# VIRTUAL ASSISTANT FOR DESKTOP

**END TERM REPORT**

**BY**

**SURAJ PATIL, ARYAN RUSIA, KAUSHAMBHA GUHA ROY, TUSHAR KUMAR SINGH**

(Section: K18JF)

(Roll Number: 50, 53, 59, 43)

**Department of Intelligent Systems**

**School of Computer Science Engineering**

**Lovely Professional University, Jalandhar**

**April-2020**

# Student Declaration

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged.
I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

SURAJ PATIL

Roll Number: 50

ARYAN RUSIA

Roll Number: 53

KAUSHAMBHA GUHA ROY

Roll Number: 59

TUSHAR KUMAR SINGH

Roll Number: 43

Place- Lovely Professional University

Date- 13-04-2020

# BONAFIDE CERTIFICATE

**Certified that this project report "VIRTUAL ASSISTANT FOR DESKTOP" is the bonafide work of "SURAJ PATIL, ARYAN RUSIA, KAUSHAMBHA GUHA ROY, TUSHAR KUMAR SINGH" who carried out the project work under my supervision.**

**MR. DIPEN SAINI**

**23681**

**Department of Intelligent Systems**

# Project Description

A virtual assistant, also called AI assistant or digital assistant, is an application program that understands natural language voice commands and completes tasks for the user.
Such tasks, historically performed by a personal assistant or secretary, include taking dictation, reading text or email messages aloud, looking up phone numbers, scheduling, placing phone calls and reminding the end user about appointments. Popular virtual assistants currently include Amazon Alexa, Apple's Siri, Google Now and Microsoft's Cortana -- the digital assistant built into Windows Phone 8.1 and Windows 10.

# Introduction

A virtual assistant is a technology based on artificial intelligence. The software uses a device’s microphone to receive voice requests while the voice output takes place at the speaker. But the most exciting thing happens between these two actions.
It is a combination of several different technologies: voice recognition, voice analysis and language processing.

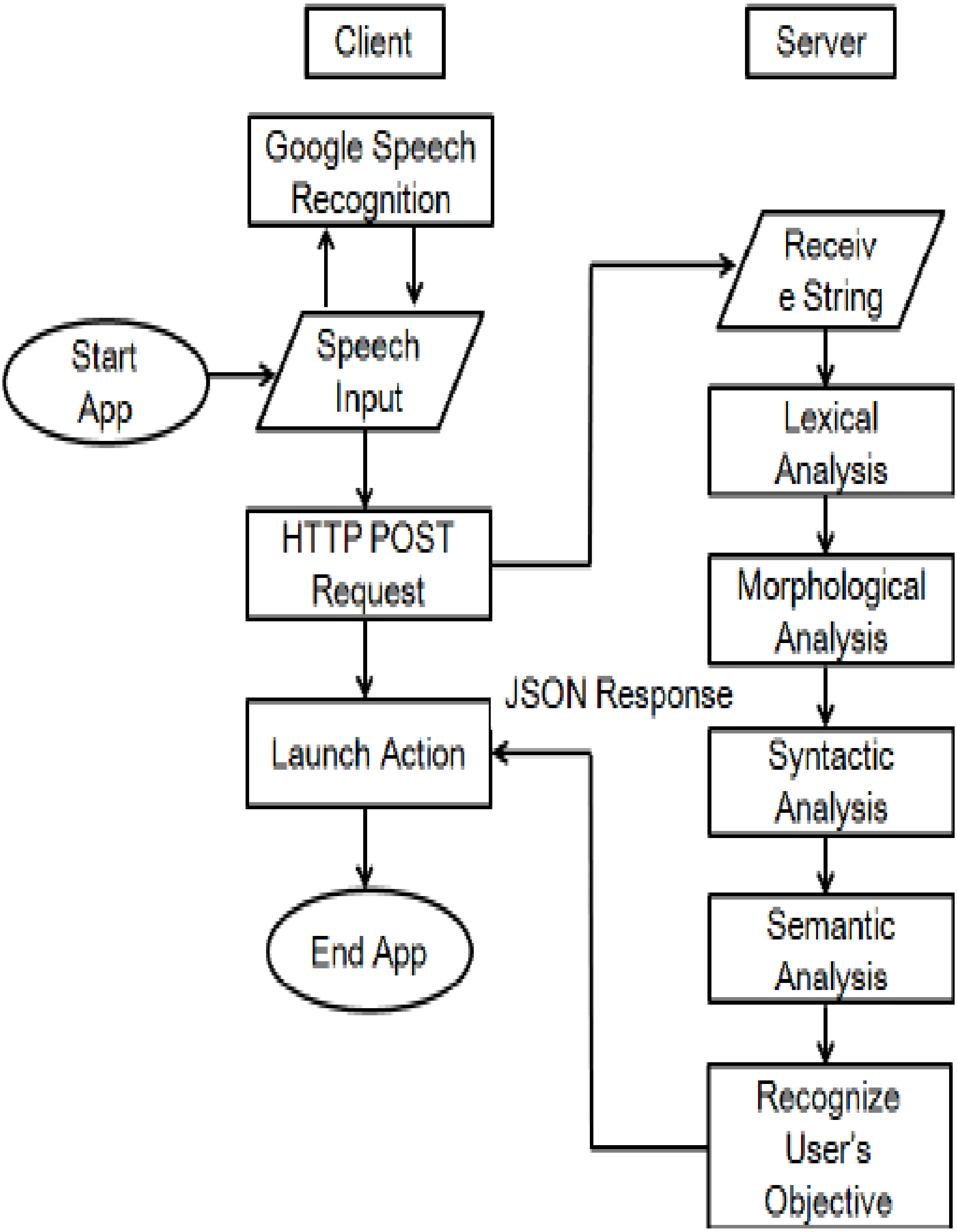
When a user asks a personal assistant to perform a task, the natural language audio signal is converted into digital data that can be analyzed by the software. Then this data is compared with a database of the software using an innovative algorithm to find a suitable answer. This database is located on distributed servers in cloud networks. For this reason, most personal assistants cannot work without a reliable Internet connection.

