**What can this A.I. assistant do for you ?**

The JARVIS can :

1. Open Google Chrome

15. Play Windows Media Player

2. Open Microsoft Edge

3. Open Mozilla Firefox

4. Open Notepad

5. Open Notepad++

6. Open Internet Explorer

7. Play VLC Media Player

8. Run Command Prompt

9. Perform Calculations

10. Open Paint

11. Open File Explorer

12. Open Camera

13. Visit Facebook & Instagram

14. Send Whatsapp Messages Directly

16. Play Video Songs

17. Open Github

18. Send Email

19. Send SMS

20. Open Youtube

21. Do Wikipedia Searches

22. Open Gmail

23. Open Linkedin Profile

24. Open Github Profile

25. Open Stack Overflow

26. Open Whatsapp Web

27. Shutdown PC

28. Restart PC

**PROJECT DESCRIPTION**

**i. Starting Jupyter Notebook**

We are going to use the Jupyter Notebook IDE in this project. Feel free to use any other IDE you are comfortable with. Started a new python file with named called JARVIS.

**ii. Defining Speak Function**

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our JARVIS talk, we will make a function called speak(). This function will take audio as an argument, and then, it will pronounce it.

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak() function we made. We are going to install a module called pyttsx3.

**What is pyttsx3 ?**

• A python library which will help us to convert text to speech. In short, it is a text-to-speech library.

• It works offline, and it is compatible with Python 2 as well the Python 3.

**Installation:**

Command : “ pip install pyttsx3 ”

In case you receive such errors:

• No module named win32com.client

• No module named win32

• No module named win32api

Then, install pypiwin32 by typing the below command in the terminal :

Command : “ pip install pypiwin32 ”.

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

**What is sapi5 ?**

• Speech API developed by Microsoft.

• Helps in synthesis and recognition of voice.

**What Is VoiceId ?**

• Voice id helps us to select different voices.

• voice[0].id = Male voice

• voice[1].id = Female voice

**Writing Our speak() Function :**

We made a function called speak() at the starting of this tutorial. Now, we will write our speak() function so that it can convert our text to speech.

**Creating Our main() function:**

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

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

**iii. Defining Wishme Function :**

Now, we are going to make a wishme() function, that will make our JARVIS wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime.

**iv. Defining Take command Function :**

The next most important thing for our A.I. assistant is that it should be able to take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will be able to return a string output by taking microphone input from the user.

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

Command : “ pip install speechRecognition ”

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

**Coding logic of JARVIS**

Now, we will develop logics for different commands such as Wikipedia searches, playing music, etc.

**v. Defining Task 1: To search something on Wikipedia**

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module :

Command : “ pip install Wikipedia ”

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

**Logic for executing tasks based on query :**

In the code we have used an if statement to check whether Wikipedia is in the search query of the user or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the help of speak function.

**vi. Defining Task 2: To open YouTube site in a web-browser**

To open any website, we need to import a module called webbrowser. It is an in-built module, and we do not need to install it with pip statement, we can directly import it into our program by writing an import statement

Here, we are using the elif loop to check whether the Youtube is in the query of the user or not. Let' suppose, the user gives command as “JARVIS open youtube.” So, open youtube will be in the user's query, and the elif condition will be true.

**vii. Defining Task 3: To open Google site in a web-browser**

We are opening Google in a web-browser by applying the same logic that we used to open youtube.

**viii. Defining Task 4: To play music**

To play music, we need to import a module called os. Import this module directly with an import statement.

In the code we first opened our music directory and then listed all the songs present in the directory with the help of the os module. With the help of **os.startfile**, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, JARVIS will play any random song from the song directory.

**ix. Defining Task 5: To know the current time**

In the code we are using datetime() function and storing the current or live of the system into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into the speech.

**x. Defining Task 6: To open the installed applications**

To open the installed application, we need the code path of the application.

Steps to get the code path of the application:

Step 1: Open the file location.

Step 2: Right-click on the application and click on properties.

Step 3: Copy the target from the target section.

After copying the target of the application, save the target into a variable. Here, I am directly opening the application without storing it’s address in any variable by using the os module to open the application.

As there are numerous of applications installed in my system, therefore I am explaining 2 of them :

1. elif "good text editor" in query or "high level text editor" in query or "external text editor" in query or "notepad plus plus" in query :

speak("Opening Notepad++")

os.system("notepad++")

2. elif "good media player" in query or "external player" in query or "vlc" in query or "good player" in query:

speak("Opening VLC Media Player")

os.system("vlc")

**xi. Defining Task 7: To send Email**

To send an email, we need to import a module called smtplib.

**What is smtplib ?**

• Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and to route emails between mail servers. An instance method called sendmail is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:

• The sender: Email address of the sender.

• The receiver:T Email of the receiver.

• The message: A string message which needs to be sent to one or more than one recipient.

**xii. Defining Send email function :**

Now, we will create a sendEmail() function, which will help us to send emails to one or more than one recipients.

**Note:** Do not forget to **'enable the less secure apps'** feature in your Gmail account. Otherwise, the sendEmail() function will not work properly.

Calling sendEmail() function inside the main() function to send Email.

We have used the try and except block to handle any possible error that can occur while sending emails.

**xiii. Defining Task 8: To send Whatsapp**

To send whatsapp message we need to import a package named **Client** from a module named **twilio.rest**.

* Your Account Sid and Auth Token from [twilio.com/console](file:///C:\Users\dell\Desktop\Notes\Python%20Project%20-%20JARVIS\twilio.com\console)
* Set the environment variables. See <http://twil.io/secure>

**xiv. Recapitulate**

1. First of all, we have created a wishme() function that gives the functionality of greeting according to the system time to our A.I.

2. After wishme() function, we have created a takeCommand() function, which helps our A.I to take command from the user. This function is also responsible for returning the user's query in a string format.

3. We developed the code logic for opening different websites like google, youtube, and stack overflow, facebook, linkedin, etc.

