



Final Report word

Project Work (Panjab University)

Voice Assistant using Python

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION

Submitted by

Vaibhav Kaura

(Roll Number: UE195126)

K Sanju Devi

(Roll Number: UE195139)

UNDER THE GUIDANCE OF

Dr. Garima Joshi

(Assistant Professor, ECE, UIET)



**UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY
(PANJAB UNIVERSITY: CHANDIGARH (160025**

March-June 2022

DECLARATION

We hereby certify that the work which is being presented in this project entitled “Voice Assistant using Python”, in partial fulfilment of the requirement for the award of degree of “Bachelor of Engineering in Electronics and Communication” submitted in University Institute of Engineering and Technology, Panjab University, Chandigarh, is an authentic record of our own work carried out .under the supervision of Dr. Garima Joshi, Assistant Professor, ECE UIET

Place: Chandigarh

Date: 5th July, 2022

(Vaibhav Kaura(UE195126

(K Sanju Devi(UE195139

ABSTRACT

The Most famous application of iPhone is “SIRI” which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is “Google Voice Search” which is used for in Android Phones. But this Application mostly works with Internet Connections. But our Proposed System has capability to work with and without Internet Connectivity. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices. The system is being designed in such a way that all the services provided by the mobile devices are accessible by the end user on the user's .voice commands

Keywords: SIRI, Google Voice Search, Mobile Device, Internet

ACKNOWLEDGEMENT

In completing this project report on project titled VIRTUAL ASSISTANT, I had to take the help and .guideline of a few respected people, who deserve my greatest gratitude

The completion of this project report gives me much Pleasure. I would like to show my gratitude to Dr. Garima mam and Neeraj sir for giving me a good guideline for project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and .indirectly guided us in writing this project report

Many people, especially my classmates and friends themselves, have made valuable comments and suggestions on this proposal which gave me inspiration to improve my project. Here I thank all the .people for their help directly and indirectly to complete this project report

Table of Contents

<i>Chapter No.</i>	<i>TITLE</i>	<i>Page NO.</i>
--------------------	--------------	-----------------

	Declaration	2
	Abstract	3
	Acknowledgement	4
1	Introduction	7
1.1	Background	8
1.2	SIRI from Apple	8
1.3	Supported Tasks(SIRI)	8
1.4	Drawbacks	9
1.5	ReQall	9
1.6	Supported Tasks	9
1.7	Drawbacks	9
1.8	Objectives	10
2	Purpose, Scope and Applicability	11
2.1	Purpose	11
2.2	Scope	11
2.3	Applicability	11
	Survey of Technologies	12
2.4	Python	12
2.5	Pytttsx	13
2.6	Speech Recognition	13
3	Requirement and Analysis	15
3.1	Problem Definition	15
3.2	Feasibility Study	16
3.3	Technical Feasibility	16
3.4	Operation Feasibility	16

3.5	Economic Feasibility	17
3.6	Organisational Feasibility	17
3.7	Cultural Feasibility	17
3.8	Hardware Requirements	18
3.9	Software Requirements	18
4	Implementation and Results	19
4.1	Code	19
4.2	Outputs	23
4.2.1	Surfing web	23
4.2.2	Setting a timer	24
4.2.3	Reading Articles	24
4.2.4	Playing songs	25
4.2.5	Doing interactive activities	25
4.2.6	Searching on google fast	26
4.2.7	Open Applications	27
	Conclusion	29
	References	30

CHAPTER - 1

INTRODUCTION

In today's era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your fingertips. These days

we aren't even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, "I'll be late today." And the text is sent. That is the task of a Voice Assistant. It also supports specialized task such as booking a flight, or finding cheapest book online from various ecommerce sites and then providing an interface to book an order are helping automate search, .discovery and online order operations

Voice Assistants are software programs that help you ease your day-to-day tasks, such as showing weather report, creating reminders, making

shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and .Microsoft's Cortana

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing

.information from online sources to the user. It is effortless to use

Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer

and the analysts are already predicting that 50% of searches will be via voice by 2020. Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver .personalized responses

This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making .intelligent decisions for routine user activities

BACKGROUND 1.1

There already exist a number of desktop virtual assistants. A few examples of Current virtual assistants available in market are discussed in this section along with the tasks they can provide and .their drawbacks

SIRI from Apple 1.2

SIRI is personal assistant software that interfaces with the user through voice interface, recognizes commands and acts on them. It learns to adapt to user's Speech and thus, improves voice recognition over time. It also tries to converse with the user when it does not identify the user request. it integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task Based .contexts, to personalize the agent behaviour specifically, to the user at a given point of Time

:Supported Tasks by SIRI 1.3

Call someone from my contacts list *Launch an application on my iPhone * Send a text message * to someone

Set up a meeting on my calendar for 9am tomorrow *Set an alarm for 5am tomorrow morning*

Play a specific song in my iTunes library*

Enter a new note*

:Drawback 1.4

SIRI does not maintain a knowledge database of its own and its understanding .comes from the information captured in domain models and data models

ReQall 1.5

ReQall is personal assistant software that runs on smartphones running Apple iOS or Google Android operating system. It helps user to recall notes as well as tasks within a location and time context. It records user inputs and converts them into commands, and monitors current stack of user tasks to proactively suggest actions while considering any changes in the environment. It also presents information based on the context of the user, as well as filter information to the user based on its learned understanding of the priority of that information

Supported Tasks 1.6

Reminders*

Email*

Calendar, Google Calendar*

Outlook*

Evernote*

Facebook, LinkedIn*

News Feeds*

Drawback 1.7

Will take some time to put all of the to-do items in – you could spend more
.time putting the entries in than actually doing the revision

OBJECTIVES 1.8

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may

have. This may be done in a business environment, for example, on the business website, with a chat interface. Virtual assistants can tremendously save you time. We spend hours in online research .and then making the report in our terms of understanding

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 are a .prerequisite for them to be adopted by consumers

