Ambedkar Institute of Technology

Jarvis 1. o

A Python Based Voice Assistant that can be used to make user task easier and faster, with different function defined inside it.

Under the Guidance of
Mr. Dipankar Mishra
HOD ITES&M

Gaurav Kumar 1702111025

DECLARATION BY THE CANDIDATE

I undersigned solemnly declare that the project report “Jarvis 1.o | A Personal Voice Assistant” is based on my own work carried out during the course of our study under the supervision of Mr. Dipankar Mishra.

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I) The work Contained in the report is original and has been done by me under the general supervision of my Teachers.

II) The work has not been submitted to any other Institution for any other degree/Diploma/Certificate in this University or any other University of India or Abroad.

III) We have followed the guidelines provided by the Teachers in writing the project report.

IV) Whenever I have used materials (Data, theoretical analysis and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Date Gaurav Kumar 1702111025

23-April-2020

Certificate

This is to certify that the thesis entitled "**Jarvis 1.o | A Personal Voice Assistant**" being submitted by Gaurav Kumar, BTE Roll NO:-1702111025 in the partial fulfillment of the requirement for the Award of three year diploma in **Information Technology Enabled Service and Management**, **Ambedkar Institute of Technology** is a record of work done by him under my supervision and guidance.

Under the Guidance of

Mr. Dipankar Mishra

Head of Department

ITES&M

ACKNOWLEDGEMENT

I express my heartiest gratitude and respectful regards to Mr. Dipankar Mishra Sir, Head of department ITES&M. My project guide for his valuable guidance, constructive criticism and consistent enthusiastic interest during the course of investigation and writing of manuscript that led this work to its successful completion. I regard my sincere thanks to the technical staff at the organization that helped me during the project and made this project successful.

Last but not the least my special thanks to all our Teacher for their helping me throughout the project and for showing support and constant encouragement during the project work.

Contents

1) Introducation
2) Conceptual Framework
3) Commands
4) Modules Used With Sexpanation
5) Funcations
6) Project Structure
7) How does Jarvis React to any query
8) Structure Diagram
9) History
10) Requirements
11) Limitations
12) Source Code
13) Reference

Project Title: **Jarvis 1.0 | A personal assistant**

Introduction:

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.

Constraints:

The application was built on and for a Microsoft Windows thus restricting it to the Windows alone. Jarvis is compatible with all the versions of the Microsoft Windows.
The system also assumes that the user has minimal English knowledge as of now.

Conceptual framework:

There is always scope for improvement, that’s why we used a simple and self-dependent function which doesn’t interfere in other function of the program or application.

The system also implements the singleton pattern and the single responsibility principle which ensure the individual functioning of the modules.

Commands :

1) Searching Queries on Wikipedia
2) Opening Different Application
3) Playing Songs
4) Making a Google Search
5) Basic Calculations
6) Downloading Youtube Videos
7) Playing Youtube Songs
8) Control Screen Brightness
9) Empty Recycle Bin
10) Organizing Desktop
11) Sending WhatsApp Messages
12) Telling Time
13) Changing Desktop Background
14) Searching for News/ Telling News from Different sources
15) Performing Basic System Operations
    a. Shutdown
    b. Restart
    c. Hibernate
    d. Screen Lock
    e. Lock Out
16) Writing Notes
17) Weather Update

Modules Used:

```
import subprocess
```
The **subprocess** module present in **Python** (both 2.x and 3.x) is used to run new applications or programs through **Python** code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands.

```
import wolfram alpha
```
Wolfram Alpha is a unique engine for computing answers and providing knowledge. It works by using its vast store of expert-level knowledge and algorithms to automatically answer questions, do analysis and generate reports.

```
import wikiquotes
```
wikiquotes is an accurate and comprehensive collection of notable quotations. ... A quotation can be notable because it has achieved fame due to its enduring relevance to many people, or because it is attributed to a notable individual, or appeared in a notable work. Quotations: wikiquotes is a collection of quotations.

```
import 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

```
import speech recognition as sr
```
Speech recognition is the process of converting spoken words to text. **Python** supports many **speech recognition** engines and APIs, including Google **Speech** Engine, Google Cloud **Speech** API, Microsoft Bing **Voice Recognition** and IBM **Speech** to Text.

```
import datetime
```
Date and datetime are an object in **Python**, so when you manipulate them, you are actually manipulating objects and not string or timestamps. Whenever you manipulate dates or **time**, you need to import datetime function. The datetime classes in **Python** are categorized into main 5 classes.

```
import 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

```
import webbrowser
```
In **Python**, **webbrowser** module provides a high-level interface which allows displaying Web-based documents to users. The **webbrowser** module can be used to launch a browser in a platform-independent manner as shown below: Code #1: import **webbrowser**. **Webbrowser.**

```
import os
```
The **OS** module in **Python** provides a way of using operating system dependent functionality. The functions that the **OS** module provides allows you to interface with the underlying operating system that **Python** is running on – be that Windows, Mac or Linux.

```
import winshell
```
The **winshell** module is a light wrapper around the Windows shell functionality. It includes convenience functions for accessing special folders, for using the shell's file copy, rename & delete functionality, and a certain amount of support for structured storage.

```
import pyjokes
```
Pyjokes is a package for fetching the perfect joke in a database. You give a sentence; you get a joke.

```
import feed parser
```
Feed parser is a **Python** library that parses feeds in all known formats, including Atom, RSS, and RDF. It runs on **Python** 2.4 all the way up to 3.3

```
import smtplib
```
Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail − import **smtplib** smtpObj = **smtplib**.

```
import ctypes
```
ctypes is a foreign function library for **Python**. It provides C compatible data types, and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure **Python**.

```
import time
```

```
import random
```
To get any random value from a list or set of variables.

```
import requests
```
The requests library is the de facto standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.

```
import fileinput
```
The **fileinput Module**. The **fileinput module** allows you to loop over the contents of one or more text files, as shown in Example 2-1. The **fileinput module** also allows you to get metainformation about the current line

```
import getpass
```
getpass() prompts the user for a password without echoing. The **getpass module** provides a secure way to handle the password prompts where programs interact with the users via the terminal. The **getpass**() function is used to prompt to users using the string prompt and reads the input from the user as Password.

```
import wmi
```
The **Python WMI** module is a lightweight wrapper on top of the pywin32 extensions, and hides some of the messy plumbing needed to get **Python** to talk to the **WMI** API. It's pure **Python** and should work with any version of **Python** from 2.1 onwards (list comprehensions) and any recent version of pywin32.

```
from pathlib import Path
```

```
from selenium import webdriver
from ecapture import ecapture as ec
from bs4 import BeautifulSoup
import win32com.client as wincl
from urllib.request import urlopen
```

Functions:
The system provides all the below stated functions.

