

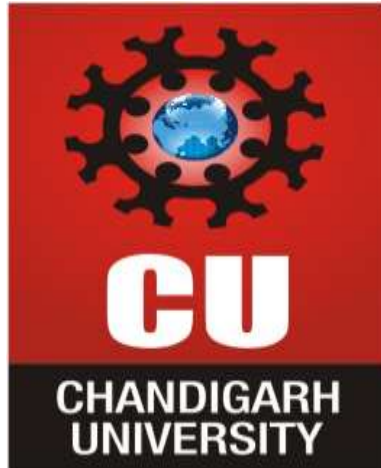
Desktop Assistant

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



Shweta
Submitted to:
(Shweta Agarwal : E9500)
Ass. Prof. Shweta Ma'am

Submitted By:

Vikash Kumar Gupta (18BCS6641)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Chandigarh University, Gharuan

April 2021

Project Design

Introduction :

This project is based on Python based libraries and provides a personal assistant using voice recognition or text mode operation. This program includes the functions and services of: calling services, text message transformation, mail exchange, alarm, event handler, location services, music player service, checking weather, Google searching engine, Wikipedia searching engine, robot chat, camera, Bing translator, Bluetooth headset support, help menu and Windows azure cloud computing.

This is a short description about “Cortana” from Wikipedia to illustrate the voice product: “Cortana” is a virtual assistant developed by Microsoft, which uses the Bing search engine to perform tasks such as setting reminders and answering questions for the user.

Cortana is currently available in English, Portuguese, French, German, Italian, Spanish, Chinese, and Japanese language editions, depending on the software platform and region in which it is used.

But we are implementing our voice assistant in American English only.

Microsoft has integrated Cortana into numerous products such as Microsoft Edge, the browser bundled with Windows 10. Microsoft's Cortana assistant is deeply integrated into its Edge browser. Cortana can find opening hours when on restaurant sites, show retail coupons for websites, or show weather information in the address bar. At the Worldwide Partners Conference 2015 Microsoft demonstrated Cortana integration with products such as GigJam. Conversely, Microsoft announced in late April 2016 that it would block anything other than Bing and Edge from being used to complete Cortana searches, again raising questions of anti competitive behavior by the company.

Microsoft's "Windows in the car" concept includes Cortana. The concept makes it possible for drivers to make restaurant reservations and see places before they go there.

Constraints:

The application was built for windows user only for different version of windows such as 10 single , home and professional versions. But our application will work all those OS having python development platform and will be easily available

The system also assumes that the user has minimal English knowledge as of now.

Aim and Purpose

Desktop Assistant was built to help the people with limited computer knowledge but it is also important to note that the other class of users might find some specific functionalities such as system logging useful.

The people who are not familiar getting around on their own on a computer can use this application as it abstracts away all the steps and presents only most important. Like they can just say to connect to a particular WIFI or blue tooth, it will connect. Want to decrease screen brightness or volume? Want to open and close an application, file? Just say it as you would to a person, it will do the work for you.

Many parents are afraid about what their child is surfing in the internet. The application can notify their parents about the activity. Even if the child deletes all the history or uses incognito mode etc. The application

logs every keystroke and website used by the child. Don't worry as they won't leave any trace while running. It completely hides and doesn't require any permissions to run.

It can also help with getting around web by opening, closing, bookmarking, reload, back, next etc a web page just by saying it to do so.

*Data extraction and interacting with a web page elements/nodes is still under working.

Finally, it is the solution for monitoring the computer. It is the solution for the parents who got a computer from their son/daughter and can't operate it to talk to them. It is the solution for the people who are willing to make a computer as their real assistant(not like fake so called assistants like Siri, Cortana etc). Even if anyone uses it, you get complete stats about what he/she was doing, what did they connected etc.

Innovation in model/design/solution

Innovation in Project:

With modest expectations in mind, each does its job relatively well. If you require more specificity, designing your own digital assistant is far from a pipe dream. Recent advances in speech recognition and converting text to speech make it viable even for hobbyists. And working in Python greatly simplifies the task, giving you the ability to make any number of customization to tailor your assistant to your needs.

