitted by me for the award of any other degree.

Signature

Place: KOLKATA

SOUVIK SAHA

Date: 27.02.2021

Univ.Roll NO. 27000317019

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

BIPA DUTTA

.....

(Asstt. Professor, E.C.E. Dept.)

Project Guide

ACKNOWLEDGEMENTS

Acknowledgement to Director, principal, HOD, Project coordinator, guide and others. It is our great fortune that we have got opportunity to carry out this project work under the supervision of **Mrs. Bipa Dutta** in the Department of **Electronics & Communication Engineering**, Brainware Group of Institutions – S.D.E.T., 398, Ramkrishnapur Road, Barasat (Near Jagadighata Market), Kolkata-700124, affiliated to Maulana Abul Kalam Azad University of Technology (**MAKAUT**), West Bengal, India. We express our sincere thanks and deepest sense of gratitude to our guide for her constant support, unparalleled guidance and limitless encouragement.

We would also like to thanks our co-guide **Mr. Rajkumar Mandal,** Assistance Professor in the Department of **Electronics & Communication Engineering**, Brainware Group of Institutions – S.D.E.T. for his constant support, unparalleled guidance and limitless encouragement.

We would also like to convey our gratitude to all the faculty members and staffs of the Department of Electronics & Communication Engineering, Brainware Group of Institutions – S.D.E.T. for their whole hearted cooperation to make this work turn into reality.

We are very thankful to our Department and to the authority of Brainware Group of Institutions – SDET for providing all kinds of infrastructural facility towards the research work.

Thanks to the fellow members of our group for working as a team.

Souvik saha (27000317019)

Sabyasachi dhar (27000317064)

Indrani Majumder (27000317049)

Rounak Choudhury (27000317035)

ASBTRACT - AI VOICE ASSISTANT

The implemented voice assistant can perform the following task it can open YouTube, Gmail, Google chrome

Predict current time, take a photo, search Wikipedia to abstract required data, predict weather in different cities , can answer computational and geographical questions too. Speech recognition is the process ofconverting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called **Speech Recognition** to allow us to convert audio into text for further processing.

The following queries of the voice assistant can be manipulated as per the users need.

Subprocess:- This module is used for getting system subprocess details which are used in variouscommands i.e Shutdown, Sleep, etc. This module comes buit-in with Python.

Wolframalpha:- It is used to compute expert-level answers using Wolfram's algorithms, knowledgebaseand AI technology. To install this module type the below command in the terminal.

pip install wolframaplha

Pyttsx3:- This module is used for conversion of text to speech in a program it works offline. To install thismodule type the below command in the terminal.

pip install pyttsx3

Tkinter:- This module is used for building GUI and comes inbuit with Python. This module comes buit - inwith Python.

Wikipedia:- As we all know Wikipedia is a great source of knowledge just like GeeksforGeeks we haveused Wikipedia module to get information from Wikipedia or to perform Wikipedia search. To install this module type the below command in the terminal.

pip install wikipedia

Speech Recognition:- Since we're building an Application of voice assistant, one of the mostimportant things in this is that your assistant recognizes your voice (means what you want to say/ ask). To install this module type the below command in the terminal.

Pip install SpeechRecognition

Web browser:- To perform Web Search. This module comes buit-in with Python.

Ecapture:- To capture images from your Camera. To install this module type the below command in theterminal.

pip install ecapture

Pyjokes:- Pyjokes is used for collection Python Jokes over the Internet. To install this module type thebelow command in the terminal.

pip install pyjokes

Datetime:- Date and Time is used to showing Date and Time. This module comes built-int with Python.

Twilio:- Twilio is used for making call and messages. To install this module type the below command inthe terminal.

pip install twilio

Requests: Requests is used for making GET and POST requests. To install this module type the belowcommand in the terminal.

pip install requests

BeautifulSoup: Beautiful Soup is a library that makes it easy to scrape information from web pages. Toinstall this module type the below command in the terminal.

pip install beautifulsoup4

Selenium : Python language bindings for Selenium WebDriver.

The selenium package is used to automate web browser interaction from Python. With selenium we can scrape web pages.

To install selenium we have to type

Pip install selenium

Open CV : OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposed to the C-based OpenCV 1.x API

Pip install openCV

Os module : This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module

It's a inbuilt module so , we don't need to install it.

CONTENTS

CHAPTER 1:

INTRODUCTION

1.1 INTRODUCTION:

A virtual assistant, also called an AI assistant or digital assistant, is an application program that understands

natural language voice commands and completes tasks for the user.

"We all know what is Virtual Assistant.

If you don't, don't worry, open your mobile and say "Ok Google" or "Hey Siri". Well, Google

Assistant, Siri, Alexa all these are example of Virtual Assistant."

Who doesn't want to have the luxury to own an assistant who always listens for your call, anticipates your every need, and takes action when necessary? That luxury is now available thanks to artificial intelligence-based voice assistants.

Voice assistants come in somewhat small packages and can perform a variety of actions after hearing your command. They can turn on lights, answer questions, play music, place online orders and do all kinds of AI-based stuff.

Voice assistants are not to be confused with virtual assistants, which are people who work remotely and can, therefore, handle all kinds of tasks. Rather, voice assistants are technology based. As voice assistants become more robust, their utility in both the personal and business realms will grow as well.

1.2 **OVERVIEW**

Nowadays, it isn't surprising to hear someone speak to someone that isn't there. We ask Alexa for the

weather and to turn the temperature down on the thermostat. Then, we ask Siri what our schedule for the day is and to call people. We are connected now more than ever using our voice and voice interface technology. We can't imagine doing things manually anymore! It's truly the future.

What is a Voice Assistant?

A voice assistant or intelligent personal assistant is a software agent that can perform tasks or services for an individual based on verbal commands i.e. by interpreting human speech and respond via synthesized voices.

Users can ask their assistants' questions, control home automation devices, and media playback via voice, and manage other basic tasks such as email, to-do lists, open or close any application etc with verbal commands.

1.3 **EXPLANATION WITH EXAMPLE**:

Let me give you the example of Braina (Brain Artificial) which is an intelligent personal assistant, human language interface, automation and voice recognition software for Windows PC. Braina is a multi-functional AI software that allows you to interact with your computer using voice commands in most of the languages of the world. Braina also allows you to accurately convert speech to text in over 100 different languages of the world.

In recent times, Voice assistants got the major platform after Apple integrated the most astonishing Virtual Assistant — Siri which is officially a part of Apple Inc. But the timeline of greatest evolution began with the year 1962 event at the Seattle World Fair where IBM displayed a unique apparatus was the actual size

of a shoebox and could perform scientific functions and can perceive 16 words and also speak them in the human recognizable voice with 0 to 9 numerical digits.

During the period of the 1970s, researchers at Carnegie Mellon University in Pittsburgh, Pennsylvania — with the considerable help of the U.S Department of Defence and its Defence Advanced Research Projects Agency (DARPA) — made Harpy. It could understand almost 1,000 words, which is approximately the vocabulary of a three-year-old child.

Big organizations like Apple and IBM sooner in the 90s started to make things that utilized voice acknowledgment. In 1993, Macintosh began to building speech recognition with its Macintosh PCs with PlainTalk.

In April 1997, Dragon NaturallySpeaking was the first constant dictation product which could comprehend around 100 words and transform it into readable content. 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.

Artificial intelligence technologies are beginning to be actively used in human life; this is facilitated by the appearance and wide dissemination of the Internet of Things (IOT). Autonomous devices are becoming smarter in their way to interact with both a human and themselves. New capacities lead to creation of various systems for integration of smart things into Social Networks of the Internet of Things. One of the relevant trends in artificial intelligence is the technology of recognizing the natural language of a human. New insights in this topic can lead to new means of natural human-machine interaction, in which the machine would learn how to understand human’s language, adjusting and interacting in it. One of such tools is voice assistant, which can be integrated into many other intelligent systems. In this paper, the principles of the functioning of voice assistants are described, its main shortcomings and limitations are given. The method of creating a local voice. assistant without using cloud services is described, which allows to significantly expand the applicability of such devices in the future.

The implemented voice assistant can perform the following task it can open YouTube, Gmail, and 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.

