IMPLEMENTATION OF DEDICATED EMPLOYEE USING SOCKET PROGRAMING

Report submitted to the SASTRA Deemed to be University

as the requirement for the course

CSE302: COMPUTER NETWORKS

Submitted by

PAUL ROY

(Reg. No.: 124003214 B. Tech – Computer Science and Engineering)

December 2022

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJAVUR – 613 401

<u>Bona-fide Certificate</u>

This is to certify that the report titled "**Implementation Of Dedicated Employee Using Socket Programing**" submitted as a requirement for the course,

CSE302: COMPUTER NETWORKS for B. Tech is a bona-fide record of the work done by **Shri. Paul Roy (Reg. No 124003214, B. tech Computer Science and Engineering)** during the academic year 2022-23, in the School of Computing.

Project Based Work *Viva voce* held on ______________

Examiner1 **Examiner2**

ACKNOWLEDGEMENTS

First of all, I would like to thank God Almighty for his endless blessings.

I would like to express my sincere gratitude to **Dr S. Vaidyasubramaniam**, Vice-Chancellor for his encouragement during the span of my academic life at SASTRA Deemed University.

I would forever remain grateful and I would like to thank **Dr A. UmaMakeswari**, Dean, School of Computing and **Dr. R. Chandiramouli**, Registrar for their overwhelming support provided during my course span in SASTRA Deemed University.

I am extremely grateful to **Dr. Shankar Sriram**, Associate Dean, School of Computing for his constant support, motivation and academic help extended for the past three years of my life in School of Computing.

I would specially thank and express my gratitude to **Dr. Kannan Balasubramanian**, Instructor of Computer Science Engineering for providing me an opportunity to do this project and for his guidance and support to successfully complete the project.

I also thank all the Teaching and Non-teaching faculty, and all other people who have directly or indirectly helped me through their support, encouragement and all other assistance extended for completion of my project and for successful completion of all courses during my academic life at SASTRA Deemed University.

Finally, I thank my parents and all others who helped me acquire this interest in the project and aided me in completing it within the deadline without much struggle.

LIST OF FIGURES

ABBREVIATIONS

DE	**-** dedicated employee
TCP	**-** Transport Control Protocol
UDP	**–** User Datagram Protocol
SK	**-** Sockets
PY	**-** Python Programming

ABSTRACT

The implementation of the dedicated digital employee is done by using python programming language. The main concept we will be using in our application is using socket programming. There will be two faces of the application where one part of the application is client side and the other is server side. These two faces of the application are connected by the transmission control protocol(TCP). The library functions are used from python to implement various functions.

In the client side of the application various functions used from the library functions imported to our program. The voice recognition is done on the client side. The client side accepts the voice from the access of the microphone in the laptop, then it processes it to words with the help of libraries installed in the python programming language. Then the command is displayed to the user to make sure the command meant by the user is the same. Then the command is sent to the server side of the Dedicated employee application using socket programing.

In the server side of the Dedicated employee application, the server side accepts the command from the client using socket programming which is connected by the Transport control protocol(TCP). Then the server side of the program access the command by checking the main keywords like music, date,
Etc. Then the server side executes the command by playing music, showing a date,etc. Then the server side of the application will automatically give the command to close the client side of the application through using sockets, where two sides of the application are connected by Transport Control Protocol(TCP) .

In case the server side of the application cannot process the command from the client side of the application, then the server side of the application gives a message to the client side of the program that the command couldn’t be processed and asks whether that the user wishes to give anymore commands. This has to be kept in mind the message is sent through sockets in the application and the server and client side of the application is connected with Transport Control protocol(TCP).
Then the program by itself will shut down .

Making it more ideal so that the user does not need to keep using the mouse or the keyboard for all the uses the User needs.

TABLE OF CONTENTS

1. INTRODUCTION

In this busy world every single person in the world tries to multitask so that they can be more efficient and be more effective to the organization he/she is working. There are limits to what a person can do with his hands and it’s time consuming for a person to type a query and find a solution on the net. Most of the time a person requires is just a small piece of information from the net which can influence the time, priority and importance of the work being done.

Some people like to keep reminders for themselves so that they can be upto date with their work and are busy enough just to set a reminder for themselves. So to summarize, that’s why busy people around the world keep a dedicated employee for themselves who can organize their work and remind them about a meeting, a deadline to compete with or the urgency of a project. But to keep a dedicated employee for themselves they need to hire a person which costs the company or the person a great deal of resources like time and money. Sometimes the people hired need to take a day or two off for their personal reasons or needs.

These problems are all solved by having AI who is always available at all your needs and all the time. These dedicated employees are preferred in such a way that the command given by the employer in the full or the half sense of the job should be done. These employees also made their way to the homes of different people helping in different ways like streaming music, helping giving instruction while cooking, taking care of the different functionalities in home like switching on/off the lights, fan, etc.

2.ABOUT TCP:

2.1.WHAT IS TCP ?

Transmission Control Protocol (TCP) is a standard that defines how to establish and maintain a network conversation by which applications can exchange data.TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other.

Together, TCP and IP are the basic rules that define the internet. The Internet Engineering Task Force (IETF) defines TCP in the Request for Comment (RFC) standards document number 793.

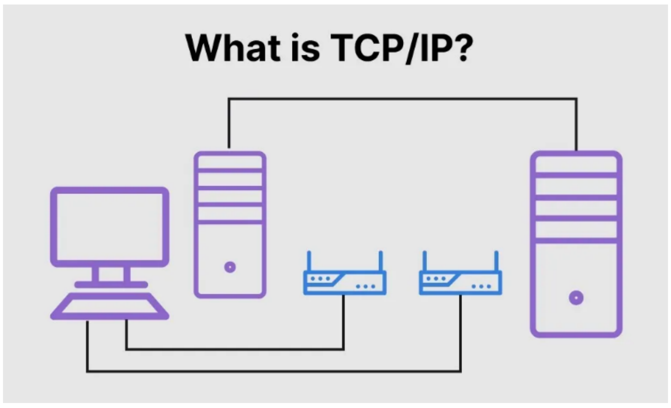


Fig 2.1: Tcp Description

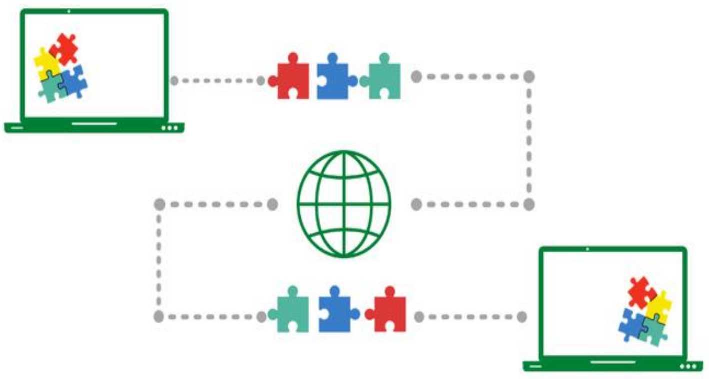


Fig 2.2: Another Tcp Description

2.2.HOW TCP WORKS ?

TCP is a connection-oriented protocol, which means a connection is established and maintained until the applications at each end have finished exchanging messages. Process that take place when tcp transmission takes place:
1.Determines how the data should broken into packets that can be transmitted
2.Sends and accepts packets from the network layer
3.Manages flow control
4.Handles the transmission of dropped or garbled packets, so that there will be error free
   transmission of the data
5.Acknowledge all the packets that arrive

When a web server sends an HTML file to a client, it uses the hypertext transfer protocol (HTTP) to do so. The HTTP program layer asks the TCP layer to set up the connection and send the file. The TCP stack divides the file into data packets, numbers them and then forwards them individually to the IP layer for delivery.

Although each packet in the transmission has the same source and destination IP address, packets may be sent along multiple routes. The TCP program layer in the client computer waits until all packets have arrived. It then acknowledges those it receives and asks for the retransmission of any it does not, based on missing packet numbers. The TCP layer then assembles the packets into a file and delivers the file to the receiving application.