4. Developed code logic for opening installed application.

5. At last, we added functionality to send emails, whatsapp, text messages.

**Query - Is it an A.I. ?**

A lot of people will argue that the virtual assistant that we have created is not an A.I, but it is the output of the bunch of the statement. But, if we look at the very basic level, the sole purpose of A.I. is to develop machines that can perform human tasks with the same effectiveness or even more effectively than humans.

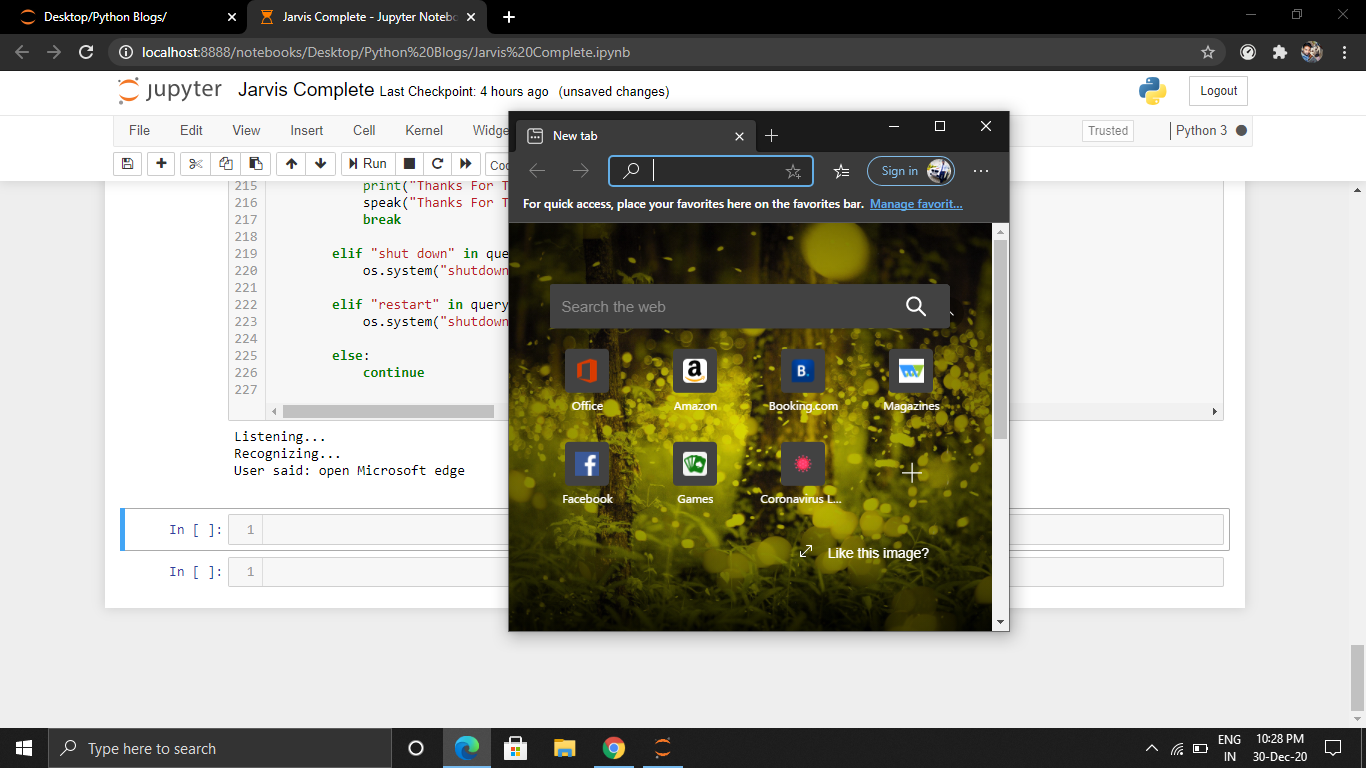
It is a fact that our virtual assistant is not a very good example of A.I., but it is an A.I. !

With this, we have successfully made your very first virtual assistant.

