PROJECT REPORT

on

AI VOICE ASSISTANT

(J.A.R.V.I.S)

By

Amitesh Srivastava (171242)

Submitted to

CERTIFICATE

I hereby declare that the work presented in this report entitled “AI Voice Assistant” in partial fulfillment of the requirements for the Major Project(8th Semester) in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from Feb 2021 to May 2021 under the supervision of Dr. Mrityunjay Singh.

Amitesh Srivastava(171242)

This is certified to be the bonafied work of the students in the Major Project Lab during academic year 2021.

[illegible]

Supervisor - Dr. Mrityunjay Singh

Date – 18/05/2020

ACKNOWLEDGEMENT

Besides the hard work of a group, the success of a project also depends highly on the encouragement and guidelines of many others. I take this opportunity to express my sincere and heartfelt gratitude to the people who have been instrumental in the successful completion of this project.

My first and foremost acknowledgement goes to our supervisor and mentor**, Dr. Mrityunjay Singh**, without whose help the completion of this project wouldn’t have been possible.
It is because of her guidance and efforts that I was able to implement a practical idea based on my field of interest.
Last but not the least I would like to acknowledge my institution **Jaypee University Of Information Technology** for giving me a platform to give me life and implementation, to the various fields I have studied till date.

TABLE OF CONTENTS:

List of Abbreviations

1. A.I. -> Artificial Intelligence

List of Figures

4.10 -> what is education google search

4.11 -> search results

4.12 -> open instagram

4.13 -> Instagram opened on chrome

4.14 -> can you open the big bang theory series for me google search

4.15 -> search results

List of Graphs

List of Tables

ABSTRACT

The project aims to develop a personal-assistant for Linux-based systems. Jarvis draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant either through voice commands or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, bing or yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks. The user statements/commands are analysed with the help of machine learning to give an optimal solution.

1. INTRODUCTION

1.1 Introduction

Just imagine having an A.I. right hand just like one in the movie Iron man. Just think of it's applications like sending e-mails without opening up your mail, searching on Wikipedia and googling and playing music on youtube without using your web browser, and other date to day tasks done on a computer. In this project, we will demonstrate how we can make our own A.I. associate using Python 3.

What can this A.I. colleague accomplish for you?

- It can answer basic questions fed to it.
- It can play music and videos on Youtube.

  Videos have remained as a main source of entertainment, one of the most prioritized tasks of virtual assistants. They are equally important for entertainment as well as educational purposes as most teaching and research activities in present times are done through Youtube. This helps in making the learning process more practical and out of the four walls of the classroom. Jarvis implements the feature through pywhatkit module. This scraps the searched YouTube query.
- It can do Wikipedia looks for you.
- It is equipped for opening sites like Google (listens to queries and searches them on Google), Youtube, and so forth, on

Chrome.

Making queries is an essential part of one's life, and nothing changes even for a developer working on Windows. We have addressed the essential part of a netizen's life by enabling our voice assistant to search the web. Here we have used webbrowser module for extracting the result from the web as well as displaying it to the user. Jarvis supports a plethora of search engines like Google, Bing and Yahoo and displays the result by scraping the searched queries.

1.2 Problem statement

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there's no such virtual assistant available for the paradise of Developers i.e. Windows platform.

PURPOSE: This Software aims at developing a personal assistant for Windows-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

PRODUCT GOALS AND OBJECTIVES: Currently, the project aims to provide the Windows Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a 6 general server administrator.

PRODUCT DESCRIPTION: As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in various search engines like Google, Bing or Yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks. The user statements/commands are analysed with the help of machine learning to give an optimal solution.

1.3 Objectives

- Allow the A.I. to speak a given piece of text.
- Make a function to open websites which asked to be opened
- Make a function which opens the latest uploaded video on Youtube with the title said by the user
- Make a function to search for the query on Google for something that the A.I. doesn’t understand.
- Feed some questions and answers to make A.I. talk like a human being.

1.4 Methodology

- Jj

1.5 Organization

- Ll
- ;ll

2. LITERATURE SURVEY

In this Project Jarvis is Digital Life Assistant which uses mainly human communication means such Twitter, instant message and voice to create two way connections between human and his apartment, controlling lights and appliances, assist in cooking, notify him of breaking news, Facebook's Notifications and many more. In our project we mainly use voice as communication means so the Jarvis is basically the Speech recognition application. The concept of speech technology really encompasses two technologies: Synthesizer and recognizer. A speech synthesizer takes as input and produces an audio stream as output. A speech recognizer on the other hand does opposite. It takes an audio stream as input and thus turns it into text transcription. The voice is a signal of infinite information. A direct analysisand synthesizing the complex voice signal is due to too much information contained in the signal. Therefore the digital signal processes such as Feature Extraction and Feature Matching are introduced to represent the voice signal. In this project we directly use speech engine which use Feature extraction technique as Mel scaled frequency cepstral. The melscaled frequency cepstral coefficients (MFCCs) derived from Fourier transform and filter bank analysis are perhaps the most widely used front- ends in state-of-the-art speech recognition systems. Our aim to create more and more functionalities which can help human to assist in their daily life and also reduces their efforts. In our test we check all this functionality is working properly. We test this on 2 speakers(1 Female and 1 Male) for accuracy purpose.

-ShrutiKa Khobragade

Department of Computer Vishwakarma Institute of InformationTechnology Pune,INDIA

3. SYSTEM DEVELOPEMENT

Tools and technologies used

Language used: Python 3

Modules used :

- pyttsx3 (imports voices and has functions related to speaking)
- datetime (#not important .)
- speech_recognition (to convert speech to text)
- wikipedia (to access Wikipedia information)
- webbrowser (to manipulate web browsing operations)
- os (for just os.clear())
- pywhatkit (for playing songs on youtube)

Functions created :

- speak() (speaks text given as argument)
- wishMe (Wishes according to the day hour)
- takeCommand() (to convert speech to text and give it as input)

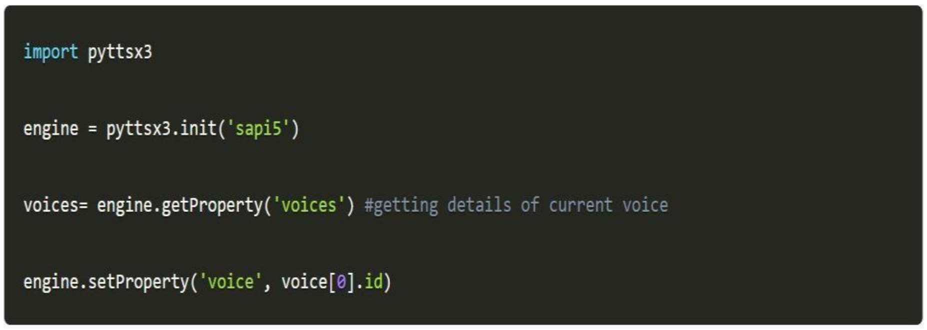
- indices(), openwebsite(), printspeak() are functionss just to shorten the code therefore, not important.

We implemented the code on Visual studio code.

Working:

What is pyttsx3?

A python library which will assist us with changing content over to discourse. So, it is a book to-discourse library.



```
import pyttsx3

engine = pyttsx3.init('sapi5')

voices= engine.getProperty('voices') #getting details of current voice

engine.setProperty('voice', voice[0].id)
```

Fig. 3.1

What is sapi5?

Discourse API created by Microsoft.

Aides in blend and acknowledgment of voice.

What Is VoiceId?

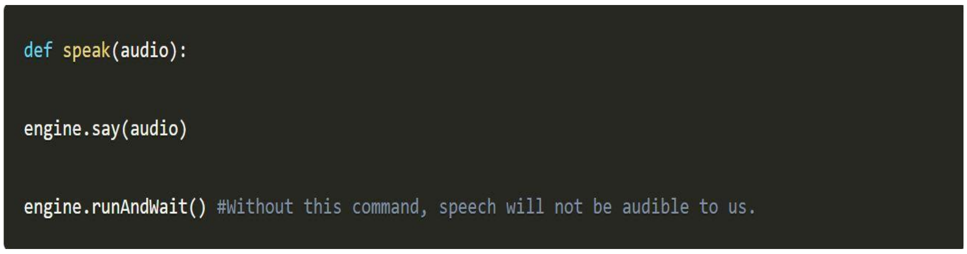
Voice id encourages us to choose various voices.

voice[0].id = Male voice

voice[1].id = Female voice

1. speak()

The most importantly thing for an A.I. right hand is that it should have the option to talk. To make our J.A.R.V.I.S. talk, we will make a capacity called talk. This capacity will accept sound as a contention,and afterward, it will articulate it.