CHAPTER 2:

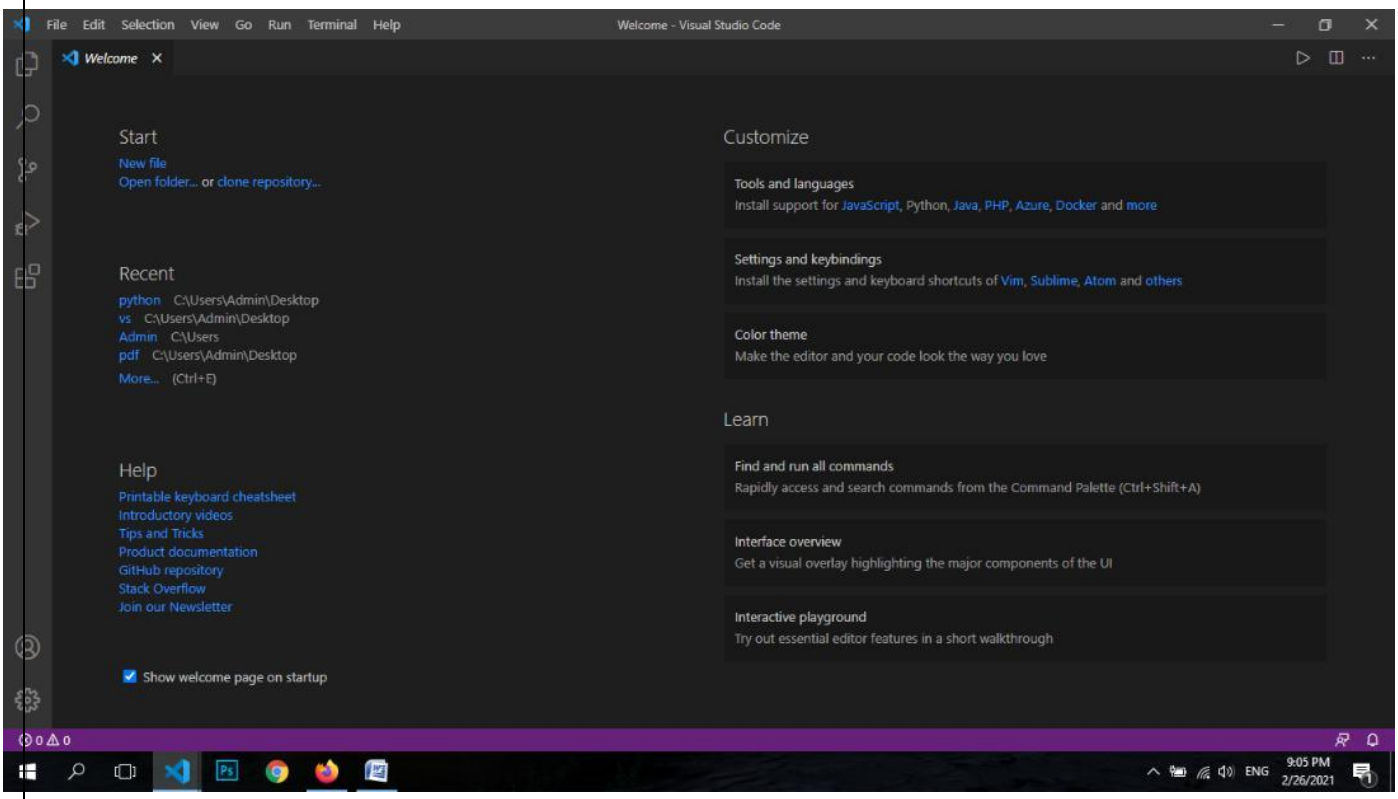
PROJECT WORK

2.1 Setting up the environment:

We used the VScode to code this up.

Here are some details about about VS code : Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). Begin our journey with VS Code.

P.S: we used 'MARSHMELLOW' name for our voice assistant.



Defining a speak function:

The first and foremost thing for an A.I. assistant is to speak. To make our bot talk, we will code a **speak()** function. This function will take audio as an argument, and then, it will pronounce it.

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
#print(voices[0].id)
engine.setProperty('voice',voices[0].id)
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

Now, the next thing we need is audio. So, we are going to install a module named **pyttsx3.**

2.2 WHAT IS PYTTSX3?

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.

The pyttsx3 module supports two voices first is female and the second is male which is provided by "sapi5" for windows.

It supports three TTS engines :

- *sapi5* – SAPI5 on Windows
- *nsss* – NSSpeechSynthesizer on Mac OS X
- *espeak* – eSpeak on every other platform

Installation:

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Admin\Desktop\python> pip install pyttsx3
```

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

Usage:

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

What is sapi5?

- Speech API developed by Microsoft.
- Helps in synthesis and recognition of voice

• The SpVoice object brings the text-to-speech (TTS) engine capabilities to applications using SAPI automation. An application can create numerous SpVoice objects, each independent of and capable of interacting with the others. An SpVoice object, usually referred to simply as a voice, is created with default property settings so that it is ready to speak immediately.

What Is Voice Id?

- Voice id helps us to select different voices.
- voice[0].id = *Male voice*
- voice[1].id = *Female voice*

Writing our speak() function

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

2.3 Creating our main function:

Now, we will create a **main()** function, and inside this **main()** function, we will call our speak function.

```
if __name__ == "__main__":
    speak('hello world')
```

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

Coding the wishme() function:

Now, we are going to make a **wishme()** function, that will make our A.I. wish or greet us according to the time on the computer.

To provide the current time to A.I., we need to import a module called datetime. Import this module to your program, by:

```
4    import datetime
```

Now, let's start defining our **wishme()** function: 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.

```
def wishMe():
    hour = int(datetime.datetime.now().hour)
    #speak(hour)
    print(hour)
    if hour >=0 and hour < 12:
        speak ("good morning .." )
    elif hour >=12 and hour <18:
        speak("good afternoon  .." )
    else:
        speak("good evening ")
```

2.4 Defining takeCommand() 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 our system. So, now we will code 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 us.

Before defining the **takeCommand()** function, we need to install a module called **speechRecognition.** Install this module by:

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

```
2   import speech_recognition as sr
```

Let's start coding our **takeCommand()** function:

```
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("listening.....")
        r.pause_threshold = 1
        r.operation_timeout = 10
        #r.dynamic_energy_threshold = True
        r.energy_threshold = 300
        # r.non_speaking_duration =
        r.operation_timeout = 5
        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')
    print(f" Boss Said : {query}\n")

except Exception as e:
    # print(e)
    speak(' i could not get it boss ... say that again')
    print("say that again...")
    return "none"
return query
```

2.5 id __name__ == __main__:

Before executing code, Python interpreter reads source file and define few special variables/global variables.

If the python interpreter is running that module (the source file) as the main program, it sets the special __name__ variable to have a value **"__main__"**. If this file is being imported from another module, __name__ will be set to the **module's name.** Module's name is available as value to __name__ global variable.

A module is a file containing Python definitions and statements. The file name is the module name with the suffix .py appended.

When we execute file as command to the python interpreter,

Advantages :

1. Every Python module has it's __name__ defined and if this is '__main__', it implies that the module is being run standalone by the user and we can do corresponding appropriate actions.
2. If you import this script as a module in another script, the __name__ is set to the name of the script/module.
3. Python files can act as either reusable modules, or as standalone programs.
4. if __name__ == "main": is used to execute some code **only** if the file was run directly, and not imported.

```
if __name__ == "__main__":
```

Task 1: To search something on Wikipedia

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

Wikipedia

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia.

Search Wikipedia, get article summaries, get data like links and images from a page, and more. Wikipedia wraps the MediaWiki API so you can focus on using Wikipedia data, not getting it.

Installing wikipedia module:

We need to type PIP install Wikipedia to install Wikipedia module.

```
4    import wikipedia
```

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

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

Now If well who is sharukh khan in wikipedia , it will give us answer

According to wikipedia……..

