```
=======================
server.py
=======================

import socket
import threading
import datetime


client_counter = 0


def handle_client(conn, addr, client_name):
    print(f"{client_name} connected from {addr[0]}:{addr[1]}")

    while True:
        data = conn.recv(1024).decode().strip()

        if not data:
            break

        if data == "QUIT":
            print(f"{client_name} disconnected.")
            conn.close()
            return

        if ":" not in data and data not in ["TIME", "DATE"]:
            response = f"ERROR: Unknown command from {client_name}."
            conn.send(response.encode())
            print(f"{client_name} sent: {data} → {response}")
            continue

        # BONUS: TIME & DATE
        if data == "TIME":
            result = datetime.datetime.now().strftime("%H:%M:%S")
            conn.send(result.encode())
            print(f"{client_name} sent: TIME → {result}")
            continue

        if data == "DATE":
            result = datetime.datetime.now().strftime("%Y-%m-%d")
            conn.send(result.encode())
            print(f"{client_name} sent: DATE → {result}")
            continue

        command, message = data.split(":", 1)
        command = command.strip()
        message = message.strip()

        # Execute commands
        if command == "UPPER":
            result = message.upper()
        elif command == "LOWER":
            result = message.lower()
        elif command == "REVERSE":
            result = message[::-1]
        elif command == "COUNT":
            result = str(len(message))
```

```python
        # BONUS: VOWELS
        elif command == "VOWELS":
            vowels = "aeiouAEIOU"
            count = sum(1 for ch in message if ch in vowels)
            result = str(count)
        else:
            result = f"ERROR: Unknown command from {client_name}."

        conn.send(result.encode())
        print(f"{client_name} sent: {data} → {result}")


def main():
    global client_counter

    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(("0.0.0.0", 2210))
    server.listen()

    print("Server is running on port 2210")

    while True:
        conn, addr = server.accept()

        client_counter += 1
        client_name = f"Client{client_counter}"

        thread = threading.Thread(target=handle_client, args=(conn, addr, cli-
ent_name))
        thread.start()


main()
```

Run in Terminal :

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS

(dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> python server.py
Server is running on port 2210
Client1 connected from 127.0.0.1:55262
Client1 sent: UPPER: Wireshark → WIRESHARK
Client1 sent: LOWER: Wireshark → wireshark
Client2 connected from 127.0.0.1:55308
Client2 sent: REVERSE: Wireshark → krahseriW
Client3 connected from 127.0.0.1:55333
Client3 sent: COUNT: Wireshark → 9
Client3 sent: VOWELS: Wireshark → 3
Client1 sent: TIME → 00:52:50
Client2 sent: DATE → 2025-11-27
Client1 disconnected.
Client3 sent: TIME → 00:53:19
Client3 sent: DATE → 2025-11-27
Client3 disconnected.
Client2 disconnected.

```
=======================
client.py
=======================

import socket

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(("127.0.0.1", 2210))

print("Connected to server.")

while True:
    msg = input("> ")

    client.send(msg.encode())

    if msg == "QUIT":
        print("Connection closed.")
        client.close()
        break

    data = client.recv(1024).decode()
    print("Response from server:", data)
```

Run in Terminal :

Client 1:

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS

```
● (dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> python client.py
  Connected to server.
  > UPPER: Wireshark
  Response from server: WIRESHARK
  > LOWER: Wireshark
  Response from server: wireshark
  > TIME
  Response from server: 00:52:50
  > QUIT
  Connection closed.
○ (dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> ▯
```

Client 2:

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS

```
● (dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> python client.py
  Connected to server.
  > REVERSE: Wireshark
  Response from server: krahseriW
  > DATE
  Response from server: 2025-11-27
  > QUIT
  Connection closed.
○ (dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> ▯
```

Client 3:

```
(dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> python client.py
Connected to server.
> COUNT: Wireshark
Response from server: 9
> VOWELS: Wireshark
Response from server: 3
> TIME
Response from server: 00:53:19
> DATE
Response from server: 2025-11-27
> QUIT
Connection closed.
(dl_env) PS C:\University\Semester 5\Computer Network\Labs\Lab 2> |
```

Wireshark:

Using filter `tcp.port == 2210` → *Shows packets that use TCP port 2210 only*