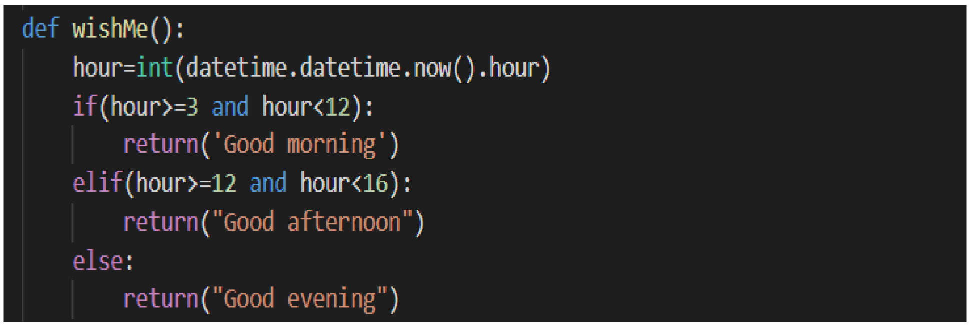


```
def speak(audio):

engine.say(audio)

engine.runAndWait() #Without this command, speech will not be audible to us.
```

Fig. 3.2

2. wishMe()

The most importantly thing for an A.I. right hand is that it should have the option to talk. To make our J.A.R.V.I.S. talk, we will make a capacity called talk. This capacity will accept sound as a contention,and afterward, it will articulate it.

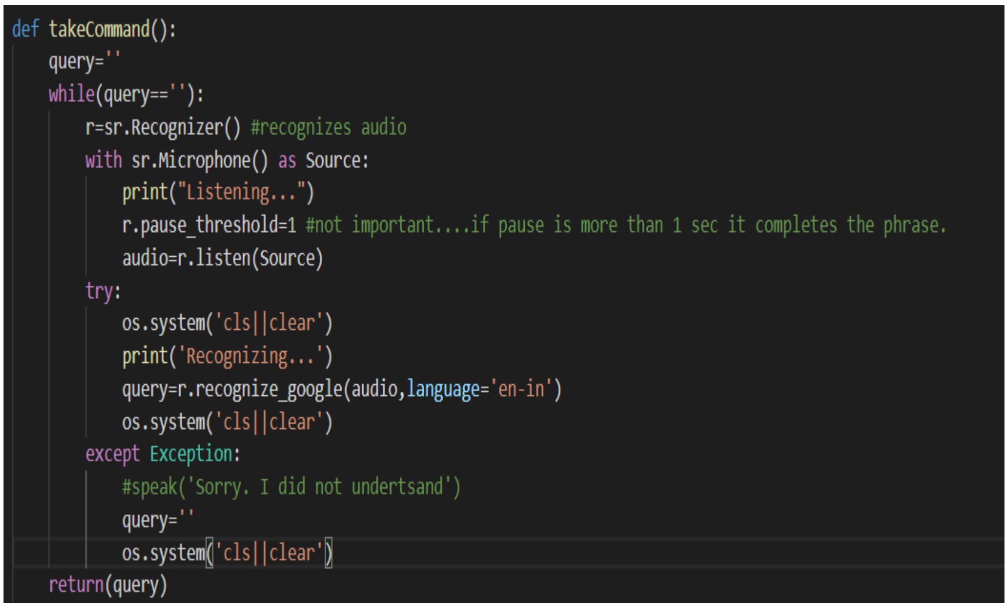


```
def wishMe():
    hour=int(datetime.datetime.now().hour)
    if(hour>=3 and hour<12):
        return('Good morning')
    elif(hour>=12 and hour<16):
        return("Good afternoon")
    else:
        return("Good evening")
```

Fig. 3.3

3. takeCommand()

How about we begin coding the takeCommand() work :



```
def takeCommand():
    query=''
    while(query==''):
        r=sr.Recognizer() #recognizes audio
        with sr.Microphone() as Source:
            print("Listening...")
            r.pause_threshold=1 #not important....if pause is more than 1 sec it completes the phrase.
            audio=r.listen(Source)
        try:
            os.system('cls||clear')
            print('Recognizing...')
            query=r.recognize_google(audio,language='en-in')
            os.system('cls||clear')
        except Exception:
            #speak('Sorry. I did not undertsand')
            query=''
            os.system('cls||clear')
    return(query)
```

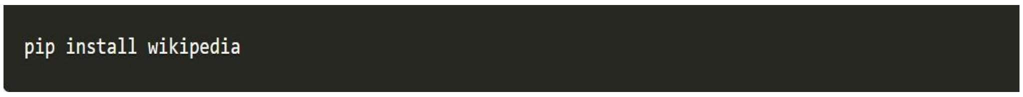
Fig. 3.4

We have effectively made our takeCommand() work.

Characterizing Task 1: To look through something on Wikipedia

To do Wikipedia look, we have to introduce and bring the Wikipedia module into our program. Type the beneath order to introduce the Wikipedia module

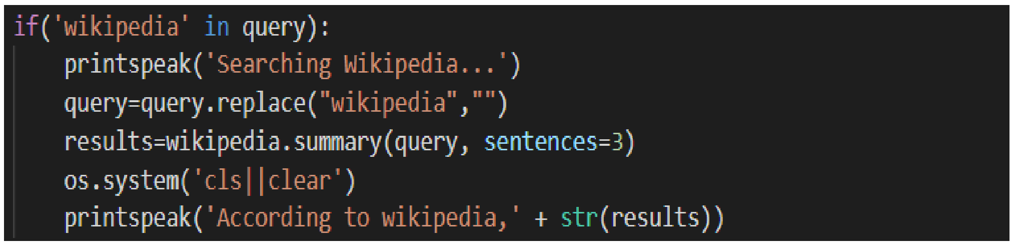
pip introduce wikipedia



```
pip install wikipedia
```

Fig. 3.5

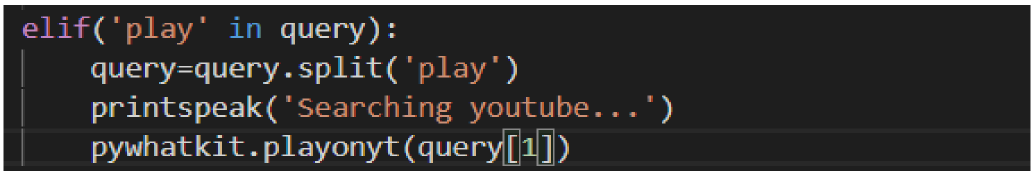
On the off chance that 'wikipedia' in inquiry: #if wikipedia found in the question then this square will be executed



```
if('wikipedia' in query):
    printspeak('Searching Wikipedia...')
    query=query.replace("wikipedia","")
    results=wikipedia.summary(query, sentences=3)
    os.system('cls||clear')
    printspeak('According to wikipedia,' + str(results))
```

Fig. 3.6

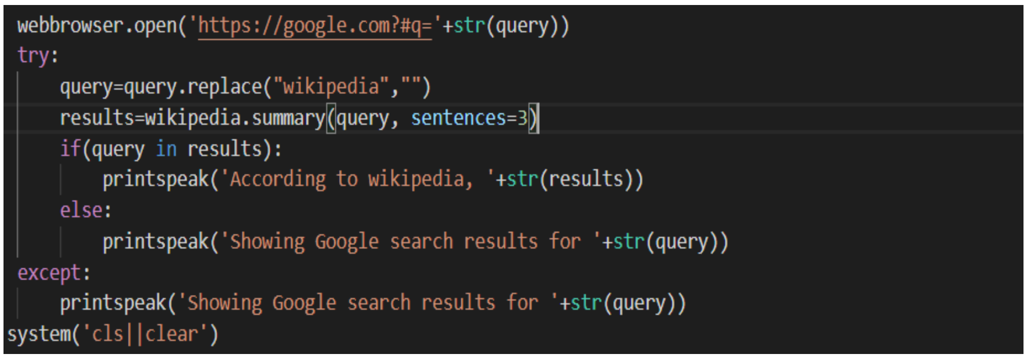
Characterizing Task 2: To play any video on youtube



```
elif('play' in query):
    query=query.split('play')
    printspeak('Searching youtube...')
    pywhatkit.playonyt(query[1])
```

Fig. 3.7

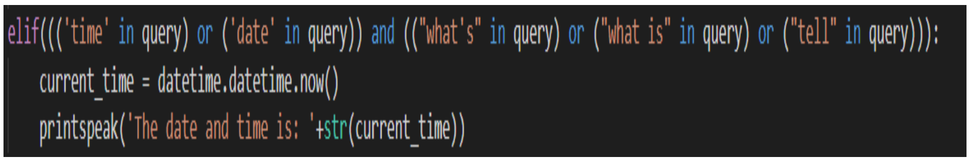
Characterizing Task 3: If any query is not understood by the assistant it look about it on google.



```
webbrowser.open('https://google.com?#q='+str(query))
try:
    query=query.replace("wikipedia","")
    results=wikipedia.summary(query, sentences=3)
    if(query in results):
        printspeak('According to wikipedia, '+str(results))
    else:
        printspeak('Showing Google search results for '+str(query))
except:
    printspeak('Showing Google search results for '+str(query))
system('cls||clear')
```