With the increasing number of queries the software’s database gets expanded and optimized, which improves voice recognition and increases the response time of the system. Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices.

There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as "what is the weather today," or "play pop music." Those requests are processed and stored in Amazon's cloud.

The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, natural language processing and speech recognition platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated algorithms to learn from data input and become better at predicting the end user's needs. Virtual assistants are quickly evolving to provide more capabilities and value to users. As speech recognition and natural language processing advances, so too will a virtual assistant's ability to understand and perform requests. And as voice recognition technology improves, virtual assistant use will move deeper into business workflows.

# FLOWCHART OF VIRTUAL ASSISTANT



# Contribution

1. Suraj Patil – Coded whole program. Make the synopsis.
2. Aryan Rusia- Help in finding data of the project and synopsis and also suggesting to Do and Don't.
3. Kaushambha Guha Roy- Help in finding the data of project and synopsis and also suggesting to Do and Don't.
4. Tushar Kumar Singh- Help in finding data of the project and synopsis and also suggesting to Do and Don't.

# Technologies and Framework

- Project wascreated using python programming language.
- We used different python libraries such as speech_recognizer for speech recognition, pyttsx for voice driver which are embedded in device and many more…
- Giving output according to query said by user
- NLP is used in in this program i.e. first voice get convert to text then vice versa.

# SWOT Analysis

- **Strength-** Program is autonomous to give output for particular input by user over voice.
- **Weakness-** According to internet connectivity program will execute, so there will be problem in some cases.
- **Opportunities-** Others will gets a good idea about new and advance strategies from this.
- **Threats-** There sometimes mismatch of word is possible so desired output will not get.

