MINOR PROJECT REPORT
ON

(VIRTUAL ASSISTANT)

Submitted by:

PRADYUMAN
PRATAP SINGH 02511104421

Under the guidance of

Dr.SHUSHMA BAHUGUNA

BANARASIDAS CHANDIWALA INSTITUTE OF INFORMATION TECHNOLOGY

(Affiliated to Guru Gobind Singh Indraprastha University)(2021-2023)

TABLE OF CONTENTS

DECLARATION

I hereby declare that the Project report entitled VIRTUAL ASSISTANT submitted to the MCA MINOR PROJECT. Project is a record of original work done by me, under the guidance and supervision of mentor
DR SHUSHMA BAHUGUNA.

I further declare that:-

1. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
2. 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.
3. We have followed the guidelines provided by the university in writing the report.

Signed :(…………………..)   Date: …………………..
PRADYUMAN PRATAP SINGH

CERTIFICATE

This is to certify that PRADYUMAN PRATAP SINGH , student of BCIIT has completed the project work and has submitted a report for the same in partial fulfilment of the Minor Project on topic of
" ***VIRTUAL ASSISTANT*** ", he has worked under the guidance as directed.

ABSTRACT

As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa , Siri , etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

ACKNOWLEDGEMENT

I thank the almighty for giving us the courage and perseverance in completing the Internship project. This project itself is acknowledgement for all those people who have give us their heartfelt co-operation in making this project a grand success.

I am greatly indebted to project guide DR SHUSHMA BAHUGUNA for providing valuable guidance at every stage of this project work. I am profoundly grateful towards the unmatched services rendered by her.

Last but not least we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the minor project.

CHAPTER 1
Introduction/Problem Definition

1.1 INTRODUCTION

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries used to make this assistant focuses on the time complexities and reduces time.

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly.

We already have multiple virtual assistants. But we hardly use it. There are number of people who have issues in voice recognition. These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system.

1.2 OBJECTIVES

The main goal of this website is:

- To create user friendly interface
- To bring out immediate and accurate results
- To save user’s time
- To decreasing human efforts
- To help in accessing routine user activities easily

One of the main advantages of voice searches is their rapidity. In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 during the same period of time15. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

1.3 METHODOLOGY

The Virtual Assistant developed using Python ,This requires good knowledge of Python.As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa , Siri , etc.

CHAPTER 2

Systems Requirement Analysis

2.1 INTRODUCTION

Voice recognition software enables hands free use of the applications, lets users to query or command the agent through voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself. Nebula Voice Assistant also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally.
Virtual assistants must provide a wide variety of services.

- can help to do lot many things like..
- can tell you the current time and date,
- can tell you the current weather,
- can tell you battery and cpu usage,
- can create the reminder list,
- can take screenshots,
- can send email to your boss or family or your friend,
- can shut down or logout or hibernate your system,
- can tell you non funny jokes,
- can open any website,
- can search the thing on wikipedia,
- can change my voice from male to female and vice-versa

2.2 LIBRARIES OR MODULES

- pyttsx3
- datetime
- speech_recognition as sr
- wikipedia
- smtplib
- webbrowser as wb
- os

- pyautogui
- psutil
- pyjokes

2.3 DESCRIPTION

- **pyttsx3**

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text- to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

- **Datetime**

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

- **speech_recognition**

This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

- **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.

- **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.

- **webbrowser as wb**

The webbrowser module is a convenient web browser controller in the Python programming language. This module offers a high-level interface that enables showing the documents based on the web. Under most circumstances, we can call the open() function from the webbrowser module to perform the right thing.

- **OS**

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules.

- **Pyautogui**

PyAutoGUI is essentially a Python package that works across Windows, MacOS X and Linux which provides the ability to simulate mouse cursor moves and clicks as well as keyboard button presses.

- **psutil**

Psutil is a Python cross-platform library used to access system details and process utilities. It is used to keep track of various resources utilization in the system. Usage of resources like CPU, memory, disks, network, sensors can be monitored.

- **Pyjokes**

Import the pyjokes module in a Python file and use the get_joke function to easily drop a random joke into your application

2.4 TECHNOLOGY USED

Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.
Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JIA, libraries used are speechrecognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc.
Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

Visual Studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

Visual Studio Code runs on macOS, Linux and Windows

Edit, build, and debug with ease#
At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

Make it your own#
Customize every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you, and we encourage you to optimize your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

Built with love for the Web#
VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio.

VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

Robust and extensible architecture#
Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron, VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML-based editor that has powered the "Monaco" cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

2.5 TECHNOLOGY USED

The software is designed to be light-weighted so that it doesn’t be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for virtual assistant.

Hardware:

- Pentium-pro processor or later.
- RAM 512MB or more.

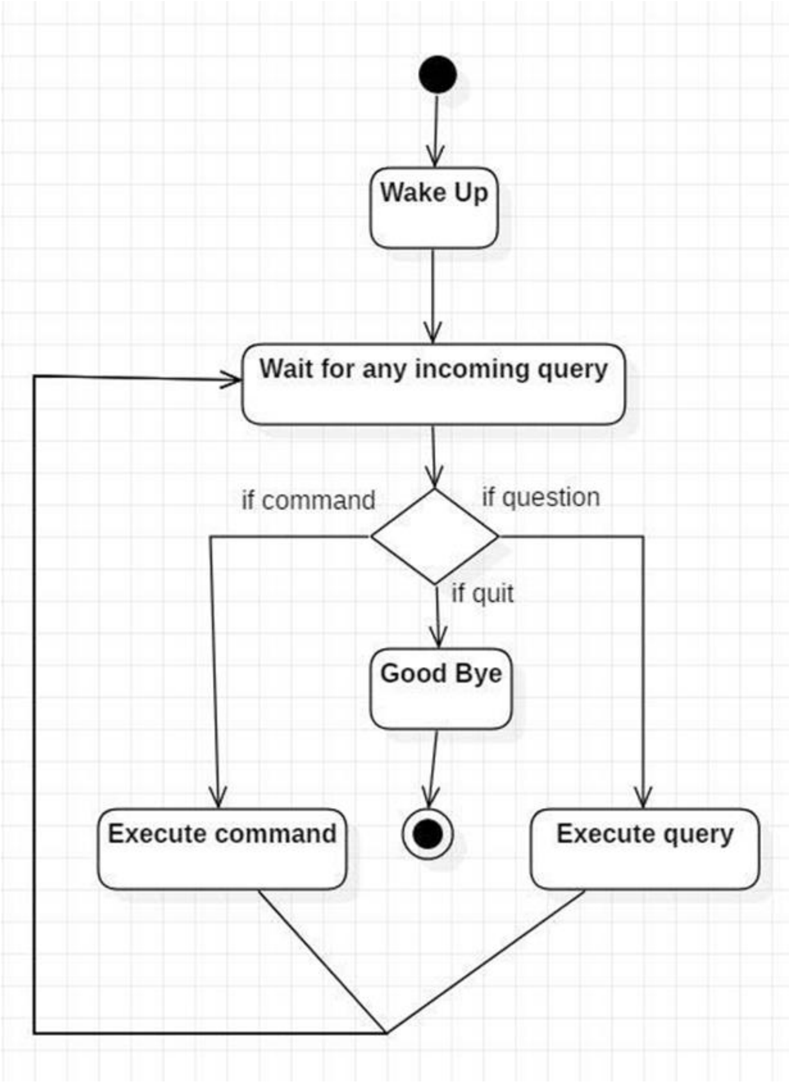
Software:

- Windows 7(32-bit) or above.
- Python 2.7 or later
- Chrome Drive
- Visual Studio Code