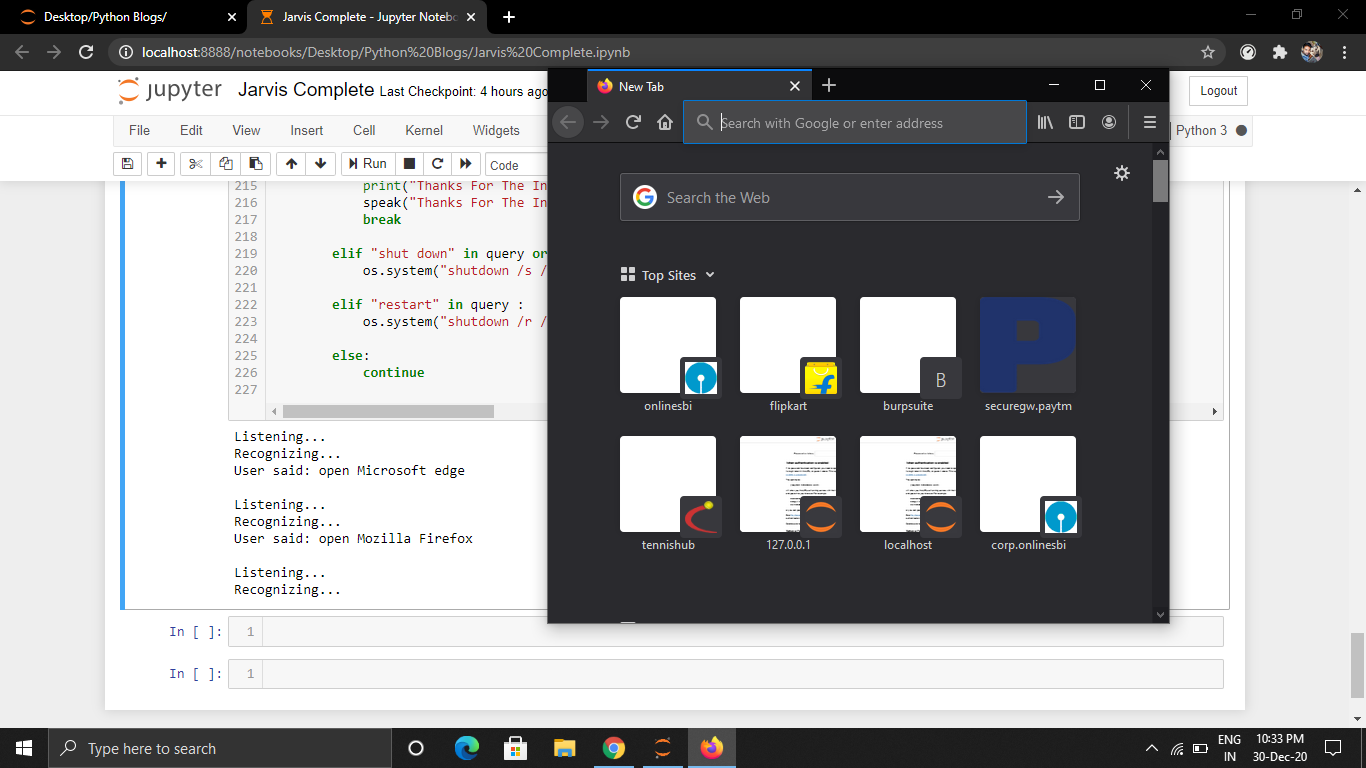
**SNAPSHOTS**

Few of the screenshots of working program are shown below :