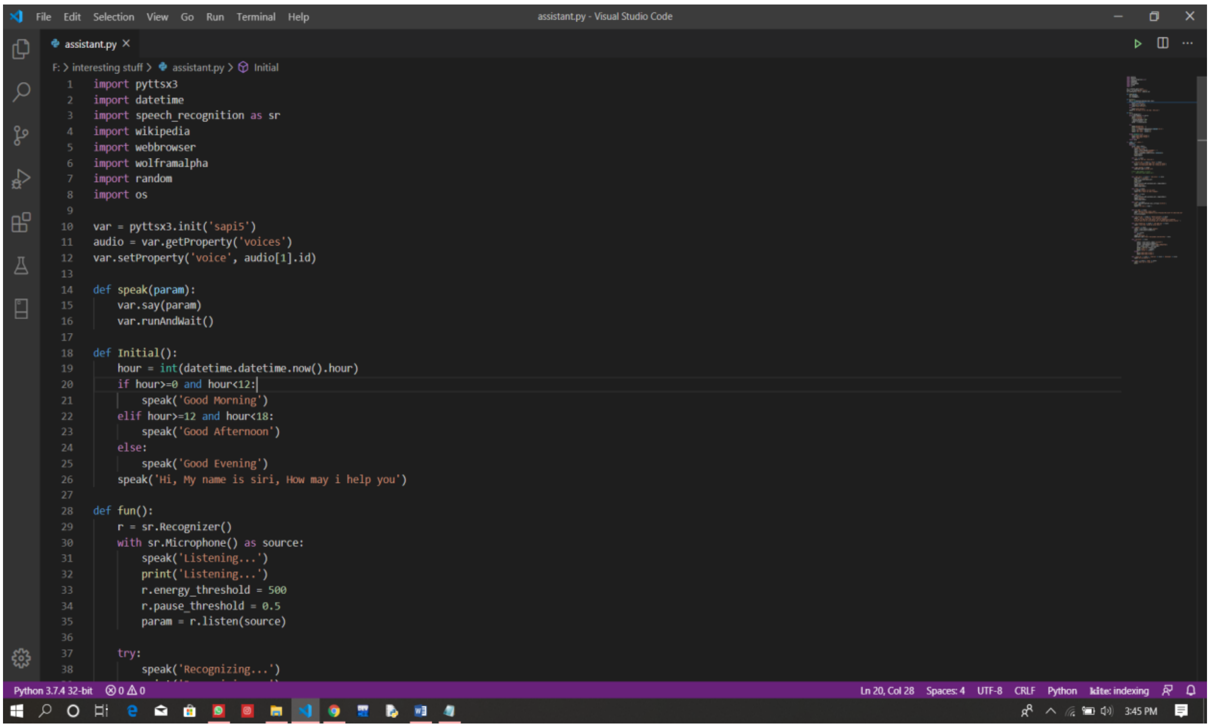
# Program

```
import pyttsx3
import datetime
import speech_recognition as sr
import wikipedia
import webbrowser
import wolframalpha
import random
import os

var = pyttsx3.init('sapi5')
audio = var.getProperty('voices')
var.setProperty('voice', audio[1].id)

def speak(param):
    var.say(param)
    var.runAndWait()

def Initial():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak('Good Morning')
    elif hour>=12 and hour<18:
        speak('Good Afternoon')
    else:
        speak('Good Evening')
    speak('Hi, My name is siri, How may i help you')

def fun():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        speak('Listening...')
        print('Listening...')
        r.energy_threshold = 500
        r.pause_threshold = 0.5
        param = r.listen(source)

    try:
        speak('Recognizing...')
        print('Recognizing...')
        voice = r.recognize_google(param,language='en-in')
        print(f'User Said : {voice}\n')
        speak(f'User Said : {voice}\n')

    except Exception as e:
        print('Speak Again Please!')
        speak('Speak Again Please!')
        return 'None'
    return voice
```