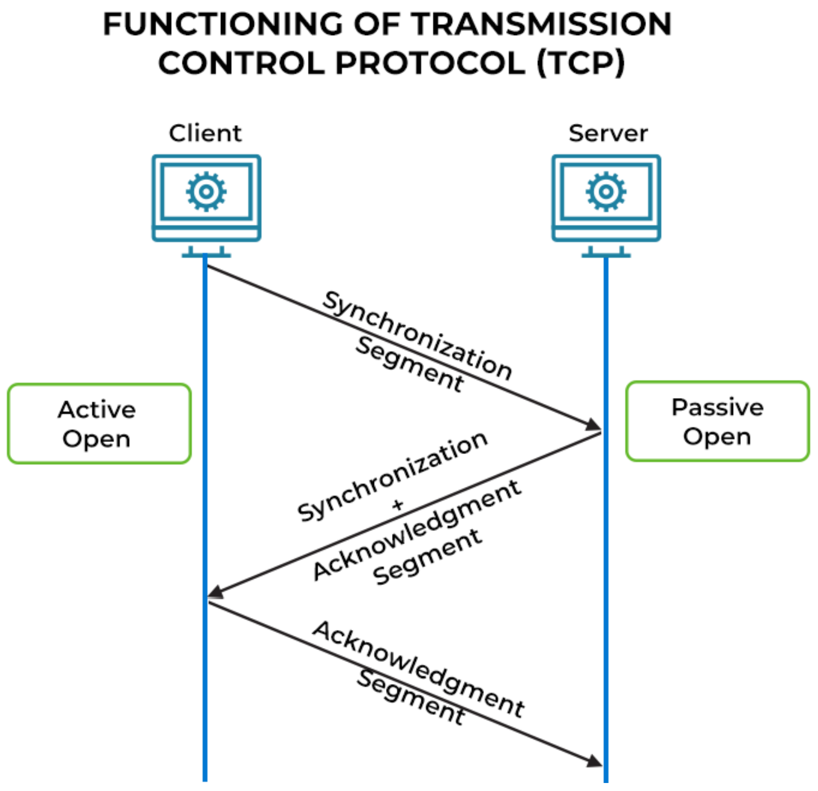

FUNCTIONING OF TRANSMISSION CONTROL PROTOCOL (TCP)
Client
Server
Synchronization Segment
Active Open
Passive Open
Synchronization + Acknowledgment Segment
Acknowledgment Segment

Fig 2.3: Functioning Of TCP

2.3.WHAT IS THE DIFFERENCE BETWEEN TCP AND UDP?

Feature	**TCP**	**UDP**
Connection status	Requires an established connection to transmit data (connection should be closed once transmission is complete)	Connectionless protocol with no requirements for opening, maintaining, or terminating a connection
Data sequencing	Able to sequence	Unable to sequence
Guaranteed delivery	Can guarantee delivery of data to the destination router	Cannot guarantee delivery of data to the destination
Retransmission of data	Retransmission of lost packets is possible	No retransmission of lost packets
Error checking	Extensive error checking and acknowledgment of data	Basic error checking mechanism using checksums
Method of transfer	Data is read as a byte stream; messages are transmitted to segment boundaries	UDP packets with defined boundaries; sent individually and checked for integrity on arrival
Speed	Slower than UDP	Faster than TCP
Broadcasting	Does not support Broadcasting	Does support Broadcasting
Optimal use	Used by HTTPS, HTTP, SMTP, POP, FTP, etc	Video conferencing, streaming, DNS, VoIP, etc

2.4.WHAT IS SOCKET PROGRAMING?

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server.

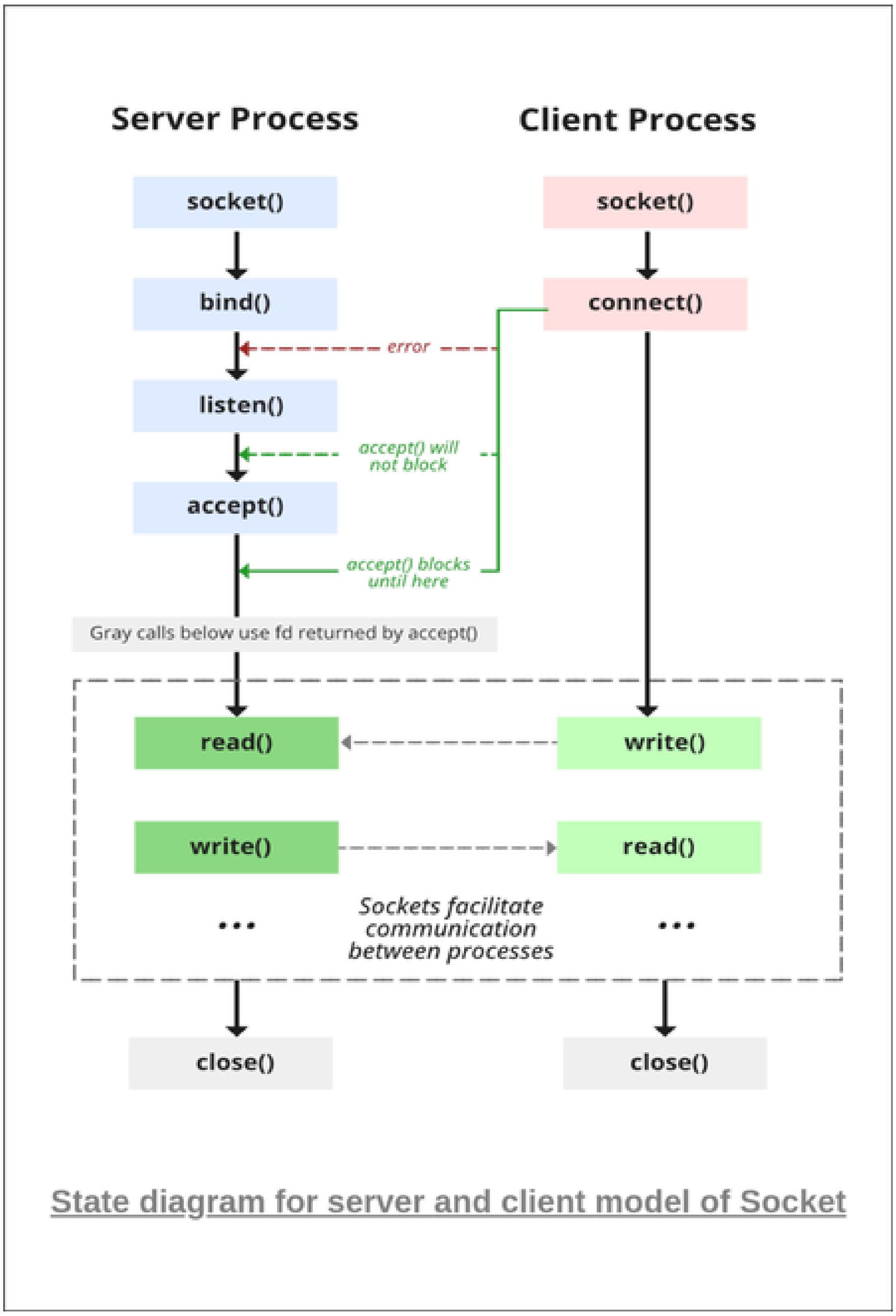

Server Process
Client Process
socket()
socket()
bind()
connect()
error
listen()
accept() will not block
accept()
accept() blocks until here
Gray calls below use fd returned by accept()
read()
write()
write()
read()
...
Sockets facilitate communication between processes
...
close()
close()
State diagram for server and client model of Socket

Fig 2.4: State Diagram of Socket Programming

2.5.HOW THE COMMUNICATION IS DONE?

To communicate both devices need to have sockets at their end. Since we're doing java programming these sockets are called java objects or socket objects. So to establish a connection and to send and receive messages and data both ends need to have a socket at the sender's end and at the receiver's end. Below is the illustration of the network:

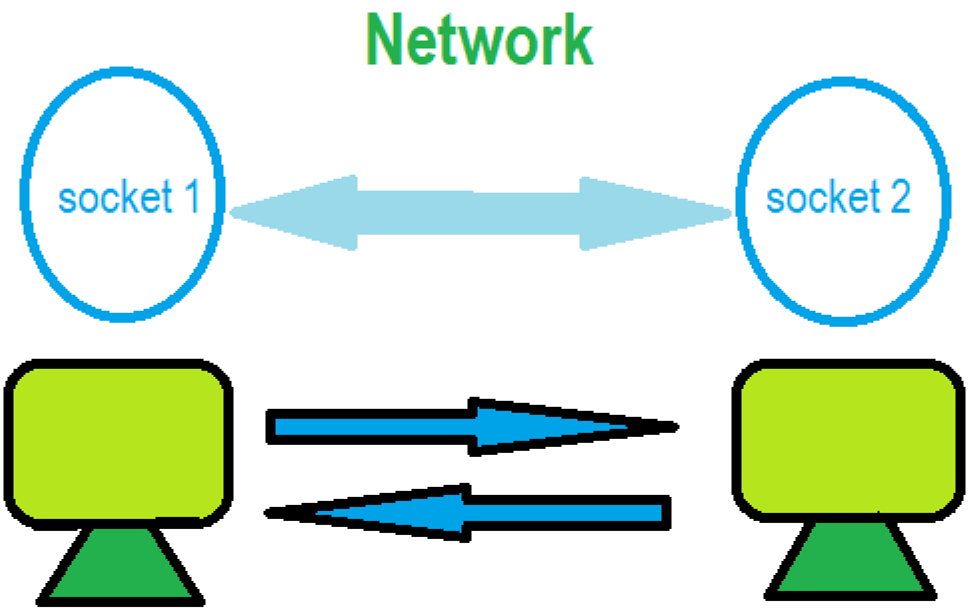


Fig 2.5: Sockets Description

2.6.WHY THE CHOICE OF SOCKETS USED IN THE PROGRAM?