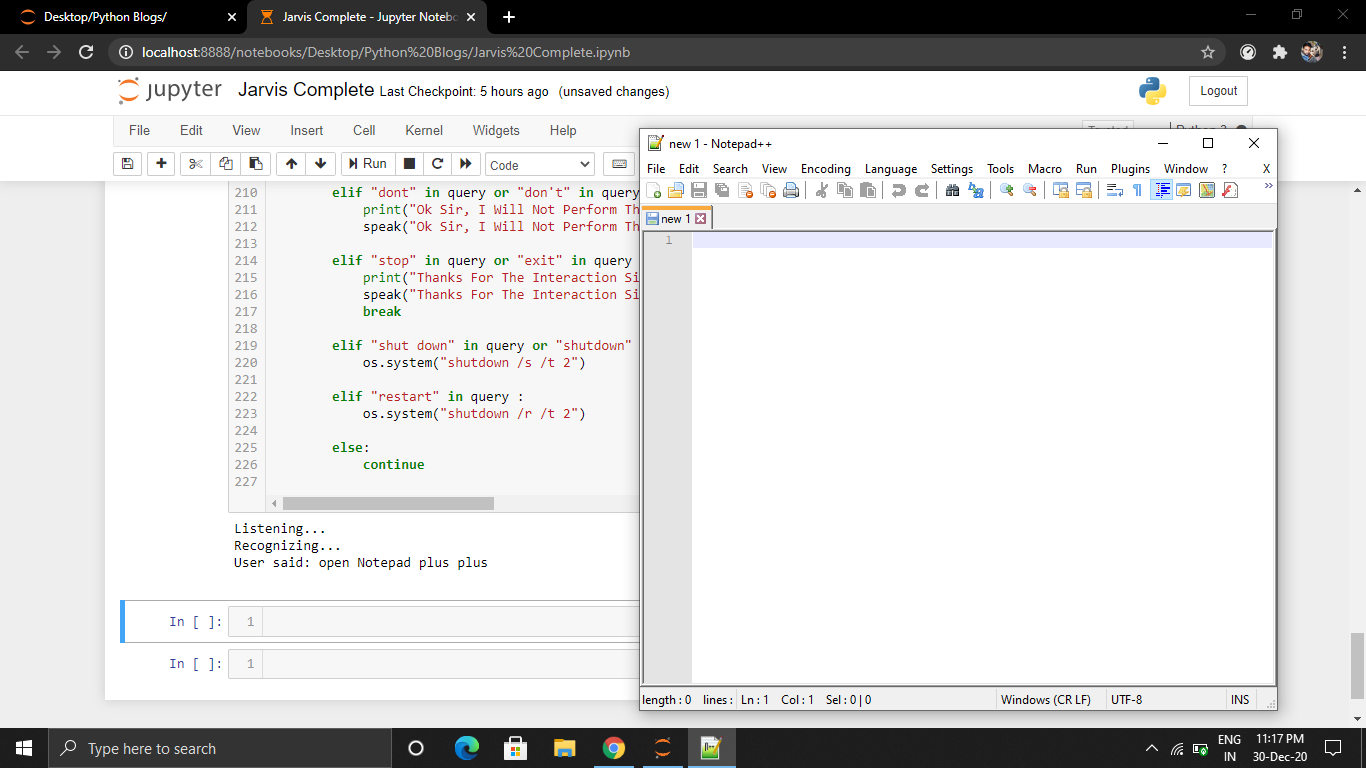
**i. Opening Microsoft Edge**

****

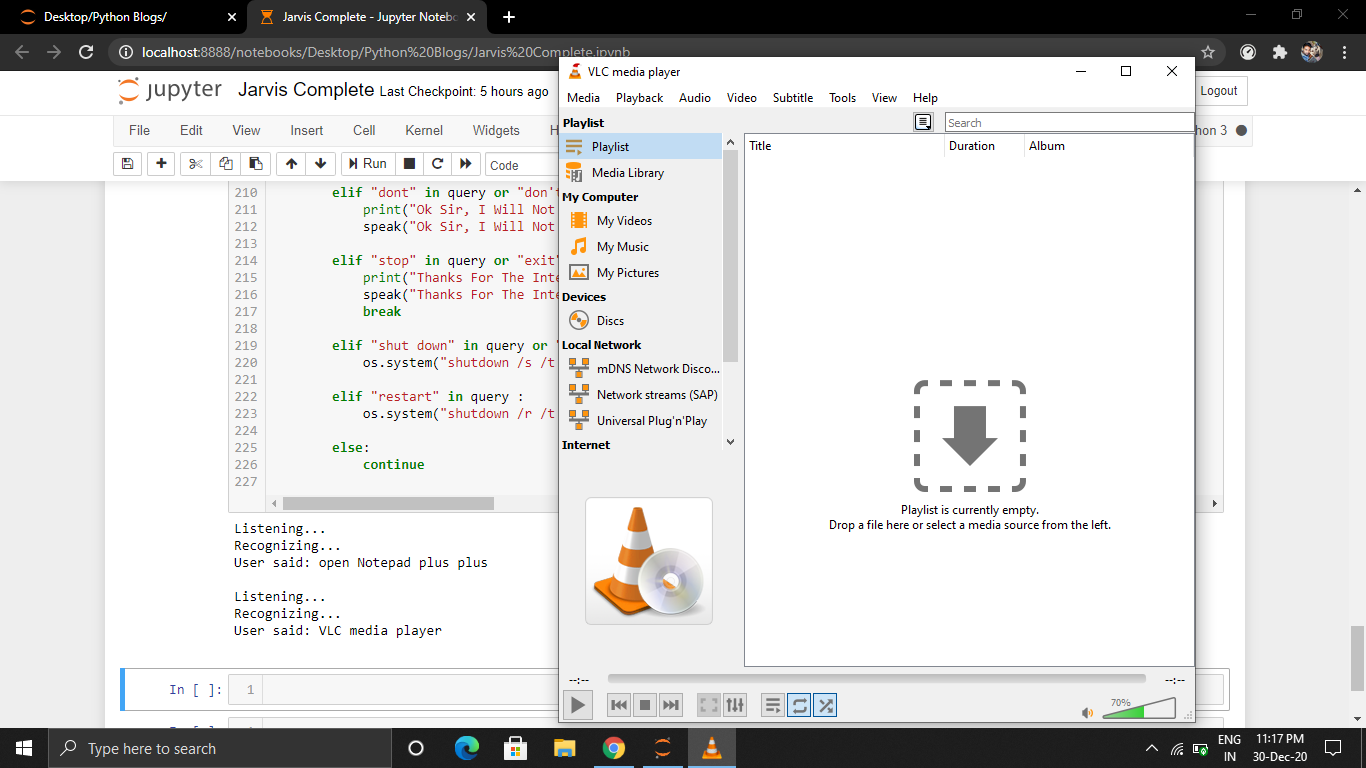
**ii. Opening Mozilla Firefox**

****

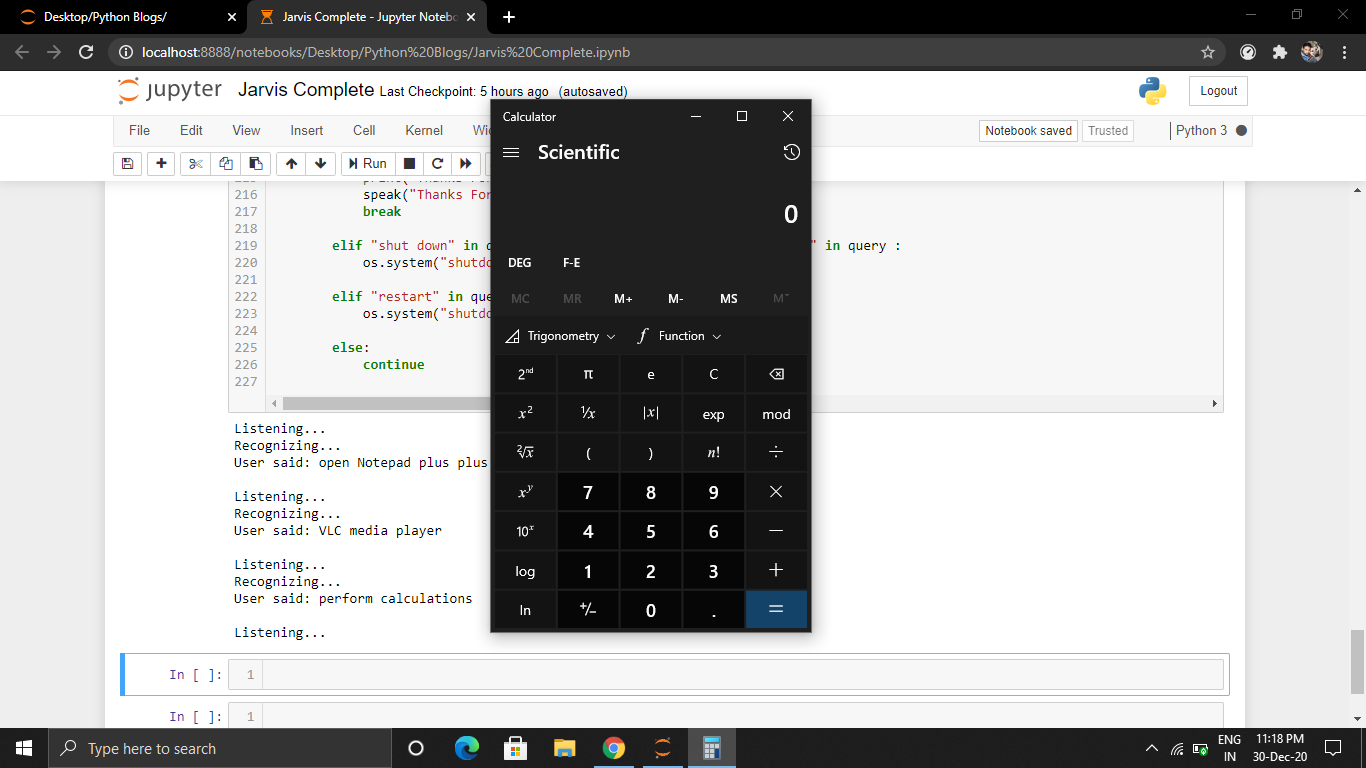
**iii. Opening Notepad++ (Text Editor)**