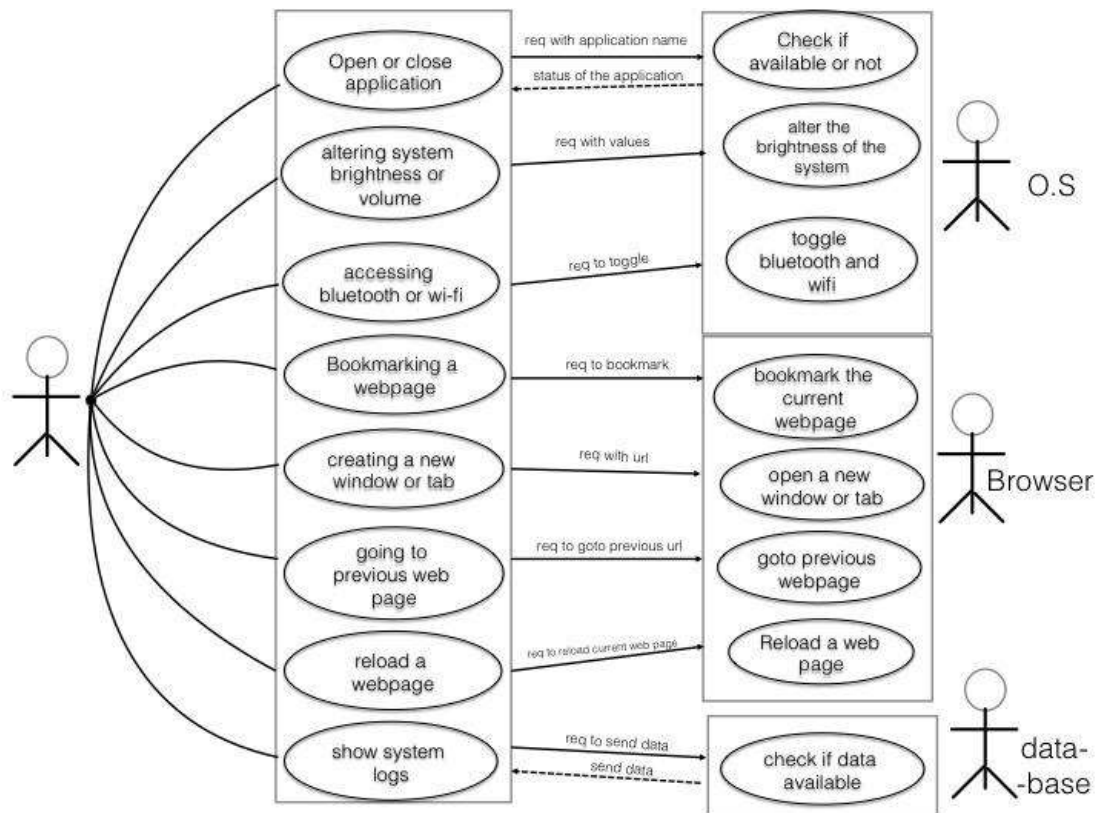
Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make 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 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 Desktop Assistant.

Diagrams:

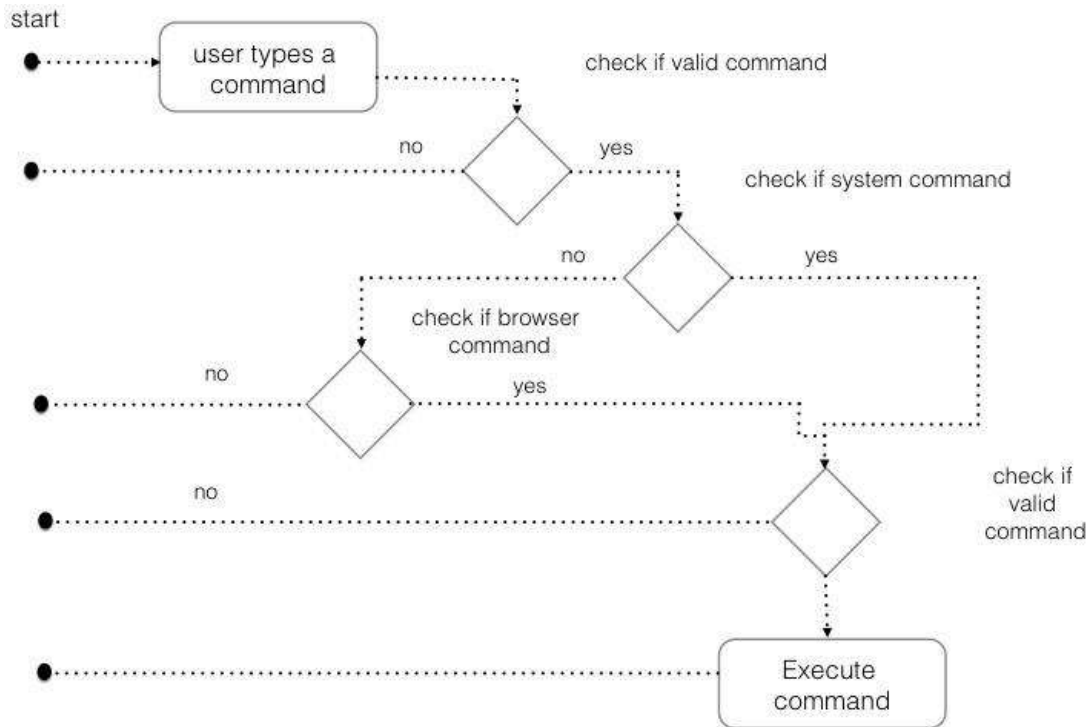
A use case is a set of scenarios that describe an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