Fig. 3.8

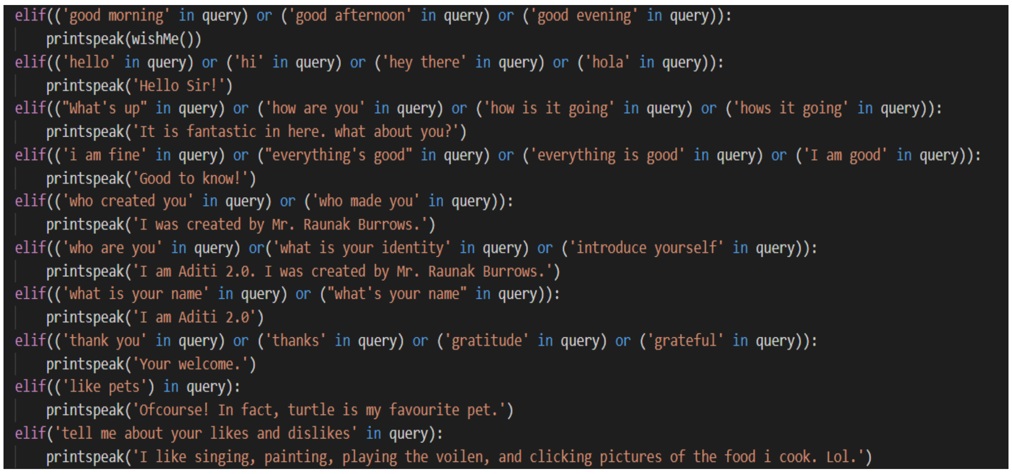
Characterizing Task 4 : To know the current time



```
elif((('time' in query) or ('date' in query)) and (("what's" in query) or ("what is" in query) or ("tell" in query))):
    current_time = datetime.datetime.now()
    printspeak('The date and time is: '+str(current_time))
```

Fig. 3.9

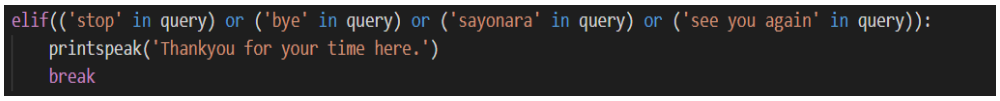
Characterizing Task 5: To ask basic questions



```
elif(('good morning' in query) or ('good afternoon' in query) or ('good evening' in query)):
    printspeak(wishMe())
elif(('hello' in query) or ('hi' in query) or ('hey there' in query) or ('hola' in query)):
    printspeak('Hello Sir!')
elif(("what's up" in query) or ('how are you' in query) or ('how is it going' in query) or ('hows it going' in query)):
    printspeak('It is fantastic in here. what about you?')
elif(('i am fine' in query) or ("everything's good" in query) or ('everything is good' in query) or ('I am good' in query)):
    printspeak('Good to know!')
elif(('who created you' in query) or ('who made you' in query)):
    printspeak('I was created by Mr. Raunak Burrows.')
elif(('who are you' in query) or('what is your identity' in query) or ('introduce yourself' in query)):
    printspeak('I am Aditi 2.0. I was created by Mr. Raunak Burrows.')
elif(('what is your name' in query) or ("what's your name" in query)):
    printspeak('I am Aditi 2.0')
elif(('thank you' in query) or ('thanks' in query) or ('gratitude' in query) or ('grateful' in query)):
    printspeak('Your welcome.')
elif(('like pets') in query):
    printspeak('Ofcourse! In fact, turtle is my favourite pet.')
elif('tell me about your likes and dislikes' in query):
    printspeak('I like singing, painting, playing the voilen, and clicking pictures of the food i cook. Lol.')
```

Fig. 3.10

Characterizing Task 6: To exit



```
elif(('stop' in query) or ('bye' in query) or ('sayonara' in query) or ('see you again' in query)):
    printspeak('Thankyou for your time here.')
    break
```

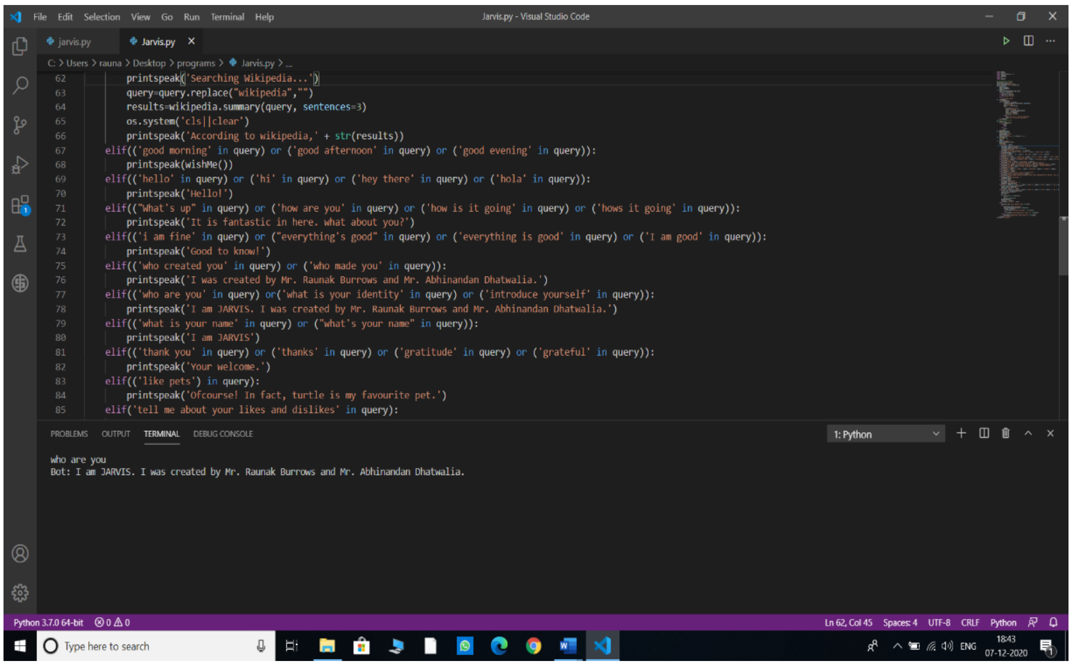
Fig. 3.11

Complete code:

```
import pyttsx3

import datetime

import speech_recognition as sr

import wikipedia

import webbrowser

import os

import pywhatkit


#initializing voice settings

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)

    if(hour>=3 and hour<12):

        return('Good morning')

    elif(hour>=12 and hour<16):
        return("Good afternoon")
    else:
        return("Good evening")
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source2:
            r.adjust_for_ambient_noise(source2, duration=0.2)
            print('Listening...')
            audio2 = r.listen(source2)
            try:
                os.system('cls||clear')
                print('Recognizing...')
                MyText = r.recognize_google(audio2)
                MyText = MyText.lower()
                print('You: '+MyText)
                os.system('cls||clear')
            except:
                print('Bot: Sorry, I could not undertsand.')
                speak('Sorry, I could not undertsand.')
    return(MyText)
#functions to shorten code
def indices(a,c):
    for i in range(len(a)):
        if(a[i]==c):
            ans=i
            break
        else:
            ans=-1
    return(ans)
def openwebsite(query):
    Query=query.split()
    x=indices(Query,'open')
    website=Query[x+1]
    webbrowser.open('https://'+str(website)+'.com')
def printspeak(s):
    print('Bot: '+s)
    speak(s)
if __name__=="__main__":
    query=takeCommand().lower()
    print(query)
    if('wikipedia' in query):
        printspeak('Searching Wikipedia...')
        query=query.replace("wikipedia","")
        results=wikipedia.summary(query, sentences=3)
        os.system('cls||clear')
        printspeak('According to wikipedia,' + str(results))
    elif(('good morning' in query) or ('good afternoon' in query) or ('good evening' in query)):
        printspeak(wishMe())
    elif(('hello' in query) or ('hi' in query) or ('hey there' in query) or ('hola' in query)):
        printspeak('Hello!')
    elif(("What's up" in query) or ('how are you' in query) or ('how is it going' in query) or ('hows it going' in query)):
        printspeak('It is fantastic in here. what about you?')
    elif(('i am fine' in query) or ("everything's good" in query) or ('everything is good' in query) or ('I am good' in query)):
        printspeak('Good to know!')
    elif(('who created you' in query) or ('who made you' in query)):
        printspeak('I was created by Mr. Amitesh Srivastava.')
    elif(('who are you' in query) or('what is your identity' in query) or ('introduce yourself' in query)):
        printspeak('I am JARVIS. I was created by Mr. Amitesh Srivastava.')
    elif(('what is your name' in query) or ("what's your name" in query)):
        printspeak('I am JARVIS')
    elif(('thank you' in query) or ('thanks' in query) or ('gratitude' in query) or ('grateful' in query)):
        printspeak('Your welcome.')
    elif(('like pets') in query):
        printspeak('Ofcourse! In fact, turtle is my favourite pet.')
    elif('tell me about your likes and dislikes' in query):
        printspeak('I like singing, painting, playing the voilen, and clicking pictures of the
food i cook. Lol. and being a machine, there is nothing I dislike.')
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query)
or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query)
or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
        try:
            query=query.replace("wikipedia","")
            results=wikipedia.summary(query, sentences=3)
            if(query in results):
                printspeak('According to wikipedia, '+str(results))
            else:
                printspeak('Showing Google search results for '+str(query))
        except:
            printspeak('Showing Google search results for '+str(query))
    os.system('cls||clear')
os.system('cls||clear')
```