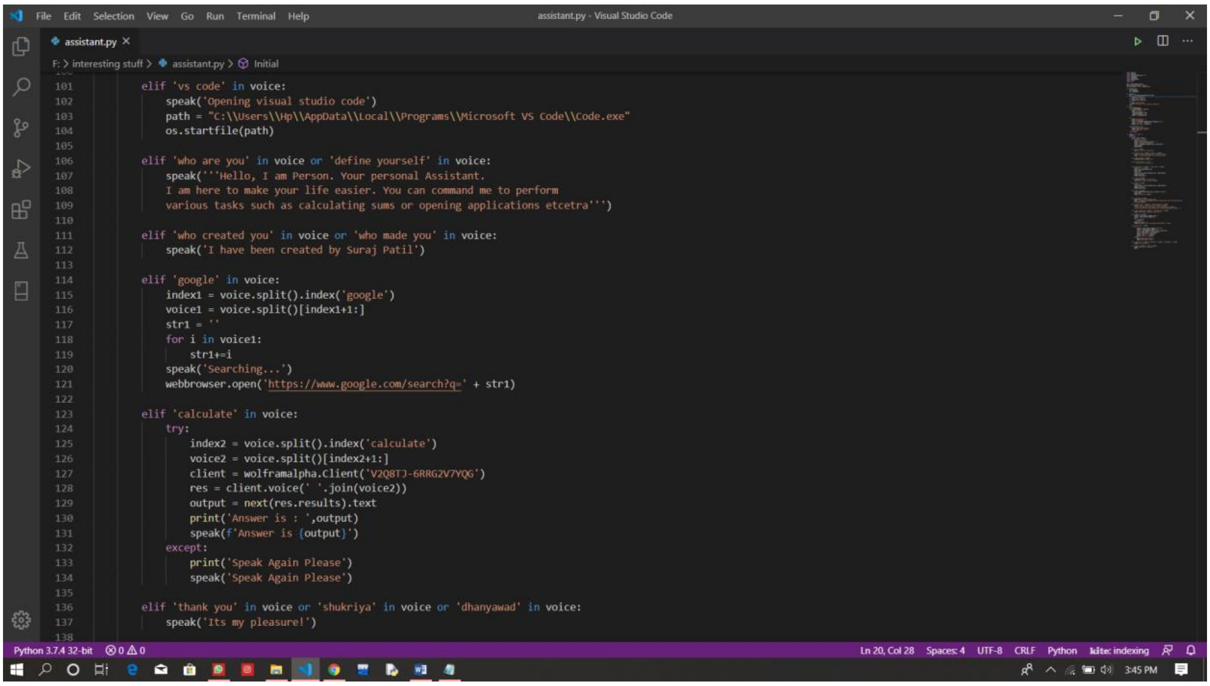
```
if __name__ == "__main__":
    Initial()
    while True:
        voice = fun().lower()
        if 'wikipedia' in voice:
            speak('Searching...')
            voice = voice.replace('wikipedia','')
            speak('According to Wkipedia')
            result = wikipedia.summary(voice, sentences=2)
            print(result)
            speak(result)

        elif 'hi' in voice:
            speak('Hi, How can I help you')

        elif 'how are you' in voice or 'kaisi' in voice:
            print('I am doing great,What can I help you with?')
            speak('I am doing great,What can I help you with?')

        elif 'open youtube' in voice:
            webbrowser.open('youtube.com')

        # elif 'open google' in voice:
        #     webbrowser.open('google.com')


        elif 'open music' in voice or 'play music' in voice:
            music_dir = 'F:\\music'
            songs = os.listdir(music_dir)
            print(songs)
            index = 0
            os.startfile(os.path.join(music_dir, songs[index]))
            speak('Playing')
            speak(songs[index])

        elif 'stop' in voice:
            os.system("taskkill /im vlc.exe")
            speak('Music Player has been stopped')

        elif 'next' in voice:
            index += 1
            os.startfile(os.path.join(music_dir, songs[index]))
            speak('Playing')
            speak(songs[index])

        elif 'time' in voice:
            time = datetime.datetime.now().strftime('%H:%M:%S')
            print(time)
            speak(f'The time is {time}')
```

```
elif 'vs code' in voice:
    speak('Opening visual studio code')
    path = "C:\\Users\\Hp\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
    os.startfile(path)

elif 'who are you' in voice or 'define yourself' in voice:
    speak('''Hello, I am Person. Your personal Assistant.
    I am here to make your life easier. You can command me to perform
    various tasks such as calculating sums or opening applications etcetra''')

elif 'who created you' in voice or 'who made you' in voice:
    speak('I have been created by Suraj Patil')

elif 'google' in voice:
    index1 = voice.split().index('google')
    voice1 = voice.split()[index1+1:]
    str1 = ''
    for i in voice1:
        str1+=i
    speak('Searching...')
    webbrowser.open('https://www.google.com/search?q=' + str1)

elif 'calculate' in voice:
    try:
        index2 = voice.split().index('calculate')
        voice2 = voice.split()[index2+1:]
        client = wolframalpha.Client('V2Q8TJ-6RRG2V7YQG')
        res = client.voice(' '.join(voice2))
        output = next(res.results).text
        print('Answer is : ',output)
        speak(f'Answer is {output}')
    except:
        print('Speak Again Please')
        speak('Speak Again Please')

elif 'thank you' in voice or 'shukriya' in voice or 'dhanyawad' in voice:
    speak('Its my pleasure!')

elif 'quit' in voice or 'band' in voice:
    speak('Thank you for using me')
    exit()
```