- Finding User Name

```
def usrname():
    speak("What should i call you sir")
    uname=takeCommandname()
    speak("Welcome Mister")
    speak(uname)
    print("#####################")
    print("Welcome Mr.",uname)
    print("#####################")
```

- Output

```
Username...
Trying to Recognizing Name...
User said: Gaurav

####################
Welcome Mr. Gaurav
####################
```

- Showing Quote of the day from "wikiquotes"

```
def quotaton():
    speak(wikiquote.quote_of_the_day())
    print(wikiquote.quote_of_the_day())
```

- Output

```
Listening...
Recognizing...
User said: yes

("A great while ago the world begun,  With hey, ho, the wind and the rain:  But that's all one, our play is done,  And we'll strive to please you every day.", 'William Shakespeare')
Listening...
Recognizing...
```

- Making Wikipedia search and giving output in Audio formate as well as Text.

```
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=3)
            speak("According to Wikipedia")
            print(results)
            speak(results)
```

- Output

```
Listening...
Recognizing...
User said: search Python on Wikipedia

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability w
ith its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Pyt
hon is dynamically typed and garbage-collected.
```

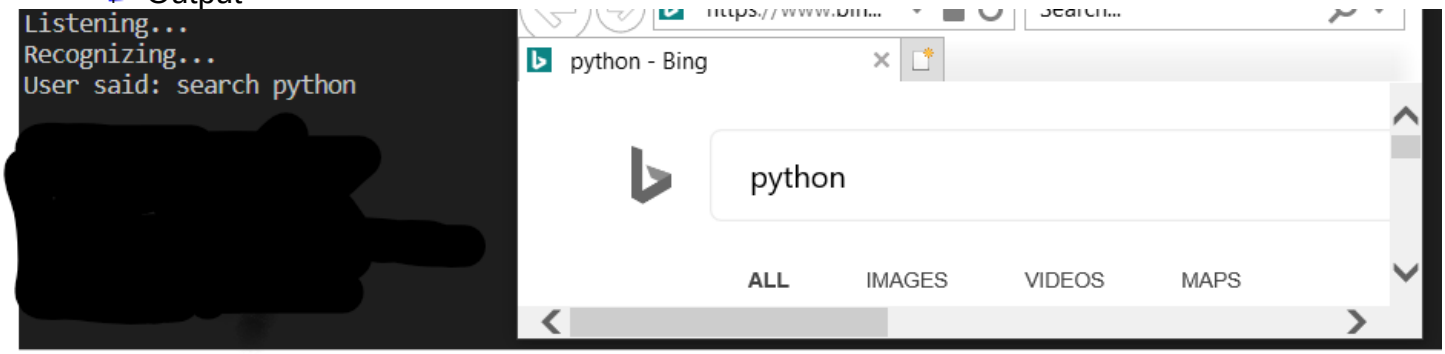
- Make Google query search

```
        elif 'search' in query or 'play' in query:
            query = query.replace("search", "")
            query = query.replace("play", "")
            webbrowser.open(query)
```

- Opening different Websites “Facebook”, “Google” , “Youtube” etc.

```
        elif 'open youtube' in query:
            speak("Here you go to Youtube\n")
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            speak("Here you go to Google\n")
            webbrowser.open("google.com")
```

- Output



- Changing System Brightness level.

```
 elif "change brightness to " in query:

            brightness = [int(word) for word in query.split() if word.isdigit()]
            c = wmi.WMI(namespace='wmi')
            methods = c.WmiMonitorBrightnessMethods()[0]
            methods.WmiSetBrightness(brightness, 0)
```

- Sending Whatsaap messages to any Group or Personally.

```
        elif "send a whatsaap message" in query or "send WhatsApp message" in query:
            driver = webdriver.Chrome('C:\\Users\\gaura\\OneDrive\\Desktop\\Major Project\\Vo
ice Assistant\\chromedriver_win32\\chromedriver.exe')
            driver.get('https://web.whatsapp.com/')
            speak("Scan QR code before proceding")
            tim=10
            time.sleep(tim)
            speak("Enter Name of Group or User")
            name = takeCommanduser()
            speak("Enter Your Message")
            msg = takeCommandmessage()
            user = driver.find_element_by_xpath('//span[@title = "{}"]'.format(name))
            user.click()
            msg_box = driver.find_element_by_class_name('_3u328')
            msg_box.send_keys(msg)
            button = driver.find_element_by_class_name('_3M-N-')
            button.click()
```

- Play music

```
        elif 'play music' in query or "play song" in query or "gaana"in query or "song" in qu
ery:
        elif 'play music' in query or "play song" in query or "gaana"in query or "song" in qu
ery:
            speak("Here you go with music")
            username = getpass.getuser()
            try:
                music_dir = "C:\\Users\\"+username+"\\Music"
            except Exception as e:
                print("Following Exception Occurred”, e)
                speak("Following Exception Occurred”, e)

            finally:
```

```
            songs = os.listdir(music_dir)
            print(songs)
            random=os.startfile(os.path.join(music_dir, songs[1]))
```

- Making Calculations

```
        elif 'add' in query or "subsctract" in query or "multiply" in query or "divide" in qu
ery:
            num1,num2 = [int(word) for word in query.split() if word.isdigit()]


            if "add" in query:
                speak(num1+num2)

            elif "substract" in query:
                speak(num1-num2)

            elif "multiply" in query:
                speak(num1*num2)

            elif "divide" in query:
                speak(num1/num2)
```

