1)

```
utsav@utsav-victus :~ $ nslookup nitk.ac.in
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name: nitk.ac.in
Address: 10.11.0.79

utsav@utsav-victus :~ $
```

2)

```
utsav@utsav-victus :~ $ ping -c 4 nitk.ac.in
PING nitk.ac.in (10.11.0.79) 56(84) bytes of data.
64 bytes from new.nitk.ac.in (10.11.0.79): icmp_seq=1 ttl=63 time=0.151 ms
64 bytes from new.nitk.ac.in (10.11.0.79): icmp_seq=2 ttl=63 time=0.220 ms
64 bytes from new.nitk.ac.in (10.11.0.79): icmp_seq=3 ttl=63 time=0.163 ms
64 bytes from new.nitk.ac.in (10.11.0.79): icmp_seq=4 ttl=63 time=0.167 ms

--- nitk.ac.in ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.151/0.175/0.220/0.026 ms
```

1)

SERVER-

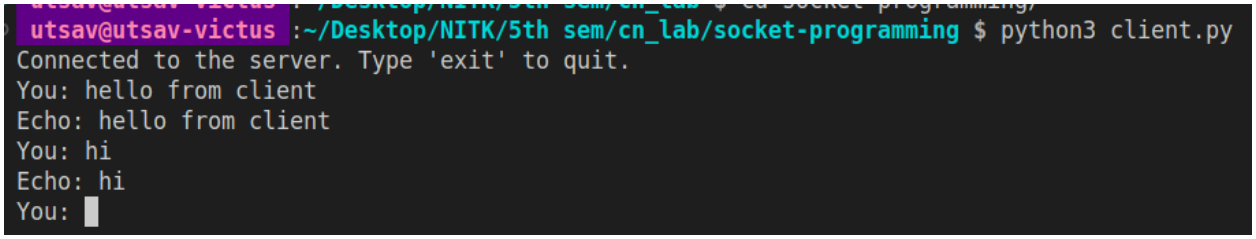
```
import socket
import threading

# Function to handle each client connection
def handle_client(client_socket, client_address):
    print(f"[+] New connection from {client_address}")

    while True:
        try:
            # Receive message from client
            message = client_socket.recv(1024).decode()
            if not message:
                break  # Client closed connection

            print(f"[{client_address}] {message}")

            # Echo the message back to client
            client_socket.send(message.encode())
```

```
        except ConnectionResetError:
            break  # Handle client crash/disconnect

    print(f"[-] Connection closed: {client_address}")
    client_socket.close()

def start_server(host="127.0.0.1", port=5555):
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((host, port))
    server.listen(5)

    print(f"[*] Server started on {host}:{port}")

    while True:
        client_socket, client_address = server.accept()
        # Create a new thread for each client
        client_thread = threading.Thread(target=handle_client, args=(client_socket,
client_address))
        client_thread.start()

if __name__ == "__main__":
    start_server()
```

CLIENT-

```
import socket

def start_client(host="127.0.0.1", port=5555):
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))

    print("Connected to the server. Type 'exit' to quit.")
    while True:
        message = input("You: ")
        if message.lower() == "exit":
            break

        client.send(message.encode())       # Send message
        echo = client.recv(1024).decode()   # Receive echo
        print("Echo:", echo)

    client.close()

if __name__ == "__main__":
```

```
start_client()
```



2)

SERVER-

```
import socket
import threading

# Hardcoded weather data
weather_data = {
    "delhi": {"temperature": "32°C", "humidity": "45%", "condition": "Sunny"},
    "mumbai": {"temperature": "28°C", "humidity": "70%", "condition": "Humid"},
    "bangalore": {"temperature": "24°C", "humidity": "55%", "condition": "Cloudy"},
    "chennai": {"temperature": "30°C", "humidity": "65%", "condition": "Rainy"},
}

def handle_client(client_socket, client_address):
    print(f"[+] Connected to {client_address}")

    while True:
        try:
            city = client_socket.recv(1024).decode().strip().lower()
            if not city:
                break

            print(f"[{client_address}] Requested weather for: {city}")

            if city in weather_data:
                report = (f"Weather in {city.capitalize()}:\n"
                          f"Temperature: {weather_data[city]['temperature']}\n"
                          f"Humidity: {weather_data[city]['humidity']}\n"
                          f"Condition: {weather_data[city]['condition']}")
            else:
                report = f"Sorry, no weather data available for {city}."

            client_socket.send(report.encode())

        except ConnectionResetError:
```

```
            break

    print(f"[-] Connection closed: {client_address}")
    client_socket.close()

def start_server(host="127.0.0.1", port=5556):
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((host, port))
    server.listen(5)

    print(f"[*] Weather Server running on {host}:{port}")

    while True:
        client_socket, client_address = server.accept()
        client_thread = threading.Thread(target=handle_client, args=(client_socket,
client_address))
        client_thread.start()

if __name__ == "__main__":
    start_server()
```

CLIENT-

```
import socket

def start_client(host="127.0.0.1", port=5556):
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((host, port))

    print("Connected to Weather Server. Type 'exit' to quit.")

    while True:
        city = input("Enter city name: ")
        if city.lower() == "exit":
            break

        client.send(city.encode())
        report = client.recv(1024).decode()
        print("\n--- Weather Report ---")
        print(report)
        print("----------------------\n")

    client.close()
```

```
if __name__ == "__main__":
    start_client()
```

```
utsav@utsav-victus :~/Desktop/NITK/5th sem/cn_lab/socket-programming $ python3 client.py
Connected to Weather Server. Type 'exit' to quit.
Enter city name: bangalore

--- Weather Report ---
Weather in Bangalore:
Temperature: 24°C
Humidity: 55%
Condition: Cloudy
----------------------
```

```
utsav@utsav-victus :~/Desktop/NITK/5th sem/cn_lab/socket-programming $ python3 server.py
[*] Weather Server running on 127.0.0.1:5556
[+] Connected to ('127.0.0.1', 52420)
[('127.0.0.1', 52420)] Requested weather for: bangalore
[('127.0.0.1', 52420)] Requested weather for: delhi
```