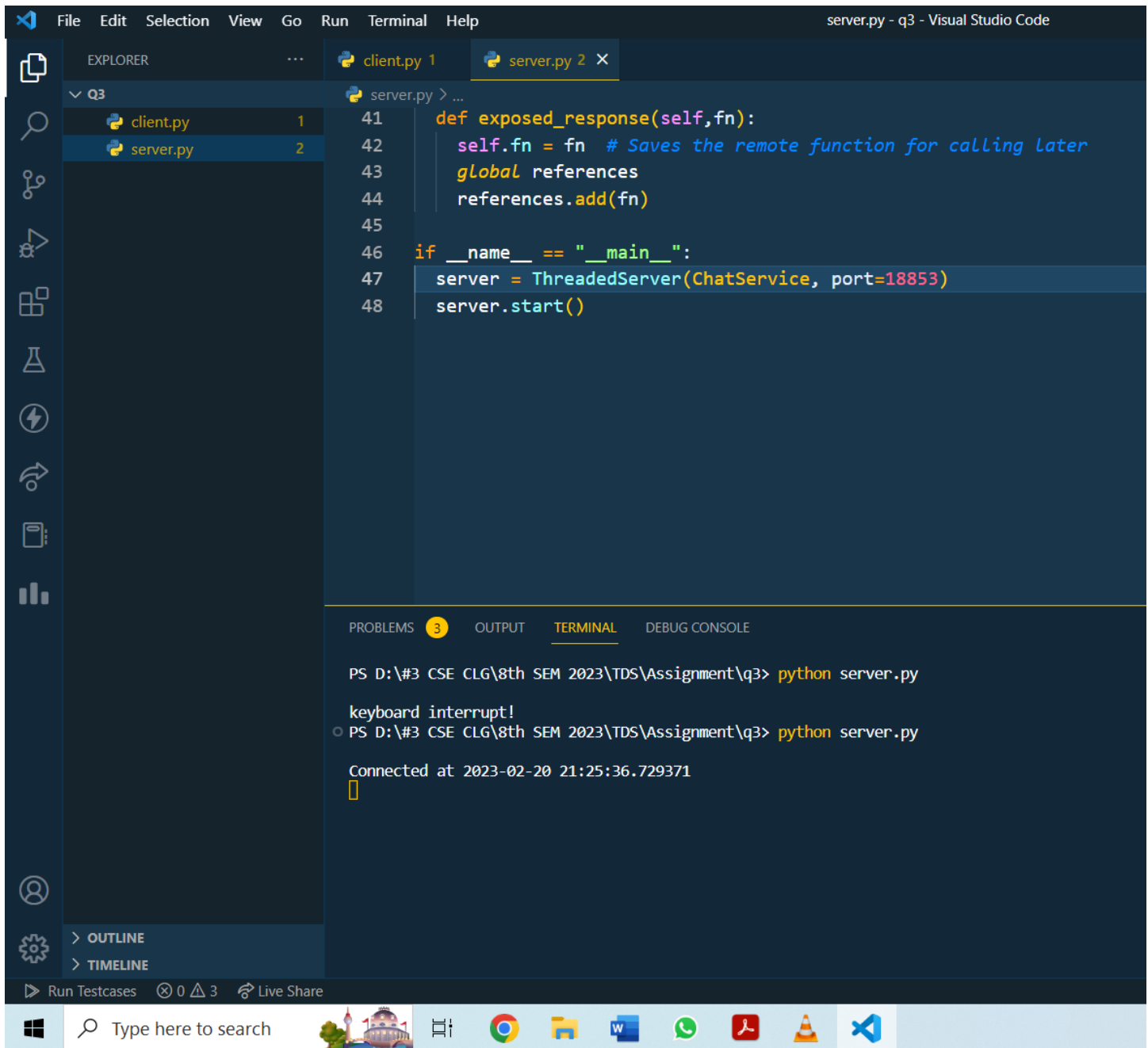


Enroll no. BT19CSE036

Name: Hasan Koser

### Question-3

Initially we will start the server then we will open a client file. The client which will get connected will look something like this