CHAPTER 3

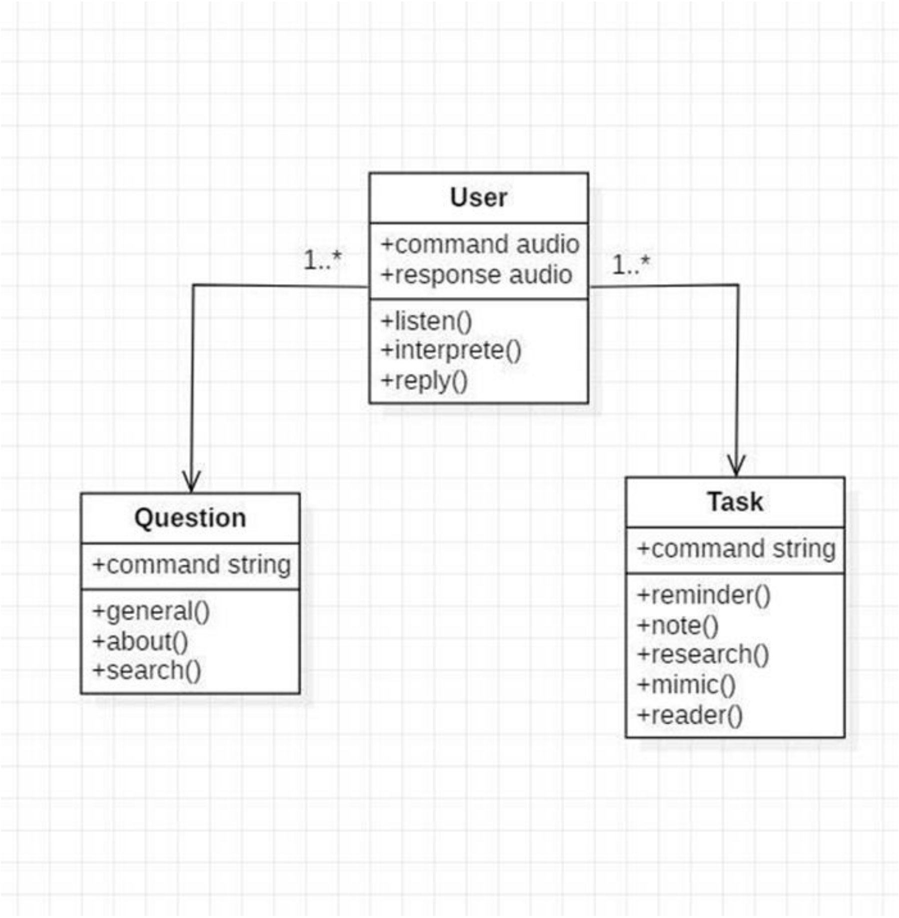
System Designs

3.1 ACTIVITY DIAGRAM



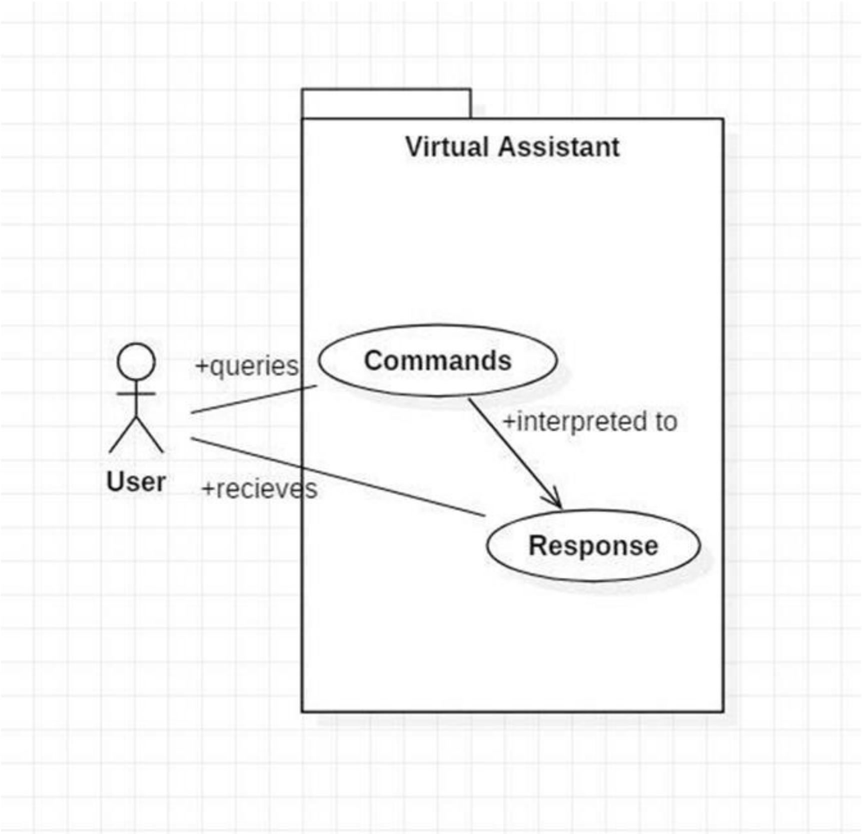
Initially, the system is in idle mode. As it receives any wake up cal it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

