client.py

server.py

--------------------server.py----------------------

import socket

from threading import Thread

# Server's IP address

SERVER\_HOST = "192.168.0.107"

SERVER\_PORT = 8080  # port we want to use

separator\_token = "<SEP>"  # we will use this to separate the client name & message

# Initialize list/set of all connected client's sockets

client\_sockets = set()

# Create a TCP socket

s = socket.socket()

# Make the port as reusable port

s.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

# Bind the socket to the address we specified

s.bind((SERVER\_HOST, SERVER\_PORT))

# Listen for upcoming connections

s.listen(5)

print(f"[\*] Listening as {SERVER\_HOST}:{SERVER\_PORT}")

def listen\_for\_client(cs):

    """

    This function keeps listening for a message from `cs` socket.

    Whenever a message is received, broadcast it to all other connected clients.

    """

    while True:

        try:

            msg = cs.recv(1024).decode()

        except Exception as e:

            print(f"[!] Error: {e}")

            client\_sockets.remove(cs)

        else:

            msg = msg.replace(separator\_token, ": ")

            for client\_socket in client\_sockets:

                client\_socket.send(msg.encode())

def accept\_connections():

    while True:

        client\_socket, client\_address = s.accept()

        print(f"[+] {client\_address} connected.")

        client\_sockets.add(client\_socket)

        t = Thread(target=listen\_for\_client, args=(client\_socket,))

        t.daemon = True

        t.start()

if \_\_name\_\_ == "\_\_main\_\_":

    accept\_connections()

--------------------client.py----------------------

import socket

from threading import Thread

# Server's IP address

SERVER\_HOST = "192.168.0.107"

SERVER\_PORT = 8080  # server's port

separator\_token = "<SEP>"  # we will use this to separate the client name & message

# Initialize TCP client socket

s = socket.socket()

# Connect to the server

s.connect((SERVER\_HOST, SERVER\_PORT))

def listen\_for\_messages():

    while True:

        message = s.recv(1024).decode()

        print("\n" + message)

# Start listening for messages in a separate thread

Thread(target=listen\_for\_messages).start()

# Prompt the client for a name

name = input("Enter your name: ")

while True:

    # Input message we want to send to the server

    to\_send = input("Enter a message: ")

    # A way to exit the program

    if to\_send.lower() == 'q':

        break

    # Add the name and the separator token to the message

    to\_send = f"{name}{separator\_token}{to\_send}"