There are various reasons for the SK programing are listed below:

1. Flexible & powerful
2. Very sufficient
3. Updated Information can be used to send only between devices
4. Low network traffic if efficient use

3.Utilities Used In The Project:

3.1 Python Programming:

Python programming is used to code this project. Python is really powerful because of it being platform independent, it run anywhere in any os. Python is a great tool which runs in the current trend and one of the latest programming language which is currently widely studied and practiced. Python has greater use of library functions available which is diverse. Python programming is a great tool to make application and makes great use of the large library functions making the programming compact and friendly with coder making readability great. Editing and increasing more functions to the program is really great in python programming.

3.2 More About Python Programming:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

3.3 Classes Used In Our Project:

A great no of functions are used from the python library classes. These functions are used and the modules are installed using pip which the functions are used in our project.

3.3.1 Classes Imported And Uses:

1.socket- Used to access the sockets for transfer of data between the server and the client

2.pyttsx3-Used to Speak to the user making the process more friendly

3.speech_recognition- Used to access the microphone and get the command from the user

4.webbrowser- Used to access the device's default browser and search for results

5.datetime- Used to access the date and time from the devices

6.pyautogui- Used to control volume of the devices

7. os- Used to access and open application for the users

8.psutil- Used to get battery related status from the device

9.pyjokes- Used to get random jokes for the users

10. pywhatkit- Used to access Youtube in the chrome browser

4. SOURCE CODE

4.1.CLIENT SIDE:

```
import socket
import pyttsx3
import speech_recognition as sr


#creating speak function using imported pyttsx3 library
def speak(audio):
    engine=pyttsx3.init()
    voices=engine.getProperty('voices')
    engine.setProperty('voices',voices[0].id)
    engine.say(audio)
    engine.runAndWait()


#creating function to take commands using speech recognition library
def takeCommand():
    r=sr.Recognizer()
    while(True):
        with sr.Microphone() as source:
            #pause threshold created so that after how much time speech hear is valid
            r.pause_threshold=0.7


            audio=r.listen(source)


            #exceptions created so that when the mic can get a clear command
            try:
                Query=r.recognize_google(audio,language='en-in')
                print(Query)
                return Query


            except Exception as e:
                speak("Could u please repeat again?")


#created so that mich will be actively listening but not respond even if the command #is valid
or not
def takeCommand1():
    r=sr.Recognizer()
    while(True):
        with sr.Microphone() as source:
            #pause threshold created so that after how much time speech hear is valid
            r.pause_threshold=0.7
```

```
          audio=r.listen(source)


          #exceptions created so that when the mic can get a clear command
          try:
             Query=r.recognize_google(audio,language='en-in')
             print(Query)
             return Query
          except Exception as e:
             continue

#the infinite loop is created so that the program will run forever helping the user
while(True):
   q=takeCommand1().lower()


   #takes command by using the defined function of take command
   if "hey" in q or "jarvis" in q:


      #the program start interacting only when the call command jarvis or hey is      #heard
      speak("Hello")
      speak("Welcome to Paul's customized dedicated Employee System..")
      speak("Please Direct me as your wish")


      #speak functions defined are used to talkback to the user
      #net infinite loop is activated for the functions which will be used in the future
      while(True):
          #socket gets initialized for the communication with the server side
          c=socket.socket()


          #it connects to the local host defined at 9999 port of the socket
          c.connect(("LocalHost",9999))


          #it takes command by using user defined functions
          query=takeCommand().lower()
          #the command taken is sent to the server to process and give the #appropriate
command back
          c.send(bytes(query,'utf-8'))


          #the talkback response from the server is received by the client from the #server side
          x=c.recv(1024).decode()

          #exits the second infinite loop if one is there in receiver command
          if "one" in x:
             speak("please Call me again if you have anyother work")
             speak("I will be here assisting with you if have any")
             break
          else:
             speak(x)
             print(x)
             #asking for new command so that the process will be continuously running
#according to the user
             speak("Do you want anything else boss?")
```

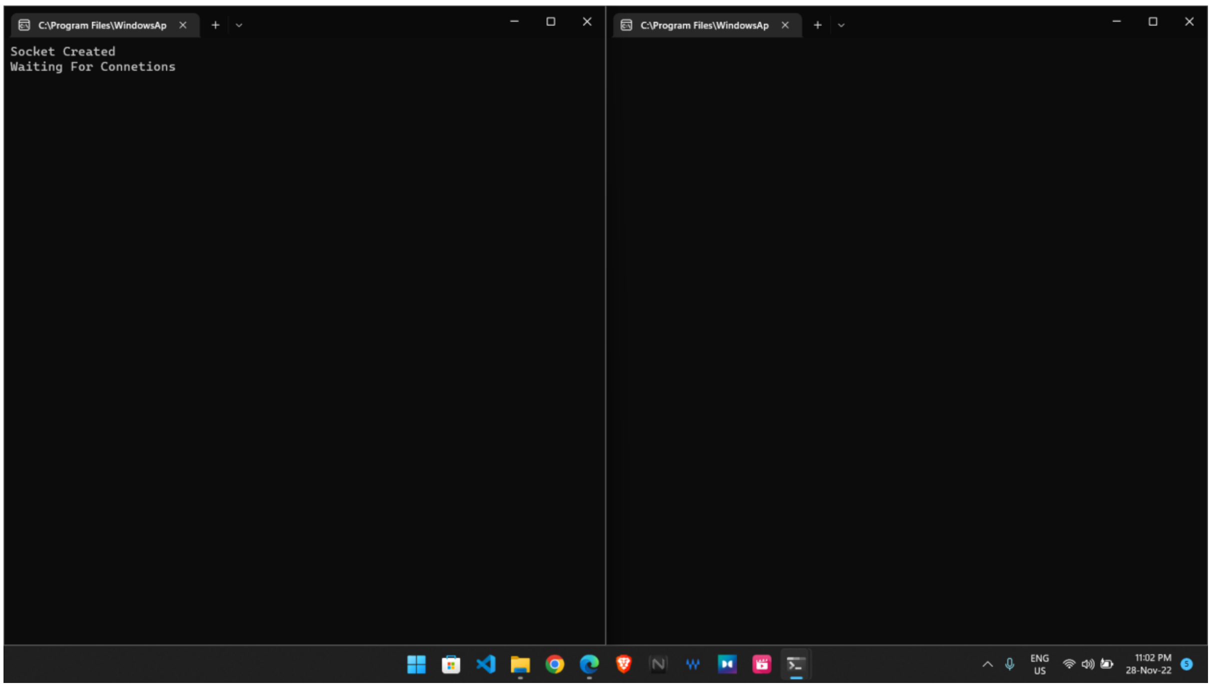
4.2.SERVER SIDE:

```
import socket
import webbrowser
import datetime
import pyautogui
import os
import psutil
import pyjokes


#socket created initialized in the ip
s=socket.socket()


#letting the user know about the course
print("Socket Created")


#binding the socket to the local host where the connection is created at 9999
s.bind(("LocalHost",9999))


#creating nof connections the server can handle simultaneously through the socket
s.listen(3)
#creating three connection for safety purpose


print("Waiting For Connetions")


#fuction being created here to convet the time which can be understood better
def convertTime(seconds):
   #dividing seconds by 60 for minutes
   minutes,seconds=divmod(seconds, 60)


   #dividing minutes by 60 for hours
   hours,minutes=divmod(minutes, 60)


   #returning the values
   return "%d:%02d:%02d"%(hours, minutes, seconds)
```

```
#infinite lop created
while True:
    #f=0 initalized so that if all the commands doesn't match it can show sum results in the
web
    f=0


    #c is initialized and address is created
    c,addr=s.accept()
    print("Connected With",addr)


    #q recives the commnd from the client side
    q=c.recv(1024).decode()


    #prints the command for furthur assistance
    print(q)


    #if the user has no other commands the givig client side to go dormant
    if "no" in q:
        c.send(bytes(" one ",'utf-8'))


        #f value changed to stop it from web searching
        f=1


    elif "date" in q:
        from datetime import datetime
        data=datetime.now()
        d=data.strftime('%m/%d/%Y')
        c.send(bytes(d,'utf-8'))


        #f value changed to stop it from web searching
        f=1

    elif "open" in q:
        if "chrome" in q or "google" in q:
            os.startfile("C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe")
            c.send(bytes("Chrome being opened for you sir....",'utf-8'))


        elif "edge" in q or "microsoft" in q:
            os.startfile("C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft
Edge.lnk")
            c.send(bytes("ms edge being opened for you sir....",'utf-8'))

        elif "brave" in q:
            os.startfile("C:\\ProgramData\\Microsoft\\Windows\\Start
Menu\\Programs\\Brave.lnk")
```