3.2 CLASS DIAGRAM



The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

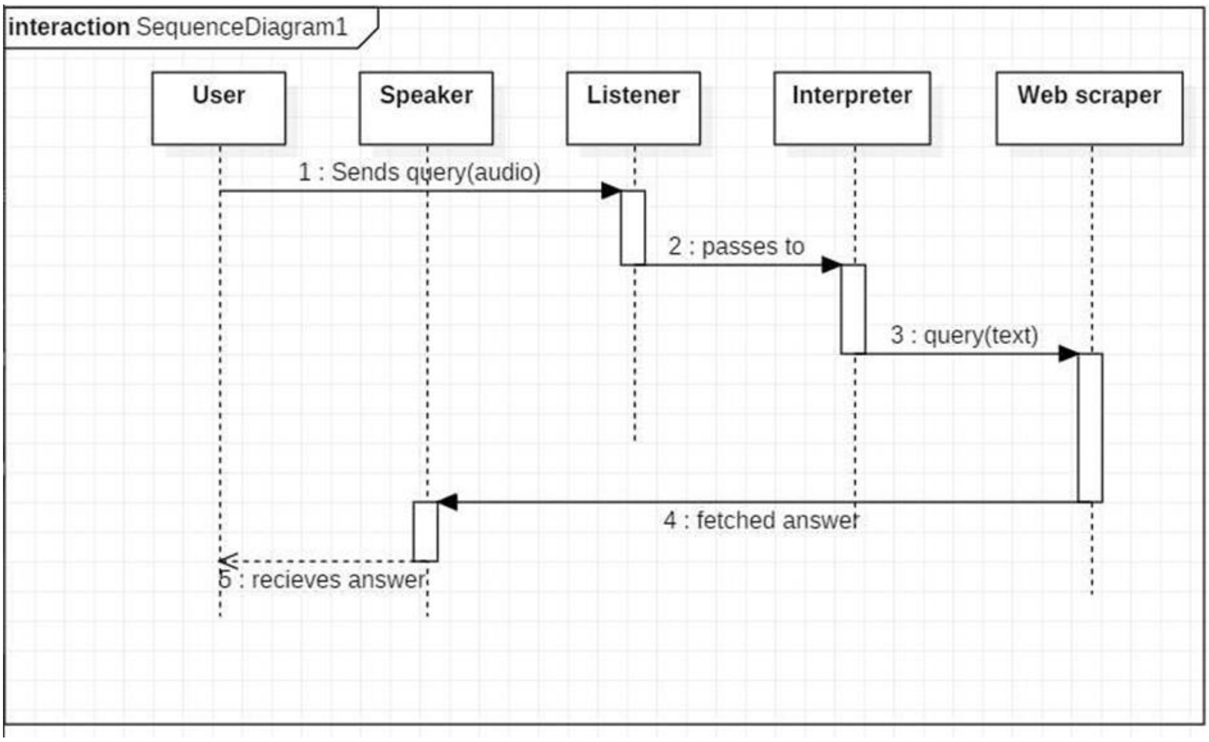
3.3 USE CASE DIAGRAM



In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

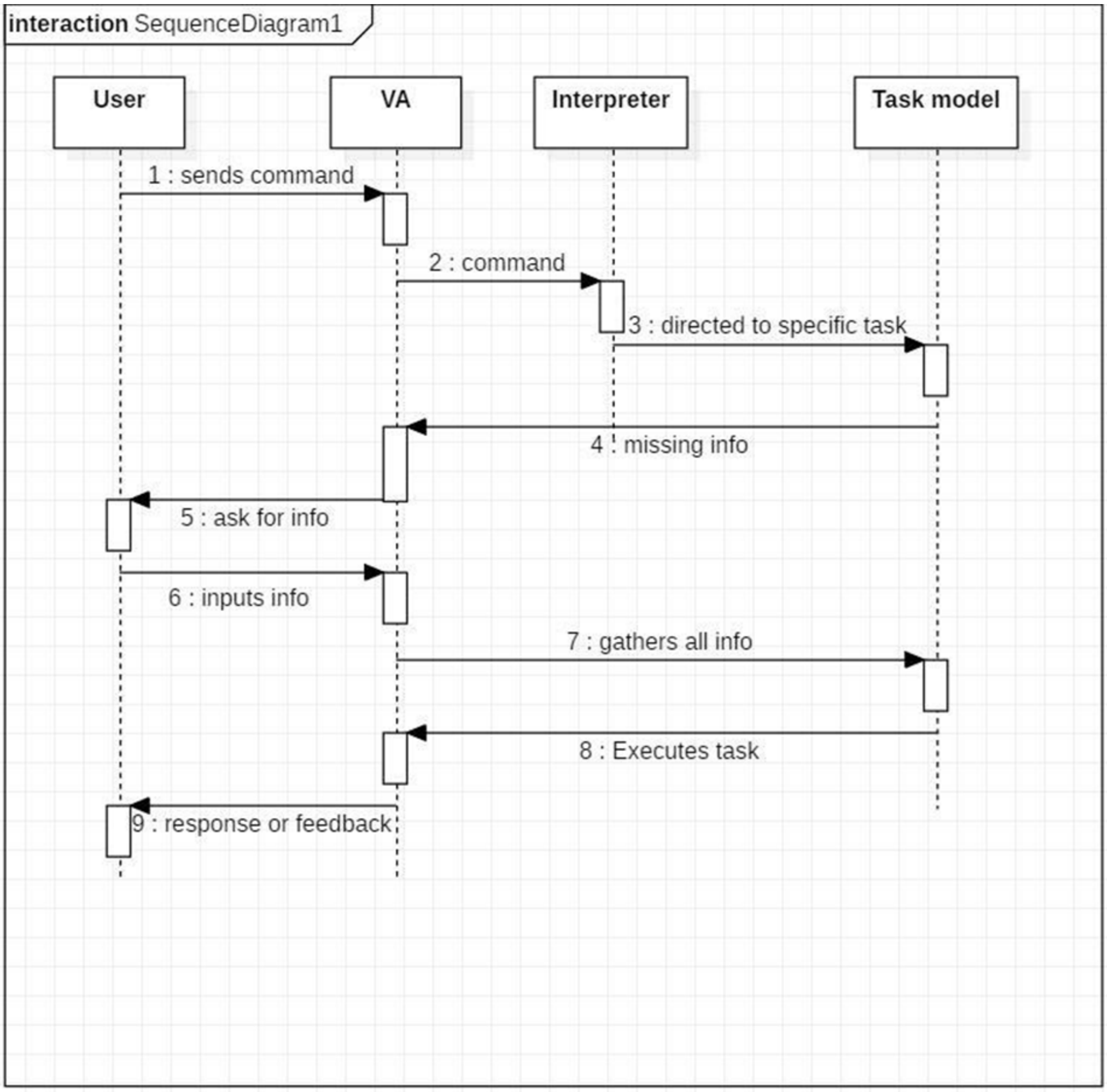
3.4 SEQUENCE DIAGRAM

3.4.1 Sequence diagram for Query-Response



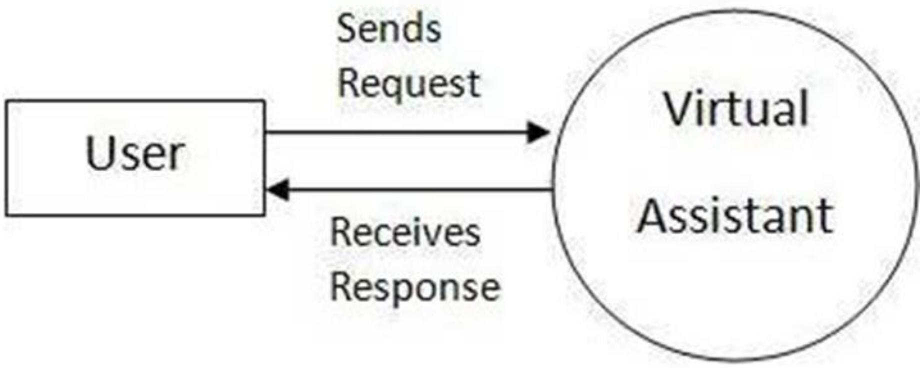
The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

3.4.2 Sequence diagram for Task Execution

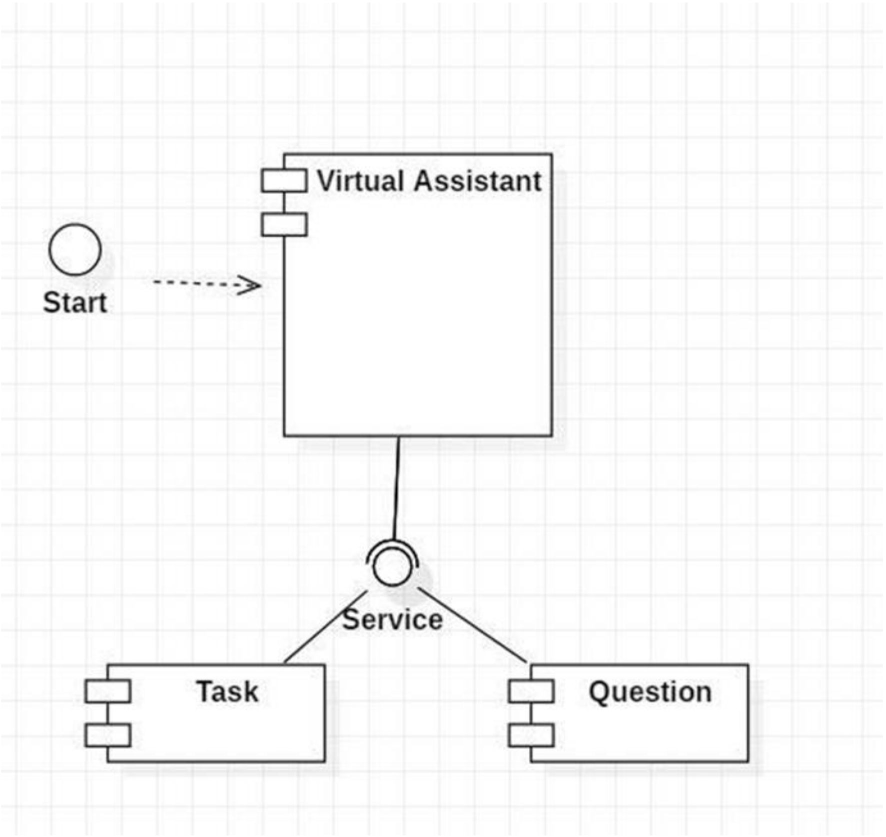


The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

3.5 DATA FLOW DIAGRAM



3.6 COMPONENT DIAGRAM



CHAPTER 4
IMPLEMENTATION

As the first step, install all the necessary packages and libraries. The command used to install the libraries is "pip install" and then import it. The necessary packages included are as follows:

4.1 LIBRARIES AND PACKAGES

pyttsx3: It is a python library which converts text to speech.

SpeechRecognition: It is a python module which converts speech to text.

Datetime: This library provides us the actual date and time.

Wikipedia: It is a python module for searching anything on Wikipedia.

Smtplib: Simple mail transfer protocol that allows us to send mails and to route mails between mail servers.

Pyjokes: It is a python libraries which contains lots of interesting jokes in it.