CHAPTER - 2

PURPOSE, SCOPE AND APPILCABILITY

:Purpose 2.1

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistant sense able users to speak natural language voice commands in order to operate the device and its apps. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being .optimized, it's clear that we are moving towards less screen interaction

:Scope 2.2

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it's not just

developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for them specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with .voice assistants is missing. Users simply cannot see or touch a voice interface

:Applicability 2.3

The mass adoption of artificial intelligence in users' everyday lives is also Fuelling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user's life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually .reserved for the rich and famous

SURVEY OF TECHNOLOGY

:Python 2.4

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 1/5 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 this project we used libraries used are .speech recognition to recognize voice, Pyttsx for text to speech

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

:Pyttsx 2.5

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 MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantages that it .works offline

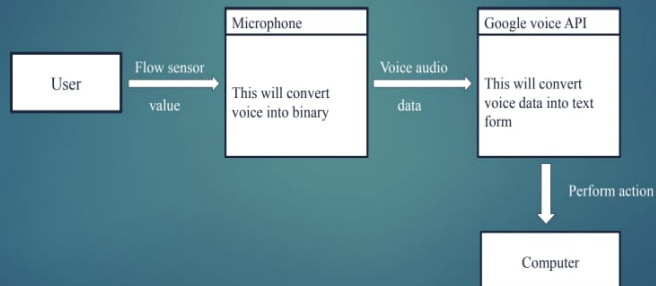
:Speech Recognition 2.6

Dataflow Diagrams

6

8

DFD - 2



2.1

2.2

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.

CHAPTER - 3

REQUIREMENT AND ANALYSIS

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to

resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. The .complete analysis is followed below

:Problem definition 3.1

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. when a virtual assistant is not able to answer questions accurately, it's because it lacks the proper context or doesn't understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviours. They require large amount of information to be fed in order for it to work efficiently. virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks

and each sub-task can have its own sub-tasks. In Such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to .recommend a particular plan

.Virtual assistants must provide a wide variety of services

:These include

.Providing information such as weather, facts from e.g., Wikipedia etc*

.Set an alarm or make to-do lists and shopping lists*

.Remind you of birthdays and meetings*

.Play music from streaming services such as Saavn and Gaana*

Play videos, TV shows or movies on televisions, streaming from e.g., Netflix or Hotstar. *Book *

.tickets for shows, travel and movies

Feasibility Study 3.2

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed .system. Five types of feasibility study are taken into consideration

:Technical feasibility 3.3

It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now adays and everyone generally, possess them. Besides, system needs internet connection. While using it, make Sure, you have a steady internet connection. It is .also not an issue in this era were almost every home or office has Wi-Fi

:Operational feasibility 3.4

It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still .don't know to write can readout problems for system and get answers

:Economic feasibility 3.5

Here, we find the total cost and benefit of the Proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. .Again, they are cheap and available. As far as maintenance is concerned, it won't cost too much

:Organizational feasibility 3.6

This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won't create any .management issues and will increase the feasibility of the project

:Cultural feasibility 3.7

It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. This project is technically feasible with no external hardware requirements. also, it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system is achievable. Decision is taken to

.proceed with the project

HARDWARE AND SOFTWARE REQUIREMENTS

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 3.8

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

:Software 3.9

Windows 7(32-bit) or above. *Python 3.9 or later*

Chrome Driver*

CHAPTER - 4

IMPLEMENTATION AND RESULTS

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor made by Microsoft for Windows, Linux and macOS. [10] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran.

:CODE 4.1

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import time
import os
import pyjokes
import subprocess as sp
```

```
MASTER = "..."
```

```

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

def speak(text):
    engine.say(text)
    engine.runAndWait()

def wishMe():
    hour = int(datetime.datetime.now().hour)

    if hour>=0 and hour<12:
        speak("Good Morning" + MASTER)
    elif hour>=12 and hour<18:
        speak("Good Afternoon" + MASTER)
    else:
        speak("Good Evening" + MASTER)
    speak("I am Alexa... How may I help you?")

def takeCommand():
    query = None
    while query is None:
        r = sr.Recognizer()
        with sr.Microphone() as source:
            r.adjust_for_ambient_noise(source, duration=5)
            print("Listening...")
            speak("Beep")
            r.pause_threshold = 1
            audio = r.listen(source)

        try:
            print("Recognising...")
            query = r.recognize_google(audio, language= 'en-in')
            ##
            #query = query.lower()
            #if 'alexa' in query:
            #    query = query.replace('alexa', '')
            print(f"user said: {query}\n")
            print(query)

        except Exception:
            print("Say that again please...")
            speak("Say that again please...")
            query = None

    if 'Wikipedia' in query:
        query = query.lower()
        speak('Searching in wikipedia...')
        query = query.replace("wikipedia", "")
        results = wikipedia.summary(query, sentences = 2)
        speak("According to wikipedia")
        print(results)
        speak(results)

    elif 'YouTube' in query:
        if 'search' in query:
            speak("{MASTER} what do you want to search")
            query2 = None
            while query2 is None:
                r2 = sr.Recognizer()
                with sr.Microphone() as source:

```