```
1
listening.....
Recognizing...
 Boss Said : he is Shahrukh Khan according to Wikipedia

According to wikipedia....
Shah Rukh Khan (pronounced ['ʃaːɦrʊx xaːn]; born 2 November 1965), also known by the initialism SRK, is an Indian actor, film p
roducer, and television personality. Referred to in the media as the "Baadshah of Bollywood" (in reference to his 1999 film Baa
dshah), "King of Bollywood" and "King Khan", he has appeared in more than 80 Hindi films, and earned numerous accolades, includ
ing 14 Filmfare Awards.
```

Task 2: To open YouTube in web-browser:

To open any website, we need to import a module called ***webbrowser.***

Web Browser :

The webbrowser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.

Under Unix, graphical browsers are preferred under X11, but text-mode browsers will be used if graphical browsers are not available or an X11 display isn't available. If text-mode browsers are used, the calling process will block until the user exits the browser.

If the environment variable BROWSER exists, it is interpreted as the os.pathsep-separated list of browsers to try ahead of the platform defaults. When the value of a list part contains the string %s, then it is interpreted as a literal browser command line to be used with the argument URL substituted for %s; if the part does not contain %s, it is simply interpreted as the name of the browser to launch.

For non-Unix platforms, or when a remote browser is available on Unix, the controlling process will not wait for the user to finish with the browser, but allow the remote browser to maintain its own windows on the display. If remote browsers are not available on Unix, the controlling process will launch a new browser and wait.

The script **webbrowser** can be used as a command-line interface for the module. It accepts a URL as the argument. It accepts the following optional parameters: -n opens the URL in a new browser window, if possible; -t opens the URL in a new browser page ("tab"). The options are, naturally, mutually exclusive. Usage example:

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.

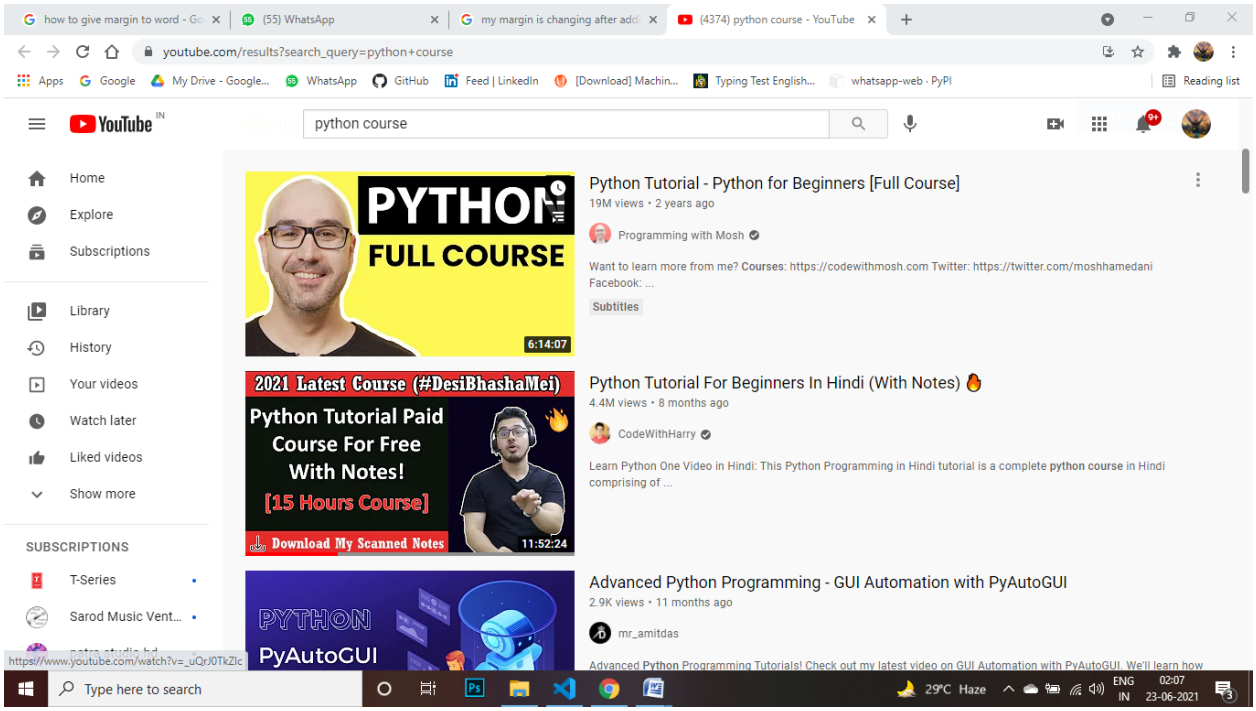
Code:

```
elif 'open youtube' in query:
    speak('what should i search boss??')
    print('what should i search boss??')
    m = takeCommand().lower()
    print(m)
    print('opening youtube...')
    speak(f'opening youtube and searching {m}')
    webbrowser.open('https://youtube.com/results?search_query=' + m)
```

Here, we are using the elif loop to check whether the Youtube is in the query of the user or not. Let' suppose, the user gives a command as "Please, open youtube."

So, open youtube will be in the user's query, and the elif condition will be true.

```
opening youtube...
listening.....
Recognizing...
listening.....
Recognizing...
 Boss Said : open YouTube

what should i search boss??
listening.....
Recognizing...
 Boss Said : Python course

python course
opening youtube...
listening.....
```



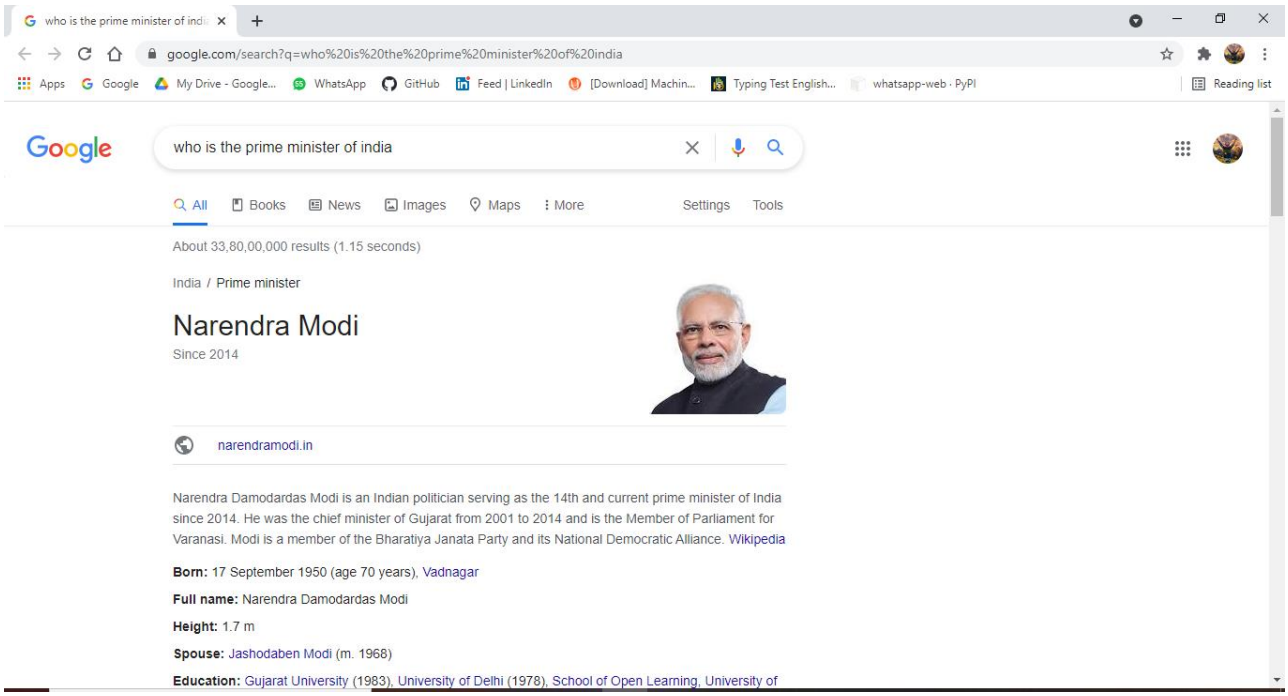
Task 3: To open Google site in a web-browser:

```
elif 'open google' in query:
    print('what should i search  boss??')
    speak('what should i search  boss??')
    z = takeCommand().lower()
    print(z)
    print('searching boss , wait a second')
    speak('searching boss , wait a second')
    webbrowser.open('https://google.com/#q=' + z)
```

We are opening Google in a web-browser by applying the same logic that we used while opening Youtube.

RESULT :

```
 Boss Said : open Google

what should i search  boss??
listening.....
Recognizing...
 Boss Said : who is the Prime Minister of India

who is the prime minister of india
searching boss , wait a second
listening.....
Recognizing...
listening.....
Recognizing...
listening.....
Recognizing...
```



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:\\my faV. songs'
    songs= os.listdir(music_dir)
    d= random.choice(songs)
    os.startfile(os.path.join(music_dir, d))
```

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. Then we used Random module to randomly select the song.

With the help of os.starfile, you can play any song of your choice. You can also play a random song with the help of a random module. Every time you command to play music, the A.I. will play any random song from the song directory.

Task 5: To know the current time:

Atfirst we need to import time module , here are some details about time module.

The **datetime**module supplies classes for manipulating dates and times.

While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

```
    elif "time" in query:
        strcurrent_time =datetime.datetime.now().strftime("%H:%M")
        print(f'the time is {strcurrent_time}')
        speak(f'boss , the time is {strcurrent_time}')
```

```
Initializing Marshmellow for Souvik.....
listening.....
Recognizing...
 Boss Said : wake up

2
listening.....
Recognizing...
listening.....
Recognizing...
 Boss Said : time

the time is 02:21
listening.....
```

In the above code, we are using **datetime()** function and storing the current time 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.