4. PERFORMANCE ANALYSIS

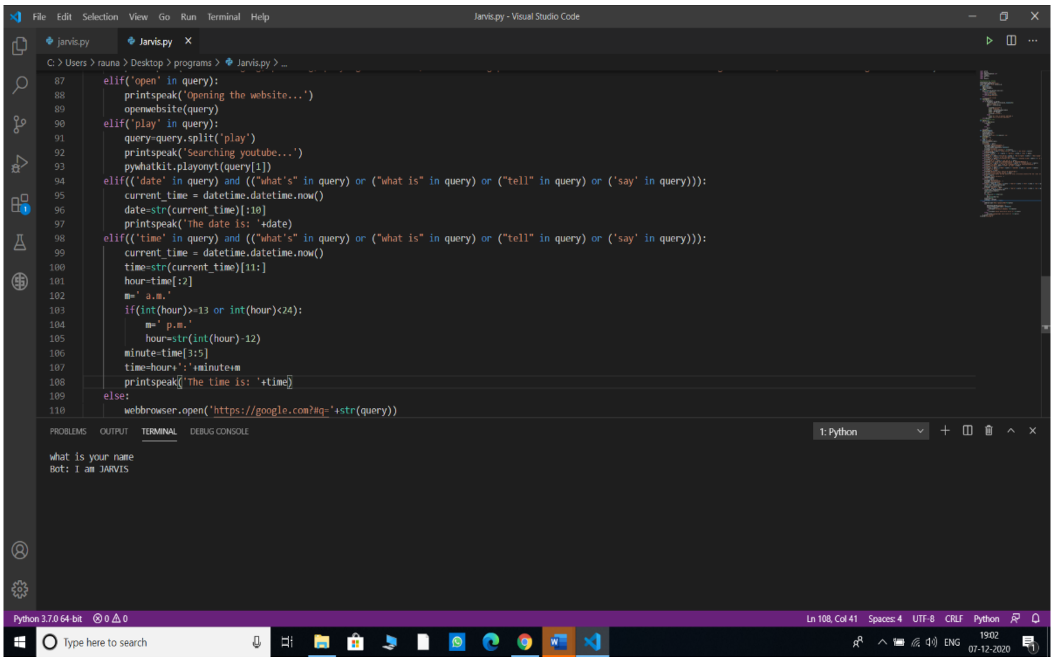
4.1 It can answer basic questions fed to it.



```
            printspeak('Searching Wikipedia...')
            query=query.replace("wikipedia","")
            results=wikipedia.summary(query, sentences=3)
            os.system('cls||clear')
            printspeak('According to wikipedia,' + str(results))
        elif(('good morning' in query) or ('good afternoon' in query) or ('good evening' in query)):
            printspeak(wishMe())
        elif(('hello' in query) or ('hi' in query) or ('hey there' in query) or ('hola' in query)):
            printspeak('Hello!')
        elif(("What's up" in query) or ('how are you' in query) or ('how is it going' in query) or ('hows it going' in query)):
            printspeak('It is fantastic in here. what about you?')
        elif(('i am fine' in query) or ("everything's good" in query) or ('everything is good' in query) or ('I am good' in query)):
            printspeak('Good to know!')
        elif(('who created you' in query) or ('who made you' in query)):
            printspeak('I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
        elif(('who are you' in query) or('what is your identity' in query) or ('introduce yourself' in query)):
            printspeak('I am JARVIS. I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
        elif(('what is your name' in query) or ("what's your name" in query)):
            printspeak('I am JARVIS')
        elif(('thank you' in query) or ('thanks' in query) or ('gratitude' in query) or ('grateful' in query)):
            printspeak('Your welcome.')
        elif(('like pets') in query):
            printspeak('Ofcourse! In fact, turtle is my favourite pet.')
        elif('tell me about your likes and dislikes' in query):
```

```
who are you
Bot: I am JARVIS. I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.
```

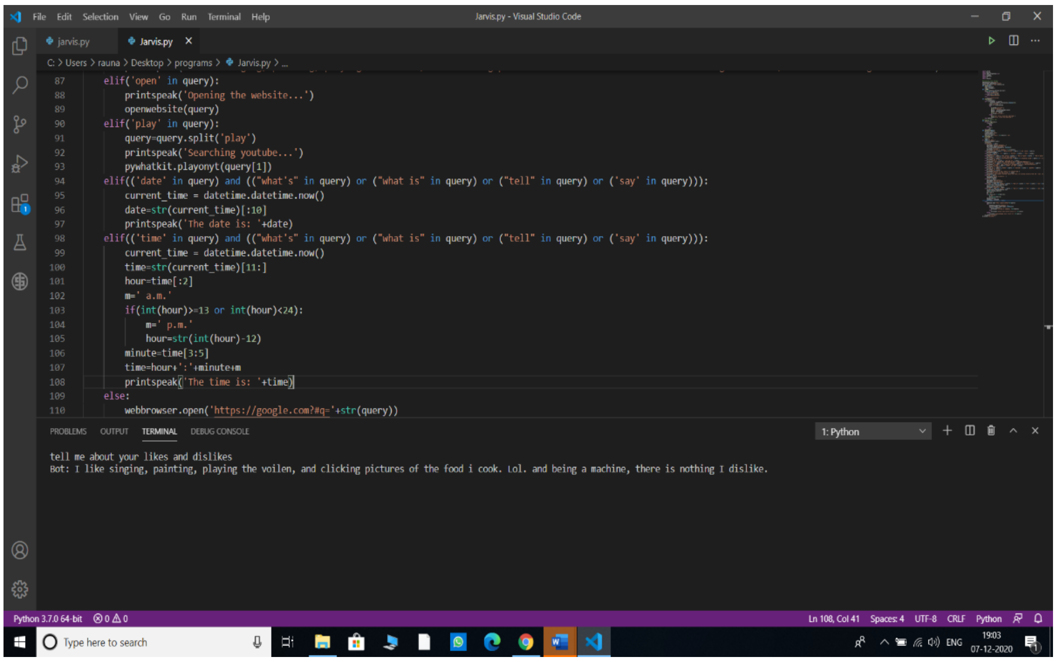
Fig. 4.1



```
        elif('open' in query):
            printspeak('Opening the website...')
            openwebsite(query)
        elif('play' in query):
            query=query.split('play')
            printspeak('Searching youtube...')
            pywhatkit.playonyt(query[1])
        elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
            current_time = datetime.datetime.now()
            date=str(current_time)[:10]
            printspeak('The date is: '+date)
        elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
            current_time = datetime.datetime.now()
            time=str(current_time)[11:]
            hour=time[:2]
            m=' a.m.'
            if(int(hour)>=13 or int(hour)<24):
                m=' p.m.'
                hour=str(int(hour)-12)
            minute=time[3:5]
            time=hour+':'+minute+m
            printspeak('The time is: '+time)
        else:
            webbrowser.open('https://google.com?#q='+str(query))
```

```
what is your name
Bot: I am JARVIS
```

Fig. 4.2



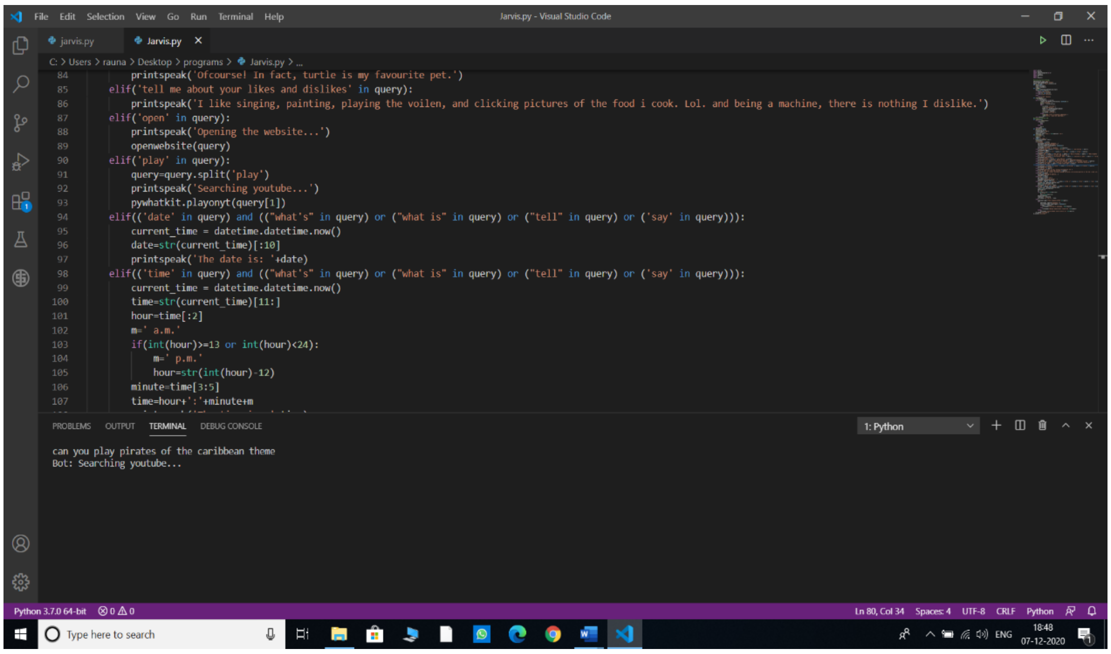
```python
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
```

```
tell me about your likes and dislikes
Bot: I like singing, painting, playing the voilen, and clicking pictures of the food i cook. Lol. and being a machine, there is nothing I dislike.
```

Fig. 4.3

As we can see, it responded with an answer that makes sense!