```

        print("Listening...")
        speak("Beep")
        audio2 = r2.listen(source, 2)
        command = r2.recognize_google(audio2)
        print(command)

    try:
        print("Recognising...")
        query2 = r2.recognize_google(audio2, language= 'en-in')
        print(f"user said: {query2}\n")
        print(query2)
        break

    except Exception:
        print("Say that again please...")
        speak("Say that again please...")
        query2 = None

    query2 = query2.replace(" ", "+")
    print("eee")
    crome_path = "C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s"
    webbrowser.get(crome_path).open(url = "https://www.youtube.com/results?search_query="+query2)
else:
    crome_path = "C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s"
    webbrowser.get(crome_path).open(url = "youtube.com")

elif 'timer' in query:
    speak("{MASTER} for how much time do you want to set?")
    query2 = None
    while query2 is None:
        r2 = sr.Recognizer()
        with sr.Microphone() as source:
            print("Listening...")
            speak("Beep")
            audio2 = r2.listen(source)
            print("Done listen")

        try:
            print("Recognising...")
            query2 = r2.recognize_google(audio2, language= 'en-in')
            print(f"user said: {query2}\n")
            print(query2)
            break

        except Exception:
            print("Say that again please...")
            speak("Say that again please...")
            query2 = None

time1 = ""

for i in query2:
    if i.isdigit():
        time1 = time1 + i
time2 = int(time1)

if 'minutes' or 'minute' in query2:
    speak("Ok, your time starts now!")
    time.sleep(60*time2)
    speak("the time is up "+MASTER)