****

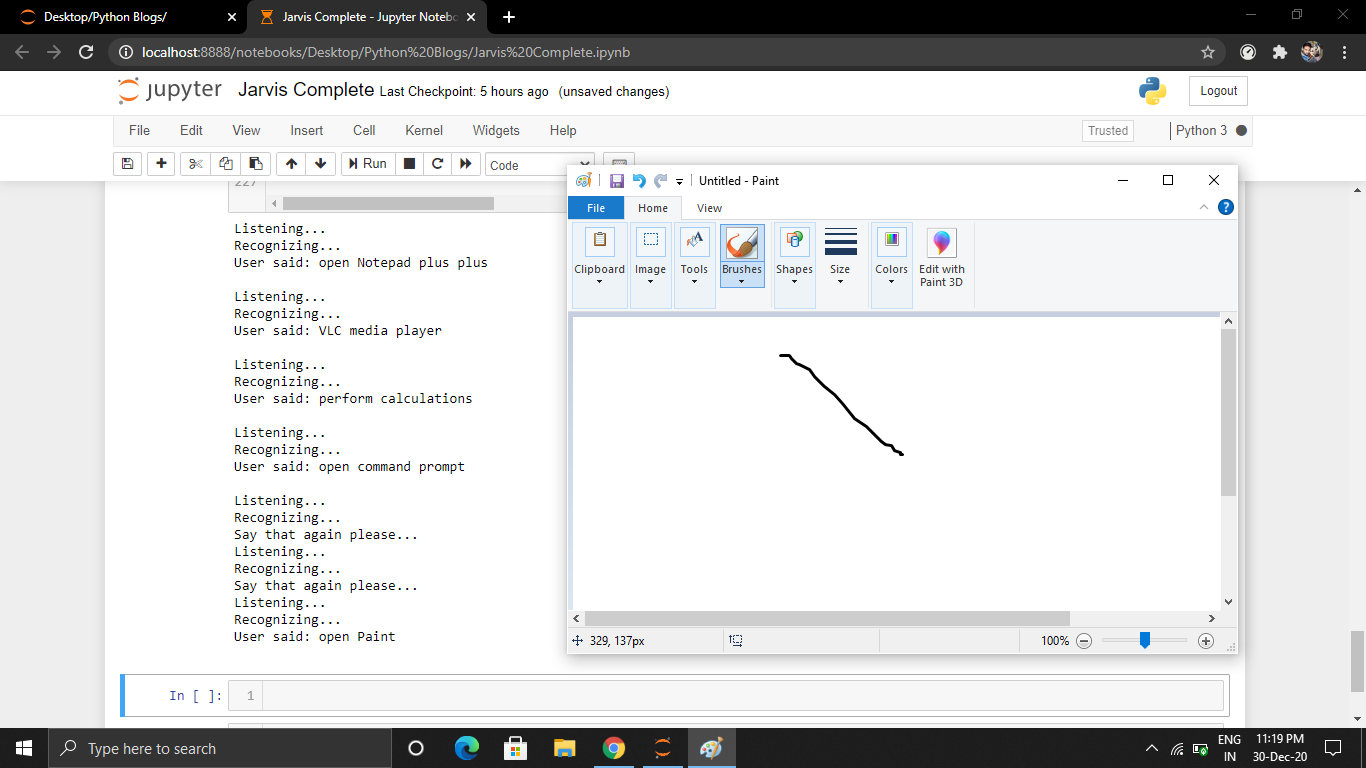
**iv. Opening VLC Media Player**

****

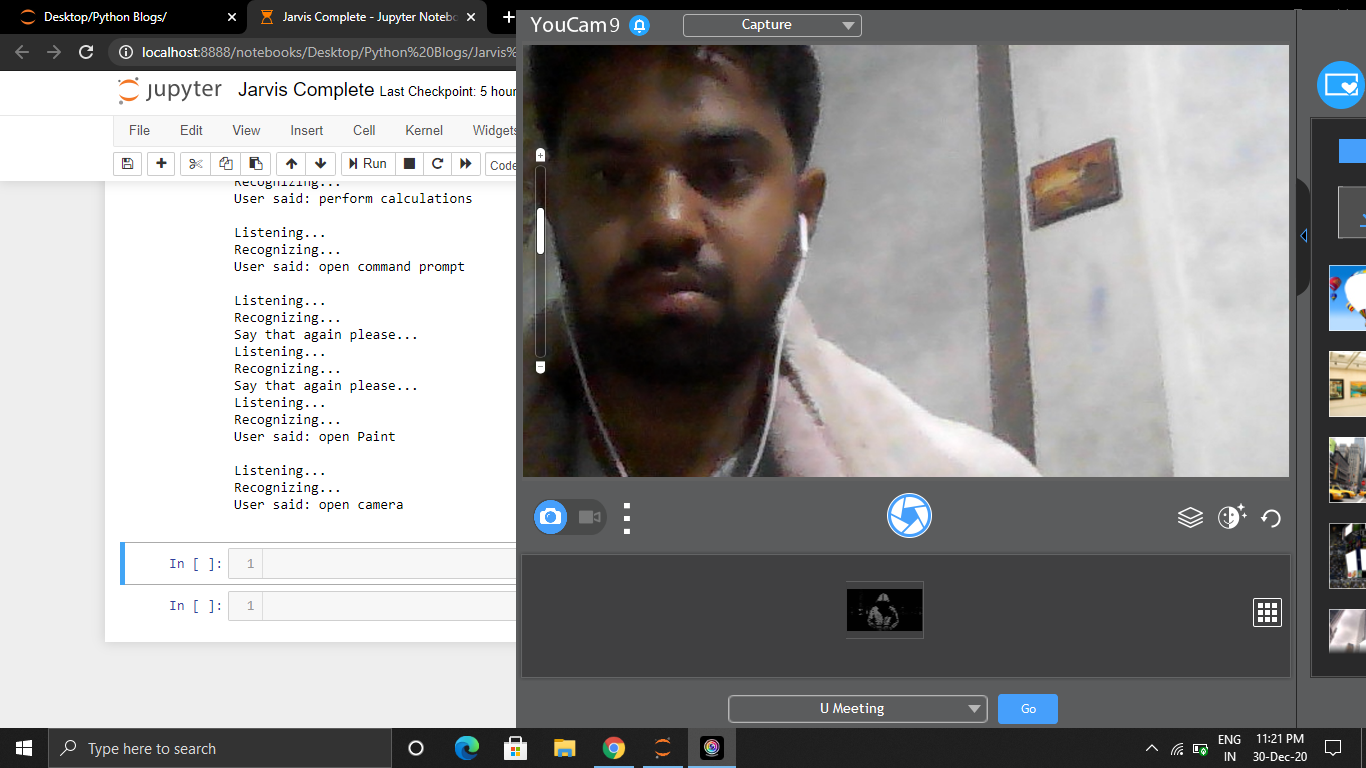