# Code Screenshots:

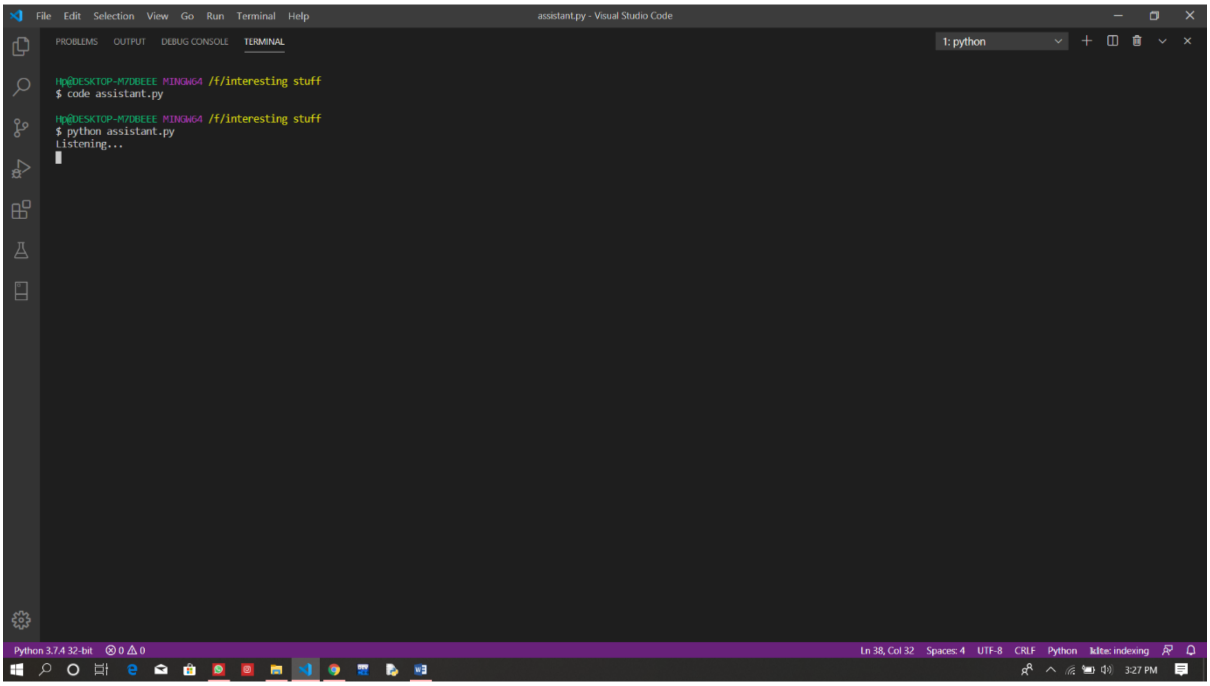


```python
import pyttsx3
import datetime
import speech_recognition as sr
import wikipedia
import webbrowser
import wolframalpha
import random
import os

var = pyttsx3.init('sapi5')
audio = var.getProperty('voices')
var.setProperty('voice', audio[1].id)

def speak(param):
    var.say(param)
    var.runAndWait()

def Initial():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak('Good Morning')
    elif hour>=12 and hour<18:
        speak('Good Afternoon')
    else:
        speak('Good Evening')
    speak('Hi, My name is siri, How may i help you')

def fun():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        speak('Listening...')
        print('Listening...')
        r.energy_threshold = 500
        r.pause_threshold = 0.5
        param = r.listen(source)

    try:
        speak('Recognizing...')
```