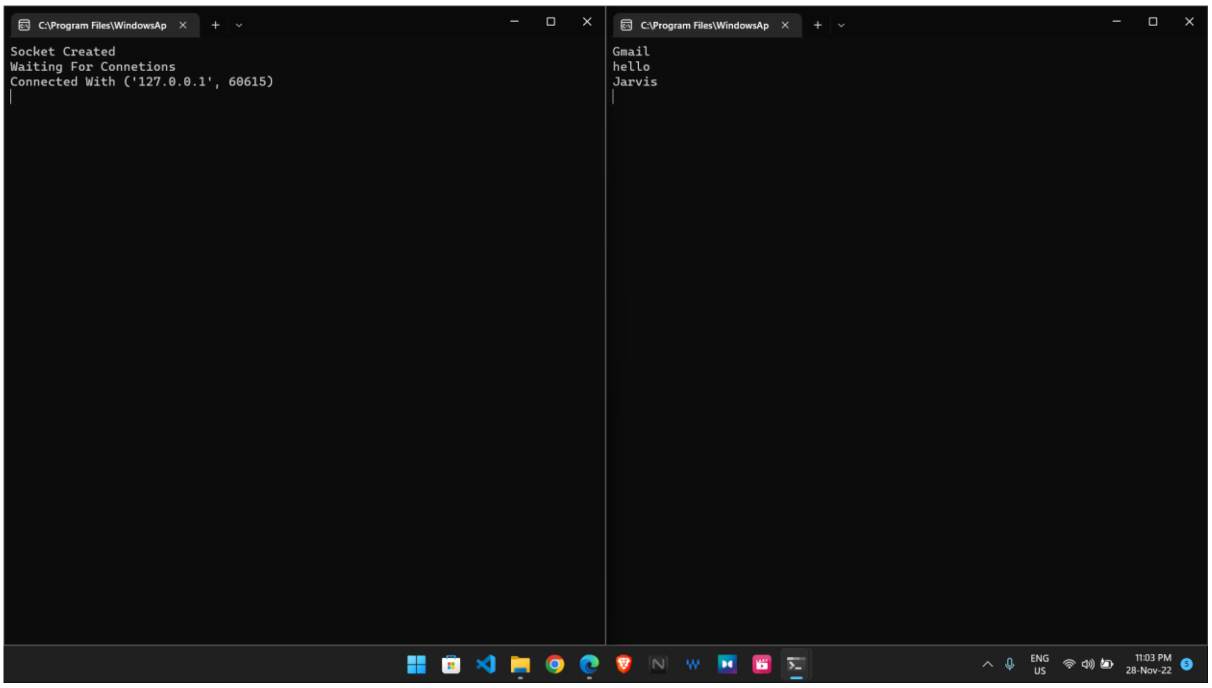
```
        c.send(bytes("Brave being opened for you sir....",'utf-8'))

    #f value changed to stop it from web searching
    f=1

#check whether there is a mention of time in the command
elif "time" in q:
    #importing datetime from datetime library in python
    from datetime import datetime

    #time getting the curent time from the system
    time= datetime.now()

    #just getting the minute and hour from the system
    current=time.strftime("%H:%M")

    #time being sent to the client side
    c.send(bytes(current,'utf-8'))

    #f value changed to stop it from web searching
    f=1

#check whether there is mention of youtube and check the search
elif "search" in q and "youtube" in q:

    #finding the where in command search is located
    n1=q.find("serach")

    #since search word has a lenght of 6
    n1=n1+6

    #finding the where in command in youtube is located
    n2=q.find("in youtube")

    #importing pywhatkit and initializing as pwt
    import pywhatkit as pwt

    #playing the youtube video from the internet
    pwt.playonyt(q[n1:n2])

    #sending the appropirate message
    c.send(bytes("this is the result we got...",'utf-8'))
    #f value changed to stop it from web searching
    f=1


#checking whether volume or sound is there in the command
elif "volume" in q or "sound" in q:


    #checking whether increase is there in the command
    if "increase" in q:


        #pyautogui class from python is used to increase the volume by 10%
        pyautogui.press("volumeup",10)


        #sending the appropirate command back to the client
        c.send(bytes("Audio Volume Increased Succesfully...",'utf-8'))

    #checking whether mute is there in the command
    elif "mute" in q:
        #pyautogui class from python is used to mute the volume
        pyautogui.press("volumemute")


    #decreasing the volume
    else:
        #pyautogui class from python is used to reduce the colume by 10%
        pyautogui.press("volumedown",10)


        #sending the appropirate command back to the client
        c.send(bytes("Audio Volume decreased Succesfully...",'utf-8'))


    #f value changed to stop it from web searching
    f=1


#checking whether battery or left is there in the command
elif "battery" in q and "left" in q:


    #using psutil class to get the battery status using sensors from the devices
    battery=psutil.sensors_battery()


    #using user defined functions to convert no of seconds left in battery drinage
    s=convertTime(battery.secsleft)


    #sending the appropirate command back to the client
    s="the time remaining for battery drinage is"+s
    c.send(bytes(s,'utf-8'))


    #f value changed to stop it from web searching
    f=1

#checking whether battery or left is there in the command
elif "percent" in q and "battery" in q:


    #using psutil class to get the battery percent using sensors from the devices
    battery = psutil.sensors_battery()


    #finding the battery percent
    p=str(battery.percent)


    #sending the appropirate command back to the client
    p="Battery percentage is ..."+p
    c.send(bytes(p,'utf-8'))


    #f value changed to stop it from web searching
    f=1


#checking whether joke is there in the command
elif "joke" in q:


    #getting the joke from the pyjoke class imported from python
    joke = pyjokes.get_joke(language="en", category="neutral")


    #sending the joke to the client side
    c.send(bytes(joke,'utf-8'))


    #f value changed to stop it from web searching
    f=1

#if f==0 then all the functions are not meant by the user is not avialable
#giving a result from the webrowser
elif f==0:
    #openning to the webrowser and inputing search term and giving the result
    webbrowser.open(q)


    #sending the appropirate command back to the client
    c.send(bytes("this is the result i got",'utf-8'))
c.close()
```

5. SCREENSHOTS:



```
Socket Created
Waiting For Conetions
```

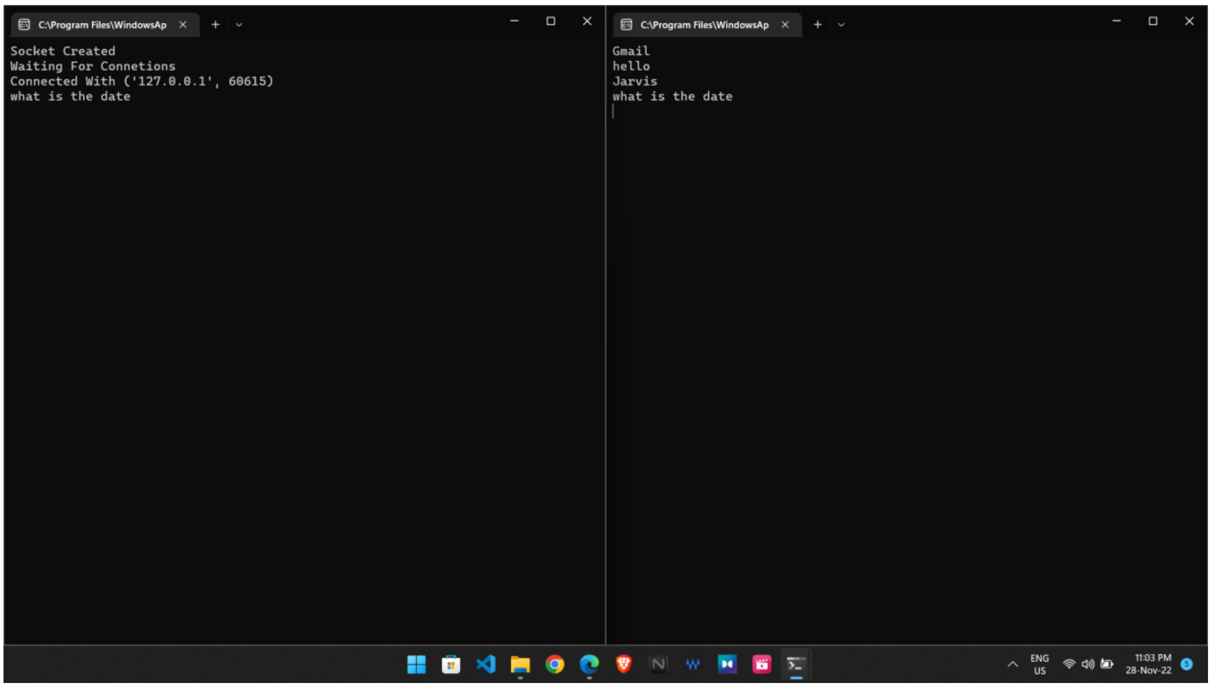
Fig 5.1: The Server is up and running socket is created and client is waiting for commands



```
Socket Created
Waiting For Conetions
Connected With ('127.0.0.1', 60615)
```

```
Gmail
hello
Jarvis
```