**v. Opening Calculator**

****

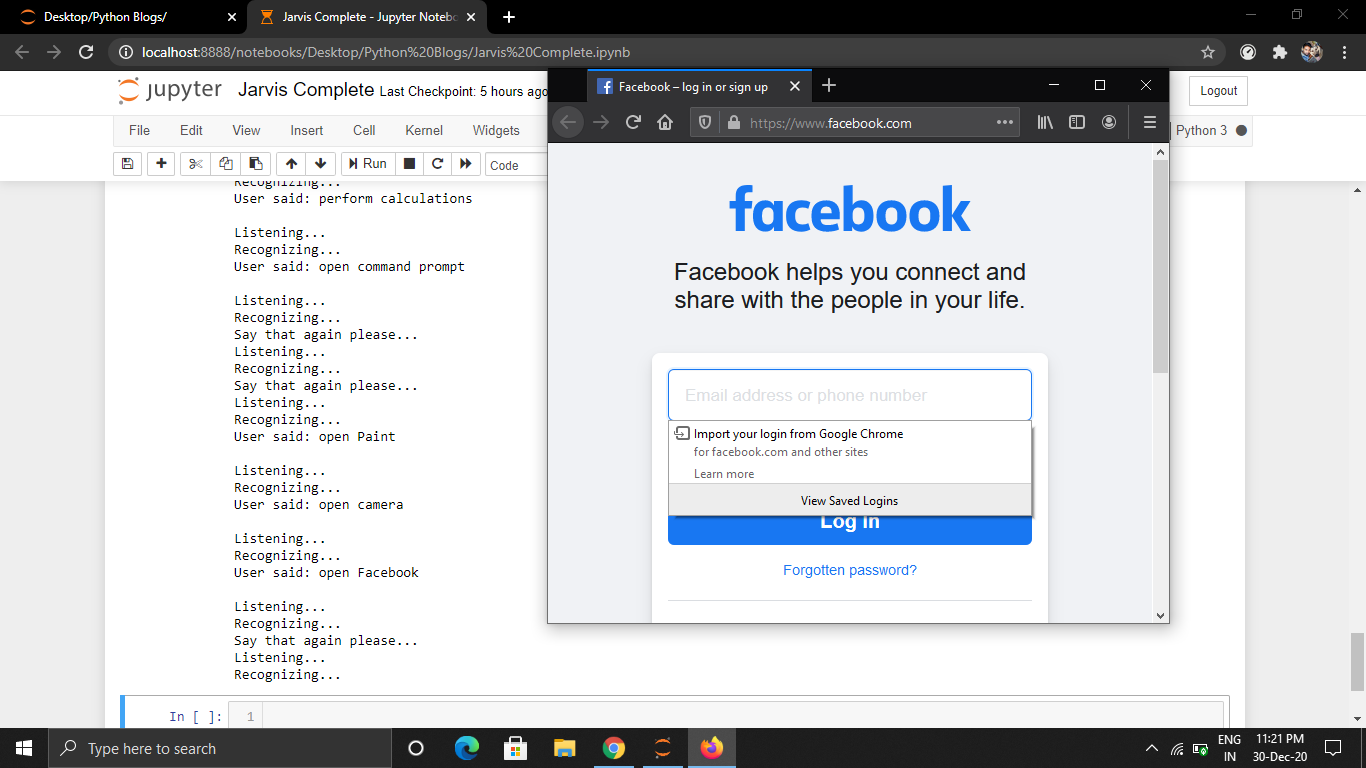
**vi. Opening MS Paint**

****

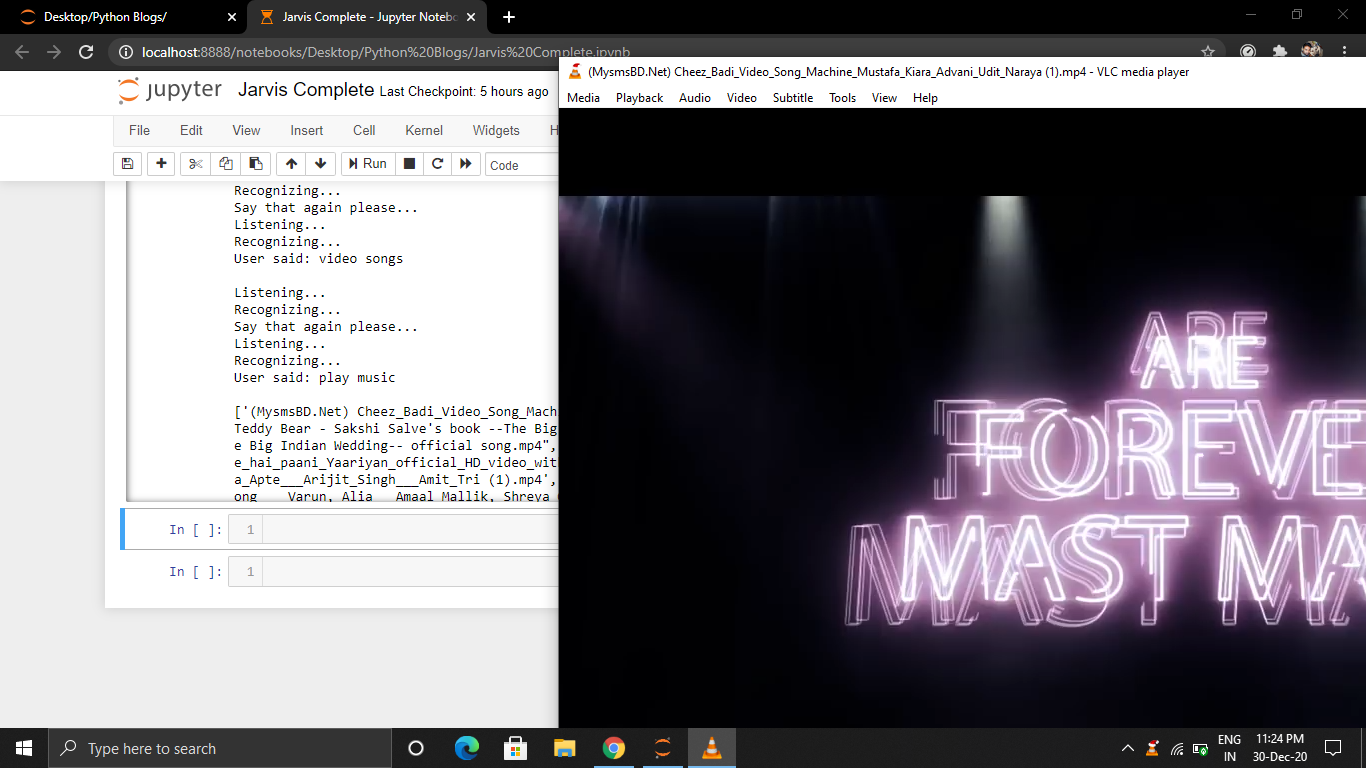
**vii. Opening Camera**

****

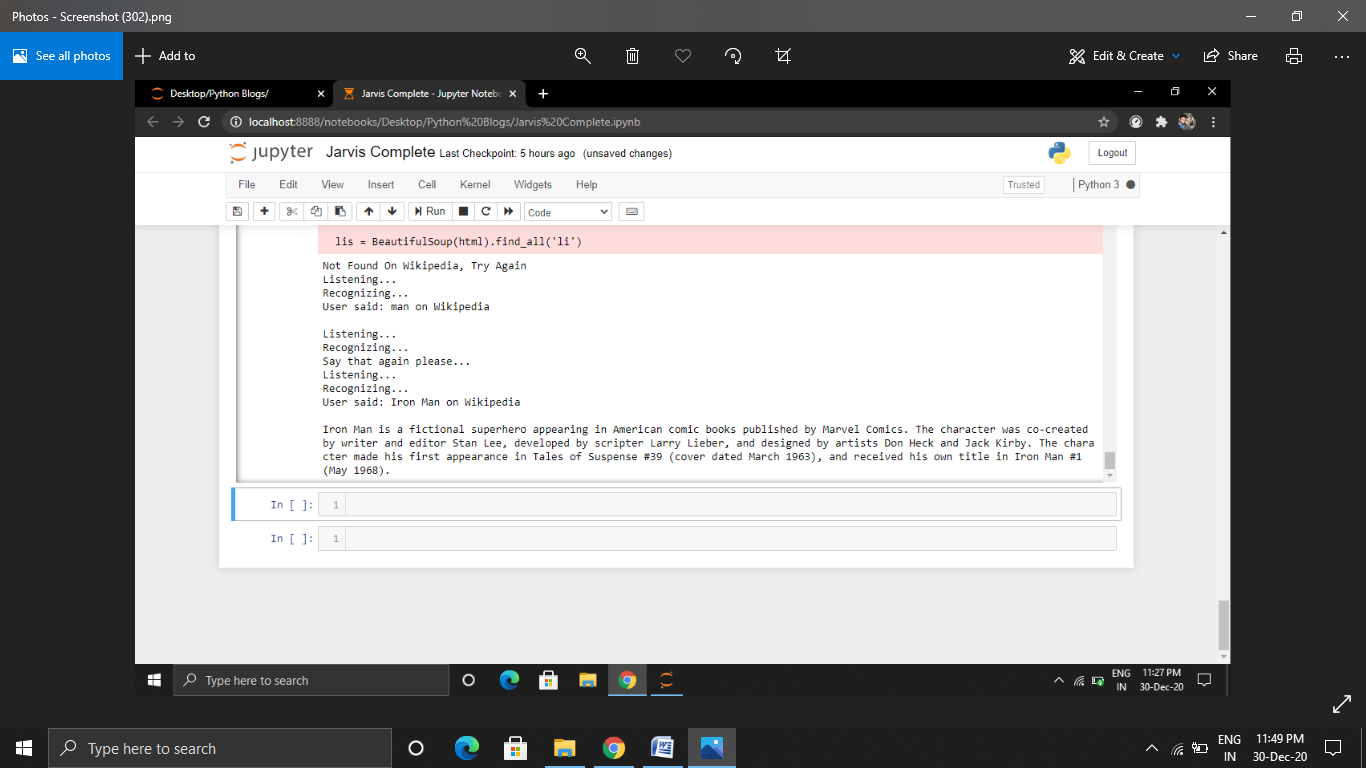
**viii. Visiting Facebook**

****

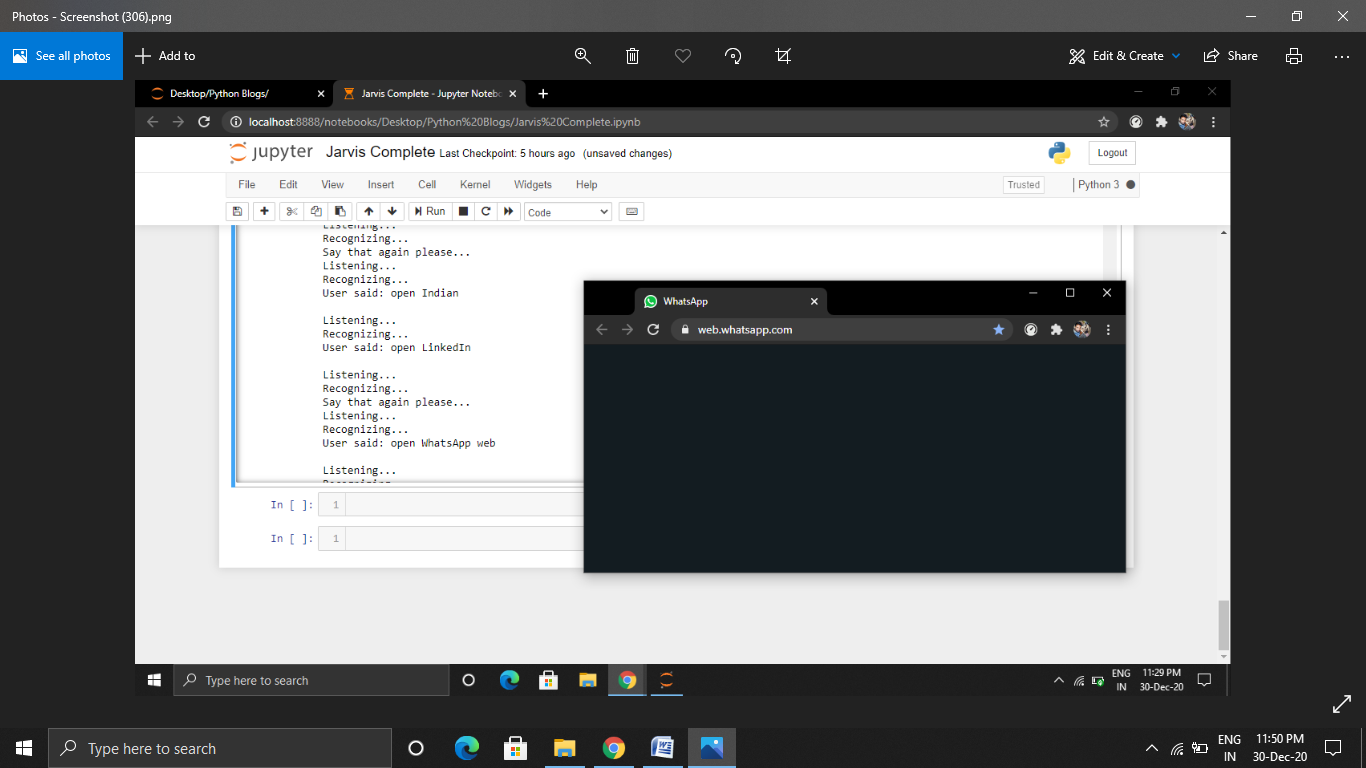