UML



Activity Diagram:

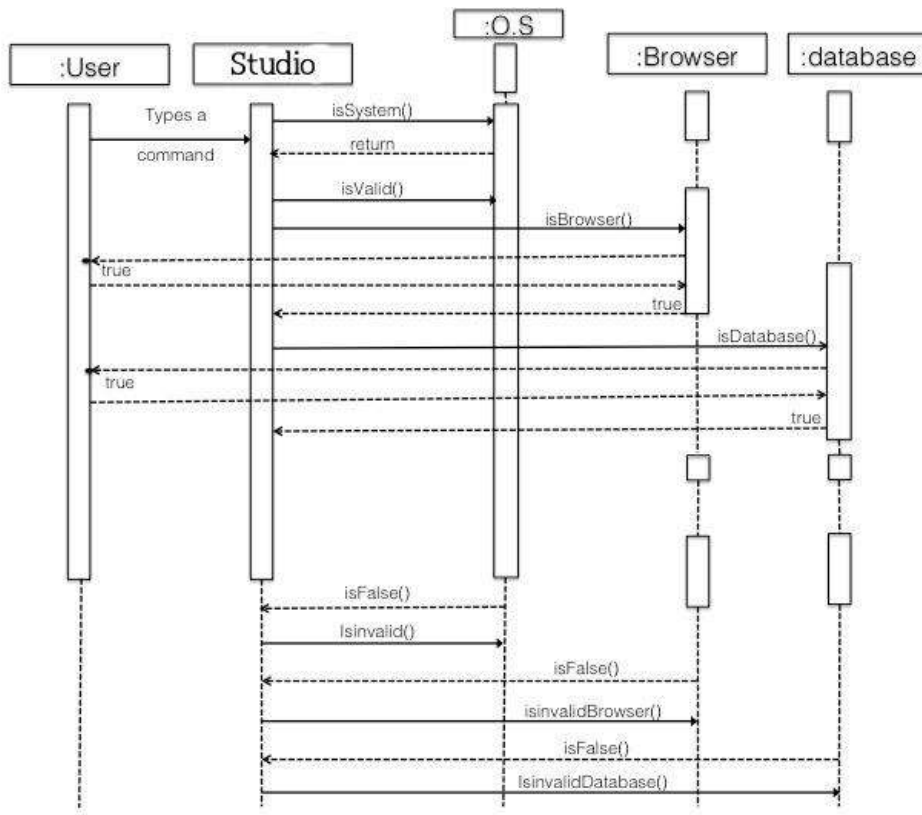
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



Sequence Diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario



Implementation

we are implementing the above application Voice assistant using python and its libraries . Anaconda environment is the main requirement for this project as its job is to integrate libraries and create and environment for their execution. It is easy to download the required libraries in Anaconda environment and then one can use jupyter lab or spider developing console for the code development .