4.2 It can play music and videos on Youtube.



```
        printspeak('Ofcourse! In fact, turtle is my favourite pet.')
    elif('tell me about your likes and dislikes' in query):
        printspeak('I like singing, painting, playing the voilen, and clicking pictures of the food i cook. Lol. and being a machine, there is nothing I dislike.')
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
```

```
can you play pirates of the caribbean theme
Bot: Searching youtube...
```

Fig. 4.4

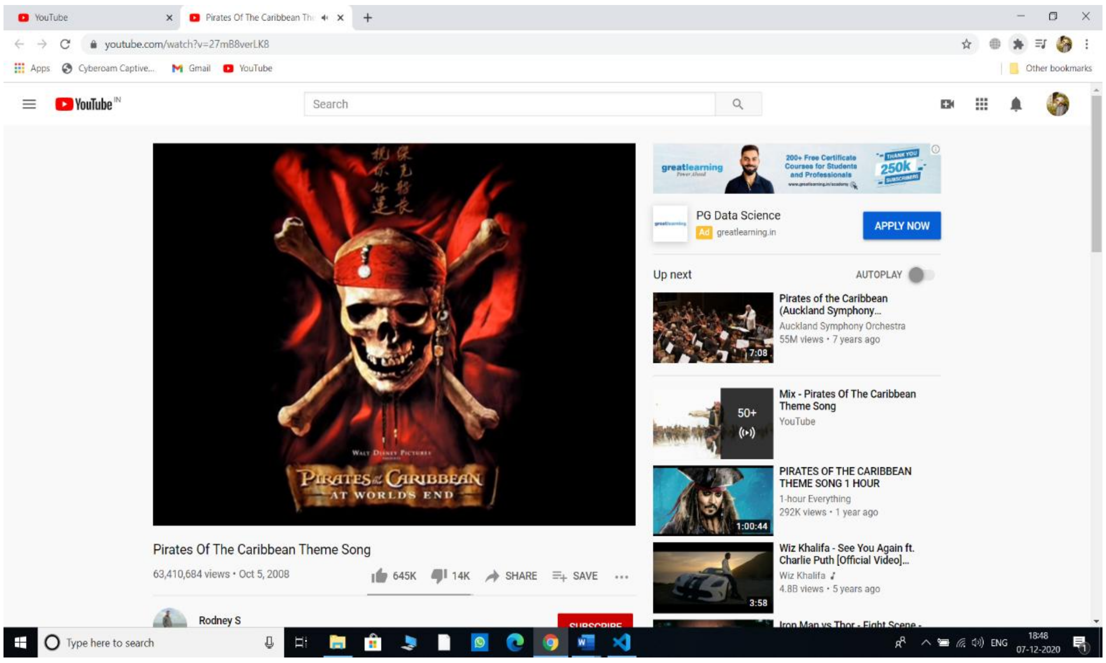
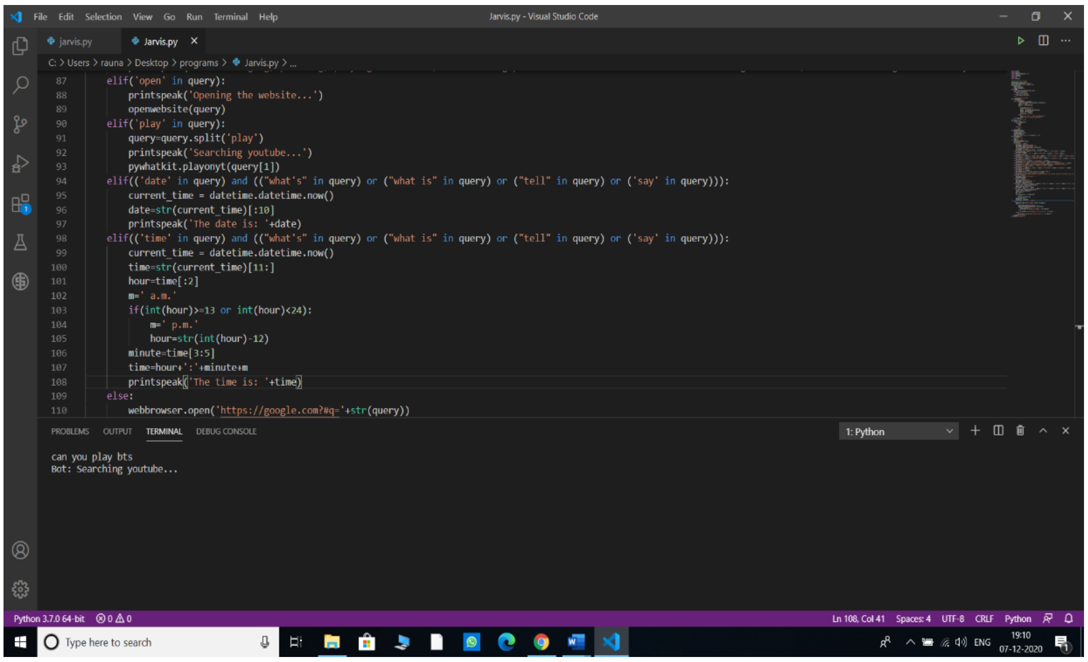

youtube.com/watch?v=27mB8verLK8
Pirates Of The Caribbean Theme Song
63,410,684 views • Oct 5, 2008
Up next
Pirates of the Caribbean (Auckland Symphony...
Mix - Pirates Of The Caribbean Theme Song
PIRATES OF THE CARIBBEAN THEME SONG 1 HOUR
Wiz Khalifa - See You Again ft. Charlie Puth [Official Video]...

Fig. 4.5



```
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
```

```
can you play bts
Bot: Searching youtube...
```

Fig. 4.6

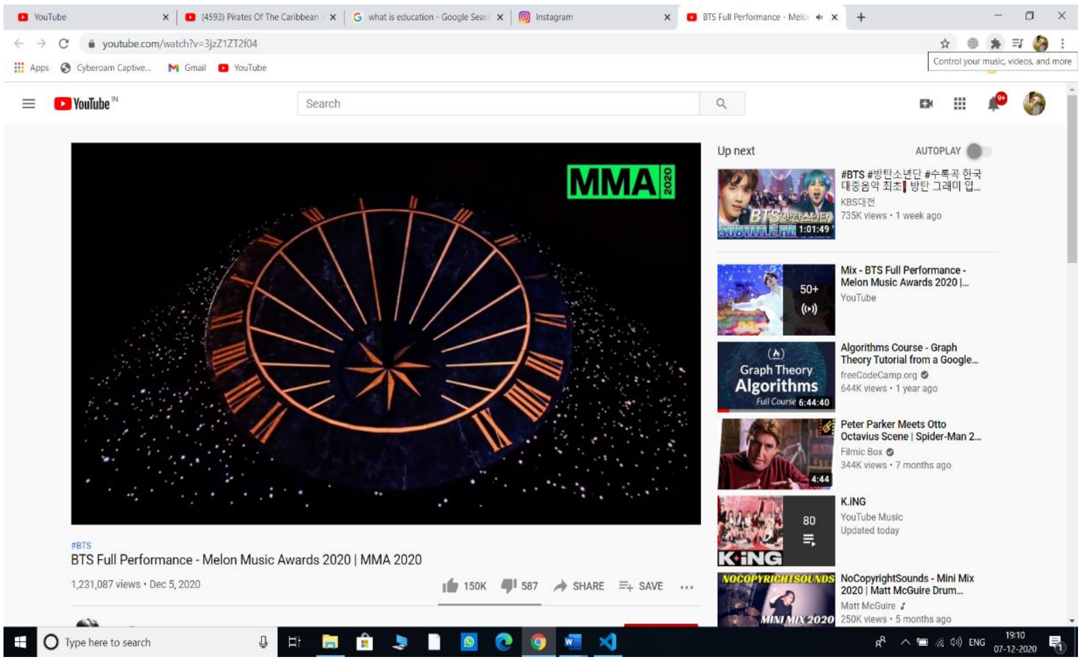
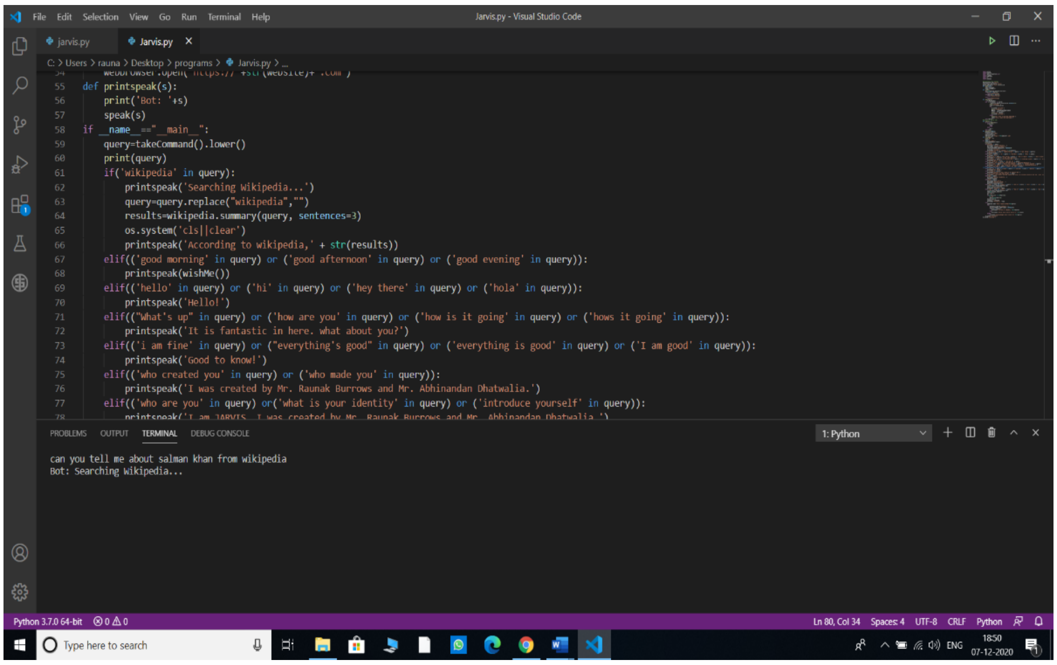

youtube.com/watch?v=3jzZ1ZT2f04
BTS Full Performance - Melon Music Awards 2020 | MMA 2020
1,231,087 views • Dec 5, 2020
Up next

Fig. 4.7

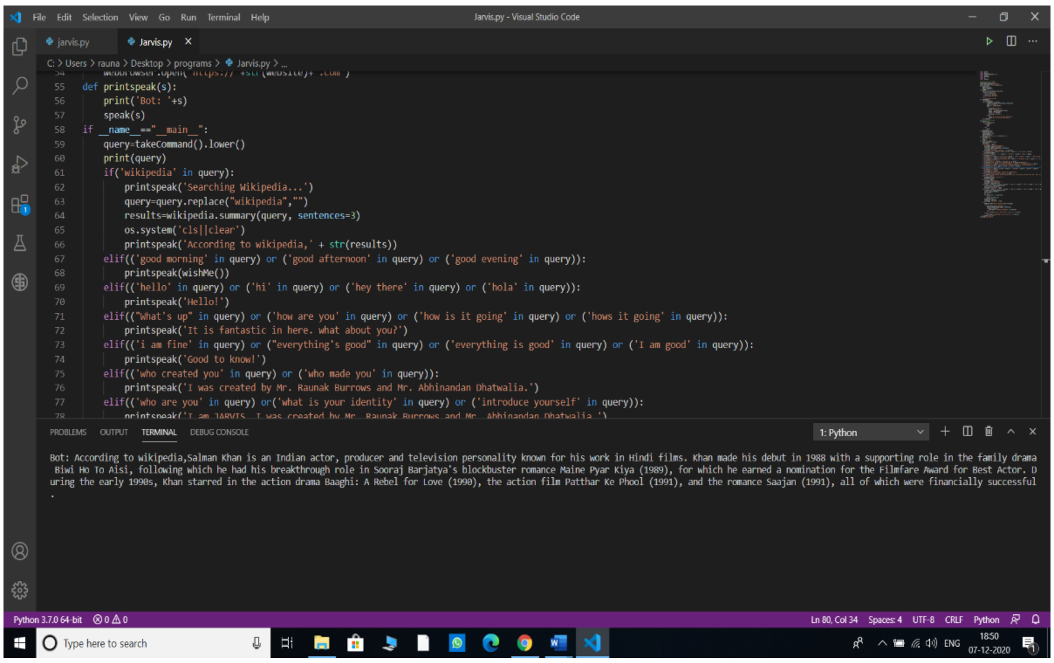
4.3 It can do Wikipedia looks for you.



```
    webbrowser.open('https://'+str(website)+'.com')
def printspeak(s):
    print('Bot: '+s)
    speak(s)
if __name__=="__main__":
    query=takeCommand().lower()
    print(query)
    if('wikipedia' in query):
        printspeak('Searching Wikipedia...')
        query=query.replace("wikipedia","")
        results=wikipedia.summary(query, sentences=3)
        os.system('cls||clear')
        printspeak('According to wikipedia,' + str(results))
    elif(('good morning' in query) or ('good afternoon' in query) or ('good evening' in query)):
        printspeak(wishMe())
    elif(('hello' in query) or ('hi' in query) or ('hey there' in query) or ('hola' in query)):
        printspeak('Hello!')
    elif(("What's up" in query) or ('how are you' in query) or ('how is it going' in query) or ('hows it going' in query)):
        printspeak('It is fantastic in here. what about you?')
    elif(('i am fine' in query) or ("everything's good" in query) or ('everything is good' in query) or ('I am good' in query)):
        printspeak('Good to know!')
    elif(('who created you' in query) or ('who made you' in query)):
        printspeak('I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
    elif(('who are you' in query) or('what is your identity' in query) or ('introduce yourself' in query)):
        printspeak('I am JARVIS. I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
```

```
can you tell me about salman khan from wikipedia
Bot: Searching Wikipedia...
```

Fig. 4.8

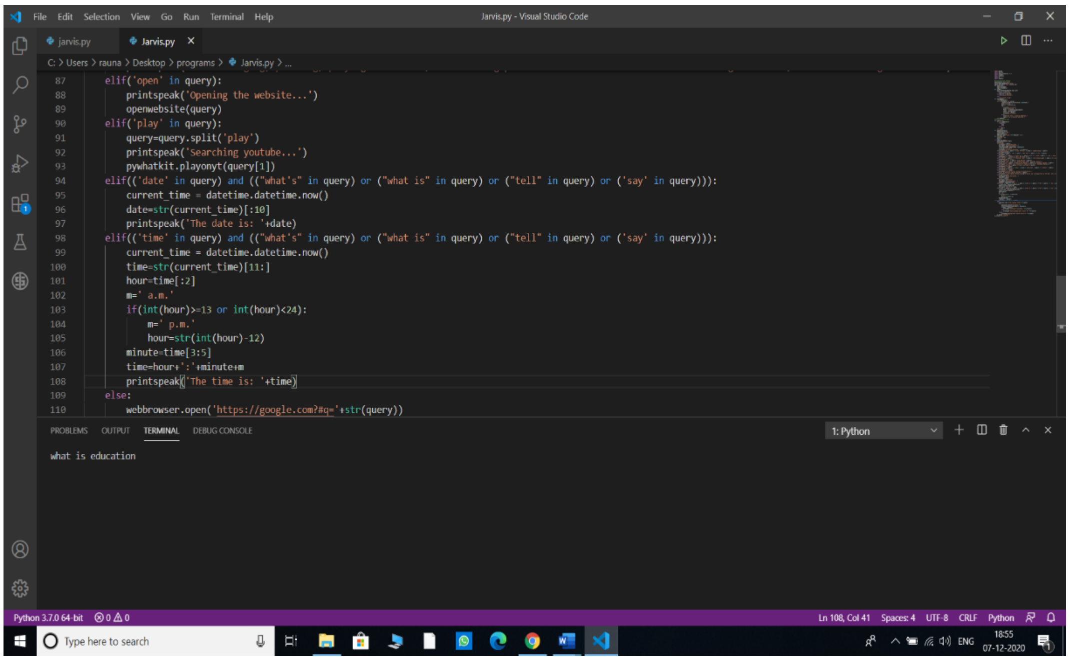


```
    webbrowser.open('https://'+str(website)+'.com')
def printspeak(s):
    print('Bot: '+s)
    speak(s)
if __name__=="__main__":
    query=takeCommand().lower()
    print(query)
    if('wikipedia' in query):
        printspeak('Searching Wikipedia...')
        query=query.replace("wikipedia","")
        results=wikipedia.summary(query, sentences=3)
        os.system('cls||clear')
        printspeak('According to wikipedia,' + str(results))
    elif(('good morning' in query) or ('good afternoon' in query) or ('good evening' in query)):
        printspeak(wishMe())
    elif(('hello' in query) or ('hi' in query) or ('hey there' in query) or ('hola' in query)):
        printspeak('Hello!')
    elif(("What's up" in query) or ('how are you' in query) or ('how is it going' in query) or ('hows it going' in query)):
        printspeak('It is fantastic in here. what about you?')
    elif(('i am fine' in query) or ("everything's good" in query) or ('everything is good' in query) or ('I am good' in query)):
        printspeak('Good to know!')
    elif(('who created you' in query) or ('who made you' in query)):
        printspeak('I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
    elif(('who are you' in query) or('what is your identity' in query) or ('introduce yourself' in query)):
        printspeak('I am JARVIS. I was created by Mr. Raunak Burrows and Mr. Abhinandan Dhatwalia.')
```

```
Bot: According to wikipedia,Salman Khan is an Indian actor, producer and television personality known for his work in Hindi films. Khan made his debut in 1988 with a supporting role in the family drama
 Biwi Ho To Aisi, following which he had his breakthrough role in Sooraj Barjatya's blockbuster romance Maine Pyar Kiya (1989), for which he earned a nomination for the Filmfare Award for Best Actor. D
uring the early 1990s, Khan starred in the action drama Baaghi: A Rebel for Love (1990), the action film Patthar Ke Phool (1991), and the romance Saajan (1991), all of which were financially successful
.
```