else:
    speak("No timer was set")

```



```

while True:
    takeCommand()
main()

```

:Outputs 4.2

Various activities/situations where our voice assistant could be used with its output image as well as results is given below

.User Input—The assistant will wait for the user to give voice command for further processing

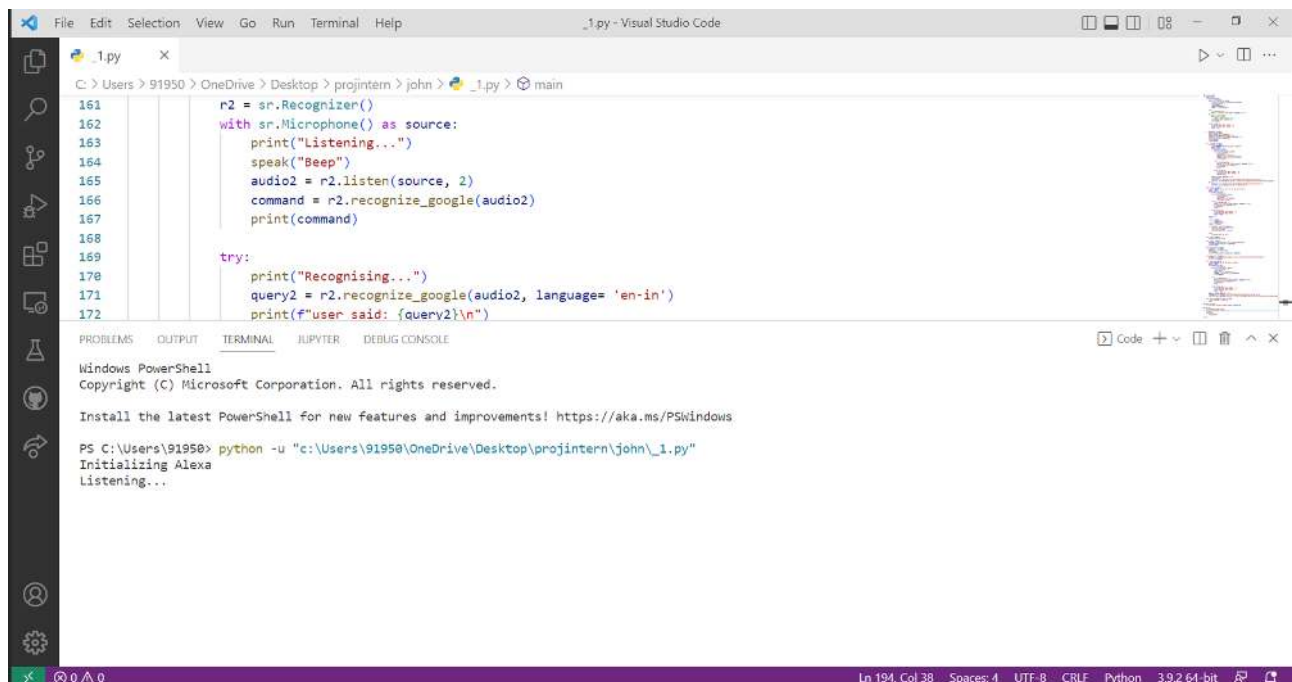


Fig 4.1

Surfing web and searching within various websites, for e.g Youtube 4.2.1

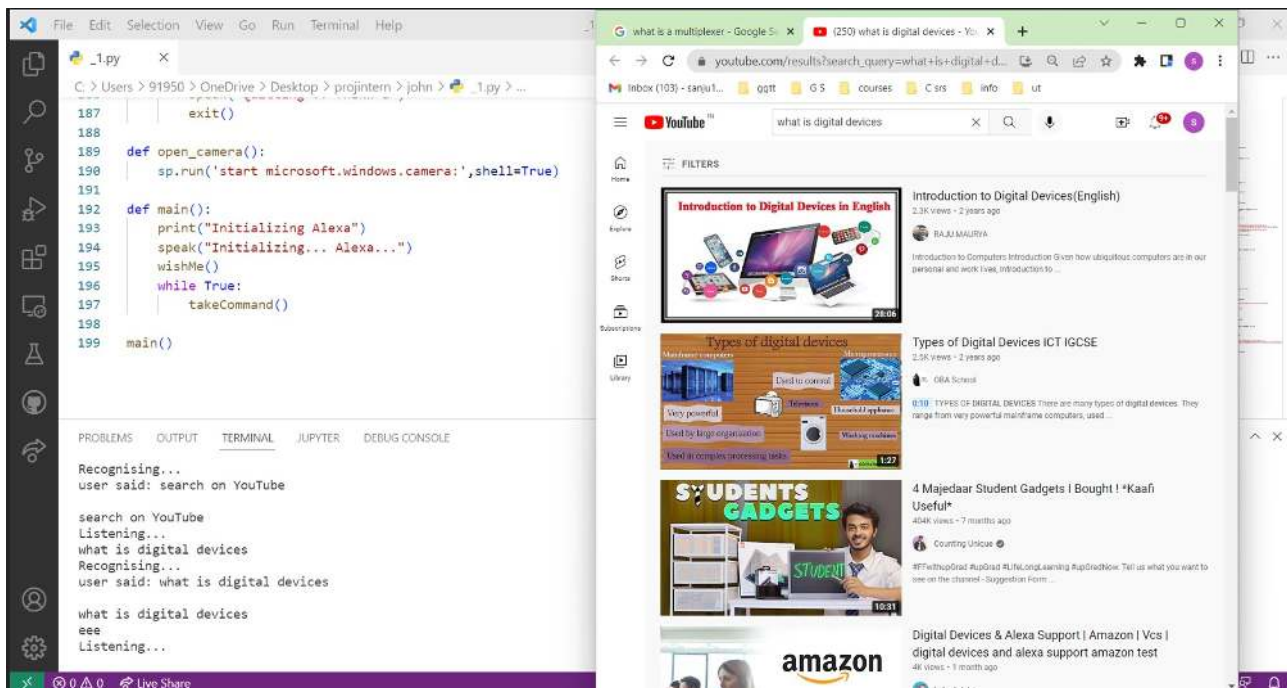


Fig 4.2

Setting a Timer/Reminder made easy using our assistant 4.2.2