- Output

```
Listening...
Recognizing...
User said: add 5 + 6

11
```

- Answering Queries through Wolfram Alpha

```
        elif "what is" in query or "who is" in query:
            client= wolframalpha.Client("WTHP37-K6P2X72X3E")
            res = client.query(query)
            try:
                print(next(res.results).text)
                speak(next(res.results).text)
            except StopIteration:
                print ("No results")
```

- Output

```
User said: what is python

kingdom | Animalia (animals)
phylum | Chordata (chordates)
class | Reptilia (reptiles)
order | Squamata (lizards and snakes)
family | Boidae (boas, pythons and anacondas)
```

- Telling PyJokes / Fun

```
        elif 'joke' in query:
            speak(pyjokes.get_joke())

        elif "boring" in query or "fun" in query:
            speak("Would you like to play RPS")
            opt=takeCommand()
            if "yes" in opt:
                speak("Rock , Paper or Scissor")
                usr=takeCommand()
                action=("Rock","Paper","Scissior","Rock","Paper","Scissior","Rock","Paper","S
cissior")
                coom=random.choices(action)
                coom = ' '.join(map(str, coom))
                print("User Symbol : ",usr)
                print("Computer Symbol : ",coom)

                if coom==usr:
                    speak("Match Draw")

                elif coom=="Scissior" and usr=="Paper":
                    speak("Computer Won")

                elif usr=="Scissior" and coom=="Paper":
                    speak("User won")

                elif usr=="Rock" and coom=="Paper":
                    speak("Computer won")

                elif coom=="Rock" and usr=="Paper":
                    speak("User won")

                elif coom=="Scissior" and usr=="Rock":
                    speak("User won")

                elif usr=="Scissior" and coom=="Rock":
                    speak("Computer won")
```

- Output

```
Listening...
Recognizing...
User said: I am feeling boring

Listening...
Recognizing...
User said: yes

Listening...
Recognizing...
User said: paper

User Symbol :  paper
Computer Symbol :  Rock
```

- Changing Background of Window

```
        elif 'change background' in query:
            ctypes.windll.user32.SystemParametersInfoW(20, 0, "C:\\Users\\GAURAV\\OneDrive\\Minor Project\\Voice\\back.jpg" , 0)
            speak("Background changed succesfully")
```

- Telling Top News from different News website i.e “The Times of India”, “Google News” and “BBC”.

-Google News

```
        elif 'google news' in query:
            try:
                jsonObj = urlopen('''https://newsapi.org/v2/top-headlines?sources=google-news-in&apiKey=5c06fed7ad3f4c78bb4c3a44255788cd''')
                data = json.load(jsonObj)
                i = 1
                speak('')
                print('''==============Google News===========''' + '\n')
                for item in data['articles']:
                    print(str(i) + '. ' + item['title'] + '\n')
                    print(item['description'] + '\n')
                    speak(str(i) + '. ' + item['title'] + '\n')
                    i += 1
            except Exception as e:
                    print(str(e))
```

- The Times of India

```
        elif 'news' in query: #samachar
            try:
                jsonObj = urlopen('''https://newsapi.org/v1/articles?source=the-times-of-india&sortBy=top&apiKey=5eeb7514007b4690b7195b4d197a75d4''')
                data = json.load(jsonObj)
                i = 1
                speak('here are some top news from the times of india')
                print('''==============TIMES OF INDIA===========''' + '\n')
                for item in data['articles']:
                    print(str(i) + '. ' + item['title'] + '\n')
                    print(item['description'] + '\n')
                    speak(str(i) + '. ' + item['title'] + '\n')
                    i += 1
            except Exception as e:
                print(str(e))
```

- BBC News

```
        elif "bbc news" in query:
            try:
                main_url = " https://newsapi.org/v1/articles?source=bbc-
news&sortBy=top&apiKey=5c06fed7ad3f4c78bb4c3a44255788cd"
                open_bbc_page = requests.get(main_url).json()
                article = open_bbc_page["articles"]
                results = []
                for ar in article:
                    results.append(ar["title"])
                for i in range(len(results)):
                    print(i + 1, results[i])
            except Exception as e:
                print(str(e))
```

■ Perfoming Varoius System (OS) Operations

- Shutdown

```
        elif 'shutdown system' in query:
            speak("Hold On a Sec! Your system is on its way to shut down")
            subprocess.call('shutdown /p /f')
```

- Lock

```
        elif 'lock window' in query or "system ko lock Karen" in query:
            speak("locking the device")
            ctypes.windll.user32.LockWorkStation()
```

- Hibernate / Sleep

```
        elif "hibernate" in query or "sleep" in query:
            speak("Hibernating")
            subprocess.call("shutdown /i /h")
```

- Clearing Recycle Bin

```
        elif 'empty recycle bin' in query:
            winshell.recycle_bin().empty(confirm=False, show_progress=False, sound=True)
            speak("Recycle Bin Recycled")
```

- Restart

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

- Log Off

```
        elif "log off" in query or "sign out" in query:
            speak("Make sure all the application are closed before sign-out")
            time.sleep(5)
            subprocess.call(["shutdown", "/l"])
```

- Locating Different Places Around the World

```
        elif "where is" in query:
            query=query.replace("where is","")
            location = query
            speak("User asked to Locate")
            speak(location)
            webbrowser.open("https://www.google.nl/maps/place/" + location + "")
```

- Writing Short Notes

```
        elif "write a note" in query:
            speak("What should i write , sir")
            note= takeCommand()
            file = open('jarvis.txt','w')
            speak("Sir, Should i include date and time")
            snfm = takeCommand()
            if 'yes' in snfm or 'sure' in snfm:
                strTime = datetime.datetime.now().strftime("%H:%M:%S")
                file.write(strTime)
                file.write(" :- ")
                file.write(note)
            else:
                file.write(note)

        elif "show note" in query:
            speak("Showing Notes")
            file = open("jarvis.txt", "r")
            print(file.read())
            speak(file.read(6))
```