Fig. 4.9

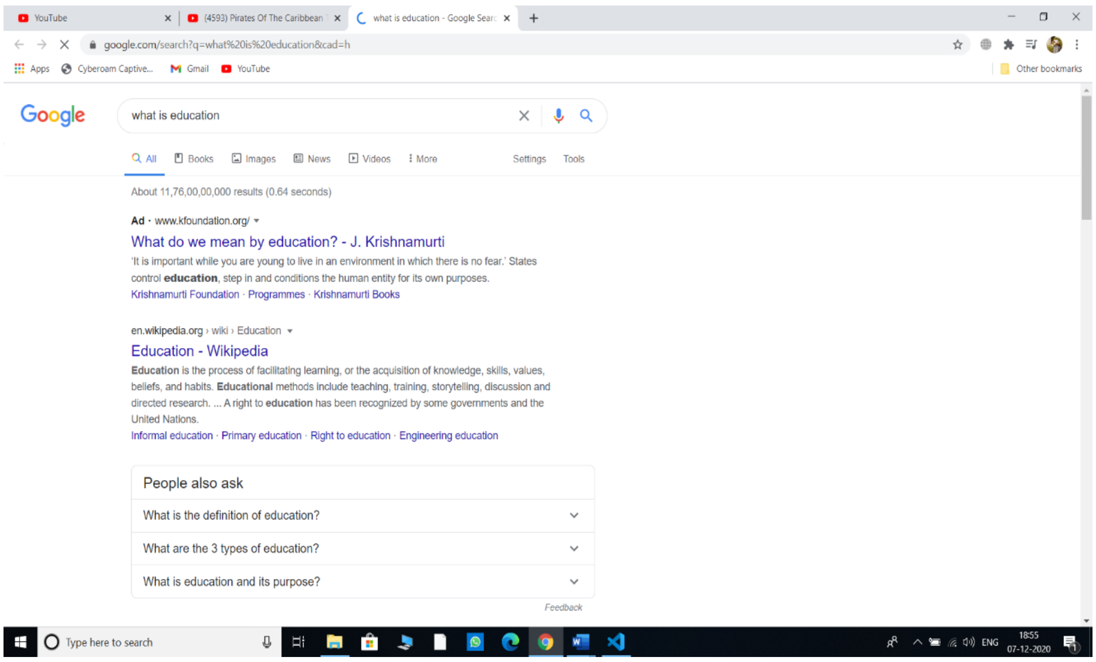
4.4 It is equipped for opening sites like Google, Youtube, and so forth, on Chrome.



```
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
```

```
what is education
```

Fig. 4.10



google.com/search?q=what%20is%20education&cad=h

what is education

About 11,76,00,00,000 results (0.64 seconds)

Ad · www.kfoundation.org/

What do we mean by education? - J. Krishnamurti

'It is important while you are young to live in an environment in which there is no fear.' States control **education**, step in and conditions the human entity for its own purposes.

Krishnamurti Foundation · Programmes · Krishnamurti Books

en.wikipedia.org › wiki › Education

Education - Wikipedia

Education is the process of facilitating learning, or the acquisition of knowledge, skills, values, beliefs, and habits. **Educational** methods include teaching, training, storytelling, discussion and directed research. ... A right to **education** has been recognized by some governments and the United Nations.

Informal education · Primary education · Right to education · Engineering education

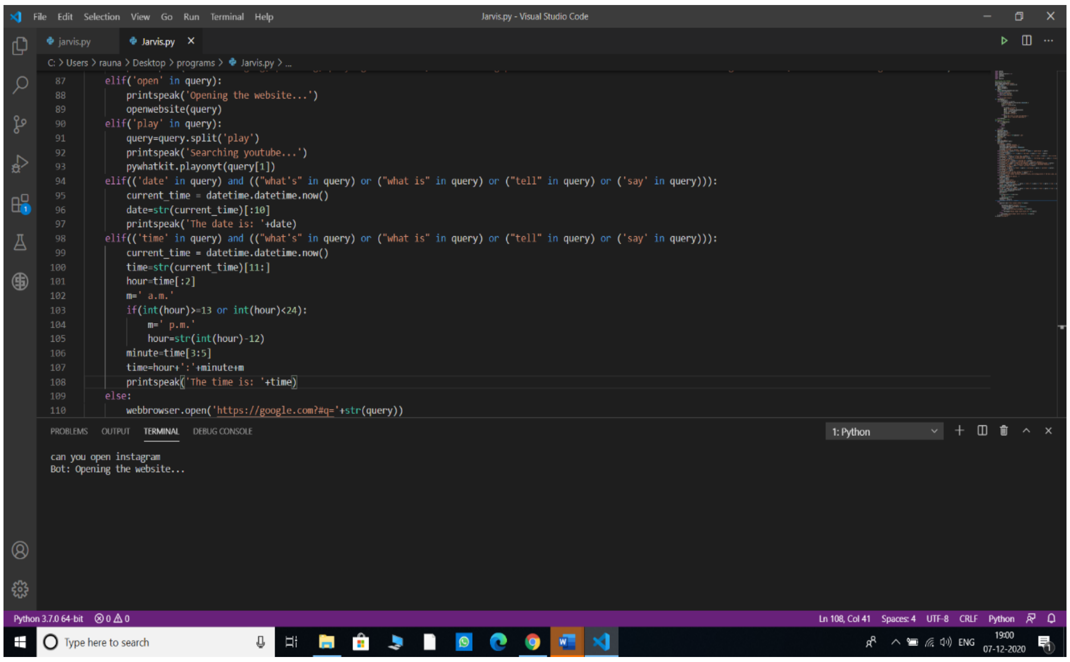
People also ask

What is the definition of education?

What are the 3 types of education?

What is education and its purpose?

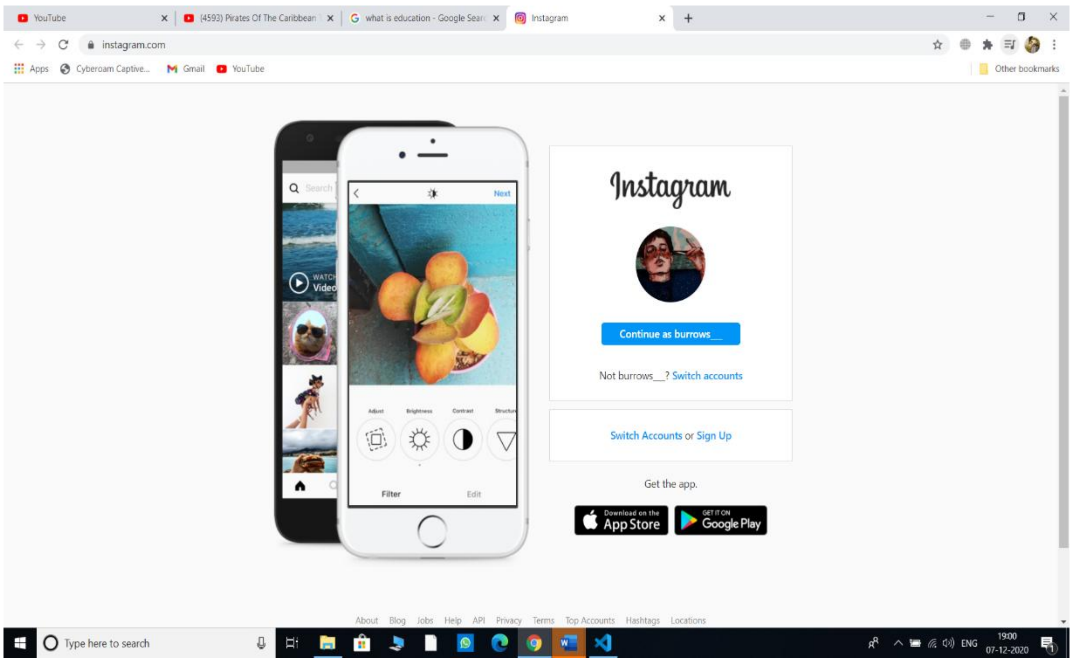
Fig. 4.11



```
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
```