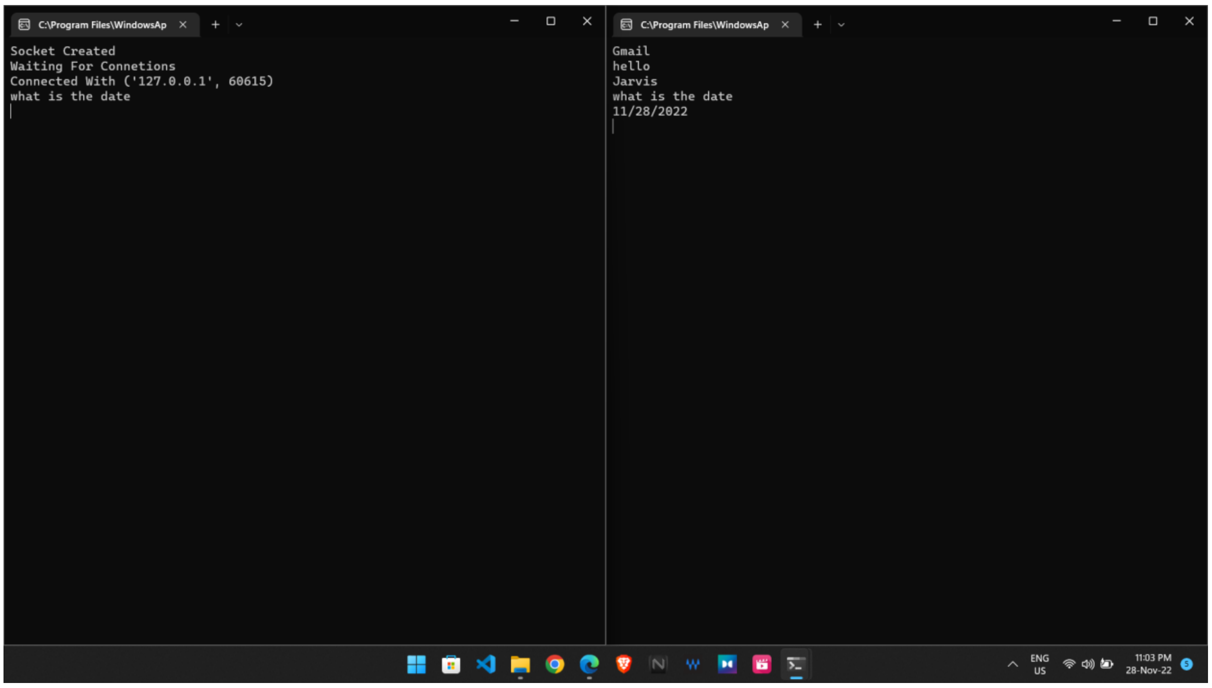
Fig 5.2: Client is catching different words but all the words which is caught doesn’t trigger the client side until the key Jarvis is noted



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
```

```
Gmail
hello
Jarvis
what is the date
```

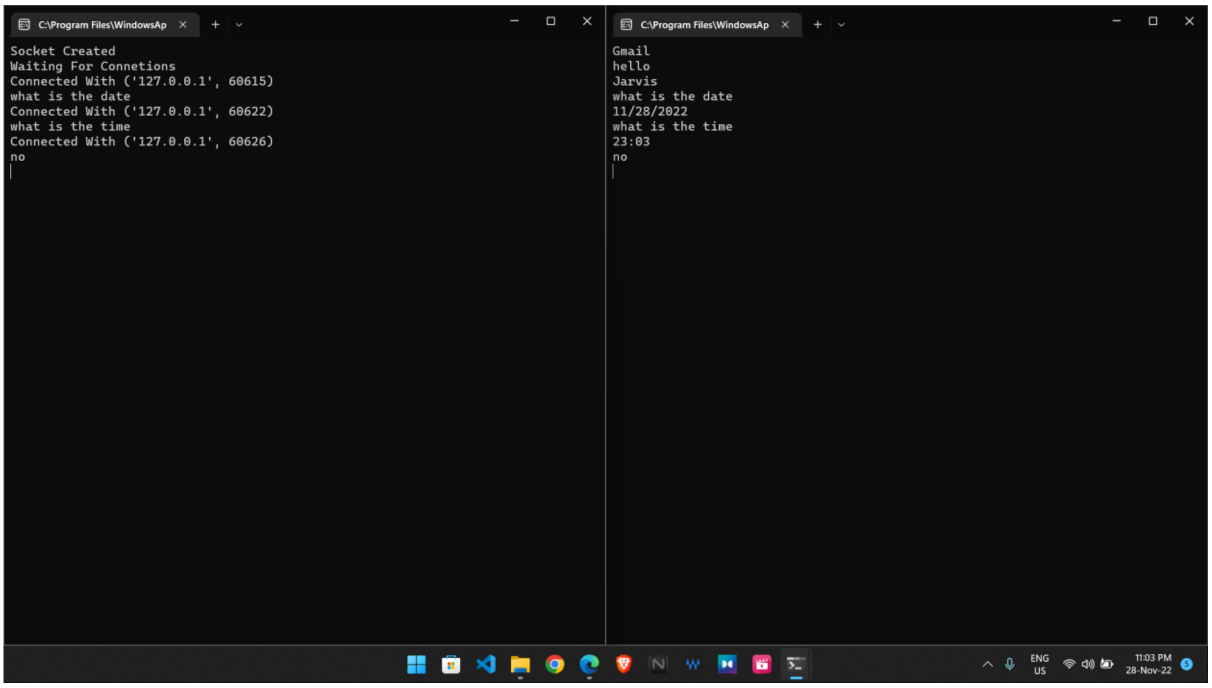
Fig 5.3: Jarvis keyword triggers the client which accepts the following command and sends to the server side



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
```

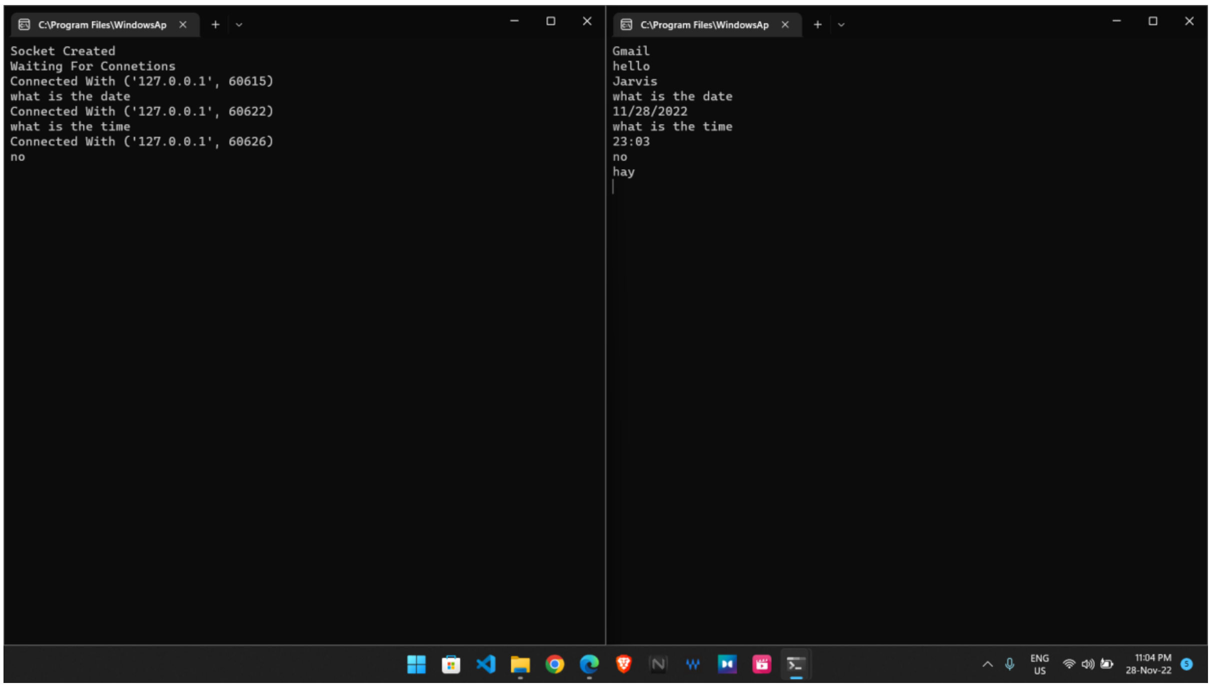
Fig 5.4: The server process the command sent to the client side which is spoken