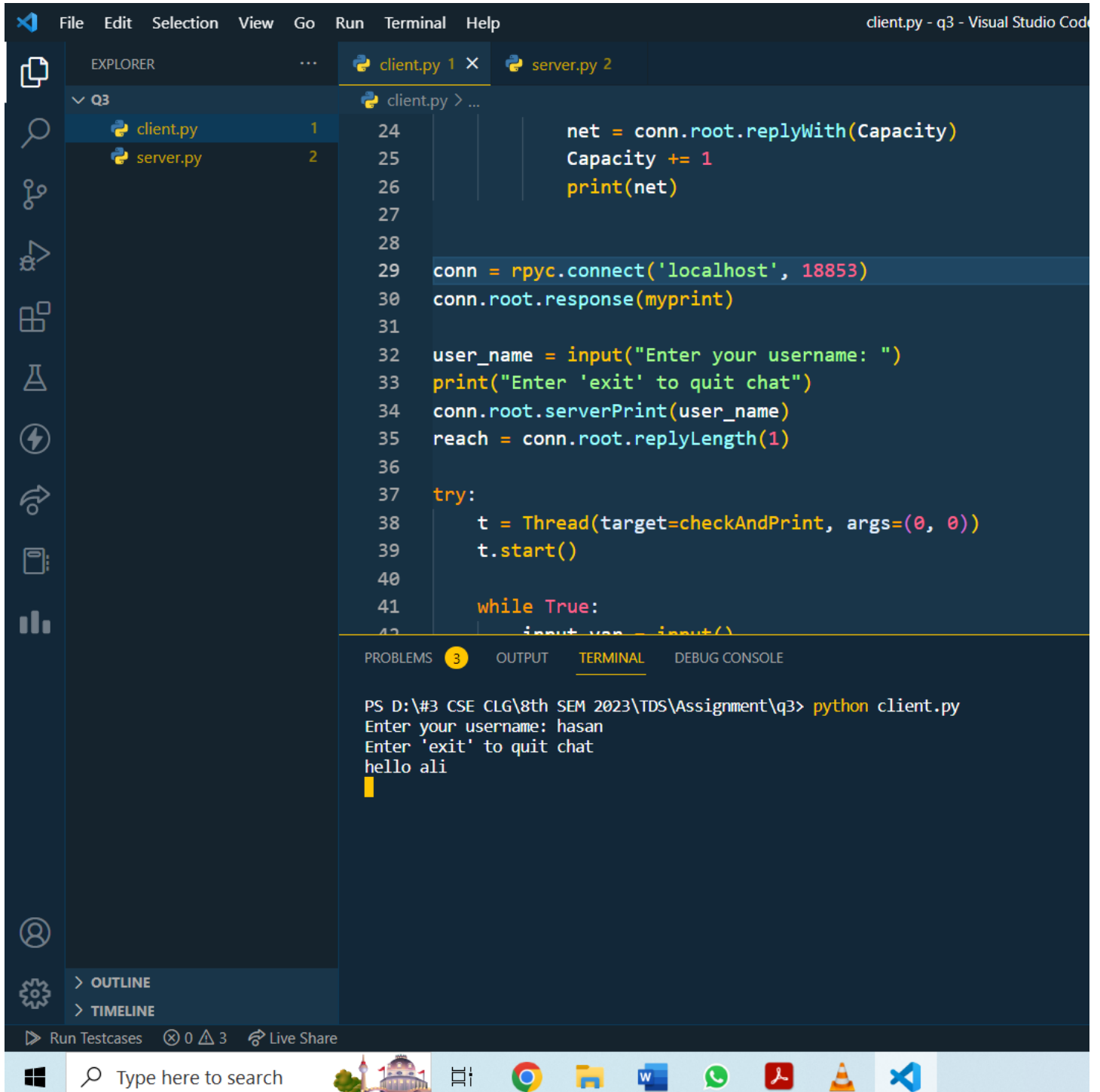
The screenshot displays the Visual Studio Code interface with two files open: `client.py` and `server.py`. The `server.py` file contains the following Python code:

```
41 def exposed_response(self,fn):
42     self.fn = fn # Saves the remote function for calling later
43     global references
44     references.add(fn)
45
46 if __name__ == "__main__":
47     server = ThreadedServer(ChatService, port=18853)
48     server.start()
```

The bottom panel shows the TERMINAL output, indicating the server has been started and is listening for connections. The output shows a keyboard interrupt followed by the server starting again and connecting to a client at 2023-02-20 21:25:36.729371.

```
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py
keyboard interrupt!
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py
Connected at 2023-02-20 21:25:36.729371
█
```

After starting the server we will type something from client side.