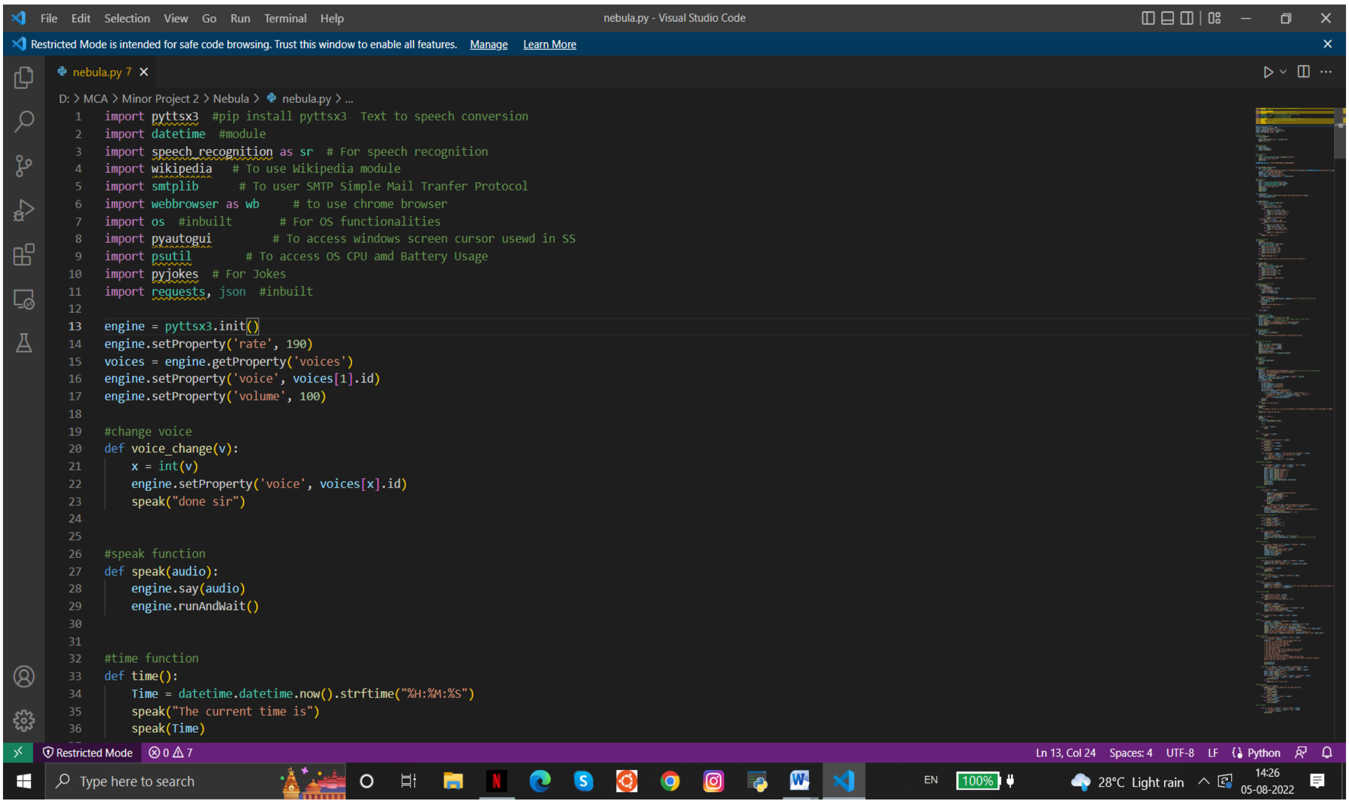
Webbrowser: It provides interface for displaying web-based documents to users.

Pyautogui: It is a python librariy for graphical user interface.

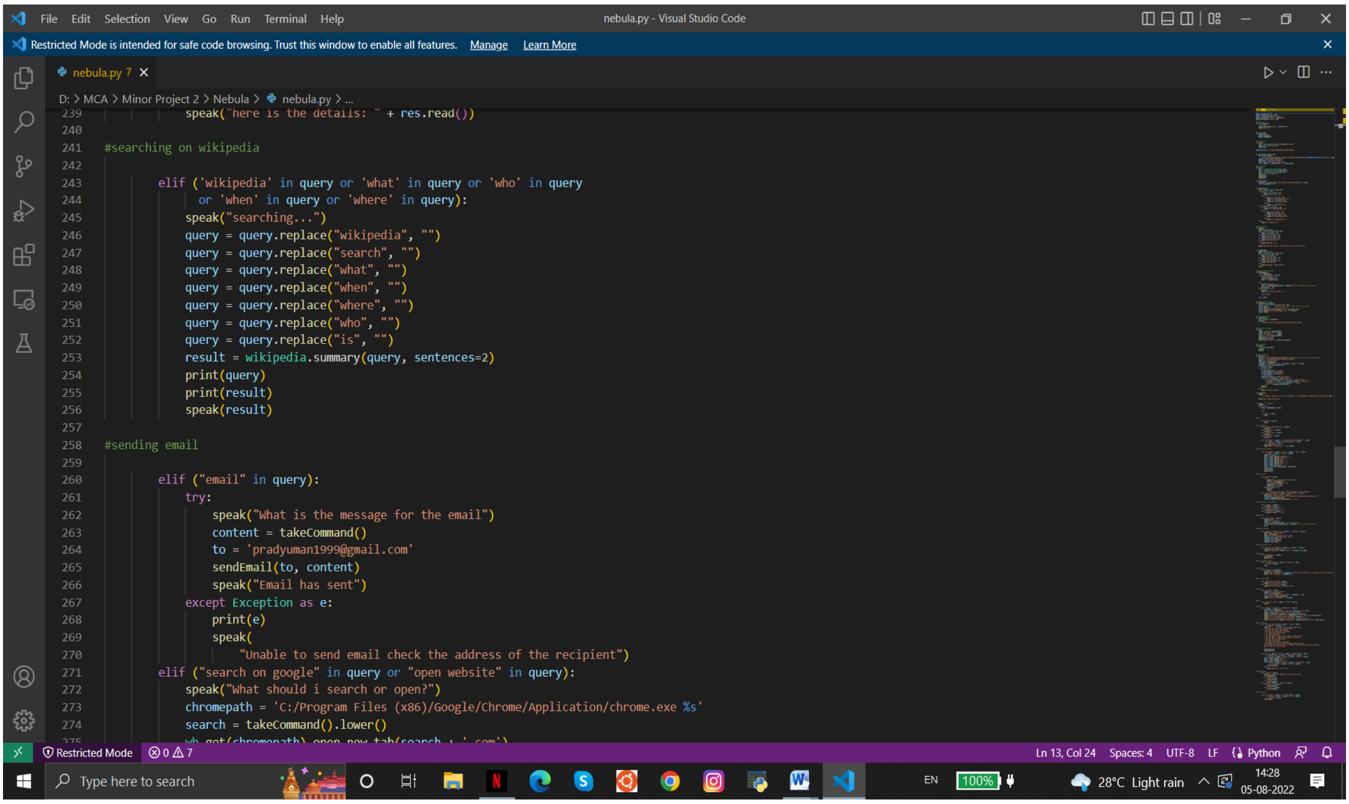
Psutil: It is a python library for getting CPU and battery usage.

os: It represents Operating System related functionality.