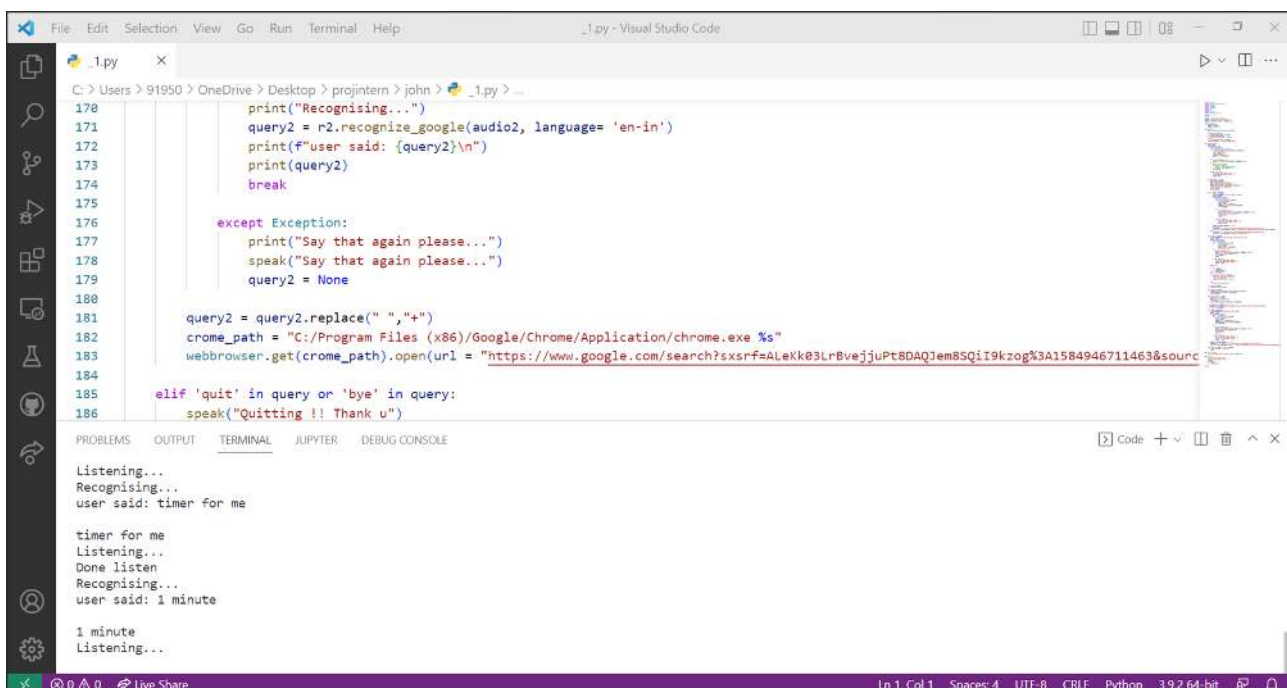


Fig 4.3

Reading knowledgeable articles for us on demand from websites like Wikipedia 4.2.3

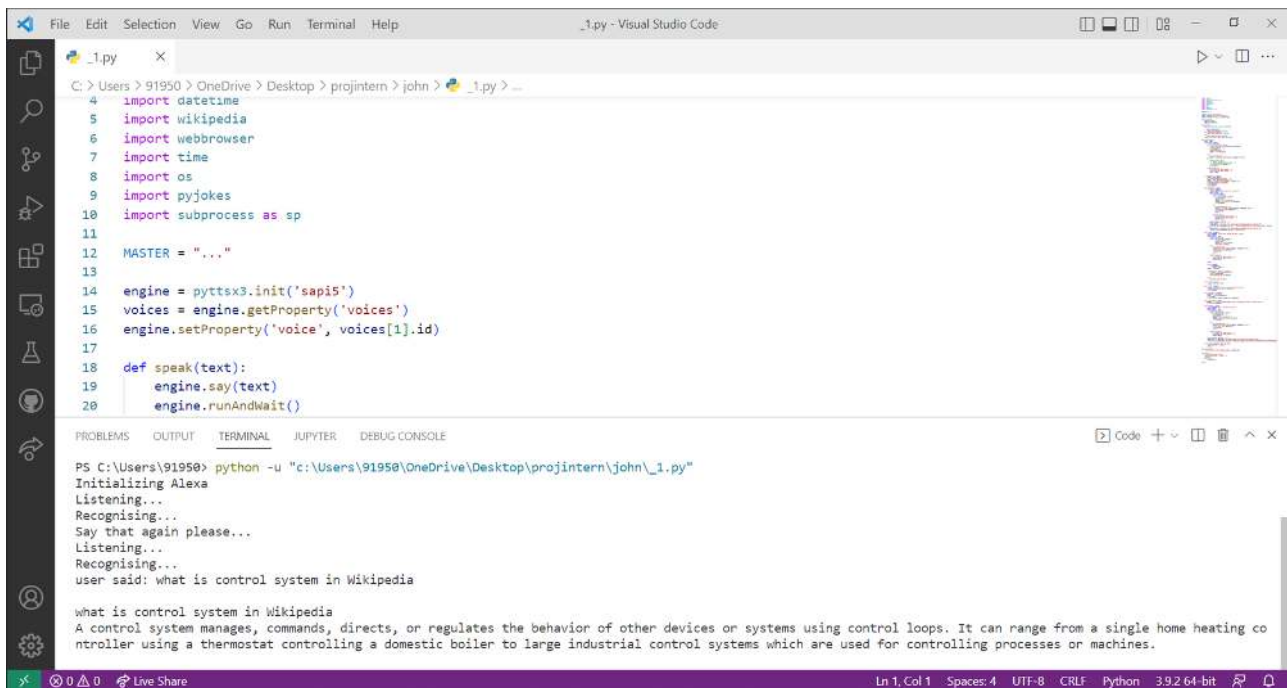


Fig 4.4

Playing songs through media player for us 4.2.4

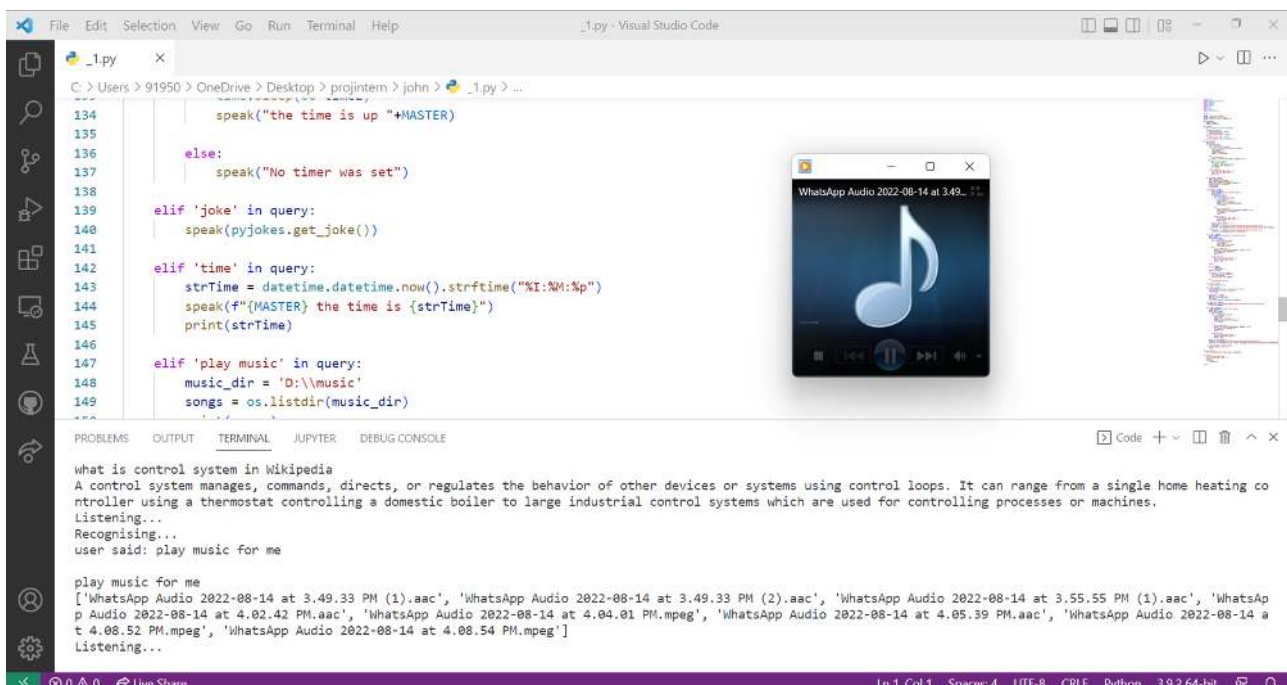
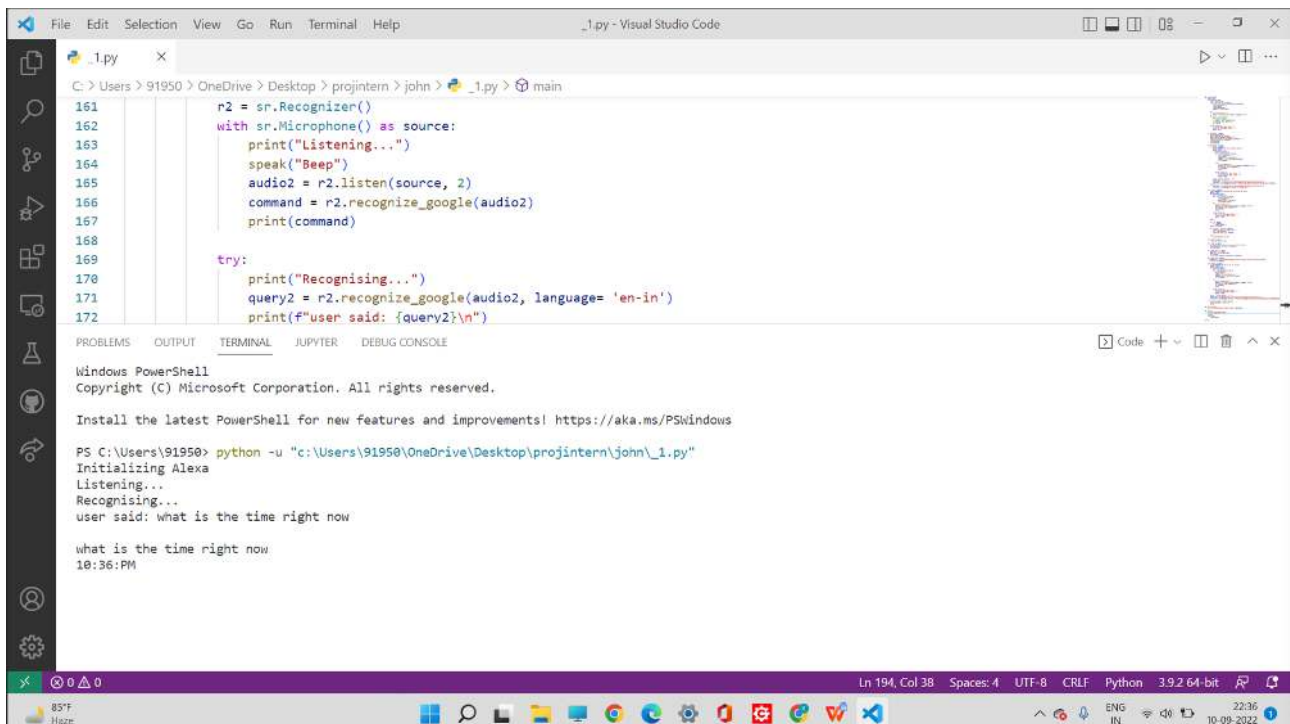


Fig 4.5

Doing Interactive activities, e.g joke telling as well as time telling 4.2.5 could be done



The screenshot shows the Visual Studio Code interface with a Python file named `_1.py` open. The code in the editor is as follows:

```
161 r2 = sr.Recognizer()
162 with sr.Microphone() as source:
163     print("Listening...")
164     speak("Beep")
165     audio2 = r2.listen(source, 2)
166     command = r2.recognize_google(audio2)
167     print(command)
168
169 try:
170     print("Recognising...")
171     query2 = r2.recognize_google(audio2, language= 'en-in')
172     print(f"user said: {query2}\n")
```

The terminal window shows the output of the script:

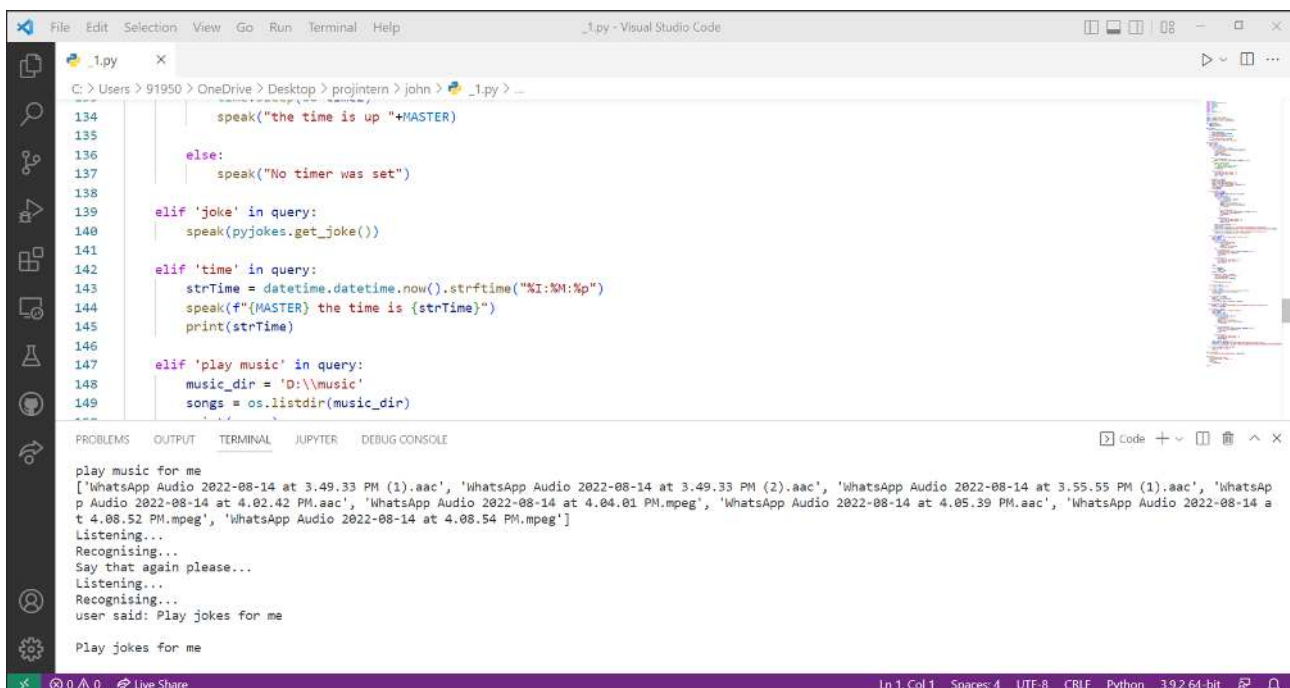
```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\91950> python -u "c:\Users\91950\OneDrive\Desktop\projintern\john\_1.py"
Initializing Alexa
Listening...
Recognising...
user said: What is the time right now