Following libraries has been use till now :

1. pyttsx3
2. pyaudio
3. google api speech recogniton

Defining Speak Function

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

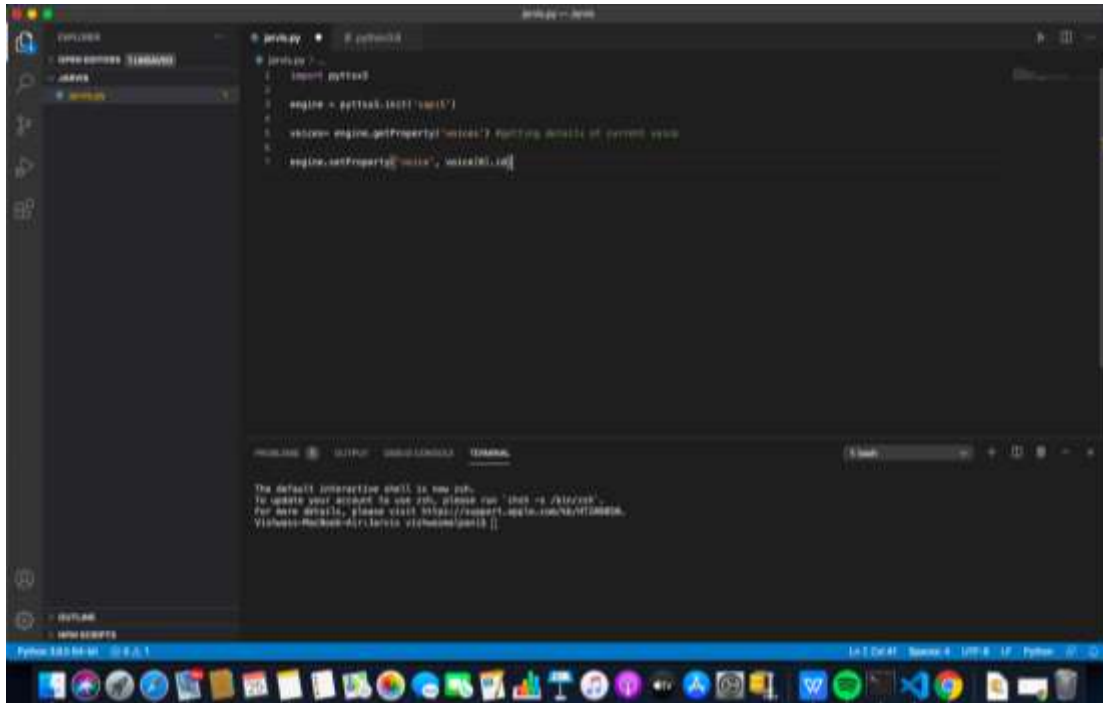
then, it will pronounce it.

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.

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 Voice Id?