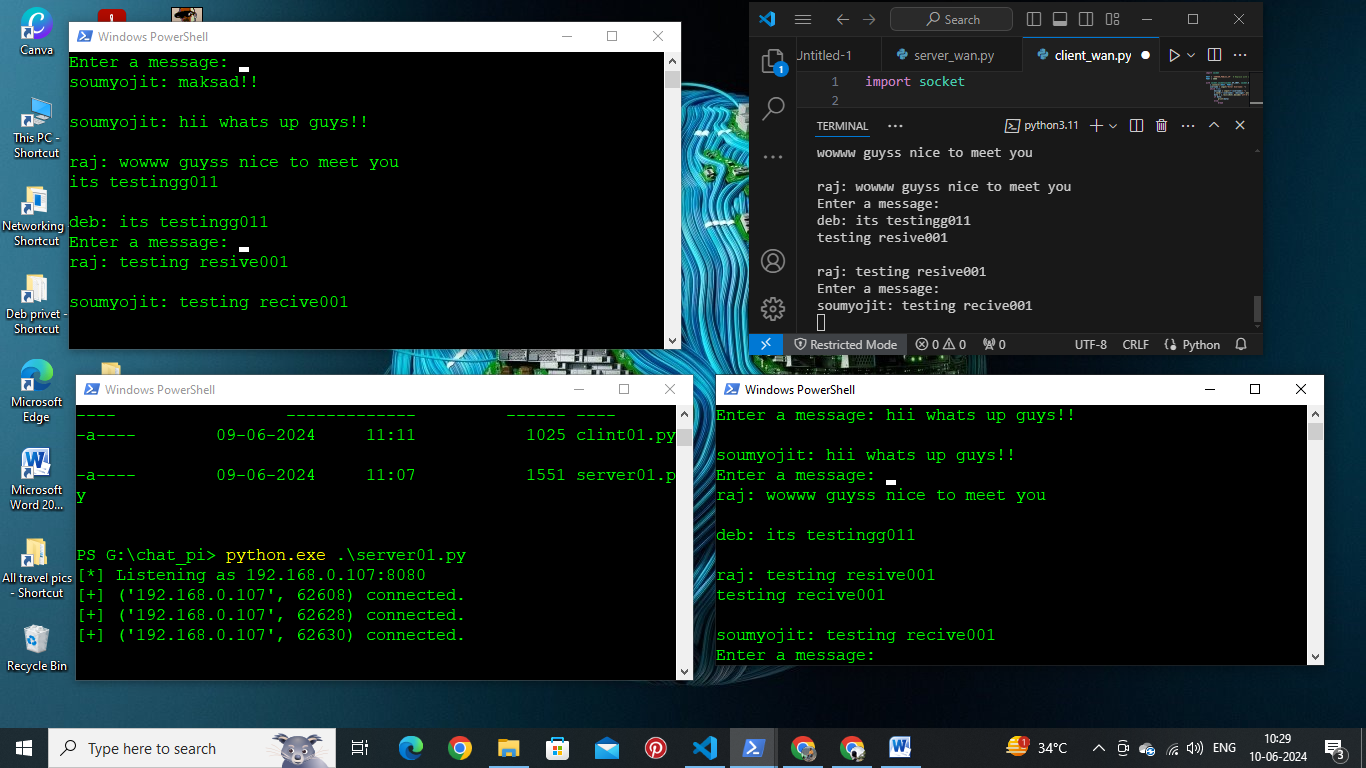
    # Finally, send the message

    s.send(to\_send.encode())