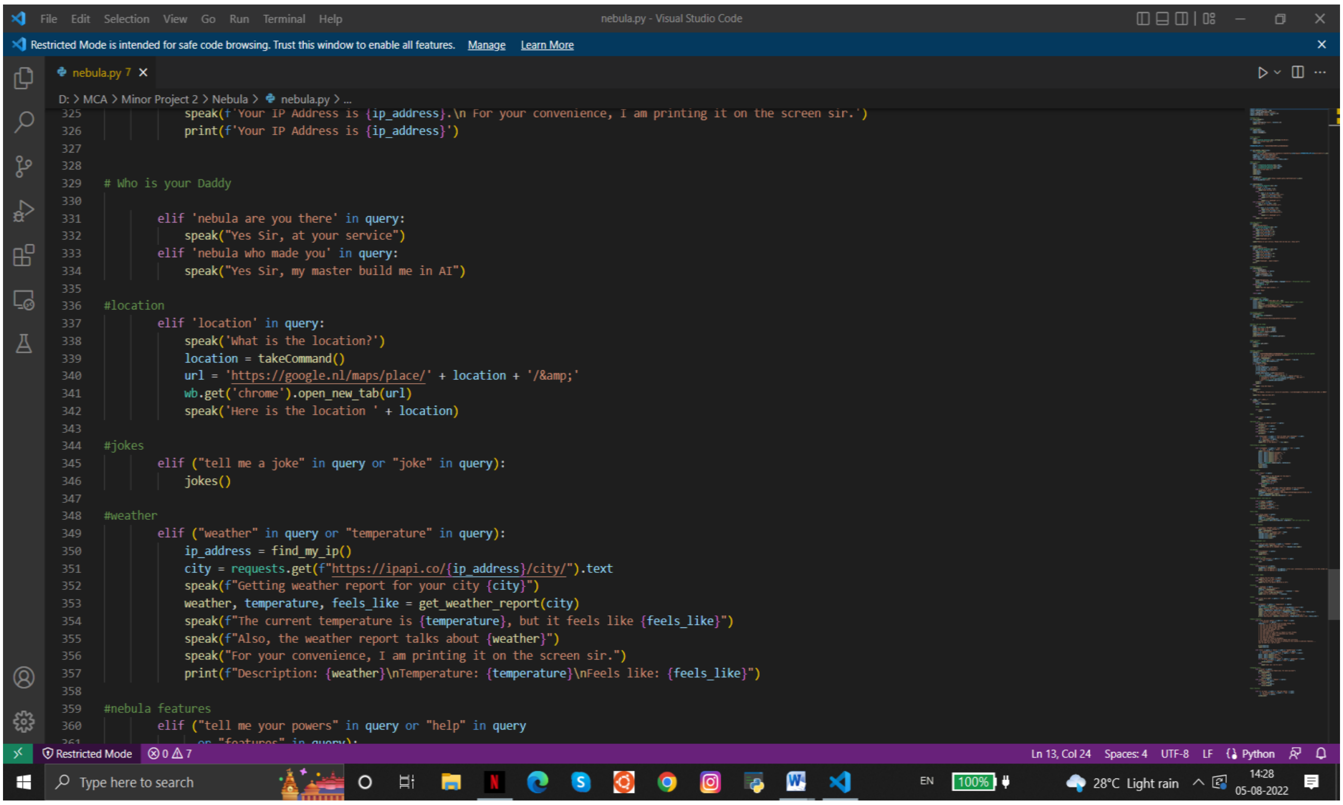
4.2 SNAPSHOTS OF CODES


File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
Restricted Mode
⊗ 0 ⚠ 7
Ln 13, Col 24 Spaces: 4 UTF-8 LF Python
Type here to search
EN 100%
28°C Light rain
14:26
05-08-2022

```
import pyttsx3  #pip install pyttsx3  Text to speech conversion
import datetime  #module
import speech_recognition as sr  # For speech recognition
import wikipedia   # To use Wikipedia module
import smtplib      # To user SMTP Simple Mail Tranfer Protocol
import webbrowser as wb     # to use chrome browser
import os  #inbuilt       # For OS functionalities
import pyautogui         # To access windows screen cursor usewd in SS
import psutil        # To access OS CPU amd Battery Usage
import pyjokes  # For Jokes
import requests, json  #inbuilt

engine = pyttsx3.init()
engine.setProperty('rate', 190)
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
engine.setProperty('volume', 100)

#change voice
def voice_change(v):
    x = int(v)
    engine.setProperty('voice', voices[x].id)
    speak("done sir")


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


#time function
def time():
    Time = datetime.datetime.now().strftime("%H:%M:%S")
    speak("The current time is")
    speak(Time)
```

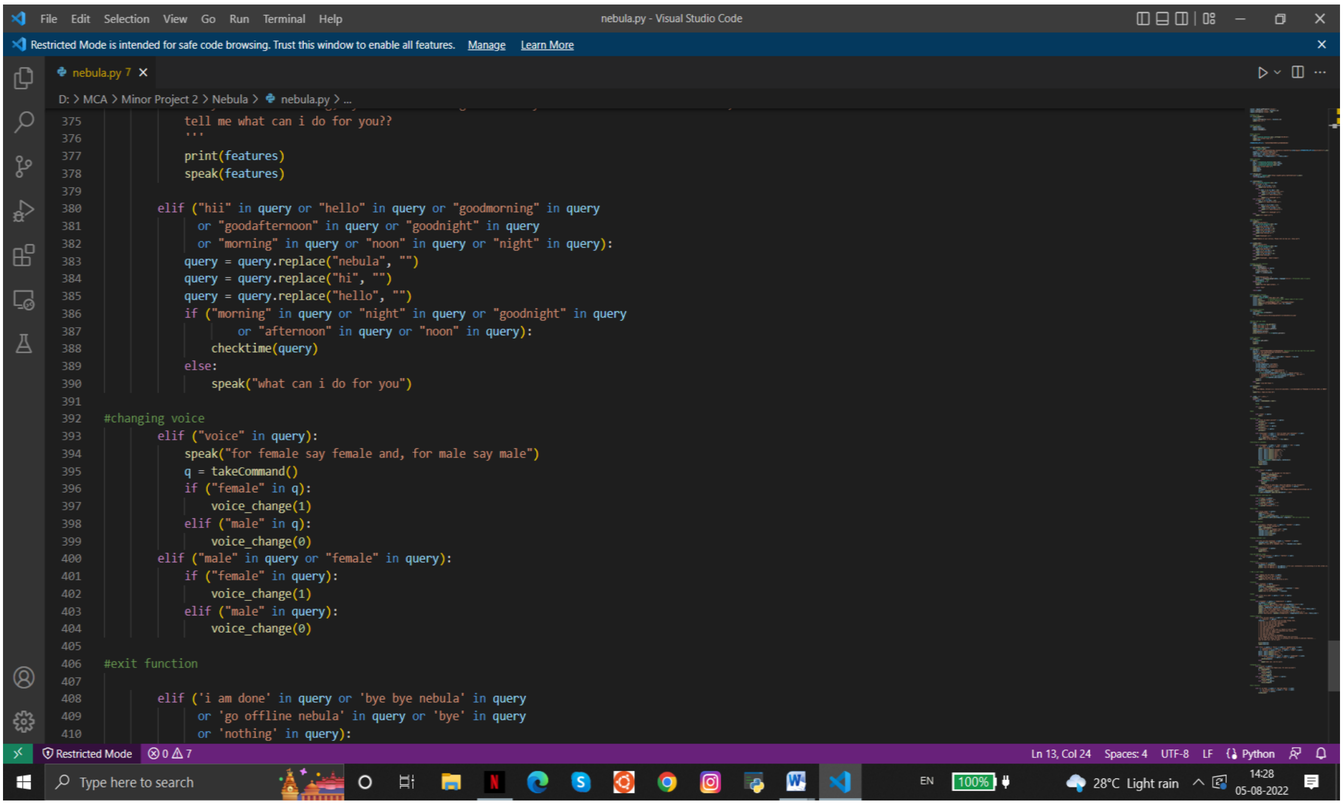

File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
Restricted Mode
⊗ 0 ⚠ 7
Ln 13, Col 24 Spaces: 4 UTF-8 LF Python
Type here to search
EN 100%
28°C Light rain
14:28
05-08-2022

```
            speak("here is the details: " + res.read())

#searching on wikipedia

        elif ('wikipedia' in query or 'what' in query or 'who' in query
              or 'when' in query or 'where' in query):
            speak("searching...")
            query = query.replace("wikipedia", "")
            query = query.replace("search", "")
            query = query.replace("what", "")
            query = query.replace("when", "")
            query = query.replace("where", "")
            query = query.replace("who", "")
            query = query.replace("is", "")
            result = wikipedia.summary(query, sentences=2)
            print(query)
            print(result)
            speak(result)

#sending email

        elif ("email" in query):
            try:
                speak("What is the message for the email")
                content = takeCommand()
                to = 'pradyuman1999@gmail.com'
                sendEmail(to, content)
                speak("Email has sent")
            except Exception as e:
                print(e)
                speak(
                    "Unable to send email check the address of the recipient")
        elif ("search on google" in query or "open website" in query):
            speak("What should i search or open?")
            chromepath = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
            search = takeCommand().lower()
            wb.get(chromepath).open_new_tab(search + '.com')
```



File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
Restricted Mode ⊗ 0 ⚠ 7
Ln 13, Col 24 Spaces: 4 UTF-8 LF Python
Type here to search
EN 100%
28°C Light rain
14:28
05-08-2022

