Name : Kansara Anjali | Class : A | Branch : Cyber Security
Semester: 5 | Enrollment No: 23162171032|Batch:52
Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub:CN
Practical 7

Aim: To implement Socket Programing

Scenario:

An organization named Albert Enterprise has established two departments for better performance of the company, as each department will be having some specific set of tasks to perform. So, this will reduce the time and increase the efficiency of the work. As both the departments are dependent on each other, they need to communicate more frequently. To solve the problem, the IT department has suggested the option to create a chat application using socket programming which will work only in the office premises. So, help the IT professionals to create the chat application.

Make sure that the application has the below mentioned features:

1) Department 1 will be set as the SERVER while department 2 will be set as a CLIENT device.

2) The message received by CLIENT or SERVER must be displayed with a time stamp.

3) If any of the devices irrespective of CLIENT or SERVER has sent the message that the “quit”, then connection should be closed on both the ends.

4) There is no restriction on the protocol selection, you can use UDP or TCP. Justify the reason for selection of the specific protocol.

Expected Output:

```
----------
  SERVER
----------
SERVER is listening..
Connection accepted from ('192.168.1.6', 53792)
CLIENT [2024-10-06 21:13:08]:  Hi I am Client 1
ENTER TEXT: I am Server
CLIENT [2024-10-06 21:13:32]:  My task is done
ENTER TEXT: quit
CLIENT [2024-10-06 21:13:36]:
```

```
----------
  CLIENT
----------
Hello there! msg from SERVER
ENTER TEXT: Hi I am Client 1
SERVER [2024-10-06 21:13:17]:  I am Server
ENTER TEXT: My task is done
SERVER [2024-10-06 21:13:36]:  quit
```

Server Code:

```
import socket

from datetime import datetime
```

```python
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(("0.0.0.0", 5050))
server.listen(1)

print("----------- SERVER -----------")
print("SERVER is listening..")

conn, addr = server.accept()
print("Connection accepted from", addr)

# Send welcome message to client
conn.send("Hello there! msg from SERVER".encode())

while True:
    msg = conn.recv(1024).decode()
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    print(f"CLIENT [{timestamp}]: {msg}")

    if msg.lower() == "quit":
        break

    text = input("ENTER TEXT: ")
    conn.send(text.encode())
```

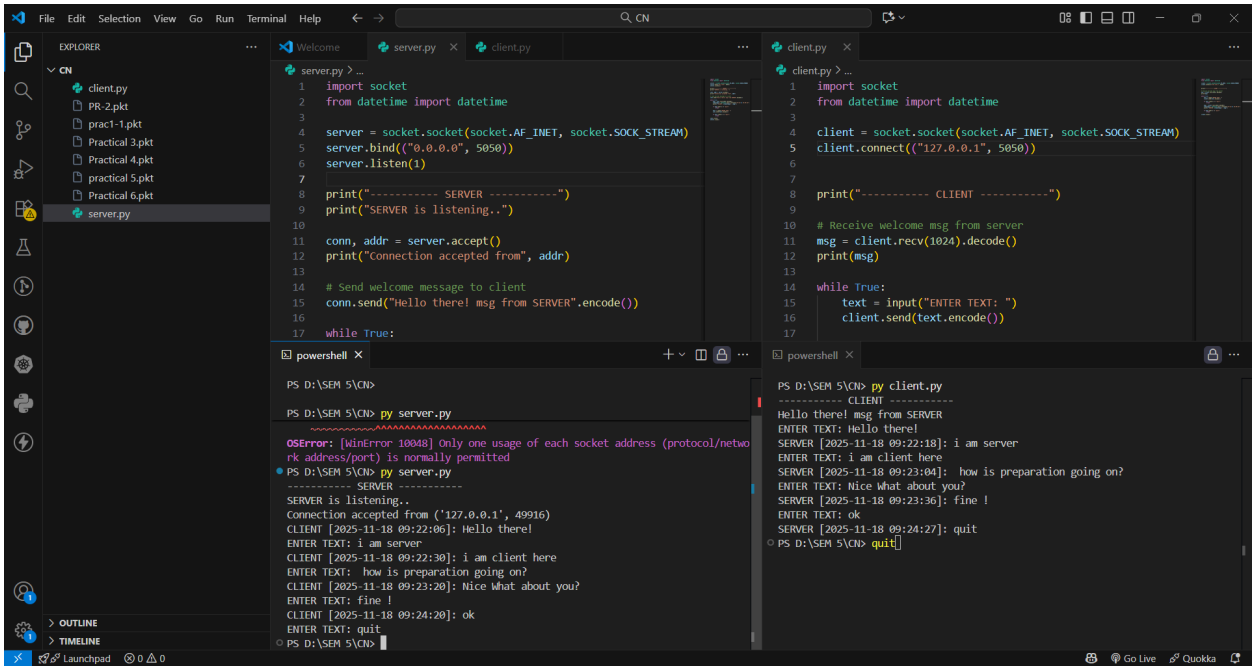
```
    if text.lower() == "quit":

        break


conn.close()

server.close()
```

Client Code:

```
import socket

from datetime import datetime


client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

client.connect(("127.0.0.1", 5050))




print("----------- CLIENT -----------")


# Receive welcome msg from server

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

print(msg)


while True:

    text = input("ENTER TEXT: ")
```

```
    client.send(text.encode())


    if text.lower() == "quit":
        break


    msg = client.recv(1024).decode()
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    print(f"SERVER [{timestamp}]: {msg}")


    if msg.lower() == "quit":
        break


client.close()
```

Output:



Conclusion:

The socket-based chat application successfully enables real-time communication between the two departments of Albert Enterprise. Department 1 operates as the SERVER while Department 2 functions as the CLIENT, and all messages are sent with proper timestamps. The application also ensures that if either side sends the message “quit”, the connection is terminated safely for both ends.