Task 6: Face recognition:

To do face recognition , we need to import a module named openCV , short name CV2

Here are some details about **openCV.**

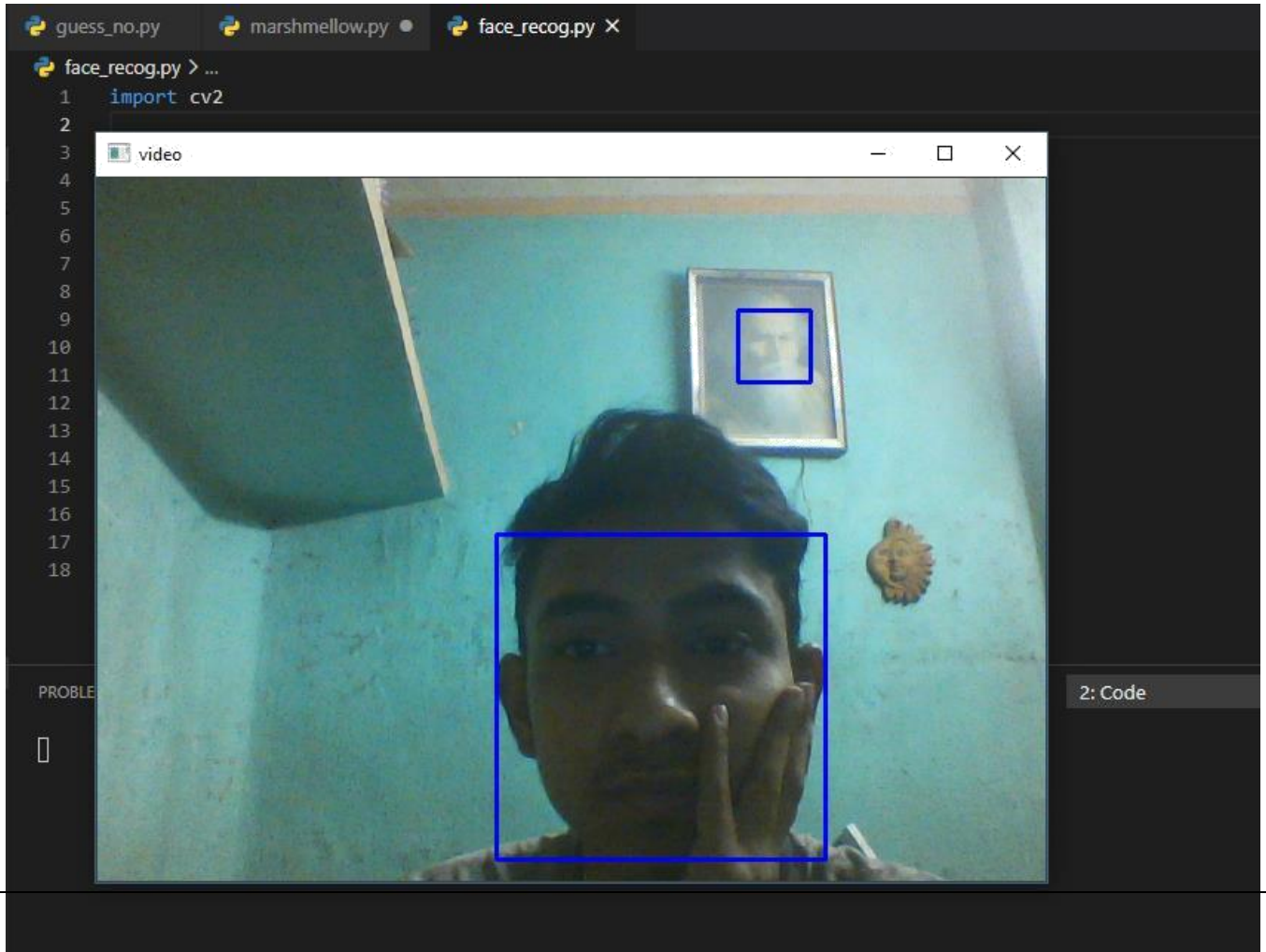
OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2. imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

```
import cv2
```

to do face recognition , we need to write some lines of code.Here, we need "haarcascade_frontal_face" to detect the face.

```
vid = cv2.VideoCapture(0)
face = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
while vid.isOpened():
    _ , frame = vid.read()
    gray = cv2.cvtColor( frame , cv2.COLOR_BGR2GRAY)
    face_1 = face.detectMultiScale(gray , 1.1 , 4)
    for (x , y , w , h) in face_1:
        cv2.rectangle( frame , (x,y), (x+w , y+h), (255 , 0 ,0) , 2)
    cv2.imshow('video' , frame)
    if cv2.waitKey(0) & 0xff == ord('q'):
        break
vid.release()
cv2.destroyAllWindows()
```

First , we need to capture the video, then we need to read the video by using cv2 module. Then we have to convert the color to gray to make it easy for our cv2 , detectMultiScale function will detect the face and cv2.rectangle function will create rectangle around face.



Task 7: Tell me a joke

For that we need to install a named pyjokes. Pyjokes is used for collection Python Jokes over the Internet.

To tell a joke we need to type this :

```
import pyjokes
```

```
    elif 'tell me a joke' in query:
        speak(pyjokes.get_jokes())
```

Task 8: Lets play a game

Lets play a number guessing game , for that we need to install random module.

```
import random
```

The random module will give random number between 1 to 10 and will tell the user the what is the correct number and number of guesses.

```
elif 'lets play a game' in query or 'play game' in query:
    s =random.randint(1 ,10)
    print(s)
    i = 0
    while True:
        i +=1
        try:
            user = int((input('enter your guess no : ')))
            if s>user:
                speak('umm !! add some number to it ')
            elif s<user:
                speak('umm !! reduce some number from it')
```

Task 9: Ask Random Question :

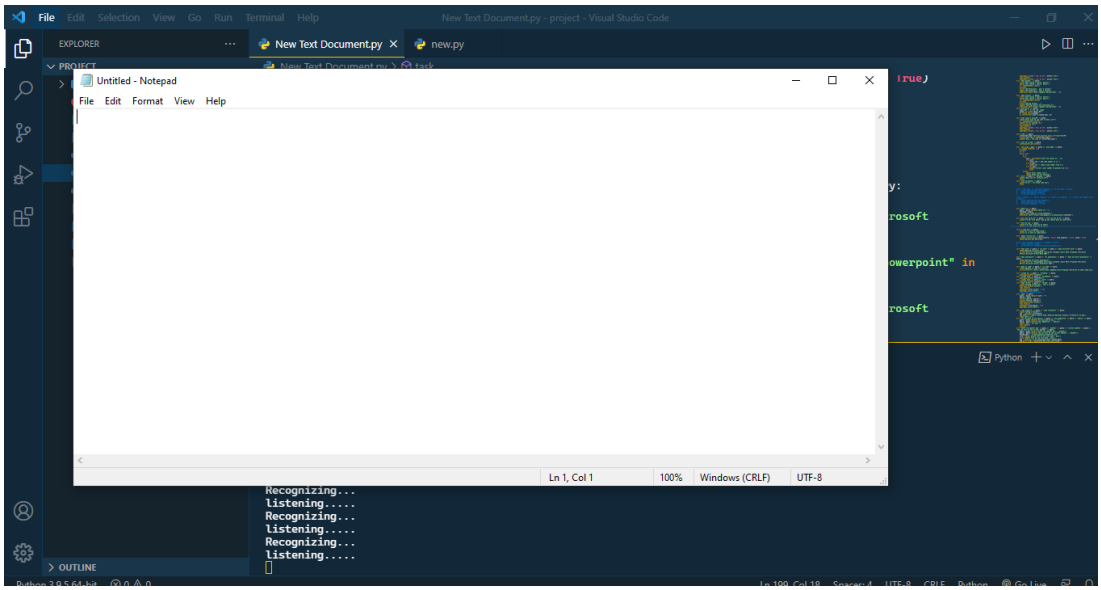
If I ask a random question and she knows the answer then she will give me the answer , like this..

```
listening.....
Recognizing...
 Boss Said : how are you

I'm fine, glad you me that
listening.....
Recognizing...
 Boss Said : love you

listening.....
Recognizing...
 Boss Said : I love you

it is hard to understand
listening.....
```

Task 10: Opening desktop Application Using voice:

If I tell my voice assistant to open Notepad , VS code or MS word or anything it will open that for me.

```
listening.....
Recognizing...
 Boss Said : open Notepad

opening  notepad
listening.....
Recognizing...
listening.....
```



Task 11: Giving random advise:

If I tell “give me any random advise” my program will give me any random advise.