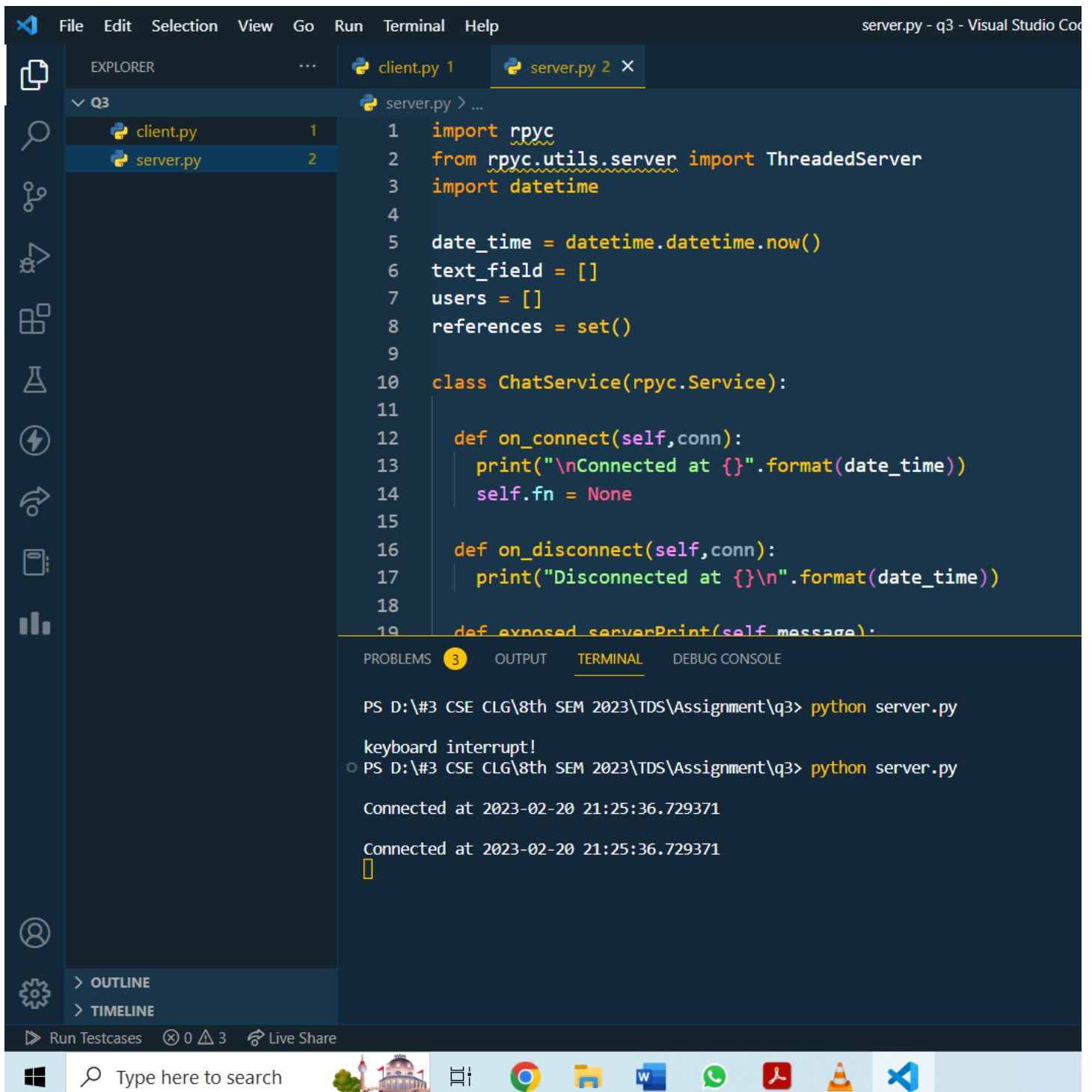
The screenshot displays the Visual Studio Code interface with a Python client program open in the editor. The Explorer sidebar on the left shows a project named 'Q3' containing two files: 'client.py' and 'server.py'. The editor window shows the code for 'client.py', which includes a function to connect to a server, send a message, and receive a response. The code is as follows:

```
24 net = conn.root.replyWith(Capacity)
25 Capacity += 1
26 print(net)
27
28
29 conn = rpyc.connect('localhost', 18853)
30 conn.root.response(myprint)
31
32 user_name = input("Enter your username: ")
33 print("Enter 'exit' to quit chat")
34 conn.root.serverPrint(user_name)
35 reach = conn.root.replyLength(1)
36
37 try:
38     t = Thread(target=checkAndPrint, args=(0, 0))
39     t.start()
40
41     while True:
42         input_name = input()
```

The TERMINAL panel at the bottom shows the execution of the program. It starts with the command 'python client.py' and shows the user input 'hasan', followed by the server response 'hello ali'.

```
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python client.py
Enter your username: hasan
Enter 'exit' to quit chat
hello ali
```

Now we will connect one more client here and see the realtime conversation.  
When a new connection is connected to server it will look something like this