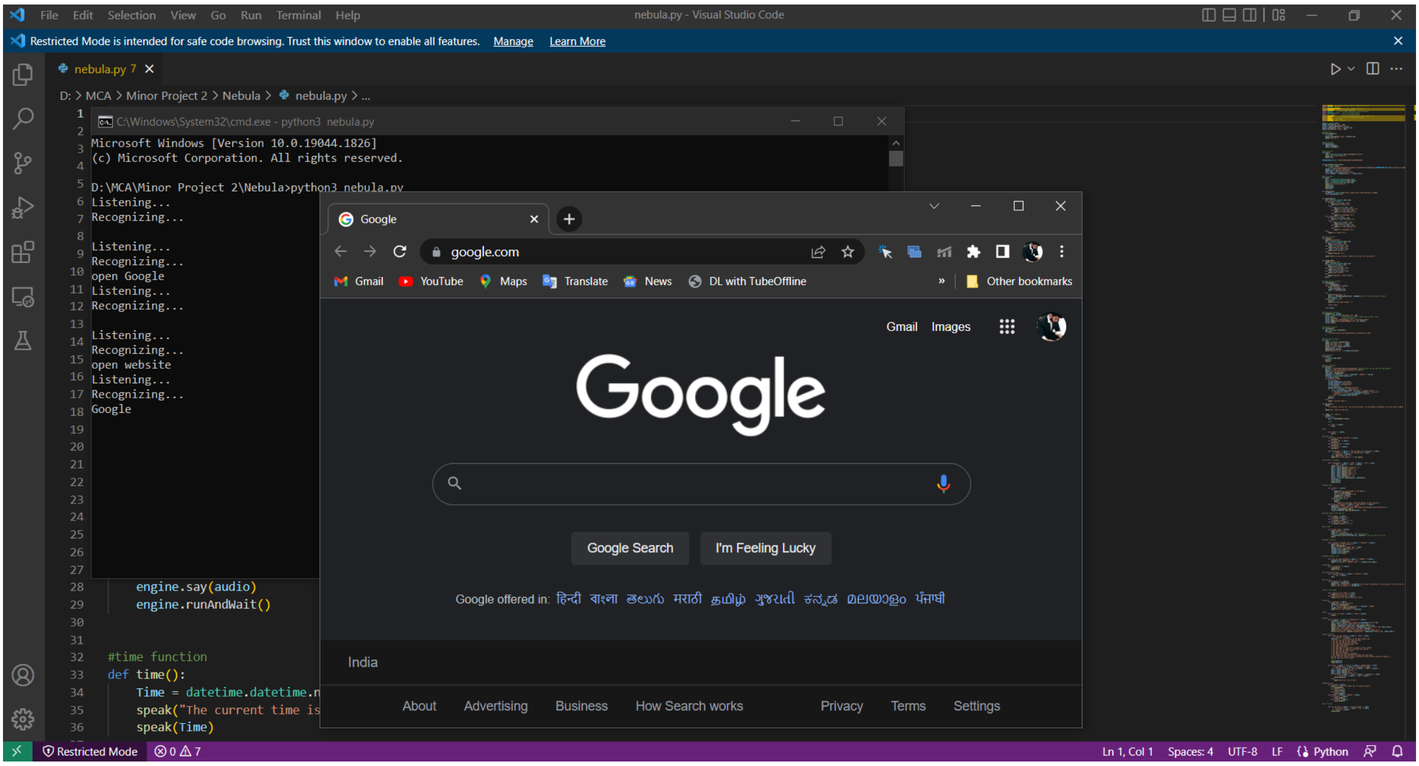
```python
            speak(f'Your IP Address is {ip_address}.\n For your convenience, I am printing it on the screen sir.')
            print(f'Your IP Address is {ip_address}')


# Who is your Daddy

        elif 'nebula are you there' in query:
            speak("Yes Sir, at your service")
        elif 'nebula who made you' in query:
            speak("Yes Sir, my master build me in AI")

#location
        elif 'location' in query:
            speak('What is the location?')
            location = takeCommand()
            url = 'https://google.nl/maps/place/' + location + '/&amp;'
            wb.get('chrome').open_new_tab(url)
            speak('Here is the location ' + location)

#jokes
        elif ("tell me a joke" in query or "joke" in query):
            jokes()

#weather
        elif ("weather" in query or "temperature" in query):
            ip_address = find_my_ip()
            city = requests.get(f"https://ipapi.co/{ip_address}/city/").text
            speak(f"Getting weather report for your city {city}")
            weather, temperature, feels_like = get_weather_report(city)
            speak(f"The current temperature is {temperature}, but it feels like {feels_like}")
            speak(f"Also, the weather report talks about {weather}")
            speak("For your convenience, I am printing it on the screen sir.")
            print(f"Description: {weather}\nTemperature: {temperature}\nFeels like: {feels_like}")

#nebula features
        elif ("tell me your powers" in query or "help" in query
              or "features" in query):
```



File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
Restricted Mode ⊗ 0 ⚠ 7
Ln 13, Col 24 Spaces: 4 UTF-8 LF Python
Type here to search
EN 100%
28°C Light rain
14:28
05-08-2022

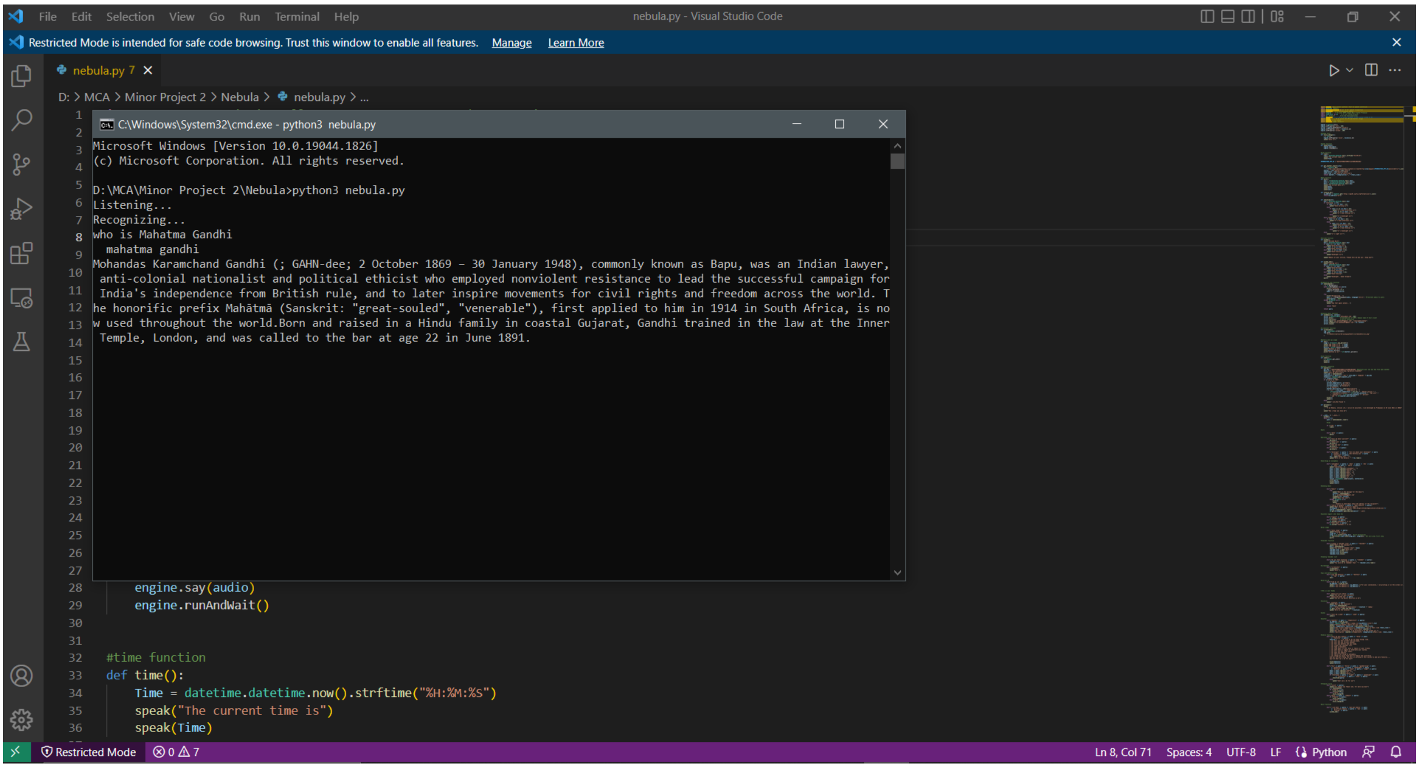
```python
            tell me what can i do for you??
            '''
            print(features)
            speak(features)

        elif ("hii" in query or "hello" in query or "goodmorning" in query
              or "goodafternoon" in query or "goodnight" in query
              or "morning" in query or "noon" in query or "night" in query):
            query = query.replace("nebula", "")
            query = query.replace("hi", "")
            query = query.replace("hello", "")
            if ("morning" in query or "night" in query or "goodnight" in query
                    or "afternoon" in query or "noon" in query):
                checktime(query)
            else:
                speak("what can i do for you")

#changing voice
        elif ("voice" in query):
            speak("for female say female and, for male say male")
            q = takeCommand()
            if ("female" in q):
                voice_change(1)
            elif ("male" in q):
                voice_change(0)
        elif ("male" in query or "female" in query):
            if ("female" in query):
                voice_change(1)
            elif ("male" in query):
                voice_change(0)

#exit function

        elif ('i am done' in query or 'bye bye nebula' in query
              or 'go offline nebula' in query or 'bye' in query
              or 'nothing' in query):
```