Here is the example -

```
        speak('Done sir')
    elif 'can you give me any advice' in query or 'any suggestion' in query or 'advice' in
        query = query.replace('can you give me any' , "")
        query = query.replace('any suggestion' , "advice")
        speak(' well ! sir wait ')
        time.sleep(1)
        advice()
        print(advice())
```

Task 12: Getting the current weather :

To get the current weather we have to tell our program , ‘whats the current weather’ then it will ask for our location , according to the location it will give us current weather update.

```
elif 'whats the weather mow' in query or 'weather' in query or 'current weather' in query or
'can you tell me the current weather' in query:
    query = query.replace('whats the weather mow' , "weather")
    query = query.replace('can you tell me the current weather' , "weather")
    chrome_path = 'E:\development\chromedriver.exe'
    city = speak('please tell me the city name , boss')
    print('please tell me the city name , boss')
    web = webdriver.Chrome(executable_path =chrome_path)
    web.get(f'https://openweathermap.org/find?q={city}')
    web1 = web.find_element_by_xpath('//*[@id="forecast_list_ul"]/table/tbody/tr/td[2]/p[1]')
    # speak(f'weather in {city} is ')
    speak(web1.text)
    print(web1.text)
    time.sleep(4)
    web.close()
```

```
Recognizing...
listening.....
Recognizing...
listening.....
Recognizing...
 Boss Said : weather

please tell me the city name , boss

DevTools listening on ws://127.0.0.1:61836/devtools/browser/788d1fd0-b7e2-45e1-95ad-c635b5a69caa
[6440:12344:0623/024331.923:ERROR:device_event_log_impl.cc(214)] [02:43:31.923] USB: usb_device_handle_win.cc:1058 Failed to re
ad descriptor from node connection: A device attached to the system is not functioning. (0x1F)
19.6°C temperature from 18.9 to 20.6 °C, wind 1.79 m/s. clouds 80 %, 1015 hpa
listening.....
```

Task 13: Setting a reminder :

To set a reminder we need to tell our program the time (hour and minute)

Then it will be able to remind something.

```
elif 'set a reminder' in query or 'set alarm' in query or 'alarm' in query:
    speak('sir , please tell me the hour , in 24 hours format')
    hour = takeCommand()
    speak('sir , please tell me the minute')
    mint = takeCommand()
    speak(f"alarm set at {hour} and {mint} minute")
    if datetime.datetime.now().hour == hour and datetime.datetime.now().minute == mint:
        speak("sir wake up")
        winsound.Beep(3000 , 10000)
    else:
        pass
```

Task 14 : exit your program :

To exit our program we need to tell our model , good bye and it will exit after telling “thank you sir for using me.

```
    elif "goodbye" in persmision:
        speak("thanks sir for using me")
        time.sleep(1)
        pyautogui.hotkey('ctrl' , 'c')
```

Task 15 : shut down or restart computer using voice :

Yes we can shut down or restart computer using voice , here is the code of the program.

Shut down code :

```
elif 'shut down' or 'shut down computer' or 'ok shut down' in query:
    print("Shutting down the computer")
    speak("Shutting the computer")
    os.system("shutdown /s /t 30")
```

Restart code :

```
elif 'restart' or 'restart computer' or 'restart the computer ' or 'restart the computer now' in
query:
    print("Shutting down the computer")
    speak("Shutting the computer")
    os.system("shutdown /r /t 30")
```

2.5 What have we done so far?

1. First of all, we have created a **wishme()** function that gives the functionality of greeting the user according to the system time.
2. 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.
3. We developed the code logic for opening different websites like google, youtube, etc.
4. Developed code logic for opening Pycharm or any other application.
5. At last, we added functionality to send emails.

2.6 Is this 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 a bunch statements. 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.

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

CHAPTER – 3

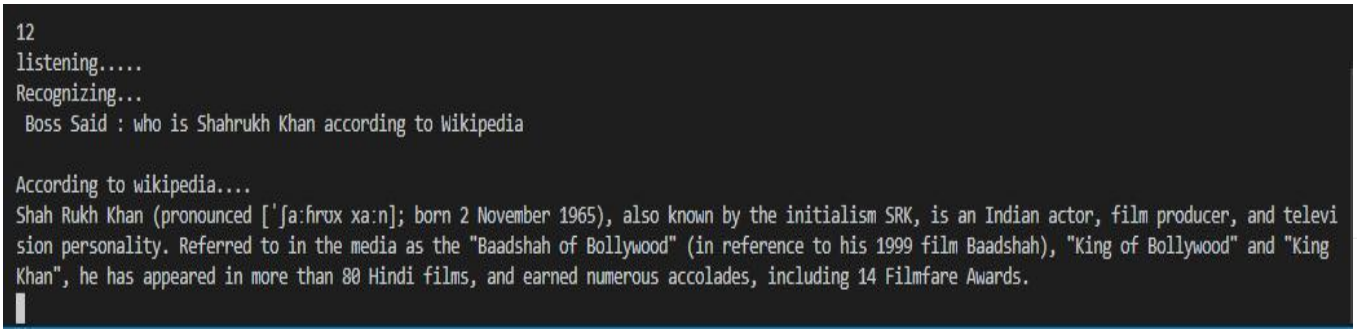
RESULTS AND DISCUSSION

3.1 Task 1: To search something on Wikipedia:

To search something on Wikipedia , first we need to tell our program, open Wikipedia then tell the name what you want to search.Suppose , you want to search ‘who is Shahrukh Khan?’

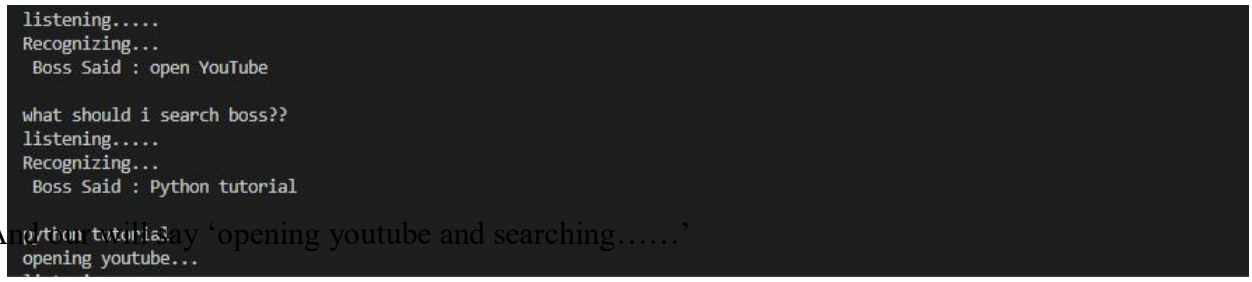
```
listening.....
Recognizing...
 Boss Said : who is Shahrukh Khan according to Wikipedia
```

Then the program will say , according to Wikipedia and the result :



TASK2: To open YouTube in web-browser:

To open youtube we need to tell our program that ‘open youtube’ and our program will say ‘what should I search?’ then you have to tell , what you want to search,

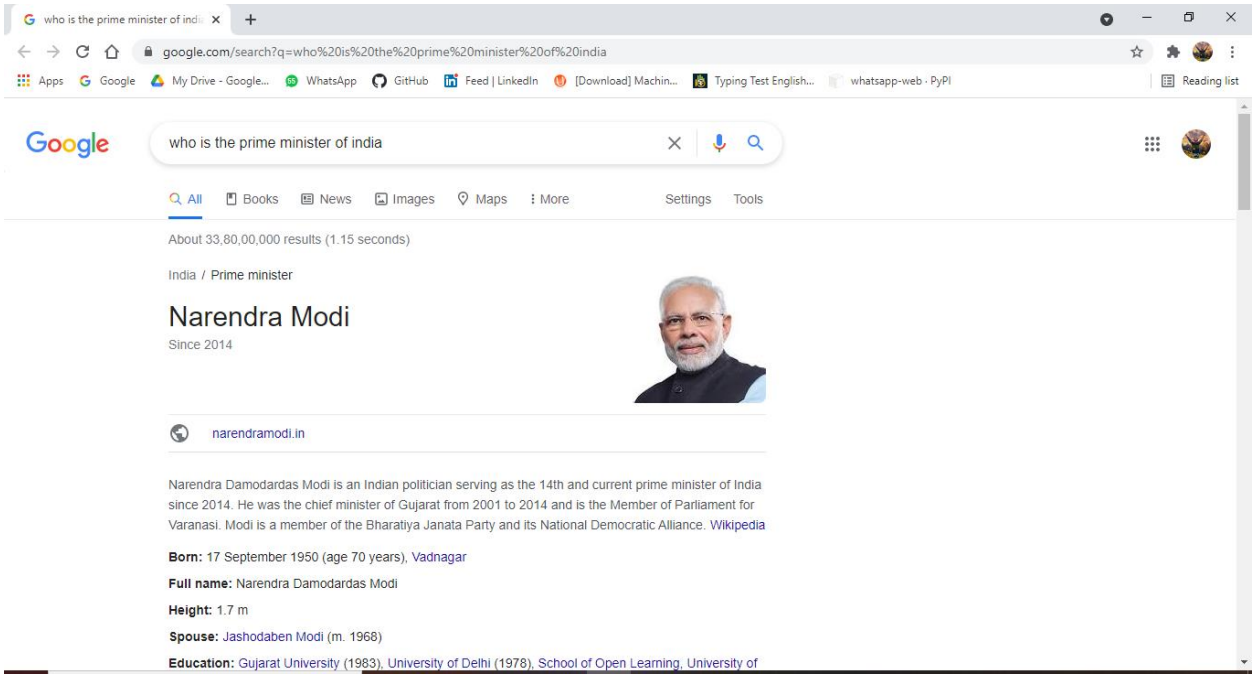


And it will say ‘opening youtube and searching……’