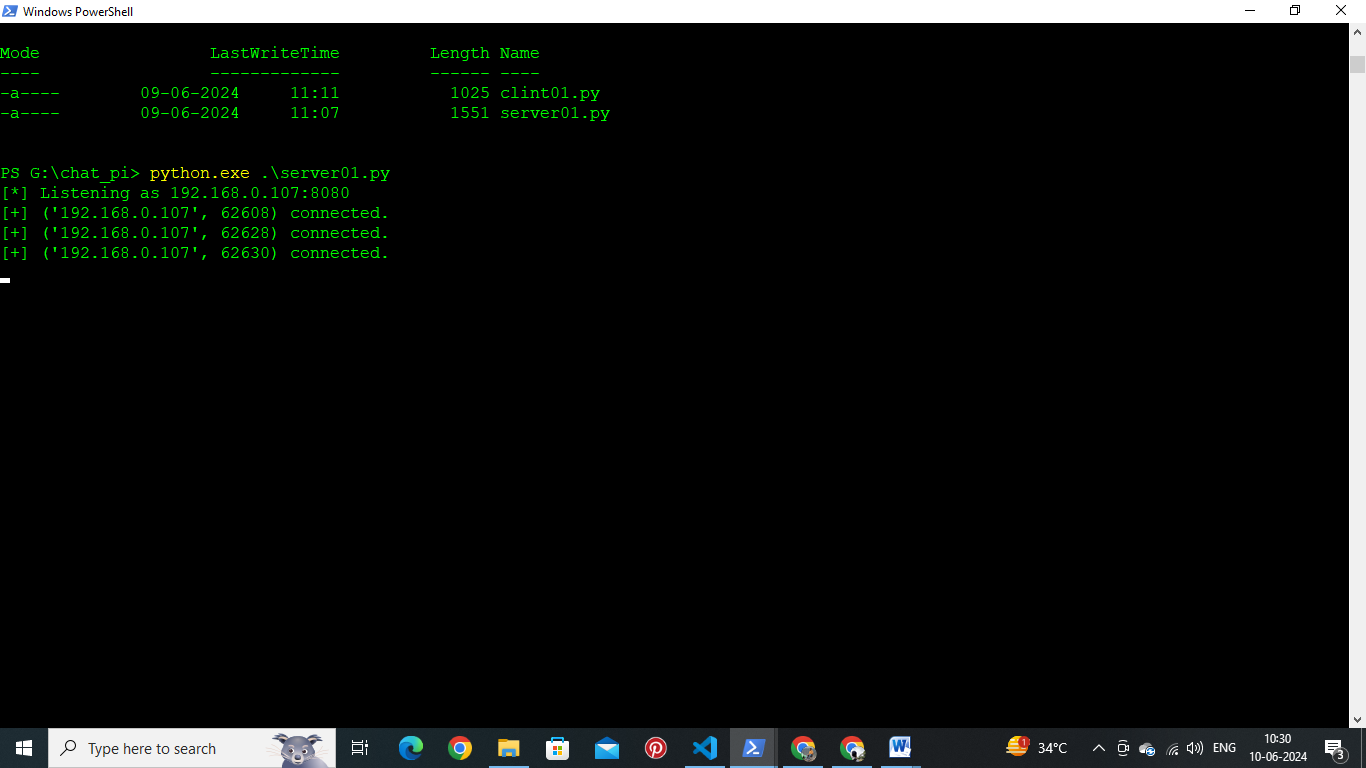
# Close the socket

s.close()

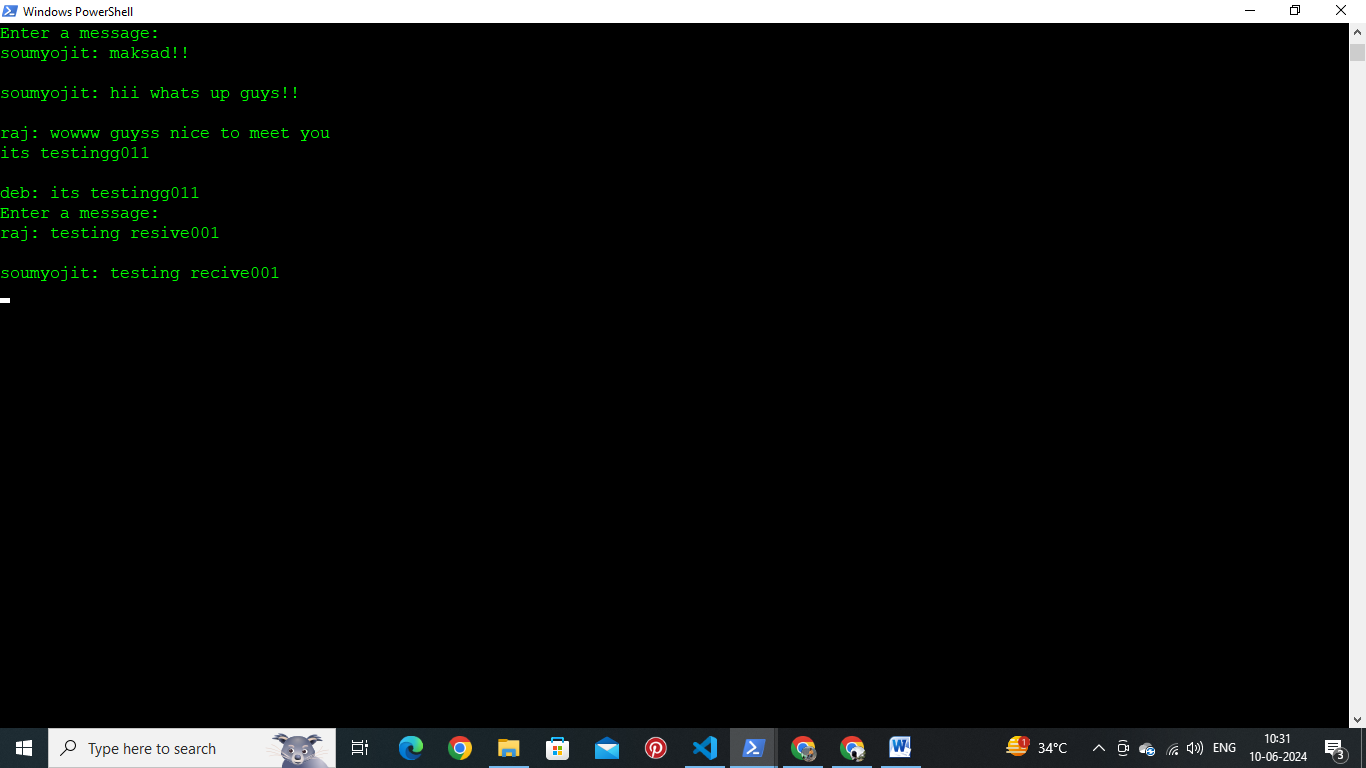
Output:



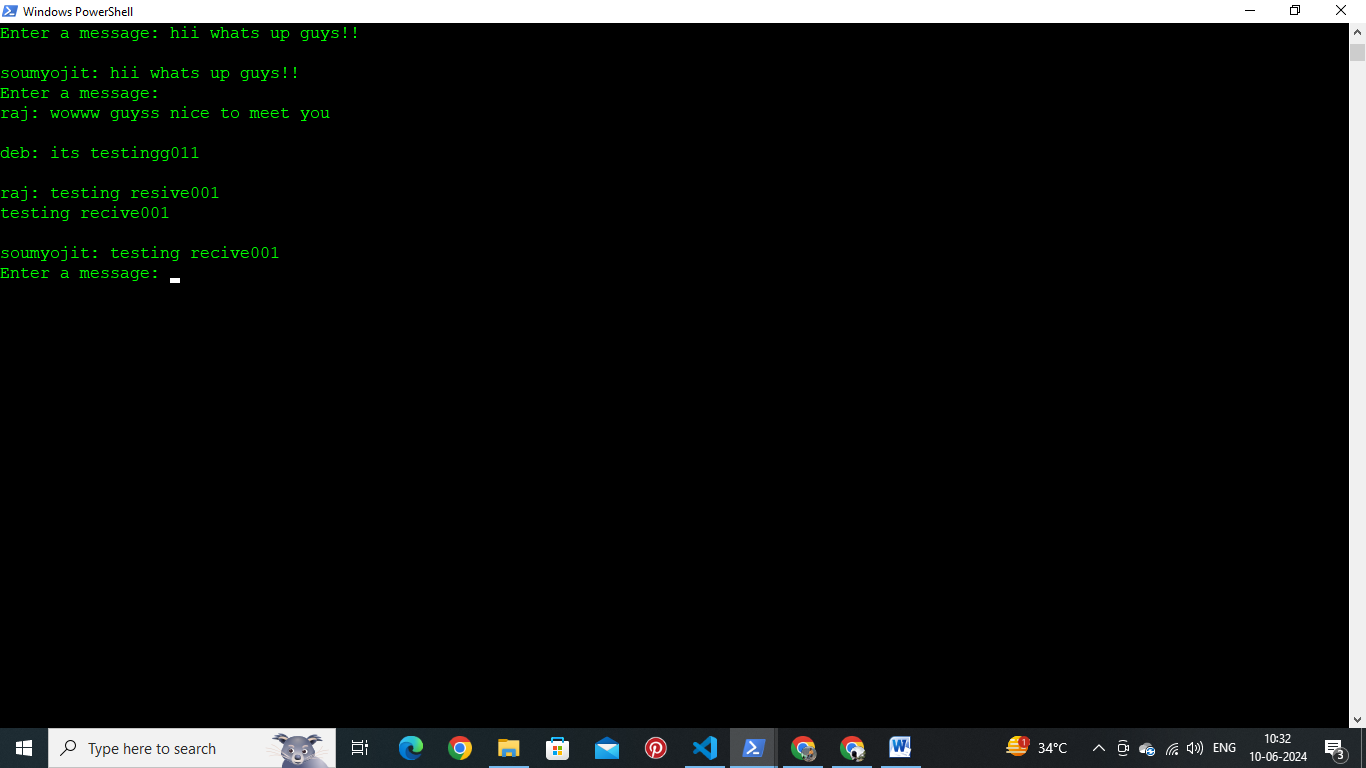
**Server:**



Client01:



Client02:



Client03:

