

Mini Project on Voice Activated Desktop Assistant

Index

- 1. Introduction**
- 2. History/existing system**
- 3. Algorithm used**
- 4. Software and Hardware requirements**
- 5. Applications**
- 6. Conclusion**
- 7. Code and output screenshots**
- 8. References**

1. Introduction

A virtual assistant or intelligent personal assistant is a software agent that can perform tasks or services for an individual based on verbal command. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some virtual assistants are able to interpret 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, and calendars with verbal commands.

As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on voice user interfaces. Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows-based personal computers, smartphones and smart speakers. Amazon has a large install base for smart speakers. Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. Many continually learn using artificial intelligence techniques including machine learning.

2. History of Voice Activated Assistants

Early 1900s – 1990:

Radio Rex was the first voice activated toy released in 1911.

Another of early tool which was enabled to perform digital speech recognition was the IBM Shoebox, presented to the general public during the 1962 Seattle World's Fair after its initial market launch in 1961. This early computer, developed almost 20 years before the introduction of the first IBM Personal Computer in 1981, was able to recognize 16 spoken words and the digits 0 to 9. The next milestone in the development of voice recognition technology was achieved in the 1970s at the Carnegie Mellon University in Pittsburgh, Pennsylvania with substantial support of the United States Department of Defense and its DARPA agency. Their tool "Harpy" mastered about 1000 words, the vocabulary of a three-year-old. About ten years later the same group of scientists developed a system that could analyze not only individual words but entire word sequences enabled by a Hidden Markov Model. Thus, the earliest virtual assistants, which applied speech recognition software were automated attendant and medical digital dictation software.

The 1990s – Present:

In the 1990s digital speech recognition technology became a feature of the personal computer with Microsoft, IBM, Philips and Lernout & Hauspie fighting for customers. Much later the market launch of the first smartphone IBM Simon in 1994 laid the foundation for smart virtual assistants as we know them today.

The first modern digital virtual assistant installed on a smartphone was Siri, which was introduced as a feature of the iPhone 4S on October 4, 2011. Apple Inc. developed Siri following the 2010 acquisition of Siri Inc., a spin-off of SRI International, which is a research institute financed by DARPA and the United States Department of Defense.

3. Algorithm Used

- The commands from user are recognized using Speech Recognition. We are using the default API key for Google Speech Recognition to recognize speech spoken by the user.
- A list of operations is provided with respect to the spoken speech. Using regular expressions, we extract information and match it with provided list of operations. Regular expression is a sequence of characters that define a search pattern. Usually this pattern is used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation.
- The operation once matched will execute the code defined in it.

The webbrowser module is used for opening a folder, to open any website and to display the current weather in a particular city in India. The selenium package is used to automate web browser interaction from Python.

To make or remove a folder the os.mkdir or os.rmdir functions of the os module are used.

The vlc module is used to play or stop music. The program is terminated using sys.exit function of exitstatus module.

- The output is given by the desktop assistant in text as well as speech medium using Text-To-Speech(TTS) methods provided by pyttsx3 module. The assistant continues to excute until it is commanded to terminate.

4. Software and Hardware requirements

The system requirements are as follows.

Hardware Used

- 1.6 GHz
- 64 Bit Operating System : Windows 10
- 8 GB RAM
- 1TB Hard-disk drive

Software Used

- Python 3.7

5. Code and Screenshots

Code

```
from gtts import gTTS

import speech_recognition as sr

import sys

from exitstatus import ExitStatus

import re

import webbrowser

import requests

import pyttsx3

from selenium import webdriver

import winsound

import vlc

import os

#for audio

def talkToMe(audio):

    print(audio)

    engine = pyttsx3.init()

    sound = engine.getProperty('voices')

    engine.setProperty('voice', sound[1].id)#for female voice

    engine.say(audio)

    engine.runAndWait()

#to listen commands

def myCommand():

    r = sr.Recognizer()
```

```

with sr.Microphone() as source:

    print('Ready...')

    r.pause_threshold = 1

    r.adjust_for_ambient_noise(source, duration=1)

    audio = r.listen(source)

    try:

        command = r.recognize_google(audio).lower()

        print('You said: ' + command + '\n')

        #loop back to continue to listen for commands if unrecognizable speech is received

    except sr.UnknownValueError:

        talkToMe('Your last command couldn\'t be heard. Try again')

        command = myCommand();

    return command

def assistant(command):

    #if statements for executing commands

    if 'what\'s up' in command:

        talkToMe('Just doing my thing')

    elif 'hey' in command:

        talkToMe('Hello. How may I help you?')

    #To open a folder on desktop

    elif 'open folder' in command:

        reg_ex = re.search('open folder (.*)',command)

        if reg_ex:

            folder = reg_ex.group(1)

            webbrowser.open("C:\\Users\\Sanika Sanaye\\Desktop\\"+folder)

```

```

talkToMe('What should I do next?')

key2 = myCommand()

#To create folder in current folder

if 'create' in key2:

    reg_ex = re.search('create (.*)', key2)

    if reg_ex:

        subfolder = reg_ex.group(1)

        os.mkdir("C:\\Users\\Sanika Sanaye\\Desktop\\"+folder+"\\"+subfolder)

#To remove folder from current folder

elif 'remove' in key2:

    reg_ex = re.search('remove (.*)', key2)

    if reg_ex:

        subfolder = reg_ex.group(1)

        os.rmdir("C:\\Users\\Sanika Sanaye\\Desktop\\"+folder+"\\"+subfolder)

talkToMe('Done')

#To open YouTube

elif 'open youtube' in command:

    url = 'https://www.youtube.com/'

    webbrowser.open(url)

    talkToMe('What would you like to do next')

    key = myCommand()

    if 'search for' in key:

        reg_ex = re.search('search for (.*)', key)

        url = 'https://www.youtube.com/results?search_query='

        if reg_ex:

```

```

        search = reg_ex.group(1)

        url = url + search

        webbrowser.open_new_tab(url)

    talkToMe('Done')

#To google search something

elif 'google search' in command:

    reg_ex = re.search('google search (.*)', command)

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

    if reg_ex:

        search = reg_ex.group(1)

        url = url + search

        webbrowser.open(url)

    talkToMe('Done')

#To open a website

elif 'open' in command

    reg_ex = re.search('open (.*)', command)

    if reg_ex:

        website = reg_ex.group(1)

        webbrowser.open("https://www."+website+".com")

#To tell current weather in a city in india

elif 'current weather in' in command:

    driver = webdriver.Chrome()

    reg_ex = re.search('current weather in (.*)', command)

    if reg_ex:

        city = reg_ex.group(1)

```

```

        driver.get("https://www.euronews.com/weather/asia/india/"+city)

        talkToMe("The current weather in %s is %s'
%(city,driver.find_element_by_xpath("//*[@class='unit_C
block__cityWeather__forecast__desc ltr no-unit']").text))

#To play music

elif 'play music' in command:

    p = vlc.MediaPlayer("C:\\Users\\Sanika Sanaye\\Downloads\\MAMAMOO_-
_Sleep_In_The_Car_(mp3.pm).mp3")

    p.play()

    print("press 1 to close music")

    key1 = input()

    if '1' in key1:

        p.stop()

#To exit the program

elif 'exit' in command:

    talkToMe("Thanks Sanika. See you next time")

    sys.exit(ExitStatus.success)

#when a command doesn't match any given commands

else:

    talkToMe('I don\'t know what you mean!')

talkToMe('Welcome to DONNA desktop assistant.')

talkToMe("1. What's up\n2. Hey\n3. Open Folder\n 3.1. Create\n 3.2. Remove\n4. Open
YouTube\n 4.1. Search for\n5. Google Search\n6. Open\n7. Current Weather in City\n8.
Play Music\n9. Exit\n")

talkToMe("How may I help you?")

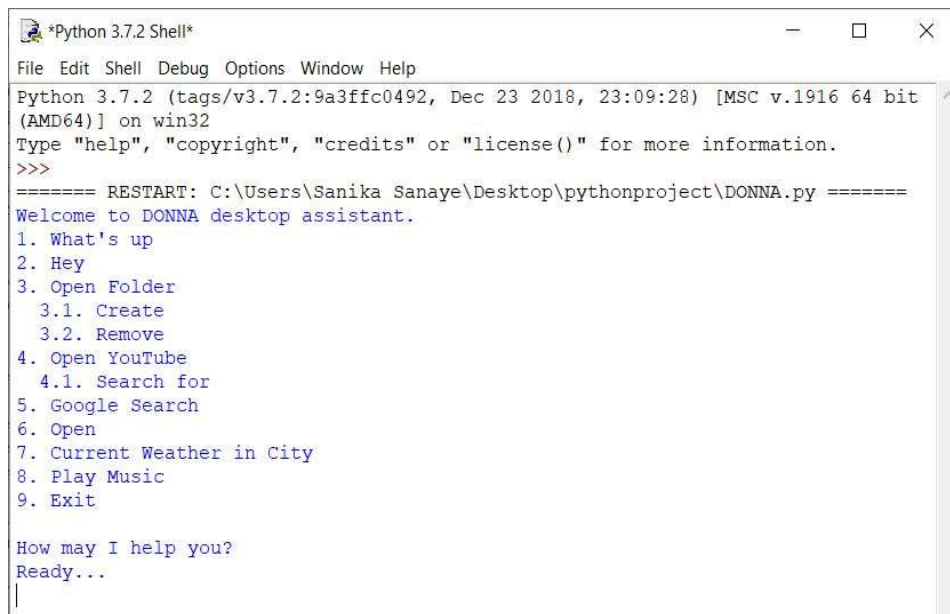
#loop to continue executing

while True:

    assistant(myCommand())

```


Screenshots



```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Sanika Sanaye\Desktop\pythonproject\DONNA.py =====
Welcome to DONNA desktop assistant.
1. What's up
2. Hey
3. Open Folder
   3.1. Create
   3.2. Remove
4. Open YouTube
   4.1. Search for
5. Google Search
6. Open
7. Current Weather in City
8. Play Music
9. Exit

How may I help you?
Ready...
|
```

Figure 1: View of assistant on boot

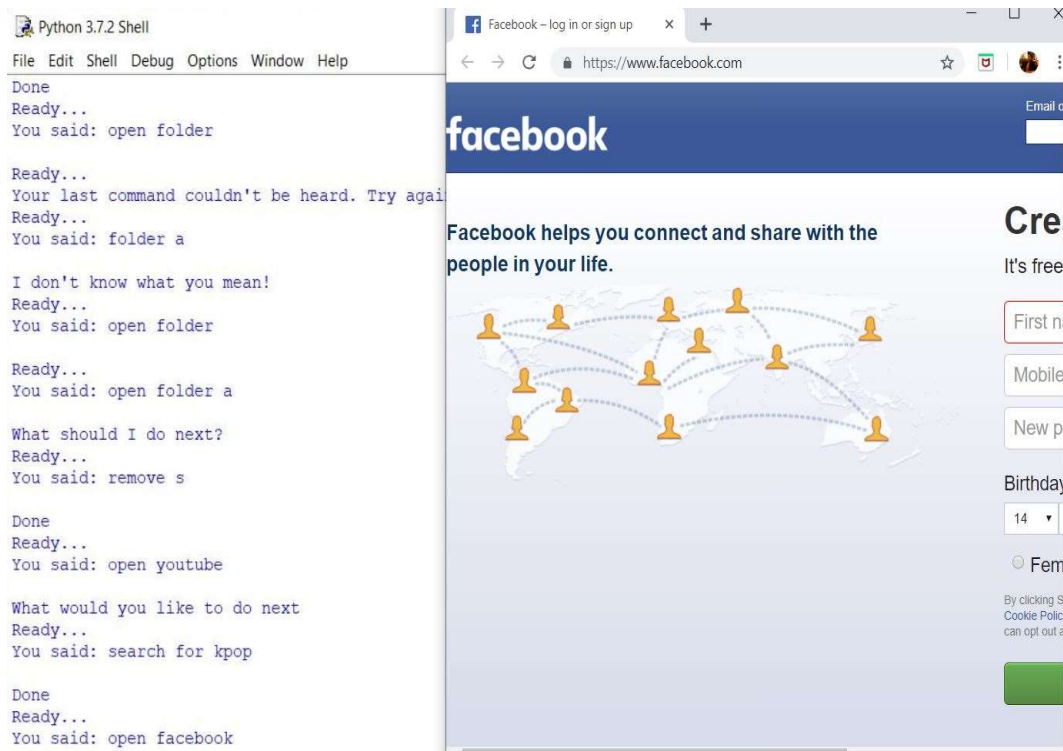


Figure 2 : Opening Facebook

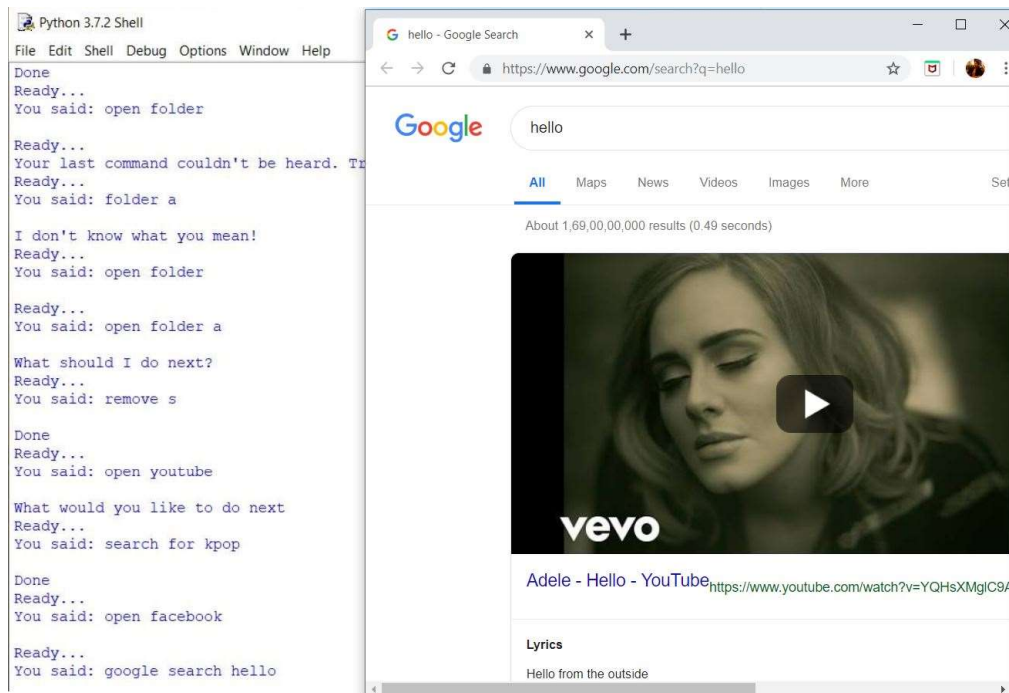


Figure 3 : Google Search

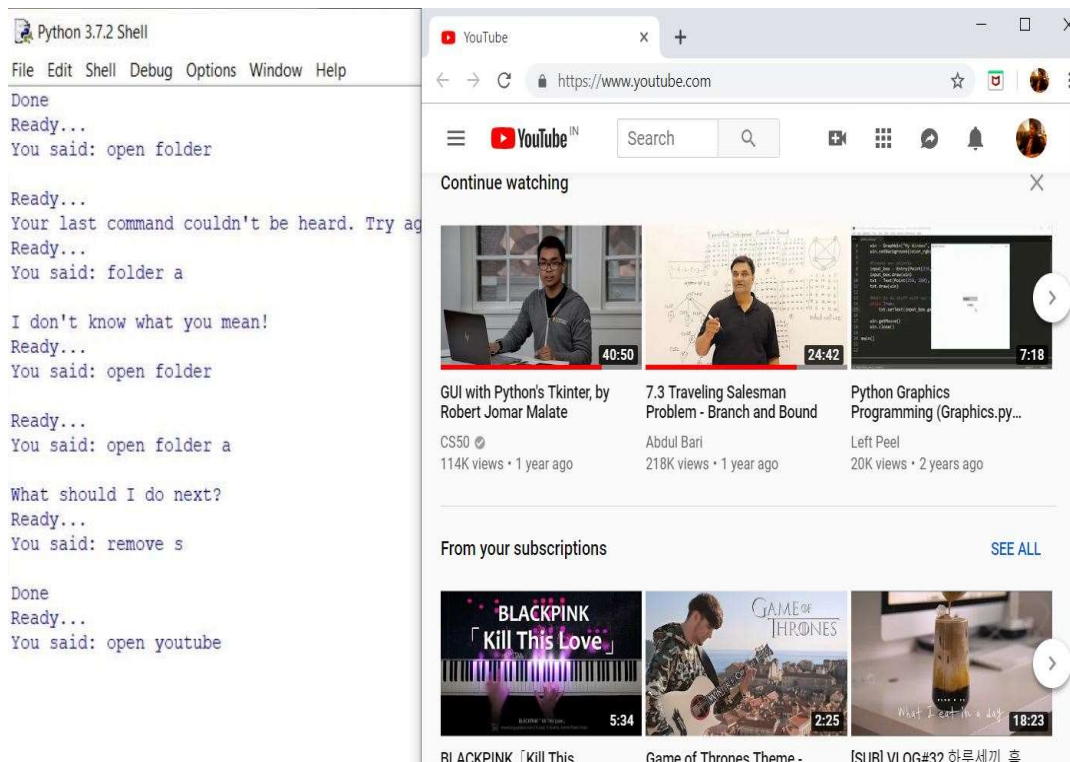


Figure 4 : Opening YouTube

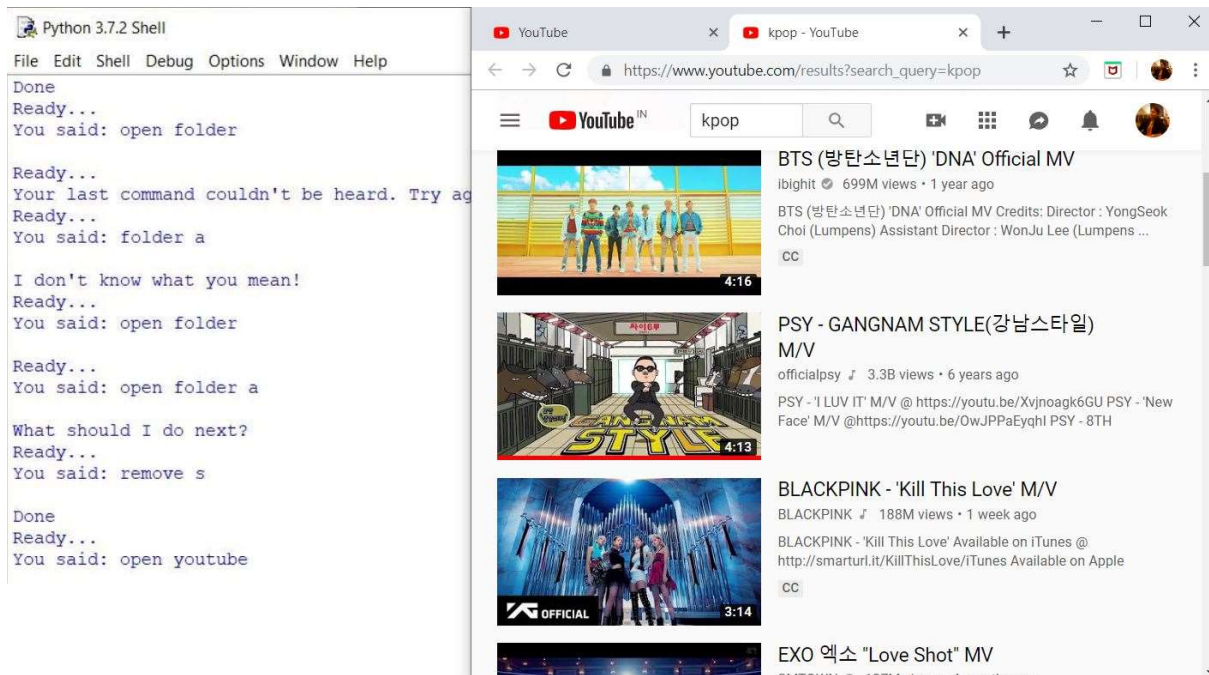


Figure 5 : Searching in YouTube

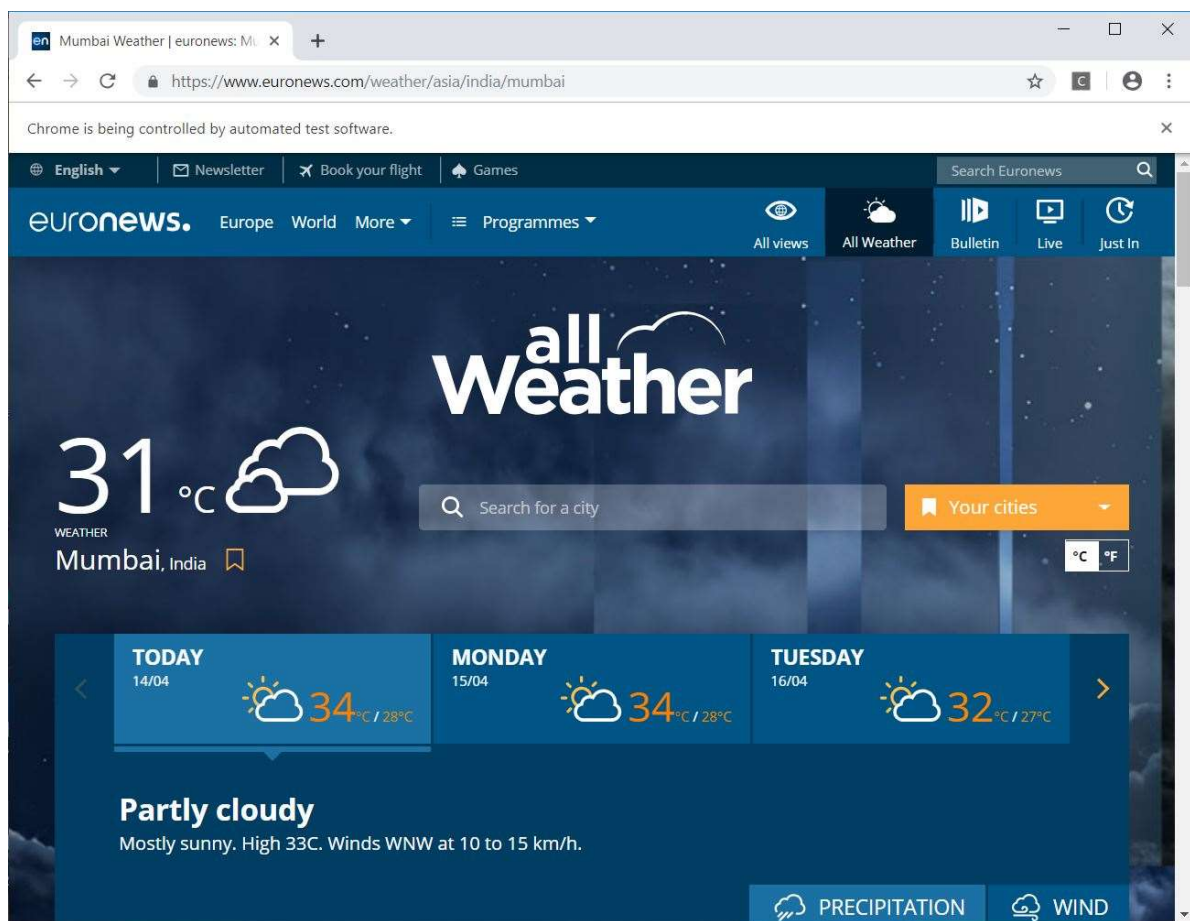


Figure 6(a) : Displaying current weather in city

```

Ready...
Your last command couldn't be heard. Try again
Ready...
You said: current weather in mumbai

The current weather in mumbai is Mostly sunny. High 33C. Winds WNW at 10 to 15 k
m/h.