The screenshot displays the Visual Studio Code interface with a Python project named 'q3'. The Explorer sidebar on the left shows two files: 'client.py' and 'server.py'. The main editor window is open to 'server.py', which contains the following code:

```
1 import rpyc
2 from rpyc.utils.server import ThreadedServer
3 import datetime
4
5 date_time = datetime.datetime.now()
6 text_field = []
7 users = []
8 references = set()
9
10 class ChatService(rpyc.Service):
11
12     def on_connect(self, conn):
13         print("\nConnected at {}".format(date_time))
14         self.fn = None
15
16     def on_disconnect(self, conn):
17         print("Disconnected at {}\n".format(date_time))
18
19     def exposed_serverPrint(self, message):
```

Below the code editor, the TERMINAL tab is active, showing the execution of the server script. The output indicates that the server was interrupted and then restarted, with two successful connections recorded at the same timestamp:

```
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py
keyboard interrupt!
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py
Connected at 2023-02-20 21:25:36.729371
Connected at 2023-02-20 21:25:36.729371
█
```

The bottom of the image shows the Windows taskbar with various application icons and a search bar.

New connection will chat for sometime and leave first

The image shows a Visual Studio Code window with the following components:

- Explorer:** Shows a folder named 'Q3' containing two files: 'client.py' (1 line) and 'server.py' (2 lines).
- Editor:** Displays the content of 'server.py' with the following code:

```
1 import rpyc
2 from rpyc.utils.server import ThreadedServer
3 import datetime
4
5 date_time = datetime.datetime.now()
6 text_field = []
7 users = []
8 references = set()
9
10 class ChatService(rpyc.Service):
11
12     def on_connect(self, conn):
13         print("\nConnected at {}".format(date_time))
14         self.fn = None
15
16     def on_disconnect(self, conn):
17         print("Disconnected at {}\n".format(date_time))
18
19     def exposed_serverPrint(self, message):
```
- Terminal:** Shows the execution of 'python client.py' and a chat session:

```
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python client.py
Enter your username: ali
Enter 'exit' to quit chat
hasan:hello ali
hello hasan
hasan:ok bye
i will leave first
exit
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3>
```
- Bottom Bar:** Includes 'Run Testcases', '0 errors', '3 warnings', 'Live Share' button, and a Windows taskbar with various application icons.

After that it will look like this in other client's screen who will leave then

server.py - q3 - Visual Stud

File Edit Selection View Go Run Terminal Help

EXPLORER

Q3

- client.py 1
- server.py 2

server.py > ...

```
1 import rpyc
2 from rpyc.utils.server import ThreadedServer
3 import datetime
4
5 date_time = datetime.datetime.now()
6 text_field = []
7 users = []
8 references = set()
9
10 class ChatService(rpyc.Service):
11
12     def on_connect(self, conn):
13         print("\nConnected at {}".format(date_time))
14         self.fn = None
15
16     def on_disconnect(self, conn):
17         print("Disconnected at {}\n".format(date_time))
18
19     def exposed_serverPrint(self, message):
```

PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python client.py  
Enter your username: hasan  
Enter 'exit' to quit chat  
hello ali  
ali has entered the chat  
ali:hello hasan  
ok bye  
ali:i will leave first  
ali has left the conversation  
okay  
exit  
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3>

> OUTLINE  
> TIMELINE

Run Testcases 0 3 Live Share

Type here to search

Now we close the server side which will look like this

The screenshot displays the Visual Studio Code interface with the following components:

- Explorer Panel:** Shows a project named 'Q3' containing two files: 'client.py' and 'server.py'.
- Code Editor:** Displays the content of 'server.py' with the following code:

```
1 import rpyc
2 from rpyc.utils.server import ThreadedServer
3 import datetime
4
5 date_time = datetime.datetime.now()
6 text_field = []
7 users = []
8 references = set()
9
10 class ChatService(rpyc.Service):
11
12     def on_connect(self, conn):
13         print("\nConnected at {}".format(date_time))
14         self.fn = None
15
16     def on_disconnect(self, conn):
17         print("Disconnected at {}\n".format(date_time))
18
19     def exposed_serverPrint(self, message):
```
- Terminal Panel:** Shows the execution of the server script. The command 'python server.py' is run, followed by a 'keyboard interrupt!' message. Subsequent runs show connection and disconnection logs:

```
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py
keyboard interrupt!
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3> python server.py

Connected at 2023-02-20 21:25:36.729371

Connected at 2023-02-20 21:25:36.729371
Disconnected at 2023-02-20 21:25:36.729371

Disconnected at 2023-02-20 21:25:36.729371

keyboard interrupt!
PS D:\#3 CSE CLG\8th SEM 2023\TDS\Assignment\q3>
```
- Bottom Bar:** Includes 'Run Testcases', '0 3' (indicating 0 errors and 3 warnings), and 'Live Share' options.