- Starting a countdown timer

```
def countdown(n) :
    while n > 0:
        print (n)
        n = n - 1
    if n ==0:
```

```
        print('BLAST OFF!')
        elif "countdown of" in query:
            contt = [int(word) for word in query.split() if word.isdigit()]
            countdown(contt)
```

- Getting weather updates

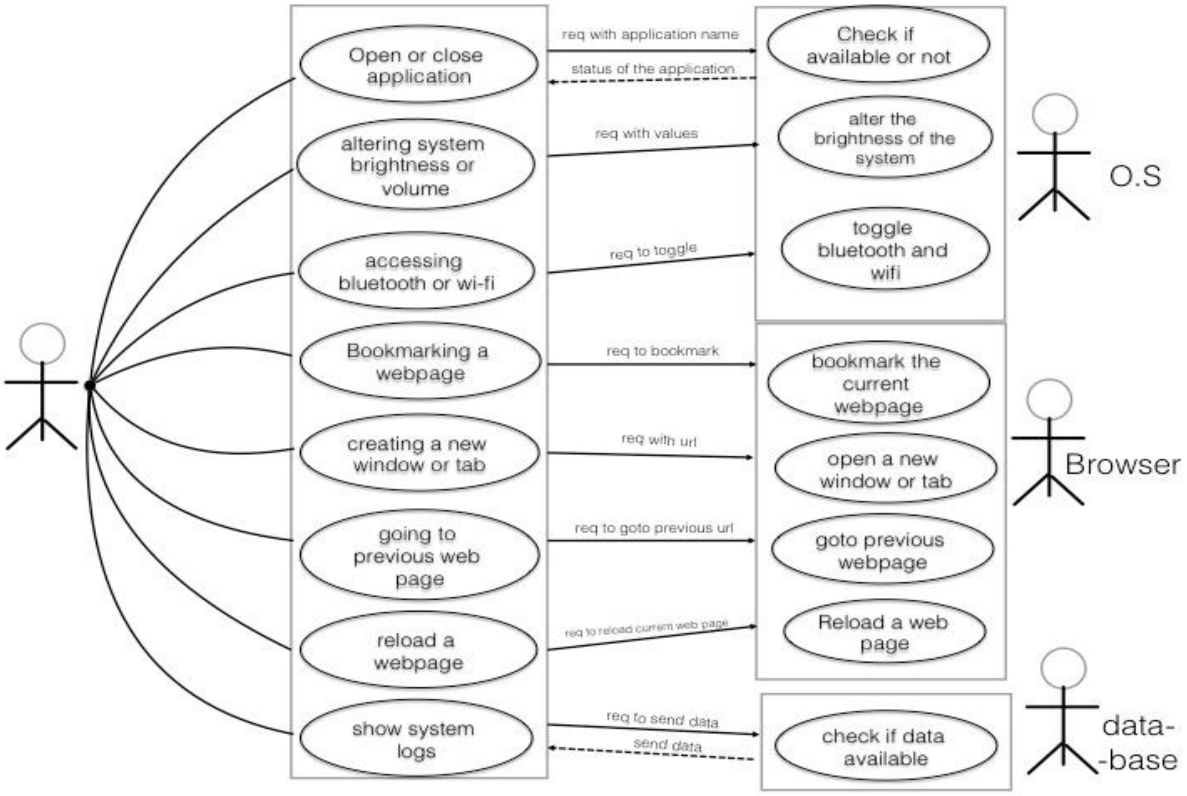
```
        elif "weather" in query:
            api_key = "139ff8e5644894750d3293adb1372433"
            base_url = "http://api.openweathermap.org/data/2.5/weather?"
            speak(" City name ")
            print("City name : ")
            city_name=takeCommand()
            complete_url = base_url + "appid=" + api_key + "&q=" + city_name
            response = requests.get(complete_url)
            x = response.json()
            if x["cod"] != "404":
                y = x["main"]
                current_temperature = y["temp"]
                current_pressure = y["pressure"]
                current_humidiy = y["humidity"]
                z = x["weather"]
                weather_description = z[0]["description"]
                print(" Temperature (in kelvin unit) = " +str(current_temperature)+"\n atmosp
heric pressure (in hPa unit) ="+str(current_pressure) +"\n humidity (in percentage) = " +str(
current_humidiy) +"\n description = " +str(weather_description))
            else:
                speak(" City Not Found ")
```

- Output

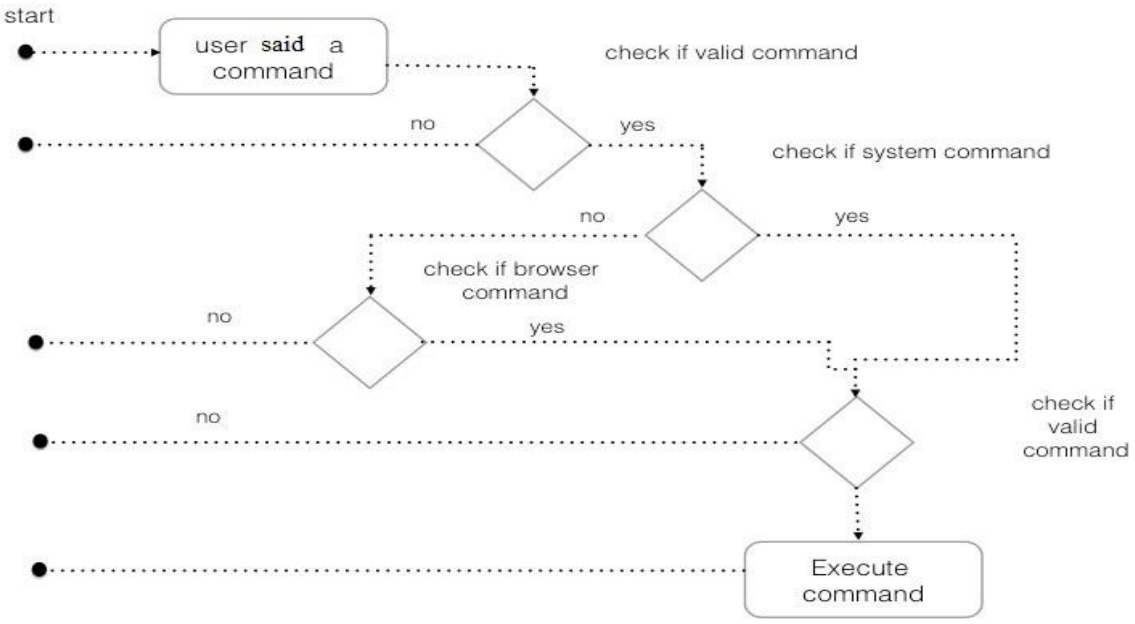
```
User said: weather update

City name :
Listening...
Recognizing...
User said: New Delhi

 Temperature (in kelvin unit) = 300.15
 atmospheric pressure (in hPa unit) =1006
 humidity (in percentage) = 42
 description = haze
```

Project Structure:

Jarvis will get all its commands from user/client asking him to perform any specific task and operation after getting user query Jarvis will search that query in all the commands that are defined their. After find query command type it will perform operation defined inside that if the query is a question then it will go to wolframalpha which will search that question inside their own server using the API key that is validated inside the program after searching the query it will return with the answer of the question which will further printed on the user screen.

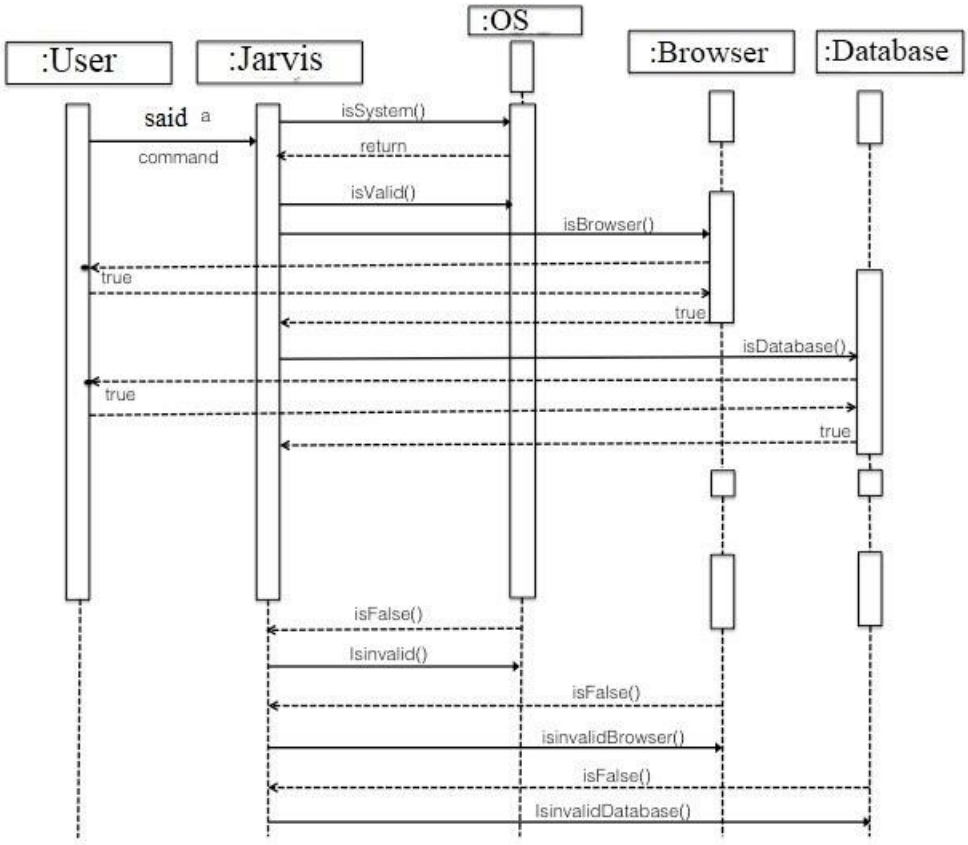


Reaction of Jarvis towards any query :

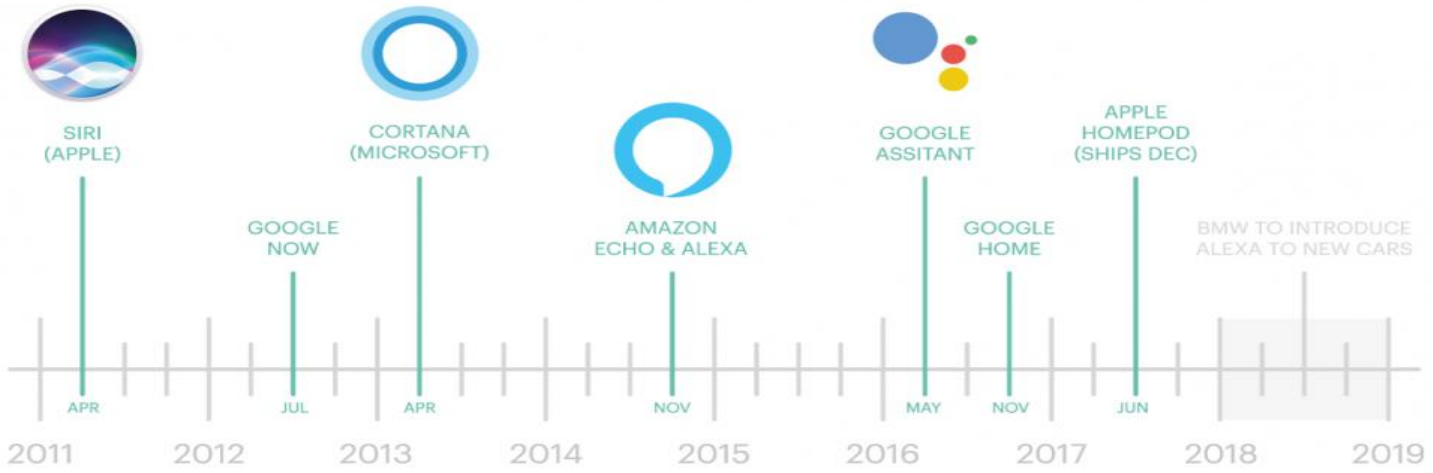


Sequence diagram:

Jarvis takes command from user through Internal Microphone or any other device that can be used as microphone after that if uses Google Text to Speech for Converting those command from Speech to Text. After that a sequence of Pre-Defined Commands are set inside the program to perform various tasks according to user command different task is performed.



History of Voice Assistant:



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

Requirement's:

Limitations:

- Need a network connectivity every time to become operational.
- Limited Commands to perform.
- Limited to Microsoft Windows Only.
- Need somewhat Noise free environment.
- To compile source code Python and many other modules needed to be installed.

- Errors occurred during the project

- Module Functionality
- Internet was one of the main errors

■ How I Solved them:

- I have made sure that there is no extra module added or any useful module removed from our systems until we have completed the project

- Since my whole program is dependent of Internet connectivity a proper and high-speed internet connection was necessary, we always kept our system in the Internet zone to make sure that all the function that we have added is working properly as it should by checking them while adding that.
-

Source Code :

```
import subprocess
import wolframalpha
import wikiquote
import pyttsx3
import json
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import winshell
import pyjokes
import feedparser
import smtplib
import ctypes
import time
import requests
import random
import fileinput
import getpass
import wmi
import os
from pathlib import Path
import tkinter as tk
from tkinter import filedialog
from selenium import webdriver
from bs4 import BeautifulSoup
import win32com.client as wincl
from urllib.request import urlopen


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

root = tk.Tk()
root.withdraw()

DIRECTORIES = {
    "HTML": [".html5", ".html", ".htm", ".xhtml"],
    "IMAGES": [".jpeg", ".jpg", ".tiff", ".gif", ".bmp", ".png", ".bpg", "svg",
               ".heif", ".psd"],
    "VIDEOS": [".avi", ".flv", ".wmv", ".mov", ".mp4", ".webm", ".vob", ".mng",
               ".qt", ".mpg", ".mpeg", ".3gp", ".mkv"],
    "DOCUMENTS": [".oxps", ".epub", ".pages", ".docx", ".doc", ".fdf", ".ods",
```

```
                ".odt", ".pwi", ".xsn", ".xps", ".dotx", ".docm", ".dox",
                ".rvg", ".rtf", ".rtfd", ".wpd", ".xls", ".xlsx", ".ppt",
                "pptx"],
    "ARCHIVES": [".a", ".ar", ".cpio", ".iso", ".tar", ".gz", ".rz", ".7z",
                 ".dmg", ".rar", ".xar", ".zip"],
    "AUDIO": [".aac", ".aa", ".aac", ".dvf", ".m4a", ".m4b", ".m4p", ".mp3",
              ".msv", "ogg", "oga", ".raw", ".vox", ".wav", ".wma"],
    "PLAINTEXT": [".txt", ".in", ".out"],
    "PDF": [".pdf"],
    "PYTHON": [".py"],
    "XML": [".xml"],
    "EXE": [".exe"],
    "SHELL": [".sh"]
}
FILE_FORMATS = {file_format: directory
                for directory, file_formats in DIRECTORIES.items()
                for file_format in file_formats}

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

def countdown(n) :
    while n > 0:
        print (n)
        n = n - 1
    if n ==0:
        print('BLAST OFF!')

def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning Sir!")

    elif hour>=12 and hour<18:
        speak("Good Afternoon Sir!")

    else:
        speak("Good Evening Sir!")

    assname=("Jarvis 1 point o")
    speak("I am your Assistant")
    speak(assname)

def usrname():
    speak("What should i call you sir")
    uname=takeCommandname()
    speak("Welcome Mister")
    speak(uname)
    print("#####################")
```

```
    print("Welcome Mr.",uname)
    print("#####################")

def quotaton():
    speak(wikiquote.quote_of_the_day())
    print(wikiquote.quote_of_the_day())

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

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

    except Exception as e:
        print(e)
        print("Unable to Recognizing your voice.")
        return "None"
    return query

def takeCommandname():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Username...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Trying to Recognizing Name...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:
        print(e)
        print("Unable to Recognizing your name.")
        takeCommandname()
        return "None"
    return query

def takeCommandmessage():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Enter Your Message")
        r.pause_threshold = 1
        audio = r.listen(source)
```

```
    try:
        query = r.recognize_google(audio, language='en-in')
        print(f'Message to be sent is : {query}\n')

    except Exception as e:
        print (e)
        print("Unable to recognize your message")
        print("Check your Internet Connectivity")
    return query

def takeCommanduser():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Name of User or Group")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        query = r.recognize_google(audio, language='en-in')
        print(f'Client to whome message is to be sent is : {query}\n')

    except Exception as e:
        print (e)
        print("Unable to recognize Client name")
        speak("Unable to recognize Client Name")
        print("Check your Internet Connectivity")
    return query

def organize():
    for entry in os.scandir():
        if entry.is_dir():
            continue
        file_path = Path(entry.name)
        file_format = file_path.suffix.lower()
        if file_format in FILE_FORMATS:
            directory_path = Path(FILE_FORMATS[file_format])
            directory_path.mkdir(exist_ok=True)
            file_path.rename(directory_path.joinpath(file_path))
    try:
        os.mkdir("OTHER")
    except:
        pass
    for dir in os.scandir():
        try:
            if dir.is_dir():
                os.rmdir(dir)
            else:
                os.rename(os.getcwd() + '/' + str(Path(dir)), os.getcwd() + '/OTHER/' + str(P
ath(dir)))
```

```
        except:
            pass

if __name__ == '__main__':
    clear = lambda: os.system('cls')
    clear()
    wishMe()
    usrname()
    speak("Can i tell you a quote of day")
    useropt=takeCommand().lower()
    if 'yes' in useropt or 'sure' in useropt:
        quotaton()
    else:
        speak("Taking you to command function")

    speak("How can i Help you, Sir")
    while True:
        query = takeCommand().lower()
        assname=("Jarvis 1 point o")
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=3)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif "good morning" in query:
            speak("A warm" +query)
            speak("How are you Mister")
            speak(assname)

        elif "wikipedia" in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=3)
            speak("According to Wikipedia")
            r = sr.Recognizer()
            print(results)
            speak(results)

        elif 'open youtube' in query:
            speak("Here you go to Youtube\n")
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            speak("Here you go to Google\n")
            webbrowser.open("google.com")

        elif "change brightness to " in query:
```

```
            brightness = [int(word) for word in query.split() if word.isdigit()]
            c = wmi.WMI(namespace='wmi')
            methods = c.WmiMonitorBrightnessMethods()[0]
            methods.WmiSetBrightness(brightness, 0)

        elif "Organize Files" in query:
            organize()

        elif 'open stackoverflow' in query:
            speak("Here you go to Stack Over flow.Happy coding")
            webbrowser.open("stackoverflow.com")

        elif "send a whatsaap message" in query or "send WhatsApp message" in query:
            driver = webdriver.Chrome('C:\\Users\\gaura\\OneDrive\\Desktop\\Major Project\\Vo
ice Assistant\\chromedriver_win32\\chromedriver.exe')
            driver.get('https://web.whatsapp.com/')
            speak("Scan QR code before proceding")
            tim=10
            time.sleep(tim)
            speak("Enter Name of Group or User")
            name = takeCommanduser()
            speak("Enter Your Message")
            msg = takeCommandmessage()
            user = driver.find_element_by_xpath('//span[@title = "{}"]'.format(name))
            user.click()
            msg_box = driver.find_element_by_class_name('_3u328')
            msg_box.send_keys(msg)
            button = driver.find_element_by_class_name('_3M-N-')
            button.click()

        elif "explorer" in query:
            speak("Opening explorer")
            subprocess.call("explorer")

        elif "excel" in query:
            speak("Mention file location")
            file_path = filedialog.askopenfilename()
            os.startfile(file_path)

        elif "word" in query:
            speak("Mention file location")
            file_path = filedialog.askopenfilename()
            os.startfile(file_path)

        elif "stackoverflow " in query:
            speak("Opening Stackoverflow")
            webbrowser.open("stackoverflow.com")

        elif 'play music' in query or "play song" in query or "gaana"in query or "song" in qu
ery:
```

```
            speak("Here you go with music")
            username = getpass.getuser()
            try:
                music_dir = "C:\\Users\\"+username+"\\Music"
            except Exception as e:
                print("Following Exception Occured",e)
                speak("Following Exception Occured",e)

            finally:
                songs = os.listdir(music_dir)
                print(songs)
                random=os.startfile(os.path.join(music_dir, songs[1]))

        elif 'the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")

        elif 'open opera' in query:
            try:
                subprocess.call("opera")
            except Exception as e:
                print(e)

        elif 'how are you' in query:
            speak("I am fine , Thank you")
            speak("How are you, Sir")

        elif "change my name to" in query:
            query=query.replace("change my name to","")
            assname=query

        elif "change name" in query:
            speak("What would you like to call me ,Sir ")
            assname = takeCommand()
            speak("Thanks for naming me")

        elif "what's your name" in query or "What is your name" in query:
            speak("My friends call me")
            speak(assname)
            print("My friends call me",assname)

        elif 'exit' in query:
            speak("Thanks for giving me your time")
            exit()

        elif "who made you" in query or "who created you" in query:
            speak("I have been created by Gaurav.")

        elif 'joke' in query:
            speak(pyjokes.get_joke())
```

```
        elif 'add' in query or "subsctract" in query or "multiply" in query or "divide" in qu
ery:
            num1,num2 = [int(word) for word in query.split() if word.isdigit()]

            try:
                if "add" in query:
                    speak(num1+num2)

                elif "substract" in query:
                    speak(num1-num2)

                elif "multiply" in query:
                    speak(num1*num2)

                elif "divide" in query:
                    speak(num1/num2)

            except Exception as e:
                print(e)

            finally:
                app_id = "WTHP37-K6P2X72X3E"
                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
                print("The answer is " + answer)
                speak("The answer is " + answer)

        elif 'search' in query or 'play' in query:
            query = query.replace("search", "")
            query = query.replace("play", "")
            webbrowser.open(query)

        elif "boring" in query or "fun" in query:
            speak("Would you like to play RPS")
            opt=takeCommand()
            if "yes" in opt:
                speak("Rock , Paper or Scissor")
                usr=takeCommand()
                action=("Rock","Paper","Scissior","Rock","Paper","Scissior","Rock","Paper","S
cissior")
                coom=random.choices(action)
                coom = ' '.join(map(str, coom))
                print("User Symbol : ",usr)
                print("Computer Symbol : ",coom)

                if coom==usr:
```

```
                    speak("Match Draw")

                elif coom=="Scission" and usr=="Paper":
                    speak("Computer Won")

                elif usr=="Scission" and coom=="Paper":
                    speak("User won")

                elif usr=="Rock" and coom=="Paper":
                    speak("Computer won")

                elif coom=="Rock" and usr=="Paper":
                    speak("User won")

                elif coom=="Scission" and usr=="Rock":
                    speak("User won")

                elif usr=="Scission" and coom=="Rock":
                    speak("Computer won")

            else:
                speak(pyjokes.get_joke())

        elif "who i am" in query:
            speak("If you talk then definately your human.")

        elif "why you came to world" in query:
            speak("Thanks to Gaurav. further It's a secret")

        elif 'power point presentation' in query:
            speak("opening Power Point presentation")
            power= r"C:\\Users\\GAURAV\\Desktop\\Minor Project\\Presentation\\Voice Assistant
.pptx"
            os.startfile(power)

        elif "who are you" in query:
            speak("I am your virtual assistant created by Gaurav")

        elif 'reason for you' in query:
            speak("I was created as a Major project by Mister Gaurav ")

        elif 'change background' in query:
            ctypes.windll.user32.SystemParametersInfoW(20, 0, "C:\\Users\\GAURAV\\OneDrive\\M
inor Project\\Voice\\back.jpg" , 0)
            speak("Background changed succesfully")

        elif 'open bluestack' in query:
            appli= r"C:\\ProgramData\\BlueStacks\\Client\\Bluestacks.exe"
            os.startfile(appli)
```

```
        elif 'google news' in query:
            try:
                jsonObj = urlopen('''https://newsapi.org/v2/top-headlines?sources=google-
news-in&apiKey=5c06fed7ad3f4c78bb4c3a44255788cd''')
                data = json.load(jsonObj)
                i = 1
                speak('')
                print('''==============Google News===========''' + '\n')
                for item in data['articles']:
                    print(str(i) + '. ' + item['title'] + '\n')
                    print(item['description'] + '\n')
                    speak(str(i) + '. ' + item['title'] + '\n')
                    i += 1
            except Exception as e:
                    print(str(e))

        elif "bbc news" in query:
            try:
                main_url = " https://newsapi.org/v1/articles?source=bbc-
news&sortBy=top&apiKey=5c06fed7ad3f4c78bb4c3a44255788cd"
                open_bbc_page = requests.get(main_url).json()
                article = open_bbc_page["articles"]
                results = []
                for ar in article:
                    results.append(ar["title"])
                for i in range(len(results)):
                    print(i + 1, results[i])
            except Exception as e:
                print(str(e))

        elif 'news' in query: #samachar
            try:
                jsonObj = urlopen('''https://newsapi.org/v1/articles?source=the-times-of-
india&sortBy=top&apiKey=5eeb7514007b4690b7195b4d197a75d4''')
                data = json.load(jsonObj)
                i = 1
                speak('here are some top news from the times of india')
                print('''==============TIMES OF INDIA===========''' + '\n')
                for item in data['articles']:
                    print(str(i) + '. ' + item['title'] + '\n')
                    print(item['description'] + '\n')
                    speak(str(i) + '. ' + item['title'] + '\n')
                    i += 1
            except Exception as e:
                print(str(e))

        elif 'lock window' in query or "system ko lock Karen" in query:
            speak("locking the device")
            ctypes.windll.user32.LockWorkStation()
```

```
        elif 'shutdown system' in query:
            speak("Hold On a Sec! Your system is on its way to shut down")
            subprocess.call('shutdown /p /f')


        elif 'empty recycle bin' in query:
            winshell.recycle_bin().empty(confirm=False, show_progress=False, sound=True)
            speak("Recycle Bin Recycled")


        elif "don't listen" in query or "stop listening" in query:
            speak("for how much time you want to stop jarvis from listening commands")
            a=int(takeCommand())
            time.sleep(a)
            print(a)

        elif "where is" in query:
            query=query.replace("where is","")
            location = query
            speak("User asked to Locate")
            speak(location)
            webbrowser.open("https://www.google.nl/maps/place/" + location + "")

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

        elif "hibernate" in query or "sleep" in query:
            speak("Hibernating")
            subprocess.call("shutdown /i /h")

        elif "log off" in query or "sign out" in query:
            speak("Make sure all the application are closed before sign-out")
            time.sleep(5)
            subprocess.call(["shutdown", "/l"])

        elif "countdown of" in query:
            contt = [int(word) for word in query.split() if word.isdigit()]
            countdown(contt)

        elif "write a note" in query:
            speak("What should i write , sir")
            note= takeCommand()
            file = open('jarvis.txt','w')
            speak("Sir, Should i include date and time")
            snfm = takeCommand()
            if 'yes' in snfm or 'sure' in snfm:
                strTime = datetime.datetime.now().strftime("%H:%M:%S")
                file.write(strTime)
                file.write(" :- ")
                file.write(note)
```

```
            else:
                file.write(note)

        elif "show note" in query:
            speak("Showing Notes")
            file = open("jarvis.txt", "r")
            print(file.read())
            speak(file.read(6))

        elif "jarvis" in query:
            wishMe()
            speak("Jarvis 1 point o in your service Mister")
            speak(assname)

        elif "weather" in query:
            api_key = "139ff8e5644894750d3293adb1372433"
            base_url = "http://api.openweathermap.org/data/2.5/weather?"
            speak(" City name ")
            print("City name : ")
            city_name=takeCommand()
            complete_url = base_url + "appid=" + api_key + "&q=" + city_name
            response = requests.get(complete_url)
            x = response.json()
            if x["cod"] != "404":
                y = x["main"]
                current_temperature = y["temp"]
                current_pressure = y["pressure"]
                current_humidiy = y["humidity"]
                z = x["weather"]
                weather_description = z[0]["description"]
                print(" Temperature (in kelvin unit) = " +str(current_temperature)+"\n atmosp
heric pressure (in hPa unit) ="+str(current_pressure) +"\n humidity (in percentage) = " +str(
current_humidiy) +"\n description = " +str(weather_description))
            else:
                speak(" City Not Found ")

        elif "wikipedia" in query:
            webbrowser.open("wikipedia.com")

        elif "will you be my gf" in query or "will you be my bf" in query:   #most asked ques
tion from google Assistant
            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 asked me that")

        elif "what is" in query or "who is" in query:
            client= wolframalpha.Client("WTHP37-K6P2X72X3E")
            res = client.query(query)
            try:
```

```
                print(next(res.results).text)
                speak(next(res.results).text)
            except StopIteration:
                print ("No results")

        elif "open Gmail" in query:
            webbrowser.open("https://mail.google.com/mail/")

        elif "open yahoo mail" in query:
            webbrowser.open("https://in.mail.yahoo.com")

        elif "Show project Report" in query:
            speak("Opening Major Project Report")
            projectre= r"C:\\Users\\GAURAV\\Desktop\\Minor Project\\Presentation\\Project Rep
ort.docx"
            os.startfile(projectre)
```

GitHub Link : https://github.com/Ghack9/Personal-Asssistant-Major-Project

Reference:

1) https://towardsdatascience.com/build-your-first-voice-assistant-85a5a49f6cc1
2) https://medium.com/@sundarstyles89/create-your-own-google-assistant-voice-based-assistant-using-python-94b577d724f9
3) https://en.wikipedia.org/wiki/Virtual_assistant
4) https://towardsdatascience.com/build-your-first-voice-assistant-85a5a49f6cc1