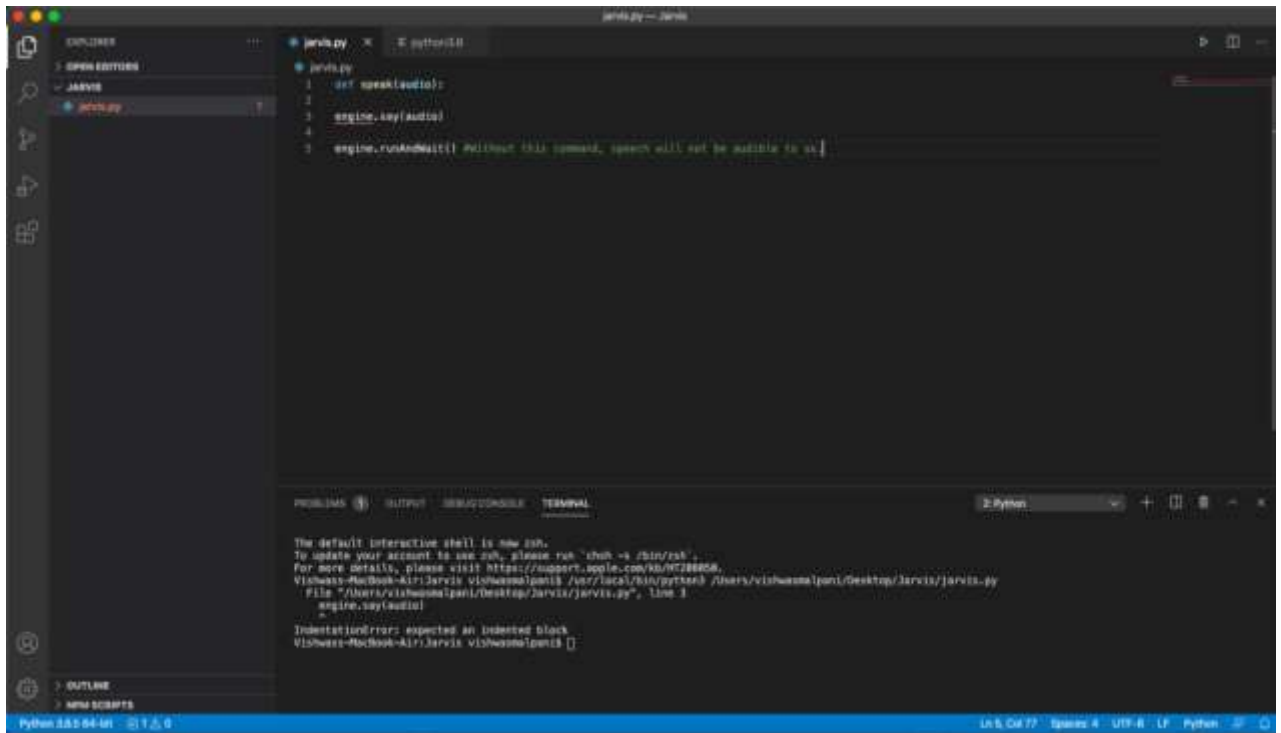
- Voice id helps us to select different voices.
- `voice[0].id` = Male voice
- `voice[1].id` = Female voice