TASK3: To open Google site in a web-browser:

First we meed to tell the program , ‘open google’ , then our program will say ‘what should I search?’then you have to tell , what you want to search…..

Then the result :



3. 2: To play music:

To play music we need tell our program ‘play music’ , then our program will say ‘what do you want to hear??’

Then it will take the name or keywords and will play song according to our choice..

TASK 5: Stop listening:

The stop listening function will stop listening for some seconds , according to the choice of user..

Sometimes , it is good for privacy

```
12
listening.....
Recognizing...
 Boss Said : stop listening

listening.....
```

Task 6: Lets play a game

For this game we need to import a module called random , first random module will a random number between 1 to 10 , then will tell th user to guess the number and we will use a counter for number of guesses ,

if the user number is great than random number , then it will tell choose a smaller number , , if the user number is smaller than random number , then it will tell choose a bigger number , and it will count the guesses.

After successfully playing the game , user will get the correct answer with number of guesses

```
Recognizing...
 Boss Said : play game

4
enter your guess no : 5
enter your guess no : 3
enter your guess no : 6
enter your guess no : 4
listening.....
```

DISCUSSION

Since we saw in our output To search something on Wikipedia , first we need to tell our program, open Wikipedia then tell the name what you want to search. Whatever thing or whosever name we search for, the program will show us an accurate information about that particular search. To open youtube we need to tell our program that ‘open youtube’ and our program will say ‘what should I search?’ then we have to tell , what you want to search, the program will give us a precise view of the search.

So, from the output it is quite obvious that whatever thing we want out voice assistant to search or to do, it will give us the information related to that particular search or thing. We can also use our voice assistant to perform certain tasks just by giving a command.

Voice Assistants can be used in our day to day lives. Besides,having a personal assistant with unlimited knowledge and information isn't it mankind really dream about?. From this outputs we can conclude that whatever command we are giving to the program, it will give us the associated results. Why Adopt A Mobile Voice Strategy?

Mobile phones are already personalized, more so than any website. Additionally, there is very little screen space on mobile, making it more difficult for users to search, or navigate. With larger product directories and more information, voice applications enable consumers to use natural language to eliminate or reduce manual effort, making it a lot faster to accomplish tasks.

Rogers has introduced voice commands to their remotes allowing customers to quickly browse and find their favorite shows or the latest movies with certain keywords, for example, an actor's name. Brands need to

focus on better mobile experiences for their consumers and voice is the way to do so. Users are searching for quicker and more efficient ways of accomplishing tasks and voice is quickly becoming the ideal channel for this.

Whether that's finding out information, making a purchase, or achieving a task, voice is the new mobile experience. With the voice and speech recognition market is expected to grow at a 17.2 percent CAGR to reach $26.8 billion by 2025, It's clear that brands are racing to figure out their voice strategy.

CHAPTER – 4

CONCLUSION

AI virtual assistants are evolving quickly. Companies are enabling them to provide more capabilities like speech recognition and natural language processing advances. It will enable them to understand and perform requests. Furthermore, improvements in voice recognition technology are allowing them to move deeper into business workflows. In the future, AI assistants will have more advanced cognitive computing technologies. These will help them carry out multi-step requests and perform more complex tasks.

Future of ai voice assistant

Artificial intelligence has truly transformed the way voice assistants are used in our daily lives, and we are only beginning to understand how they will be integrated into all of our activities in the years to come Report after report is predicting voice assistants will soar and that means the tools and technologies behind these devices are shaping the internet of skills. We are talking about the next generation of tools to spark. growth in retail, logistics, healthcare, smart cities, manufacturing, and autonomous vehicles, among many others.

A recent survey from PWCreveals voice assistants have been used in a host of ways during the past decade and they will continue to mold our very essence. Here's what some of the numbers are showing:

- 90% of people recognize voice assistants
- 72% had used a voice assistant
- 57% top commends come from a smartphone
- 27% issue commands to a speaker
- 20% issue commands for vehicle navigation purposes

What's more, adoption of voice-assistant technologies is highest among 18-24-year-olds. But the age group that uses voice assistants most frequently is the 25-49-year-old group, with 65% of them being considered "heavy" users that issue voice commands to a device at least once a day.

More importantly, let's consider what voice assistants are used for today. According to PWC's survey, the

most common tasks people ask of their voice assistants are to search for information on the internet, answering a question, providing weather or the news, playing music, and setting a timer or reminder. In addition, the report shows the slightly less common tasks include sending a text or email and checking traffic. Interestingly, 50% or more of people say they never do include buying or ordering something via their voice assistant and using it to control other smart devices.

As for growth, Juniper Researchsays there will be 8 billion digital voice assistants in use by 2023. That means the stage is set for something significant. But what does this growth mean for chat bots and more?

The first hurdle, awareness, has been cleared during the past decade of usage. Now the second hurdle, achieving acceptance and basic use across different demographics, has also been cleared during the same period. The technology has come a long way in a relatively short amount of time too. The next hurdle, though, will have to do with user trust.

There is a lot of work that still needs to be achieved yet. We need to be asking ourselves, why aren't people using voice assistants to accomplish more complex tasks? The PWC report shows when it comes to more complex tasks and involve people's hard-earned cash, people prefer to use

methods.they know and trust. That means voice assistants do not appear on the list just yet. But this isn't true for everybody.

people in this survey, about 50%, say they have made purchases using a voice assistant. Purchases include food (34%), groceries (31%), books (24%), and transportation (21%).

But about 25% came out saying they wouldn't even consider using voice assistants to make purchases. The top reason is because folks just don't trust their voice assistants to correctly interpret and process purchases.

It's all about stakes. The stakes just aren't that high when you're asking your GoogleHome mini to play a certain playlist on Spotify or to tell you what the temperature is outside. But if you're

asking AmazonAlexa on your echo to buy you a replacement air filter and ship it to your house, you're asking it to spend real money.

You're trusting that AI to understand your request, get the right filter, charge you the right price, and send it to the right house. Even if, in this case, we're talking less than $50, it's still a much greater risk. Gaining consumers' trust is going to be the next big task for voice-assistant tech companies. And shopping is just one example of what people aren't doing.

Even fewer people, according to PWC's survey, are using voice assistants to control their smart homes, and this is perhaps the biggest the crux of the problem. In the future, we need to look at what voice assistants will be capable of doing, and it's going to require user trust. For example, one prediction for voice assistants will be their growing use in healthcare scenarios. Voice assistants have the ability to help in so many opportunities. The real question now is how long will it take before we really trust voice assistants to do our "bidding" for us.

7 Key Predictions for the Future of A.I Voice Assistant

When voice technology began to emerge in 2011 with the introduction of Siri, no one could have predicted that this novelty would become a driver for tech innovation. Now, a decade later, it's estimated that every 1 in 4 U.S. adults own a smart speaker (i.e., Google Home, Amazon Echo) and eMarketer forecasts that nearly 92.3 percent smartphone users will be using voice assistants by 2023.

Brands such as Amazon, and Google are continuing to fuel this trend as they compete for market share. Voice interfaces are advancing at an exponential rate in all industries, with notable growth in healthcare to banking, as companies are racing to release their own voice technology integrations to keep pace with consumer demand.

What's Causing The Shift Towards Voice?

The main driver for this shift towards voice user interfaces is changing user demands. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized.

The mass adoption of artificial intelligence in users' everyday lives is also fueling the shift towards voice applications. The number of IoT devices such as smart thermostats,

appliances, 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, however, it only starts there. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years.