what is the time right now
10:36:PM
```

Fig 4.6



The screenshot shows the Visual Studio Code interface with a Python file named `_1.py` open. The code in the editor is as follows:

```
134 speak("the time is up "+MASTER)
135
136 else:
137     speak("No timer was set")
138
139 elif 'joke' in query:
140     speak(pyjokes.get_joke())
141
142 elif 'time' in query:
143     strTime = datetime.datetime.now().strftime("%I:%M:%p")
144     speak(f"{MASTER} the time is {strTime}")
145     print(strTime)
146
147 elif 'play music' in query:
148     music_dir = 'D:\\music'
149     songs = os.listdir(music_dir)
```

The terminal window shows the output of the script:

```
play music for me
['WhatsApp Audio 2022-08-14 at 3.49.33 PM (1).aac', 'WhatsApp Audio 2022-08-14 at 3.49.33 PM (2).aac', 'WhatsApp Audio 2022-08-14 at 3.55.55 PM (1).aac', 'WhatsApp Audio 2022-08-14 at 4.02.42 PM.aac', 'WhatsApp Audio 2022-08-14 at 4.04.01 PM.mpeg', 'WhatsApp Audio 2022-08-14 at 4.05.39 PM.aac', 'WhatsApp Audio 2022-08-14 at 4.08.52 PM.mpeg', 'WhatsApp Audio 2022-08-14 at 4.08.54 PM.mpeg']
Listening...
Say that again please...
Listening...
Recognising...
user said: Play jokes for me

Play jokes for me
```

Fig 4.7

Searching on Google the exact thing we require making googling faster 4.2.6

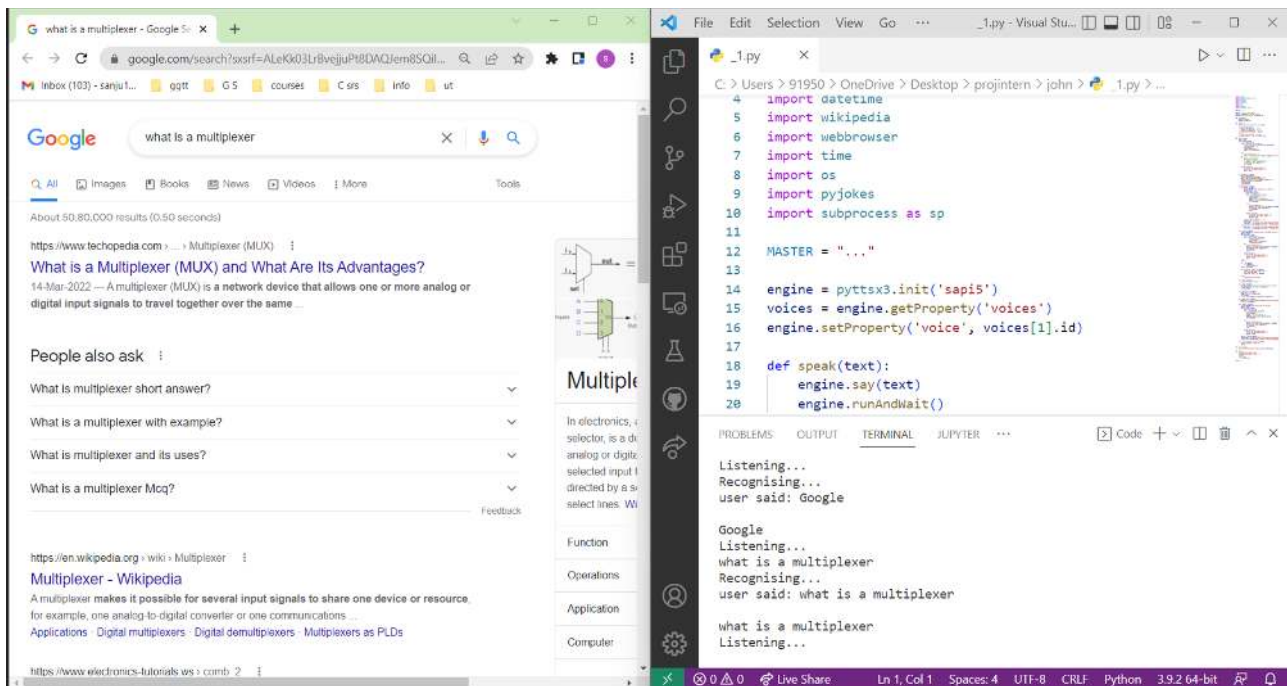


Fig 4.8

: Open Applications 4.2.7

VS code

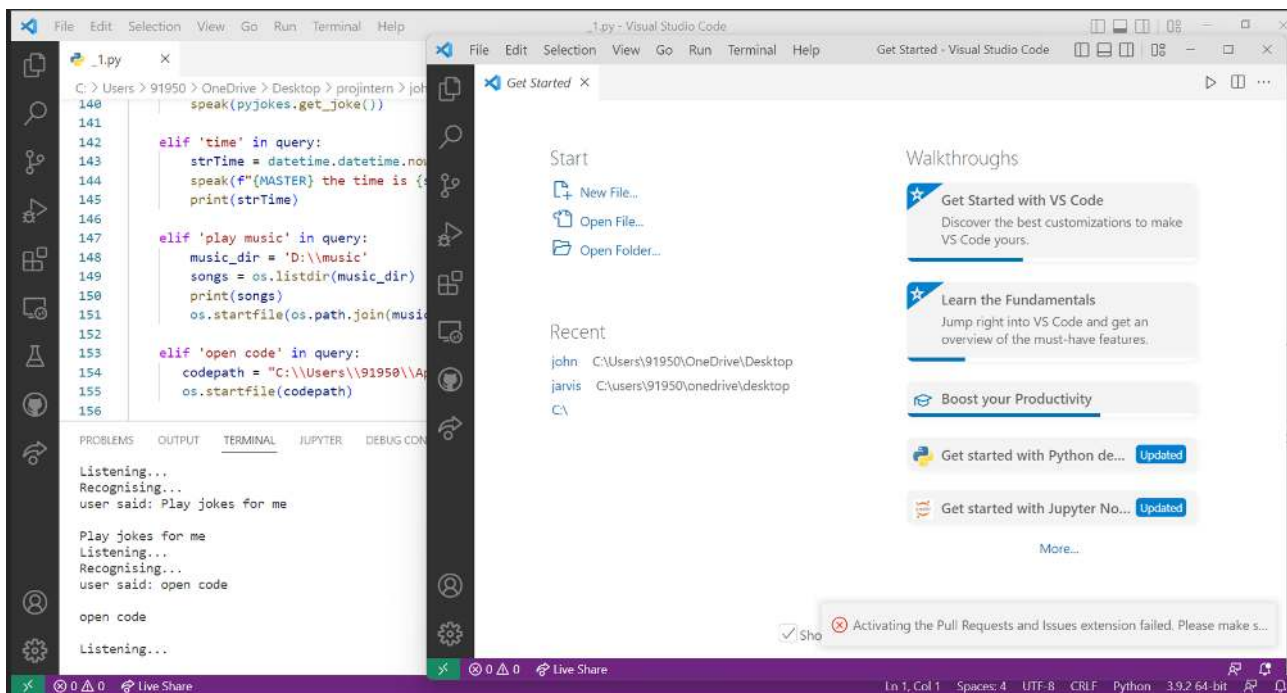
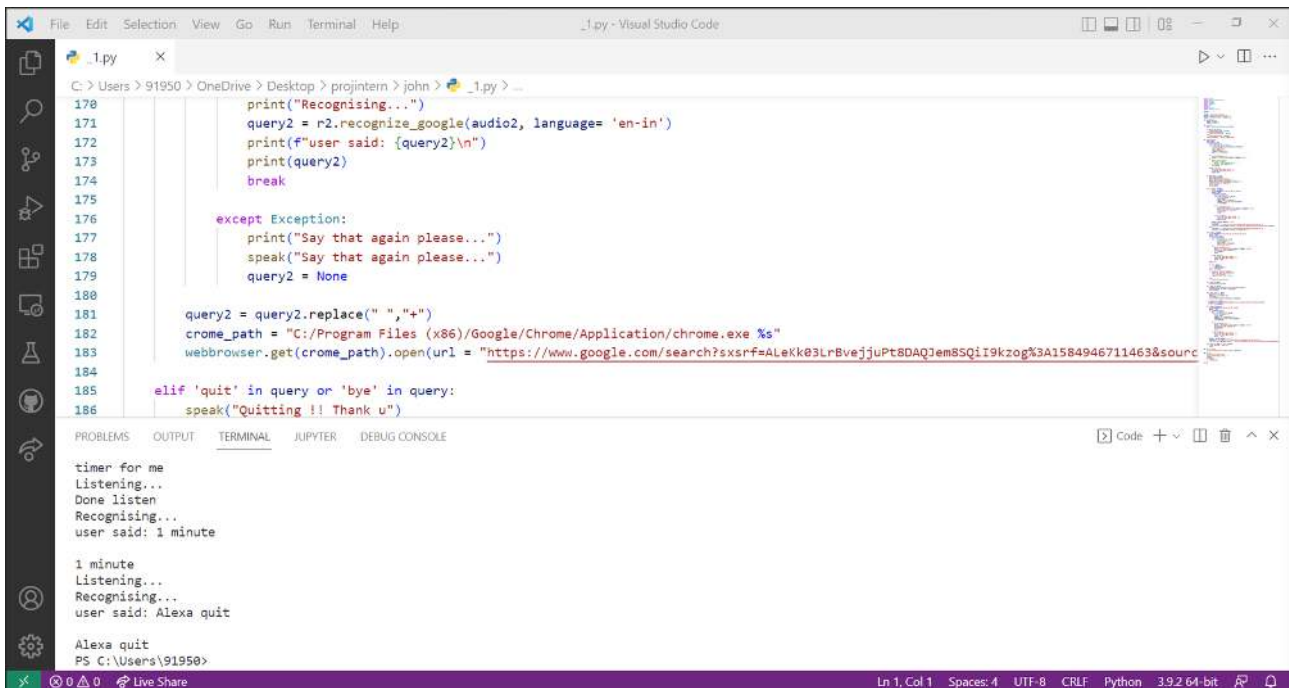


Fig 4.9

: Quit



The screenshot shows a Visual Studio Code window with a Python file named `_1.py` open. The code is a script that interacts with Google Assistant (r2) to recognize voice commands and perform actions like opening a web browser. The terminal output shows the script's execution, including listening for commands, recognizing "1 minute" and "Alexa quit", and displaying the current directory.