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
Connected With ('127.0.0.1', 60622)
what is the time
Connected With ('127.0.0.1', 60626)
no
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
what is the time
23:03
no
```

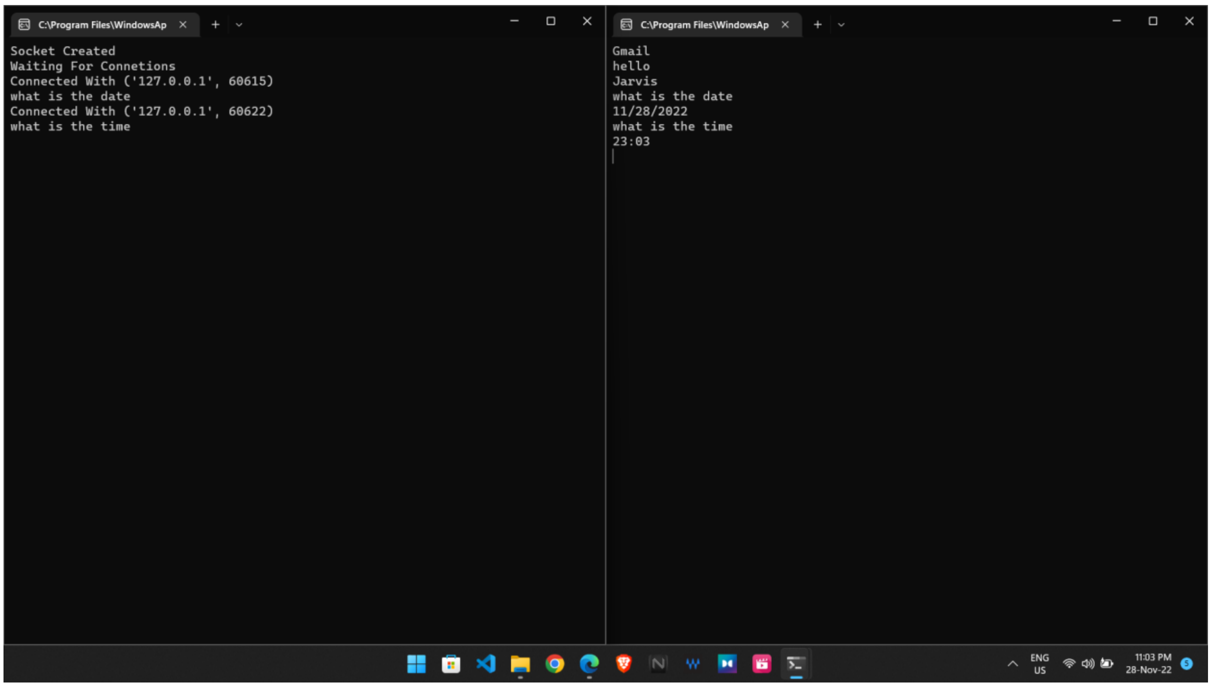
Fig 5.5:New connection is established waiting for the getting the next command



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
Connected With ('127.0.0.1', 60622)
what is the time
Connected With ('127.0.0.1', 60626)
no
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
what is the time
23:03
no
hay
```

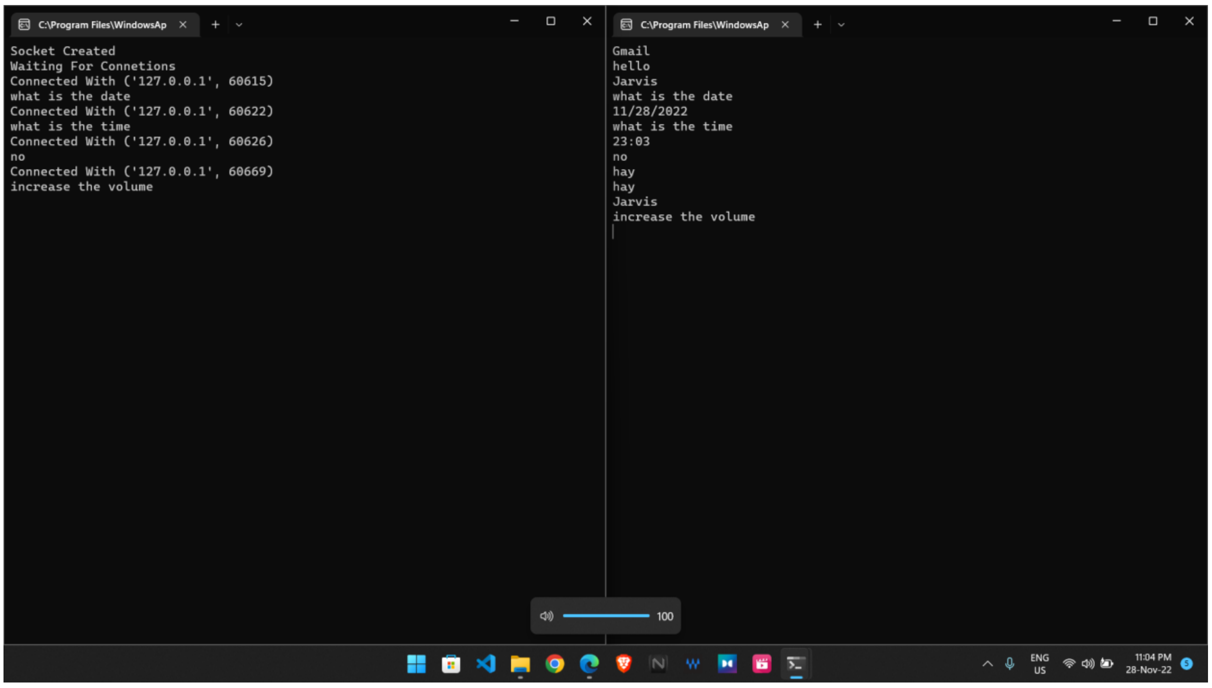
Fig 5.6: The client side going dormant



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
Connected With ('127.0.0.1', 60622)
what is the time
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
what is the time
23:03
```

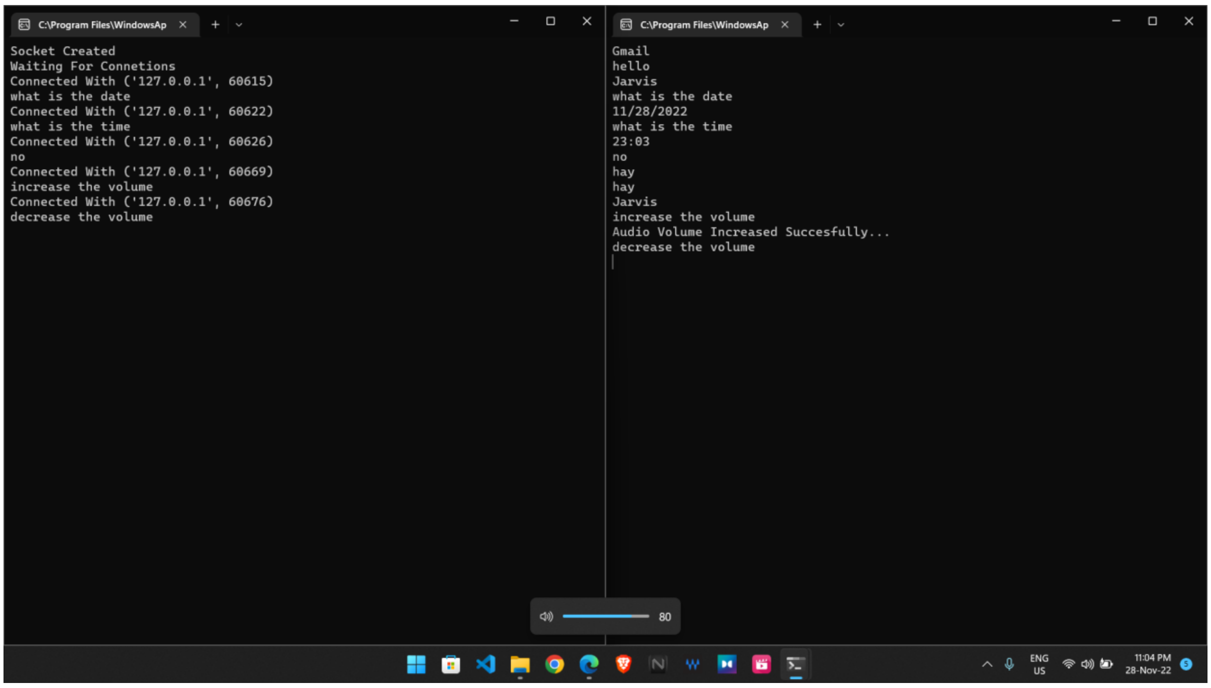
Fig 5.7: Executing time command



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
Connected With ('127.0.0.1', 60622)
what is the time
Connected With ('127.0.0.1', 60626)
no
Connected With ('127.0.0.1', 60669)
increase the volume
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
what is the time
23:03
no
hay
hay
Jarvis
increase the volume
```

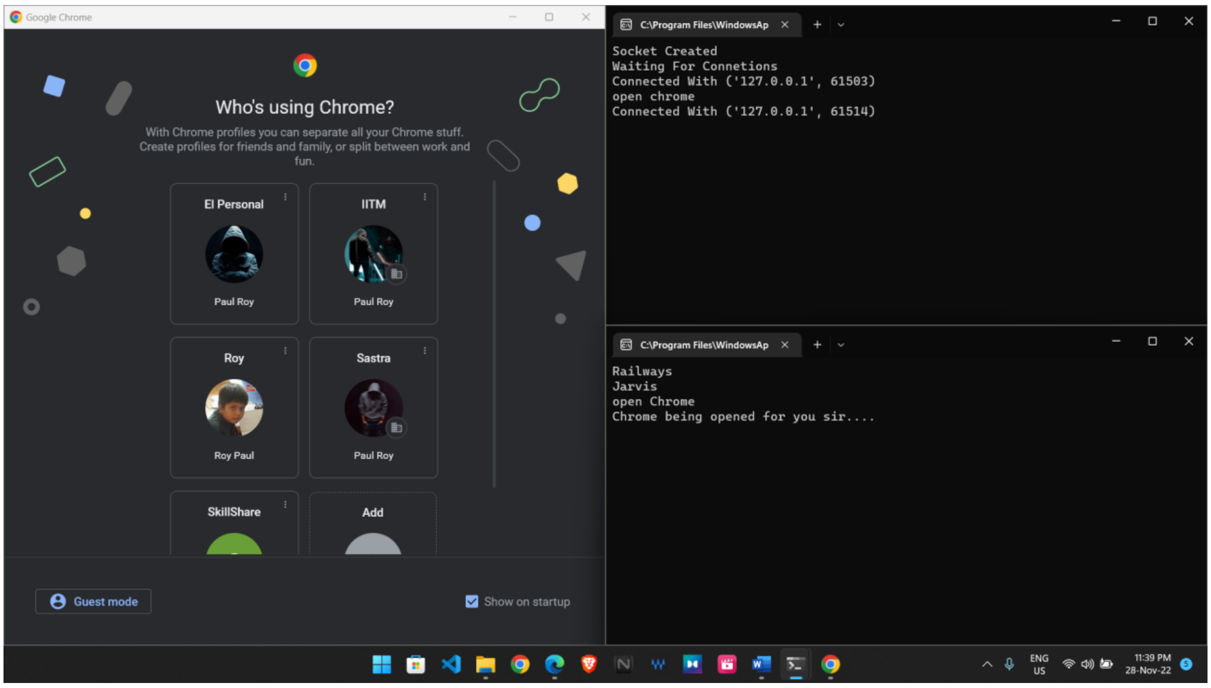
Fig 5.8:Executing Volume increase Function