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.



```
1 def speak(audio):
2     engine.say(audio)
3     engine.runAndWait() #Without this command, speech will not be audible to us.
4
5 def main():
6     speak("Jarvis is ready to serve you.")
7
8 if __name__ == '__main__':
9     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit <https://support.apple.com/kb/HT208050>.
Vishwas-MacBook-Air:Jarvis vishwas@vishwas: /usr/local/bin/python3 /Users/vishwas@vishwas/Desktop/Jarvis/Jarvis.py
File "/Users/vishwas@vishwas/Desktop/Jarvis/Jarvis.py", line 3
 engine.say(audio)
 ^
IndentationError: expected an indented block
Vishwas-MacBook-Air:Jarvis vishwas@vishwas []

Whatever you will write inside this speak() function will be converted into speech.

Congratulations! With this, our Vice Assistant has its own voice, and it is ready to speak.

Here, we have stored the integer value of the current hour or time into a variable named hour.

Now, we will use this hour value inside an if-else loop.

Defining Take command Function :

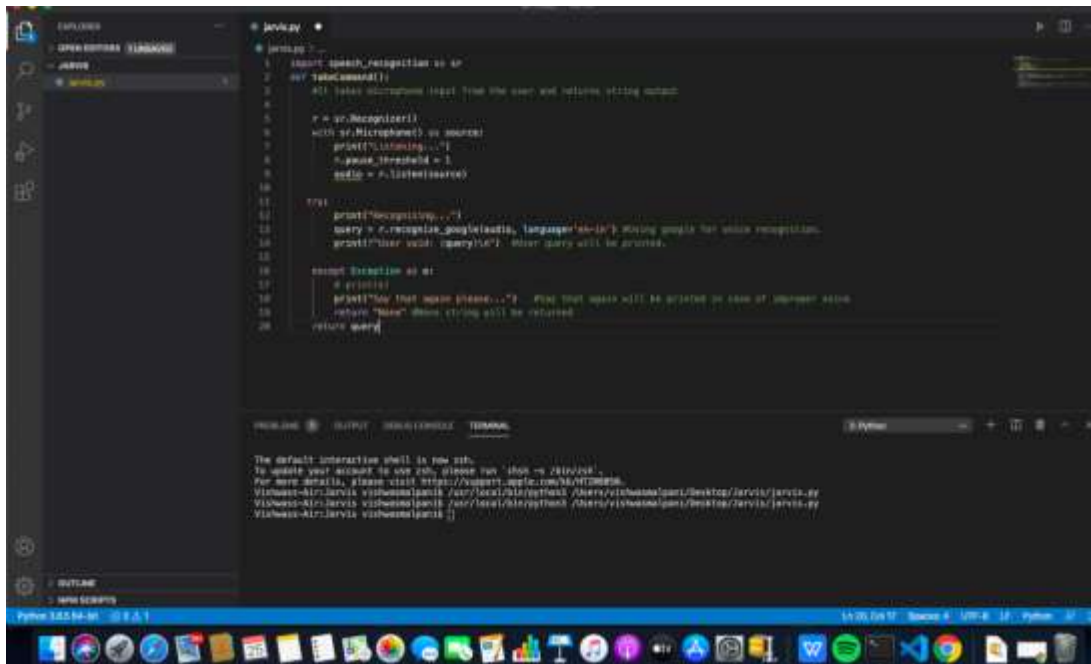
The next most important thing for our A.I. assistant is that it should be able to take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will be able to return a string output by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

Before defining the takeCommand() function, we need to install a module called speech

Recognition. Install this module by:

pip install speechRecognition

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



We have successfully created our takeCommand() function. Now we are going to add a try and except block to our program to handle errors effectively.

Coding logic of Voice Assistant

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

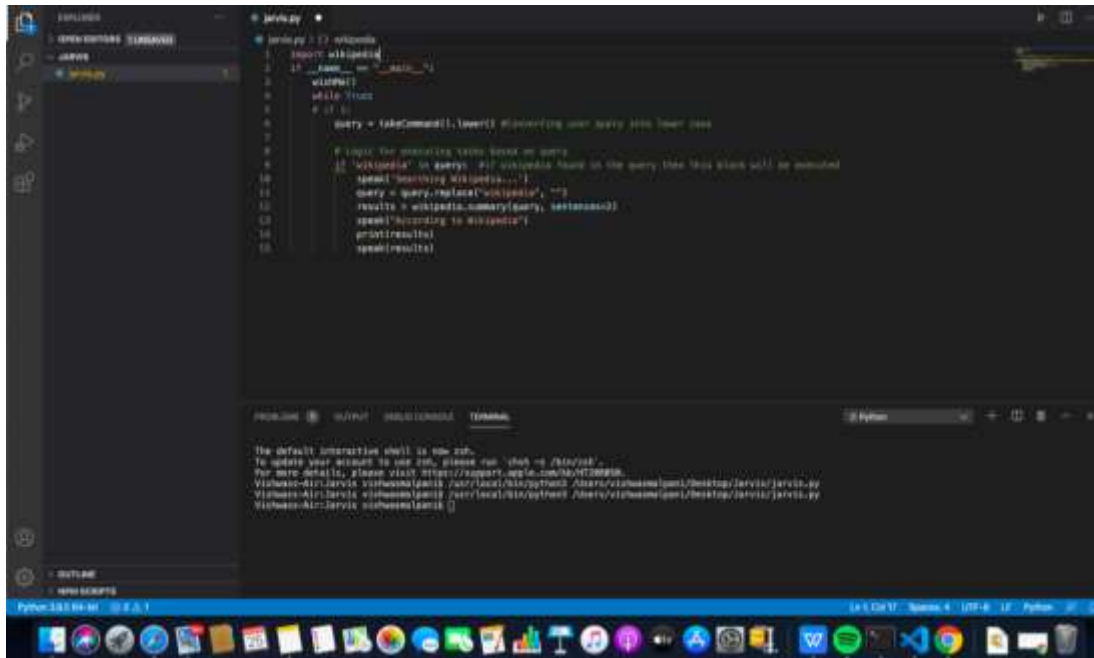
Defining Task 1: To search something on Wikipedia

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 :

pip install wikipedia

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