```
C:\Users\91950> OneDrive\> Desktop\> projintern > john > _1.py > -
170     print("Recognising...")
171     query2 = r2.recognize_google(audio2, language= 'en-in')
172     print(f"user said: {query2}\n")
173     print(query2)
174     break
175
176     except Exception:
177         print("Say that again please...")
178         speak("Say that again please...")
179         query2 = None
180
181     query2 = query2.replace(" ", "+")
182     chrome_path = "C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s"
183     webbrowser.get(chrome_path).open(url = "https://www.google.com/search?xsrf=ALekK03LrBveJjuPt8DAQiem8SQiI9kzog%3A1584946711463&source=hp&hl=en&gl=us&sa=X&ved=269q7j894qQ10wEoAEQ6AFgg:AOvVaw132n8=&as_sqs=AOvVaw132n8=")
184
185     elif 'quit' in query or 'bye' in query:
186         speak("Quitting !! Thank u")
```

terminal

```
timer for me
Listening...
Done listen
Recognising...
user said: 1 minute

1 minute
Listening...
Recognising...
user said: Alexa quit

Alexa quit
PS C:\Users\91950>
```

4.10

Conclusion

Voice Assistant is successfully made. Voice Assistant Controlled Personal Assistant System will use the natural language processing (Python speech to text) and uses python speech interpreter integrated with various code techniques to achieve a smart assistant that can control the computer .and applications and even solve user queries using web searches

It can be designed to minimise the human efforts to interact with many other subsystems, which would otherwise have to be performed manually. By achieving this, the system will make human .life comfortable

References

:Research paper

<https://www.ijeat.org/wp-content/uploads/papers/v10i4/D24250410421.pdf>

Python releases 3.10.7, 3.9.14, 3.8.14, and 3.7.14 are now available". 7 September 2022. "
.Retrieved 7 September 2022

:Books

Python Cookbook, Third Edition by David Beazley and Brian K . Jones .1

Python for Geeks by Muhammad Asif .2

:Websites

www.geeksforgeeks.com .1

www.w3schools.com .2

www.python.org .3