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 60615)
what is the date
Connected With ('127.0.0.1', 60622)
what is the time
Connected With ('127.0.0.1', 60626)
no
Connected With ('127.0.0.1', 60669)
increase the volume
Connected With ('127.0.0.1', 60676)
decrease the volume
```

```
Gmail
hello
Jarvis
what is the date
11/28/2022
what is the time
23:03
no
hay
hay
Jarvis
increase the volume
Audio Volume Increased Succesfully...
decrease the volume
```

Fig 5.9: Executing volume decrease function

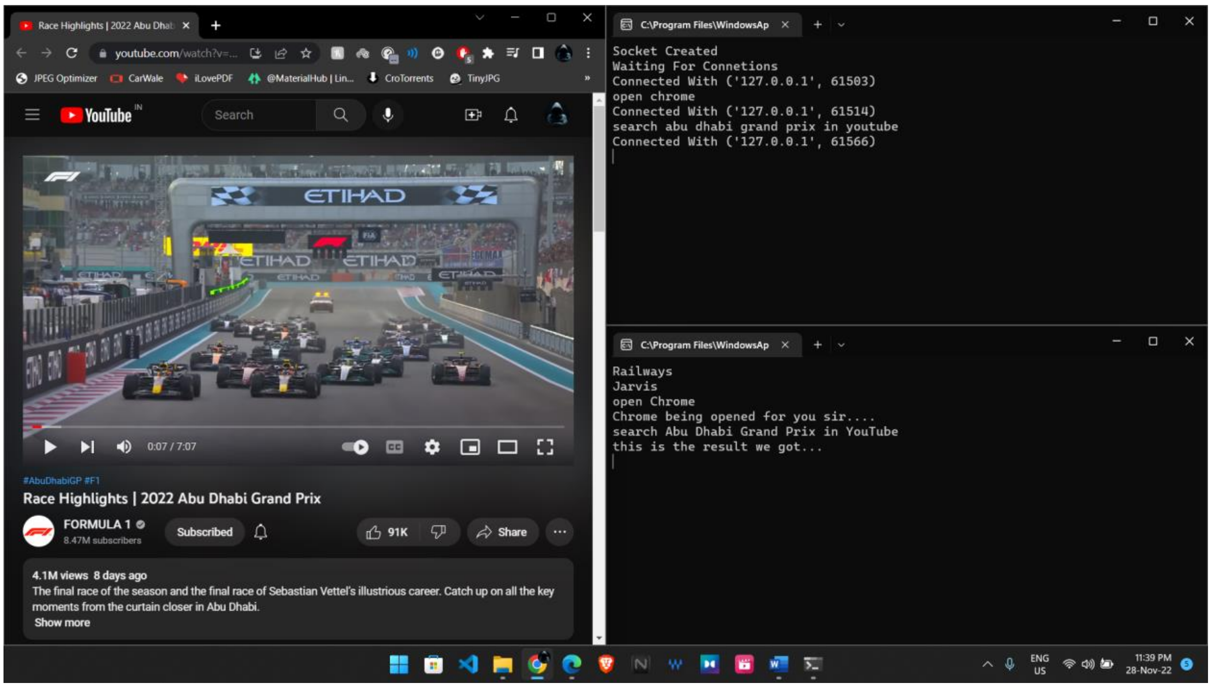


Google Chrome
Who's using Chrome?
With Chrome profiles you can separate all your Chrome stuff. Create profiles for friends and family, or split between work and fun.
El Personal
Paul Roy
IITM
Paul Roy
Roy
Roy Paul
Sastra
Paul Roy
SkillShare
Add
Guest mode
Show on startup
C:\Program Files\WindowsAp
ENG US
11:39 PM
28-Nov-22

```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 61503)
open chrome
Connected With ('127.0.0.1', 61514)
```

```
Railways
Jarvis
open Chrome
Chrome being opened for you sir....
```

Fig 5.10: Executing Open application functions

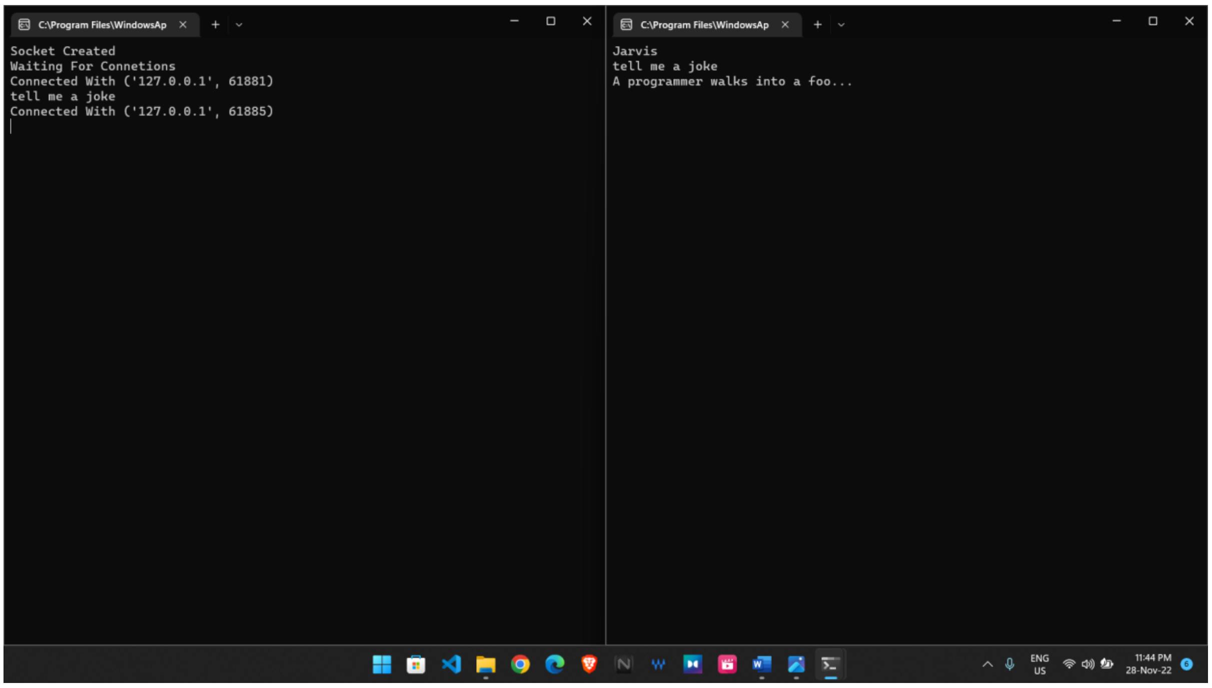


```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 61503)
open chrome
Connected With ('127.0.0.1', 61514)
search abu dhabi grand prix in youtube
Connected With ('127.0.0.1', 61566)
```