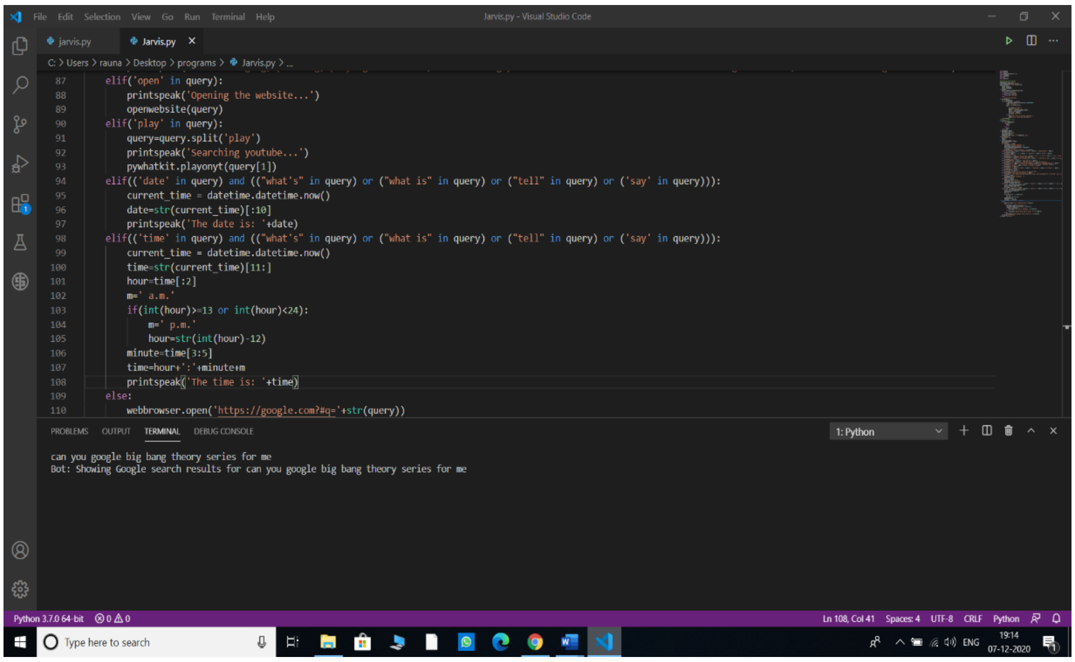
```
can you open instagram
Bot: Opening the website...
```

Fig. 4.12



instagram.com
Instagram
Continue as burrows__
Not burrows__? Switch accounts
Switch Accounts or Sign Up
Get the app.
Download on the App Store
GET IT ON Google Play

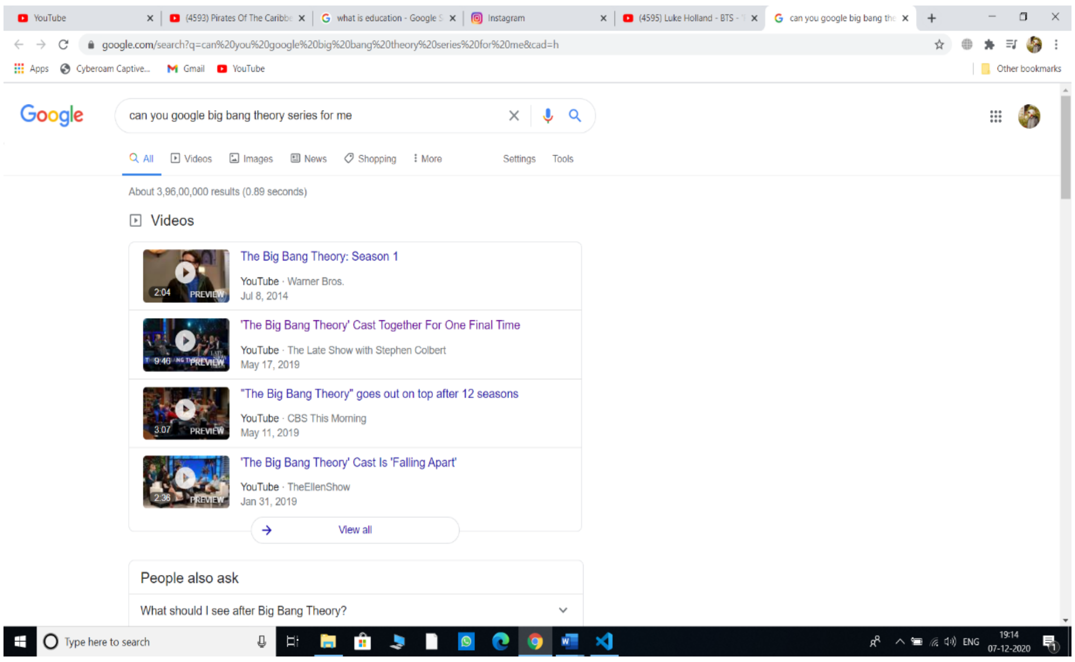
Fig. 4.13


Jarvis.py - Visual Studio Code
jarvis.py
Jarvis.py
C: > Users > rauna > Desktop > programs > Jarvis.py > ...

```python
    elif('open' in query):
        printspeak('Opening the website...')
        openwebsite(query)
    elif('play' in query):
        query=query.split('play')
        printspeak('Searching youtube...')
        pywhatkit.playonyt(query[1])
    elif(('date' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        date=str(current_time)[:10]
        printspeak('The date is: '+date)
    elif(('time' in query) and (("what's" in query) or ("what is" in query) or ("tell" in query) or ('say' in query))):
        current_time = datetime.datetime.now()
        time=str(current_time)[11:]
        hour=time[:2]
        m=' a.m.'
        if(int(hour)>=13 or int(hour)<24):
            m=' p.m.'
            hour=str(int(hour)-12)
        minute=time[3:5]
        time=hour+':'+minute+m
        printspeak('The time is: '+time)
    else:
        webbrowser.open('https://google.com?#q='+str(query))
```

```
can you google big bang theory series for me
Bot: Showing Google search results for can you google big bang theory series for me
```

Fig. 4.14


YouTube
(4593) Pirates Of The Caribb
what is education - Google S
Instagram
(4595) Luke Holland - BTS
can you google big bang the
google.com/search?q=can%20you%20google%20big%20bang%20theory%20series%20for%20me&cad=h
Apps
Cyberoam Captive...
Gmail
YouTube
Other bookmarks
can you google big bang theory series for me
All
Videos
Images
News
Shopping
More
Settings
Tools

About 3,96,00,000 results (0.89 seconds)

Videos

The Big Bang Theory: Season 1
YouTube · Warner Bros.
Jul 8, 2014

'The Big Bang Theory' Cast Together For One Final Time
YouTube · The Late Show with Stephen Colbert
May 17, 2019

"The Big Bang Theory" goes out on top after 12 seasons
YouTube · CBS This Morning
May 11, 2019

'The Big Bang Theory' Cast Is 'Falling Apart'
YouTube · TheEllenShow
Jan 31, 2019

View all

People also ask

What should I see after Big Bang Theory?

Fig. 4.15

5. CONCLUSIONS

5.1 Conclusions

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. The future plans include integrating Jarvis with mobile using React Native to provide a synchronised experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

5.2 Future Scope

We plan to Integrate Jarvis with mobile using react native, to provide a synchronized experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

Functional Requirements:

- Proper Internet Connection
- Github Credentials
- Python 3.7
- pyttsx3 module for voice support (Text-to-Speech)

- Chromium-based browser, like Chrome, Edge
- Node JS with npm Non-Functional Requirements:

The non-functional requirements of the system include:

- The system ensures safety, security and usability, which are observable during operation (at run time).
- The system is adaptable to different situations.
- The project has good and compact UI using AngularJS with responsive interface.
- The project is light on resources.

5.3 Applications Contributions

- Allow the A.I. to speak a given piece of text.

REFERENCES: www.youtube.com

codewithharry.com

kaggle

towardsdatascience.com