4.3 SNAPSHOTS


File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
C:\Windows\System32\cmd.exe - python3 nebula.py
Google
google.com
Gmail
YouTube
Maps
Translate
News
DL with TubeOffline
Other bookmarks
Gmail
Images
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
India
About
Advertising
Business
How Search works
Privacy
Terms
Settings
Restricted Mode
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Python

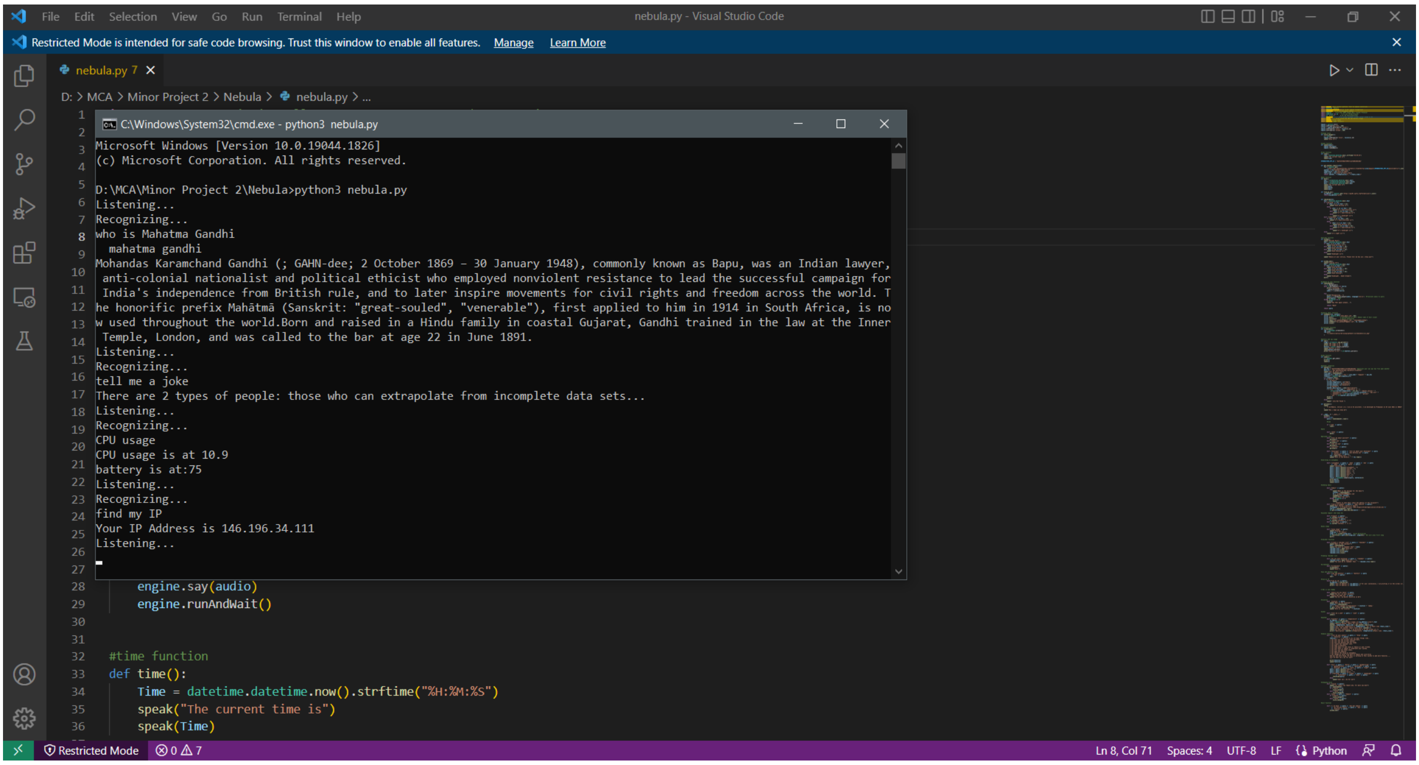
```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

D:\MCA\Minor Project 2\Nebula>python3 nebula.py
Listening...
Recognizing...

Listening...
Recognizing...
open Google
Listening...
Recognizing...

Listening...
Recognizing...
open website
Listening...
Recognizing...
Google
```

```
    engine.say(audio)
    engine.runAndWait()


#time function
def time():
    Time = datetime.datetime.n
    speak("The current time is
    speak(Time)
```


File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
C:\Windows\System32\cmd.exe - python3 nebula.py
Restricted Mode
Ln 8, Col 71
Spaces: 4
UTF-8
LF
Python

```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

D:\MCA\Minor Project 2\Nebula>python3 nebula.py
Listening...
Recognizing...
who is Mahatma Gandhi
  mahatma gandhi
Mohandas Karamchand Gandhi (; GAHN-dee; 2 October 1869 – 30 January 1948), commonly known as Bapu, was an Indian lawyer,
 anti-colonial nationalist and political ethicist who employed nonviolent resistance to lead the successful campaign for
 India's independence from British rule, and to later inspire movements for civil rights and freedom across the world. T
he honorific prefix Mahātmā (Sanskrit: "great-souled", "venerable"), first applied to him in 1914 in South Africa, is no
w used throughout the world.Born and raised in a Hindu family in coastal Gujarat, Gandhi trained in the law at the Inner
 Temple, London, and was called to the bar at age 22 in June 1891.
```

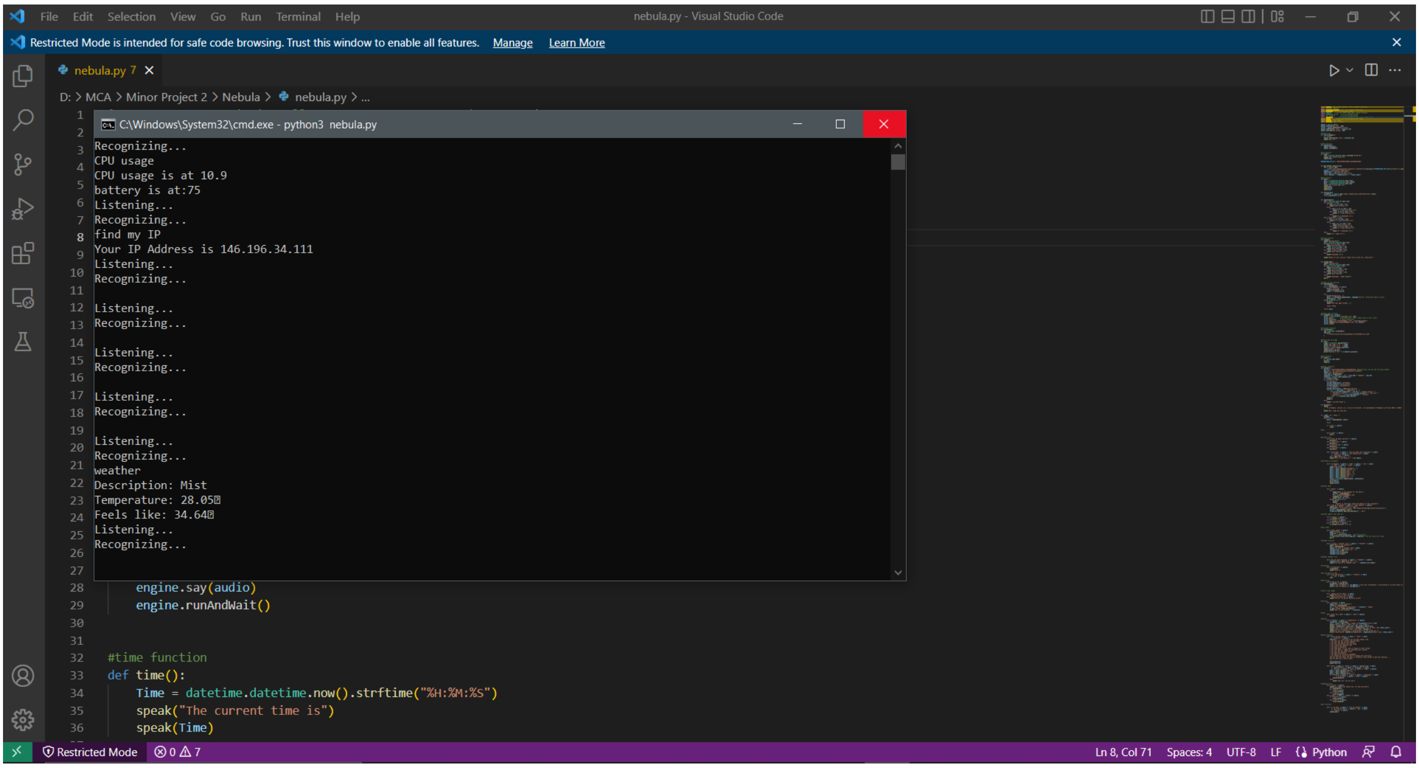
```
    engine.say(audio)
    engine.runAndWait()


#time function
def time():
    Time = datetime.datetime.now().strftime("%H:%M:%S")
    speak("The current time is")
    speak(Time)
```