```
1 import wikipedia
2 if __name__ == '__main__':
3     wikipedia
4     while True:
5         # if {}
6         query = takeCommand().lower() #lowering user query into lower case
7
8         # logic for searching wikipedia on query
9         if 'wikipedia' in query: #if wikipedia found in the query then this block will be executed
10             query = query.replace('wikipedia', '')
11             results = wikipedia.summary(query, sentences=10)
12             speak("According to wikipedia")
13             print(results)
14             speak(results)
```

In the above 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.

Defining Task 2: To open YouTube site in a web browser

To open any website, we need to import a module called web browser. 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 "Open youtube." So, open youtube will be in the user's query, and the elif condition will be true.

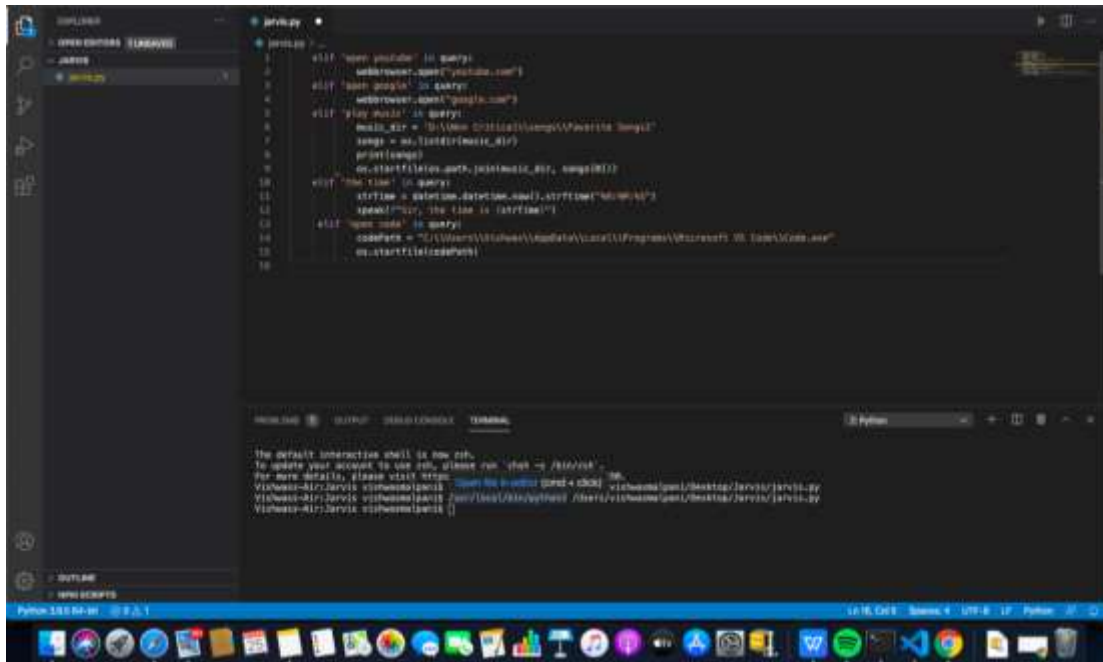
Defining Task 3: To open Google site in a web browser

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

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.starfile, 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, voice Assistant will play any random song from the song directory.



The image shows a screenshot of a Visual Studio Code editor window. The editor is open to a file named 'jarkov.py'. The code in the file is as follows:

```
1 elif 'see youtube' in query:
2     webbrowser.open("youtube.com")
3 elif 'see google' in query:
4     webbrowser.open("google.com")
5 elif 'play music' in query:
6     music_dir = "D:\\Music\\Songs\\Favorite Songs"
7     songs = os.listdir(music_dir)
8     print(songs)
9     randomfile=os.path.join(music_dir, songs[0])
10
11 elif 'the time' in query:
12     strftime = datetime.datetime.now().strftime("%H:%M:%S")
13     speak(f"Sir, the time is {strftime}")
14 elif 'open code' in query:
15     codepath = "C:\\Users\\Vishwas\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
16     os.startfile(codepath)
17
```

The terminal window at the bottom shows the output of the script, which is a list of songs in the 'Favorite Songs' directory:

```
the default interaction shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://wiki.archlinux.org/title/zsh
Vishwas-Mir:Jarkov vishwas@pavils ~$ python3 jarkov.py
D:\\Music\\Songs\\Favorite Songs
['01 - The Time (Official Music Video).mp3', '02 - The Time (Official Music Video).mp3', '03 - The Time (Official Music Video).mp3', '04 - The Time (Official Music Video).mp3', '05 - The Time (Official Music Video).mp3', '06 - The Time (Official Music Video).mp3', '07 - The Time (Official Music Video).mp3', '08 - The Time (Official Music Video).mp3', '09 - The Time (Official Music Video).mp3', '10 - The Time (Official Music Video).mp3']
```

Implementation (50%)