Applications of this technology are seen everywhere, so where will it take us in 2021 and beyond? We provide a high-level overview of the potential that voice has and 7 key predictions we think will take off in the coming years.

7 Future Predictions of Voice Assistants in 2021:

Mobile App Integration

Integrating voice-tech into mobile apps has become the hottest trend right now, and will remain so because voice is a natural user interface (NUI).

Voice-powered apps increase functionality, and save users from complicated app navigation. Voice-activated apps make it easier for the end-user to navigate an app — even if they don't know the exact nameof the item they're looking for or where to find it in the app's menu. While at this stage, voice integration may be seen as a nice-to-have by users, this will soon become a requirement that users will expect.

Voice-Tech in Healthcare

In 2020, AI-powered chatbots and virtual assistants played a vital role in the fight against COVID-19. Chatbots helped screen and triage patients, and Apple's Siri now walks users through CDC COVID-19 assessment questions and then recommends telehealth apps. Voice and conversational AI have made health services more accessible to everyone who was unable to leave their home during COVID-19 restrictions. Now that patients have a taste for what is possible with voice and healthcare, behaviors are not likely to go back to re-pandemic norms. Be prepared to see more investment in voice-tech integration in the healthcare industry in the years to come.

Search Behaviours Will Change

Voice search has been a hot topic of discussion. Visibility of voice will undoubtedly be a challenge. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice

interface unless it is connected to the Alexa or Google Assistant app. Search behaviors, in turn, will see a big change. In fact, if tech research firm Juniper Research is correct, voice-based ad revenue could reach $19 billion by 2022, thanks in large part to the growth of voice search apps on mobile devices.

Brands are now experiencing a shift in which touchpoints are transforming to listening points, and organic search will be the main way in which brands have visibility. As voice search grows in popularity, advertising agencies and marketers expect Google and Amazon will open their platforms to additional forms of paid messages.

Individualized Experiences

Voice assistants will also continue to offer more individualized experiences as they get better at differentiating between voices. Google Home is able to support up to six user accounts and detect unique voices, which allows Google Home users to customize many features. Users can ask “What’s on my calendar today?” or “tell me about my day?” and the assistant will dictate commute times, weather, and news information for individual users. It also includes features such as nicknames, work locations, payment information, and linked accounts such as Google Play, Spotify, and Netflix. Similarly, for those using Alexa, simply saying “learn my voice” will

allow users to create separate voice profiles so the technology can detect who is speaking for more individualized experiences.

Voice Cloning

Machine learning tech and GPU power development commoditize custom voice creation and make the speech more emotional, which makes this computer-generated voice indistinguishable from the real one.

You just use a recorded speech and then a voice conversion technology transforms your voice into another. Voice cloning becomes an indispensable tool for advertisers, filmmakers, game developers, and other content creators.

Smart Displays

Last year smart displays were on the rise as they expanded voice-tech's functionality. Now, the demand for these devices is even higher, with consumers showing a preference for smart displays over regular smart speakers. In the third quarter of 2020, the sales of smart displays rose year-on-year by 21 percent to 9.5 million units, while basic smart speakers fell by three percent. In 2021, we expect for there to be a lot of innovation in the world of smart displays to integrate more advanced technology and more customization. Smart displays, like the Russian Sber portal or a Chinese smart screen Xiaodu, for example, are already equipped with a suite of upgraded AI-powered functions, including far-field voice interaction, facial recognition, hand gesture control, and eye gesture detection.

Voice in The Gaming Industry

It takes a lot of time and effort to record a voice for spoken dialogues within the game for each of the characters. In the upcoming year, developers will be able to use sophisticated neural networks to mimic human voices. In fact, looking a little bit ahead, neural networks will be able to even create appropriate NPC

responses. Some game design studios and developers are working hard to create and embed this dialogue block into their tools, so seeing games include dynamic dialogues isn't too far off.

Voice User Interface (VUI) Will Continue to Advance

Even with just that handful of simple scenarios, it's easy to see why voice assistants are shaping up to become the hubs of our connected homes and increasingly connected lives.

Voice technology is becoming increasingly accessible to developers. For example, Amazon offers

Transcribe, an automatic speech recognition (ASR) service that enables developers to add speech-to-text capability to their applications. Once the voice capability is integrated into the application, users can analyze audio files and in return, receive a text file of the transcribed speech.

Google has made moves in making Assistant more ubiquitous by opening the software development kit through Actions, which allows developers to build voice into their own products that support artificial intelligence. Another one of Google's speech-recognition products is the AI-driven Cloud Speech-to-Text tool which enables developers to convert audio to text through deep learning neural network algorithms.

This is only the beginning of voice technology as we will see major advancements in the user interface in the years to come. With the advancements in VUI, companies need to start educating themselves on how they can best leverage voice to better interact with their customers. It's important to ask what the value of adding voice will be as it doesn't always make sense for every brand to adopt. How can you provide value to your customers? How are you solving their pain points with voice? Will voice enhance the user experience or frustrate the user?

In 2021, voice-enabled apps will not only accurately understand what we are saying, but how we are saying it and the context in which the inquiry is made.

However, there are still a number of barriers that need to be overcome before voice applications will see mass adoption. Technological advances are making voice assistants more capable particularly in AI, natural language processing (NLP), and machine learning. To build a robust speech recognition experience, the artificial intelligence behind it has to become better at handling challenges such as accents and background noise. And as consumers are becoming increasingly more comfortable and reliant upon using voice to talk to their phones, cars, smart home devices, etc., voice technology will become a primary interface to the digital world and with it, expertise for voice interface design and voice app development will be in greater demand.

Voice is the Future Of Brand Interaction And Customer Experience

Advancements in a number of industries are helping digital voice assistants become more sophisticated and useful for everyday use. Voice has now established itself as the ultimate mobile experience. A lack of skills and knowledge make it particularly hard for companies to adopt a voice strategy. There is a lot of opportunity for much deeper and much more conversational experiences with customers. The question is, is your brand willing to jump on this opportunity?

REFERENCE :

1. Hirschberg, Julia, and Christopher D. Manning. “Advances in Natural Language Processing.” Science 349 (6245): 261–266. doi:10.1126/science.aaa8685.

2. Moore, Clayton. “The Most Useful Skills for Google Home.” Digital Trends . https://www.digitaltrends.com/home/google-home-most-useful-skills/.

3. Lee, Nicole. “Google Assistant on the iPhone Is Better than Siri, but Not Much.”

Engadget.https://www.engadget.com/2017/05/17/google-assistant-iphonehands-on/.

4. Hachman, Mark. “The Microsoft-Amazon Deal Leaves Cortana Speakers with One Advantage: Skype.” PCWorld . https://www.pcworld.com/article/ 3221284/windows/the-microsoft-amazon-deal-leaves-cortana-speakers-with-one-advantageskype.html.

5. Spence, Ewan. “Windows Phone Is Dead, Long Live Microsoft’s Smartphone Dream.). https://www.forbes.com/sites/ewanspence/2017/07/12/microsoftwindows-phone-windows10-mobile-strategy/

. 6. Tilley, Aaron. “How A Few Words to Apple’s Siri Unlocked a Man’s Front Door.” Forbes .

https://www.forbes.com/sites/aarontilley/2016/09/21/applehomekit-siri-security/.

7. Liptak, Andrew. “Amazon’s Alexa Started Ordering People Dollhouses after Hearing Its Name on TV.” The Verge . https://www.theverge./ 14200210/amazon-alexa-tech-news-anchor-order-dollhouse.

8. Zhang, Guoming, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. “DolphinAttack: Inaudible Voice Commands.” ACM SIGSAC Conference on Computer and Communications Security, New York, NY, 103–117. doi:10.1145/3133956.3134052.

9. Tung, Liam. “Google Home Mini Flaw Left Smart Speaker Recording Everything.” . http://www.zdnet.com/article/google-home-mini-flaw-left-smartspeaker-recording-everything/.

Buhr, Sarah. “An Amazon Echo May Be the Key to Solving a Murder Case.” TechCrunch. http://social.techcrunch.com/an-amazon-echo-may-bethe-key-to-solving-a-murder-case/.