File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
C:\Windows\System32\cmd.exe - python3 nebula.py
Restricted Mode ⊗ 0 ⚠ 7
Ln 8, Col 71 Spaces: 4 UTF-8 LF Python

```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

D:\MCA\Minor Project 2\Nebula>python3 nebula.py
Listening...
Recognizing...
who is Mahatma Gandhi
  mahatma gandhi
Mohandas Karamchand Gandhi (; GAHN-dee; 2 October 1869 – 30 January 1948), commonly known as Bapu, was an Indian lawyer,
 anti-colonial nationalist and political ethicist who employed nonviolent resistance to lead the successful campaign for
 India's independence from British rule, and to later inspire movements for civil rights and freedom across the world. T
he honorific prefix Mahātmā (Sanskrit: "great-souled", "venerable"), first applied to him in 1914 in South Africa, is no
w used throughout the world.Born and raised in a Hindu family in coastal Gujarat, Gandhi trained in the law at the Inner
 Temple, London, and was called to the bar at age 22 in June 1891.
Listening...
Recognizing...
tell me a joke
There are 2 types of people: those who can extrapolate from incomplete data sets...
Listening...
Recognizing...
CPU usage
CPU usage is at 10.9
battery is at:75
Listening...
Recognizing...
find my IP
Your IP Address is 146.196.34.111
Listening...
```

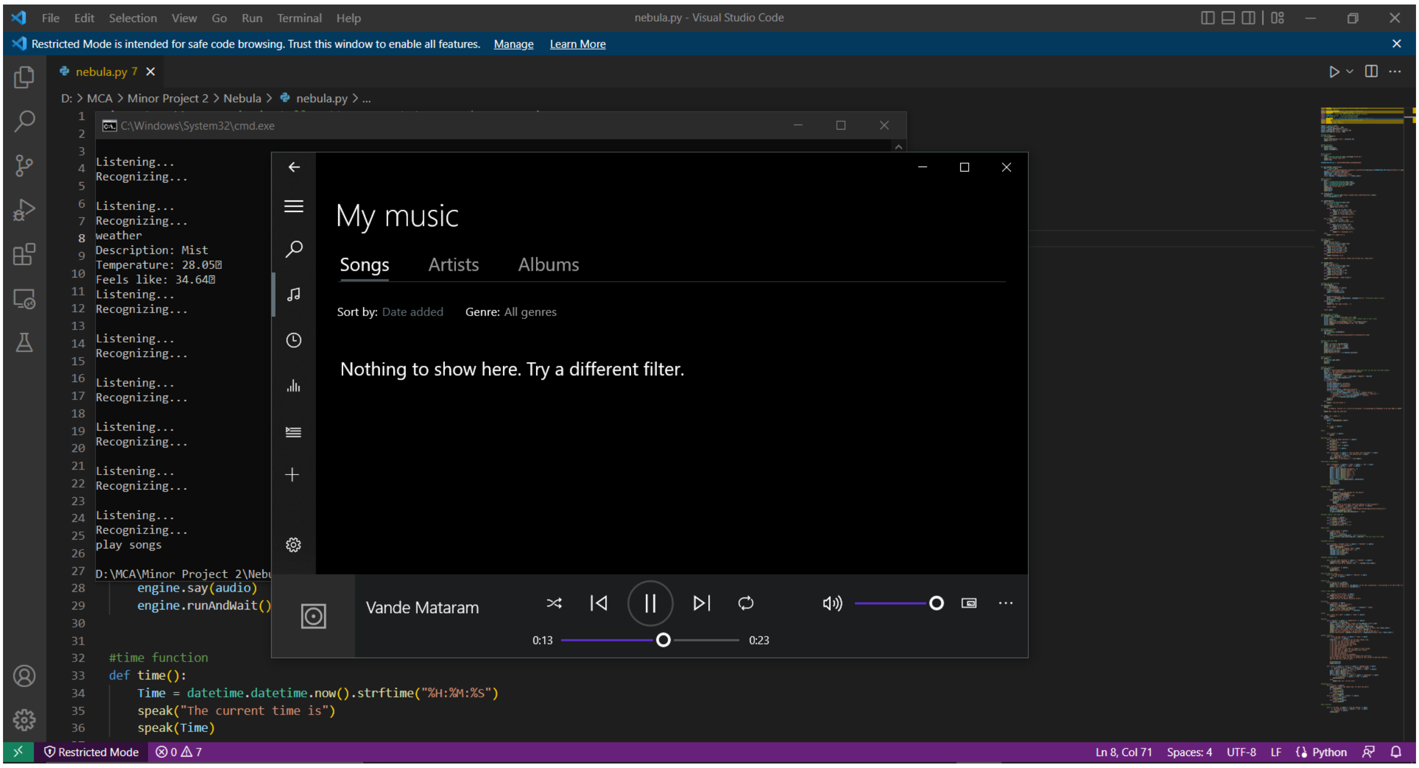
```python
    engine.say(audio)
    engine.runAndWait()


#time function
def time():
    Time = datetime.datetime.now().strftime("%H:%M:%S")
    speak("The current time is")
    speak(Time)
```



File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
C:\Windows\System32\cmd.exe - python3 nebula.py
Restricted Mode ⊗ 0 ⚠ 7
Ln 8, Col 71 Spaces: 4 UTF-8 LF Python

```
Recognizing...
CPU usage
CPU usage is at 10.9
battery is at:75
Listening...
Recognizing...
find my IP
Your IP Address is 146.196.34.111
Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...
weather
Description: Mist
Temperature: 28.05□
Feels like: 34.64□
Listening...
Recognizing...
```

```python
    engine.say(audio)
    engine.runAndWait()


#time function
def time():
    Time = datetime.datetime.now().strftime("%H:%M:%S")
    speak("The current time is")
    speak(Time)
```



File Edit Selection View Go Run Terminal Help
nebula.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
nebula.py 7
D: > MCA > Minor Project 2 > Nebula > nebula.py > ...
C:\Windows\System32\cmd.exe
My music
Songs
Artists
Albums
Sort by: Date added
Genre: All genres
Nothing to show here. Try a different filter.
Vande Mataram
0:13
0:23
Restricted Mode
⊗ 0 ⚠ 7
Ln 8, Col 71
Spaces: 4
UTF-8
LF
Python

```
Listening...
Recognizing...

Listening...
Recognizing...
weather
Description: Mist
Temperature: 28.05□
Feels like: 34.64□
Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...

Listening...
Recognizing...
play songs

D:\MCA\Minor Project 2\Nebu
```

```
        engine.say(audio)
        engine.runAndWait()


    #time function
    def time():
        Time = datetime.datetime.now().strftime("%H:%M:%S")
        speak("The current time is")
        speak(Time)
```

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

5.1 Conclusion:

NEBULA is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below:

5.2 LIMITATIONS

Security is somewhere an issue, there is no voice command encryption in this project.

- Background voice can interfere
- Misinterpretation because of accents and may cause inaccurate results.
- NEBULA cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!”

5.3 SCOPE FOR FUTURE WORK

- Make NEBULA to learn more on its own and develop a new skill in it.
- NEBULA android app can also be developed.
- Make more Nebula voice terminals.
- Voice commands can be encrypted to maintain security.

CHAPTER 6

REFERENCES

References and Websites:

- **Websites referred**
  - www.stackoverflow.com
  - www.pythonprogramming.net
  - www.codecademy.com
  - www.tutorialspoint.com
  - www.google.co.in
- **YouTube Channels referred**
  - CS Dojo
  - edureka!