**ix. Playing Music**

****

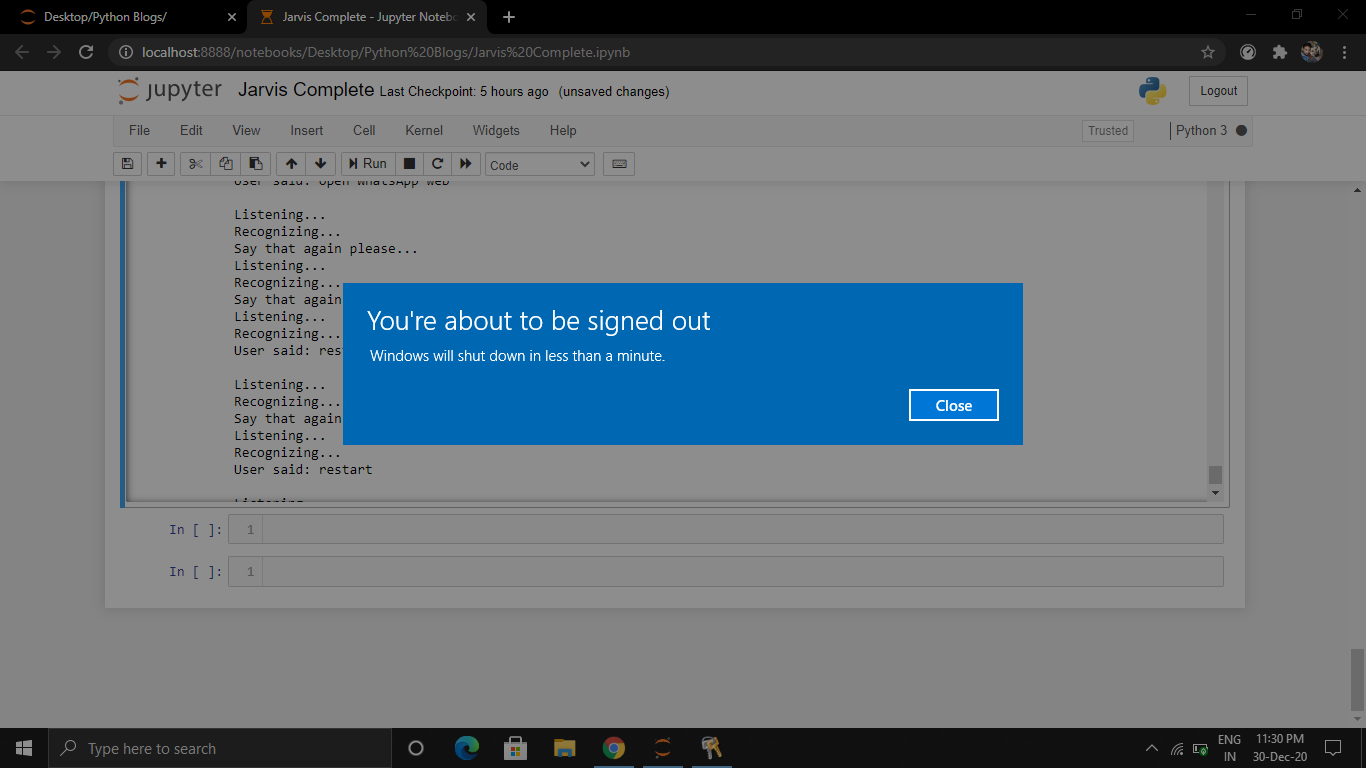
**x. Wikipedia Searches**

****

**xi. Opening Whatsapp Web**

****

**xii Restarting PC**

****