```
Railways
Jarvis
open Chrome
Chrome being opened for you sir....
search Abu Dhabi Grand Prix in YouTube
this is the result we got...
```

Race Highlights | 2022 Abu Dhabi Grand Prix
FORMULA 1
8.47M subscribers
Subscribed
91K
Share
4.1M views 8 days ago
The final race of the season and the final race of Sebastian Vettel's illustrious career. Catch up on all the key moments from the curtain closer in Abu Dhabi.
Show more

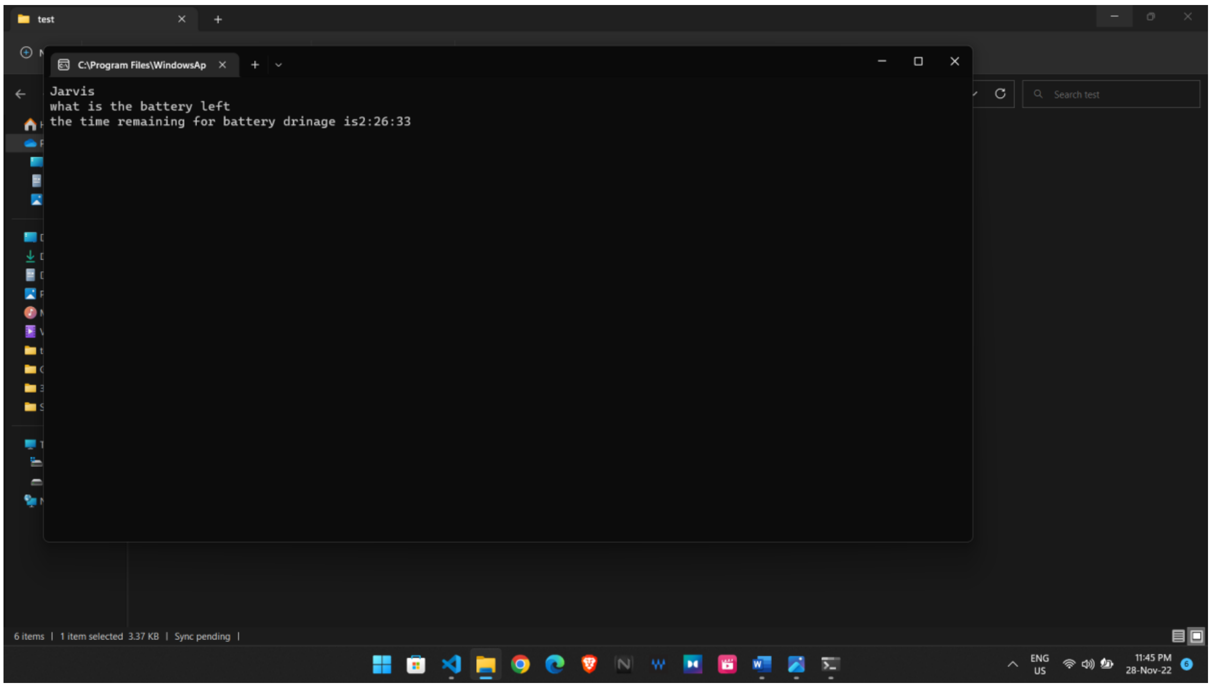
Fig 5.11:Opening Youtube video function



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 61881)
tell me a joke
Connected With ('127.0.0.1', 61885)
```

```
Jarvis
tell me a joke
A programmer walks into a foo...
```

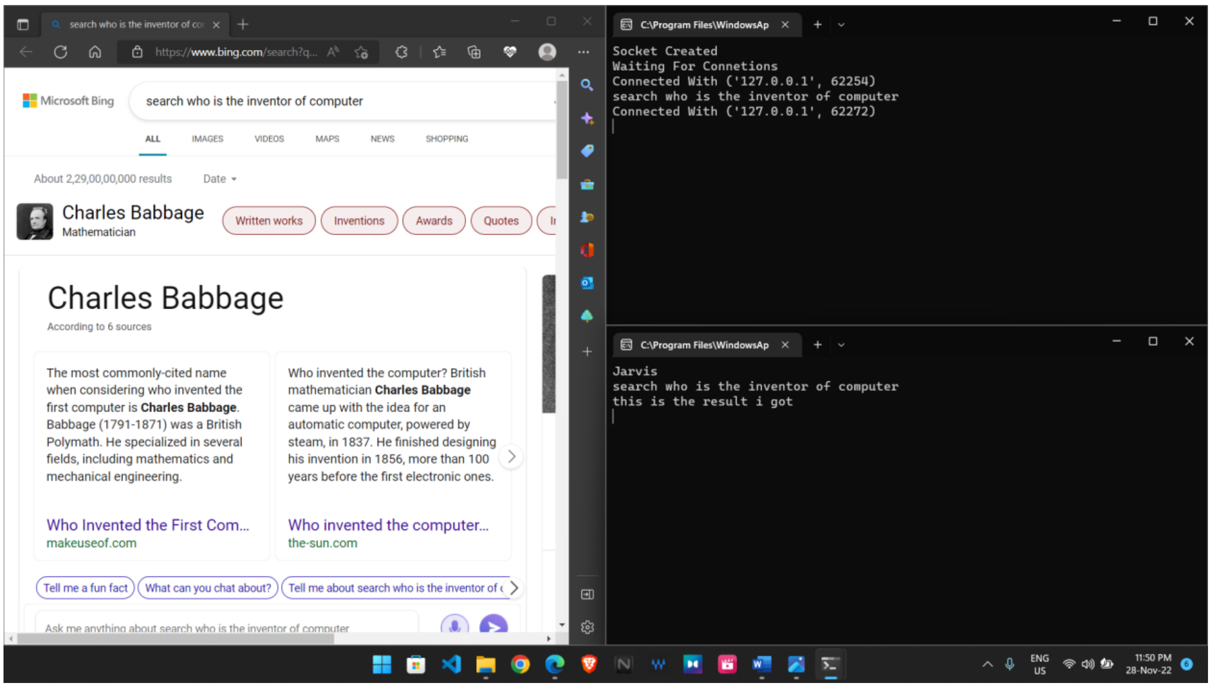
Fig 5.12: Telling a joke function



```
Jarvis
what is the battery left
the time remaining for battery drinage is2:26:33
```

test
C:\Program Files\WindowsAp
Search test
6 items | 1 item selected 3.37 KB | Sync pending |
11:45 PM
28-Nov-22

Fig 5.13: Battery Left function



```
Socket Created
Waiting For Connetions
Connected With ('127.0.0.1', 62254)
search who is the inventor of computer
Connected With ('127.0.0.1', 62272)
```

```
Jarvis
search who is the inventor of computer
this is the result i got
```

Charles Babbage

According to 6 sources

The most commonly-cited name when considering who invented the first computer is **Charles Babbage**. Babbage (1791-1871) was a British Polymath. He specialized in several fields, including mathematics and mechanical engineering.

Who Invented the First Com...
makeuseof.com

Who invented the computer? British mathematician **Charles Babbage** came up with the idea for an automatic computer, powered by steam, in 1837. He finished designing his invention in 1856, more than 100 years before the first electronic ones.

Who invented the computer...
the-sun.com

search who is the inventor of co
https://www.bing.com/search?q...
Microsoft Bing
search who is the inventor of computer
ALL
IMAGES
VIDEOS
MAPS
NEWS
SHOPPING
About 2,29,00,00,000 results
Date
Charles Babbage
Mathematician
Written works
Inventions
Awards
Quotes
Tell me a fun fact
What can you chat about?
Tell me about search who is the inventor of
Ask me anything about search who is the inventor of computer
C:\Program Files\WindowsAp
11:50 PM
28-Nov-22

Fig 5.14: Search Function

6. MERITS AND DEMERITS

MERITS OF DEDICATED EMPLOYEE USING SOCKET PROGRAMING:

1. Before getting the command the connection is made making it reliable.
2. Multiple Users can access the server making it reusable.
3. Lots of customization is available because its open source.
4. Performance of the app never degrade because it is developed using a powerful framework.
5. Keeps running in the background which able the user to multitask.

DEMERITS OF DEDICATED EMPLOYEE USING SOCKET PROGRAMING:

1. Since the connection is made earlier before the command it has tendencies close the program before the command is accepted.
2. Open-source code can also be vulnerable to tampers.
3. Too much load to the server can crash the server

7. CONCLUSION AND FUTURE SCOPE

There is always some place for enhancements in any software application, however good and efficient the application may be.

Right now, we are dealing with immediate needs of the user from the Dedicated Employee. In the future the application may further developed to include some features such as

1. Reminder Creation.
2. Making it more interactive.
3. Calling functions.
4. Messaging
5. Message Readout
6. More application to start as users downloads
7. Access more remote Electronics
8. More friendly voices

Since the use of the dedicated employee increases day by day it should regularly updated with features and should introduces market first features to keep the user hooked to our services. There is buck load of other Dedicated Employee System in market we should compete with. So the updates regularly and suggestion from the user should be reguraliry heard.

8. REFERENCES

1. Peters, Tim (19 August 2004). "PEP 20 – The Zen of Python". *Python Enhancement Proposals*. Python Software Foundation. Archived from the original on 26 December 2018. Retrieved 24 November 2008.

2. *"General Python FAQ — Python 3.9.2 documentation"*. docs.python.org. *Archived* from the original on 24 October 2012. *Retrieved 28 March 2021*

3. Mathis, Matt; Dukkipati, Nandita; Cheng, Yuchung (May 2013). "RFC 6937 - Proportional Rate Reduction for TCP". Retrieved 6 June 2014.

4. "www-306.ibm.com - AnyNet Guide to Sockets over SNA". Archived from the original on 2008-05-03. Retrieved 2006-09-07.

5. Kongthon, Alisa; Sangkeettrakarn, Chatchawal; Kongyoung, Sarawoot; Haruechaiyasak, Choochart (1 January 2009). *Implementing an Online Help Desk System Based on Conversational Agent*. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*. MEDES '09. New York, NY, USA: ACM. pp. 69:450–69:451. doi:10.1145/1643823.1643908. ISBN 9781605588292. S2CID 1046438.