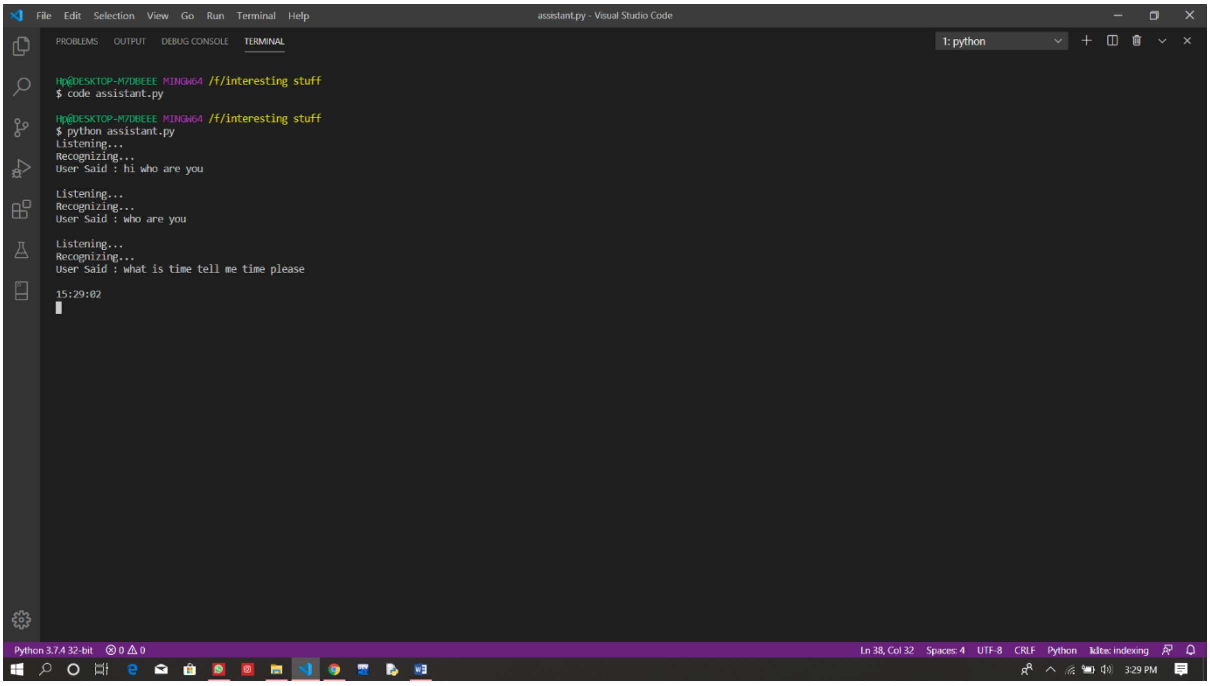


```python
        elif 'vs code' in voice:
            speak('Opening visual studio code')
            path = "C:\\Users\\Hp\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"
            os.startfile(path)

        elif 'who are you' in voice or 'define yourself' in voice:
            speak('''Hello, I am Person. Your personal Assistant.
            I am here to make your life easier. You can command me to perform
            various tasks such as calculating sums or opening applications etcetra''')

        elif 'who created you' in voice or 'who made you' in voice:
            speak('I have been created by Suraj Patil')

        elif 'google' in voice:
            index1 = voice.split().index('google')
            voice1 = voice.split()[index1+1:]
            str1 = ''
            for i in voice1:
                str1+=i
            speak('Searching...')
            webbrowser.open('https://www.google.com/search?q=' + str1)

        elif 'calculate' in voice:
            try:
                index2 = voice.split().index('calculate')
                voice2 = voice.split()[index2+1:]
                client = wolframalpha.Client('V2Q8TJ-6RRG2V7YQG')
                res = client.voice(' '.join(voice2))
                output = next(res.results).text
                print('Answer is : ',output)
                speak(f'Answer is {output}')
            except:
                print('Speak Again Please')
                speak('Speak Again Please')

        elif 'thank you' in voice or 'shukriya' in voice or 'dhanyawad' in voice:
            speak('Its my pleasure!')
```

## Output:



```
Hp@DESKTOP-M7DBEEE MINGW64 /f/interesting stuff
$ code assistant.py

Hp@DESKTOP-M7DBEEE MINGW64 /f/interesting stuff
$ python assistant.py
Listening...
```



```
Hp@DESKTOP-M7DBEEE MINGW64 /f/interesting stuff
$ code assistant.py

Hp@DESKTOP-M7DBEEE MINGW64 /f/interesting stuff
$ python assistant.py
Listening...
Recognizing...
User Said : hi who are you

Listening...
Recognizing...
User Said : who are you

Listening...
Recognizing...
User Said : what is time tell me time please

15:29:02
```