```

Figure 6(b) : Tells the current weather in city

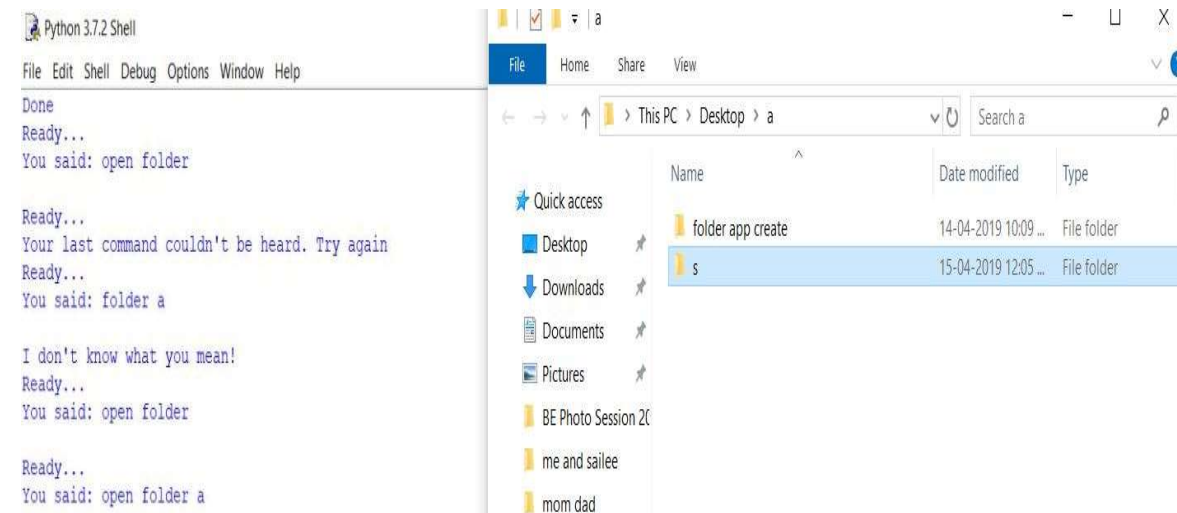


Figure 7 : Opens folder

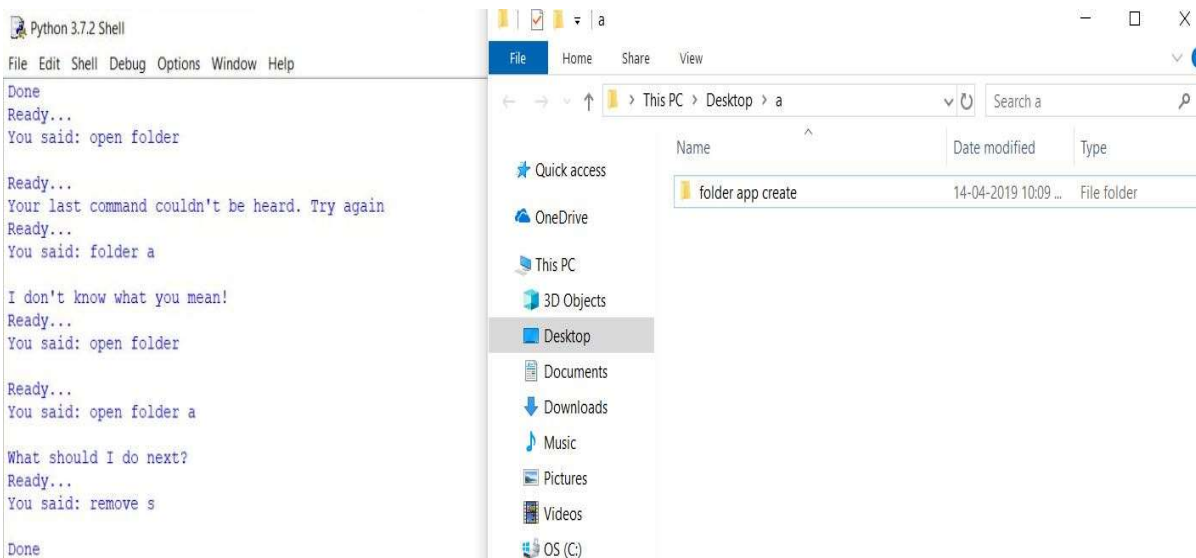


Figure 8 : Removes specified folder

```
Ready...
Your last command couldn't be heard. Try again
Ready...
You said: current weather in mumbai

The current weather in mumbai is Mostly sunny. High 33C. Winds WNW at 10 to 15 k
m/h.
Ready...
You said: play music

press 1 to close music
1
Ready...
You said: exit

Thanks Sanika. See you next time
>>>
```

Figure 9 : Play music operation and exit operation

Conclusion

Voice activated assistants are in wide use in technology, gaming and media industries. Popularity of Amazon's Alexa, iPhone's Siri and Microsoft's Cortana only goes forward to prove the versatility of this software. The DONNA desktop assistant aims to capture the operativity of these renowned softwares in the most basic terms.

References

1. <https://github.com/jg-fisher/desktopAssistant/blob/master/desktopAssistant.py>
2. <https://pypi.org/project/pyttsx3/>
3. <https://pypi.org/project/selenium/>
4. https://github.com/the-javapocalypse/weather-forecaster-using-selenium/blob/master/weather_forecast.py