10. Selenium documentation : https://www.selenium.dev/
11. Python documentation : https://www.python.org/doc/

APPENDIX

CODING:

```
from urllib.parse import quote

import pyautogui

import pyttsx3

import speech_recognition as sr

import wikipedia

import time

import datetime

import pyaudio

import webbrowser

import os

import random

# import sys

import requests
```

```
import pywhatkit as kit

import random

import pyjokes

import os

import winshell

import sys

import winsound

# import cv2

from selenium import webdriver

# from playsound import playsound

import winsound


print("Initializing Marshmellow for Souvik.....")

NAME = "souvik"


engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

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


def speak(audio):

engine.say(audio)
```

```
engine.runAndWait()


def wishMe():
hour = int(datetime.datetime.now().hour)
mint = int(datetime.datetime.now().minute)
second= int(datetime.datetime.now().second)
#speak(hour)
print(hour)
if hour >=0 and hour < 12:
speak ("good morning .." )
elif hour >=12 and hour <18:
speak("good afternoon  .." )
else:
speak("good evening ")
def Marshmellow():
if Marshmellow==takeCommand().lower():
speak('yes boss')
else:
speak('sorry boss')
def takeCommand():
```

```
r = sr.Recognizer()


with sr.Microphone(device_index = 0) as source:

r.adjust_for_ambient_noise(source )

print("listening.....")

audio = r.listen(source)

try:

print("Recognizing...")

query = r.recognize_google(audio, language='en-in')

print(f" Boss Said : {query}\n")


except:

# speak(' i could not get it boss ... say that again')

# print("say that again...")

return 'None'


return query

def advice():

url = 'https://www.boredapi.com/api/activity'

response = requests.get(url)

data = response.json()
```

```
activity = data['activity']

type = data['type']

speak(f'if you are getting bored you can {activity} , and its a type of {type}...')




def task():

# wishMe()

speak("hi boss ")

# speak("how can i help you??....")

while (1):

query = takeCommand().lower()

if 'wikipedia' in query:

speak('searching wikipedia...')

query = query.replace("wikipedia","")

results = wikipedia.summary(query, sentences=2)

print("According to wikipedia....")

speak("According to wikipedia...")

print(results)

speak(results)

elif 'open youtube' in query or 'youtube' in query:
```

```
speak('what should i search boss??')

print('what should i search boss??')

m = takeCommand().lower()

print(m)

print('opening youtube...')

speak(f'opening youtube and searching {m}')

webbrowser.open('https://youtube.com/results?search_query=' + m)

time.sleep(10)

pyautogui.click(x = 410, y =319 , button='left')

time.sleep(8)

pyautogui.click(x = 729, y =435 , button='left')

elif 'open google'  in query:

print('what should i search  boss??')

speak('what should i search  boss??')

z = takeCommand().lower()

print(z)

print('searching boss , wait a second')

speak('searching boss , wait a second')

webbrowser.open('https://google.com/search?q=' + z)


elif 'open browser' in query:
```

```
print('what should i search  boss??')

speak('what should i search  boss??')

z = takeCommand().lower()

print(z)

print('opening browser...')

speak(f'opening browser and searching {z}')

webbrowser.open('https://google.com/search?q=' + z)

elif 'play music' in query:

music_dir = 'D:\\my faV. songs'

songs= os.listdir(music_dir)

d= random.choice(songs)

os.startfile(os.path.join(music_dir, d))


elif "play song on youtube" in query:

speak('which song do you want to hear sir??')

z = takeCommand().lower()

speak(f'ok sir playing {z}')

kit.playonyt(f'{z}')

time.sleep(12)

pyautogui.click(x = 410, y =319 , button='left')

time.sleep(8)
```

```
pyautogui.click(x = 764, y =484 , button='left')


elif "time" in query:

strcurrent_time =datetime.datetime.now().strftime("%H:%M")

print(f'the time is {strcurrent_time}')

speak(f'boss , the time is {strcurrent_time}')


elif 'tell me a joke' in query:

speak(pyjokes.get_jokes())


elif 'lets play a game' in query or 'play game' in query:

s =random.randint(1 ,10)

print(s)

i = 0

while True:

i +=1

try:

user = int((input('enter your guess no : ')))

if s>user:

speak('umm !! add some number to it ')

elif s<user:
```

```
speak('umm !! reduce some number from it')

elif s==user:

speak(f'correct ,your number of guesses are {i}')

break

except:

speak('enter number only')

speak('thank you for playing')

elif 'exit' in query or 'goodbye' in query:

speak('thank you for using me,sir')

exit()

elif 'stop listening' in query:

speak("ok sir , its break time now")

break



# elif 'shut down' or 'shut down computer' or 'ok shut down' in query:

#     print("Shutting down the computer")

#     speak("Shutting the computer")

#     os.system("shutdown /s /t 30")


# elif 'restart' or 'restart computer' or 'restart the computer ' or 'restart the computer now' in query:
```

```
#     print("Shutting down the computer")

#     speak("Shutting the computer")

#     os.system("shutdown /r /t 30")



elif "where is" in query:

query = query.replace("where is", "")

location = query

speak(f"boss asked to Locate{location}")

webbrowser.open(f"https://www.google.co.in/maps/place/{location}")



elif "will you be my gf" in query or "will you be my bf" in query:

speak("I'm not sure about, may be you should give me some time")



elif "how are you" in query:

speak("I'm fine, glad you me that")

print("I'm fine, glad you me that")



elif "i love you" in query:

speak("It's hard to understand")

print('it is hard to understand')
```

```
elif 'empty recycle bin' in query:

winshell.recycle_bin().empty(confirm = False, show_progress = False, sound = True)

speak("Recycle Bin Recycled")


# elif "open notepad" in query or "notepad" in query:

#     speak("Opening notepad")

#     os.startfile('C:\\Windows\\system32\\notepad.exe')


elif "open word" in query or "ms word" in query or "open microsoft word" in query:

speak("Opening microsoft word")

os.startfile('C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office Word 2007')


elif "open powerpoint" in query or "ms powerpoint" in query or "open microsoft powerpoint" in query:

speak("Opening microsoft powerpoint")

os.startfile('C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office PowerPoint 2007')


elif "open vs code" in query or "vs code" in query:

speak("Opening vs code for souvik")
```

```
os.startfile('C:\\Users\\Souvik Saha\\AppData\Local\Programs\\Microsoft VS Code\\Code.exe')


elif 'volume up' in query or 'volumeup' in query:

pyautogui.hotkey('volumeup')

elif 'volume down' in query or 'volumedown' in query:

pyautogui.hotkey('volumedown')

elif 'volume mute' in query or 'mute' in query:

pyautogui.hotkey('volumemute')

elif 'open chrome' in query or 'chrome' in query:

speak('openimg chrome boss , wait a second')

pyautogui.hotkey('winleft')

time.sleep(2)

pyautogui.write('chrome' , 0.5)

pyautogui.press('enter')

# pyautogui.hotkey('winleft')

elif 'open' in query:

query = query.replace("open", "")

search = query

print(f"opening {search}")

speak(f"opening {search}")

pyautogui.hotkey('winleft')
```

time.sleep(2)

pyautogui.write(search , 0.3)

pyautogui.press('enter')

elif 'take snapsort' in query or 'take screensort' in query:

speak('taking screensort')

im1 = pyautogui.screenshot()

im1.save(r'C:\\Users\\Souvik Saha\\OneDrive\Desktop\\project\\screensort\\ss.png')

speak('Done sir')

elif 'can you give me any advice' in query or 'any suggestion' in query or 'advice' in query:

query = query.replace('can you give me any' , "")

query = query.replace('any suggestion' , "advice")

speak(' well ! sir wait ')

time.sleep(1)

advice()

print(advice())

elif 'whats the weather mow' in query or 'weather' in query or 'current weather' in query or 'can you tell me the current weather' in query:

query = query.replace('whats the weather mow' , "weather")

query = query.replace('can you tell me the current weather' , "weather")

chrome_path = 'E:\development\chromedriver.exe'

```
city = speak('please tell me the city name , boss')

print('please tell me the city name , boss')

web = webdriver.Chrome(executable_path =chrome_path)

web.get(f'https://openweathermap.org/find?q={city}')

web1 = web.find_element_by_xpath('//*[@id="forecast_list_ul"]/table/tbody/tr/td[2]/p[1]')

# speak(f'weather in {city} is ')

speak(web1.text)

print(web1.text)

time.sleep(4)

web.close()


elif 'set a reminder' in query or 'set alarm' in query or 'alarm' in query:

speak('sir , please tell me the hour , in 24 hours format')

hour = takeCommand()

speak('sir , please tell me the minute')

mint = takeCommand()

speak(f"alarm set at {hour} and {mint} minute")

if datetime.datetime.now().hour == hour and datetime.datetime.now().minute == mint:

speak("sir wake up")

winsound.Beep(3000 , 10000)

else:
```

```
pass


if __name__ == "__main__":
while True:
persmision = takeCommand()
if "wake up" in persmision:
task()
elif "goodbye" in persmision:
speak("thanks sir for using me")
time.sleep(1)
pyautogui.hotkey('ctrl' , 'c')
```