6/10/2025

Practical XII

AIM: Implement echo client server using TCP/UDP sockets.

```
import socket
import time

def ping_server(host="127.0.0.1", port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK DGRAM) as s:
        try:
            s.sendto(b"Hello", (host, port))
        except s.timeout:
            print("Request timed out")
ping_server()
```

```
import socket
def start_server(host="127.0.0.1", port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK DGRAM) as s:
        s.bind((host, port))
        print(f"Server listening on {host}:{port}")
        while True:
            data, addr = s.recvfrom(1024)
            print(f"Received from {addr}: {data.decode()}")
            s.sendto(b'pong', addr)
start_server()
```

Output:

```
Server listening on 127.0.0.1:12345
Received from ('127.0.0.1', 57621): Hello
Received from ('127.0.0.1', 53967): Hello
```

AIM: Implement chat client server using TCP/UDP sockets

```
import socket
def client (host='127.0.0.1', port=12345):
    with socket.socket (socket.AF_INET, socket.SOCK_DGRAM) as s:
        while True:
            msg=input ("You: ")
            s.sendto(msg.encode(),(host, port))
            data, addr = s.recvfrom (1024)
            print (f"Server: {data.decode()}")
client()
```

```
import socket
def server (host='127.0.0.1', port=12345):
    with socket.socket (socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.bind ((host, port))
        print (f"Server listening on {host}: {port}")
        while True:
            data, addr = s.recvfrom (1024)
            print (f"Client: {data.decode()}")
            msg= input ("You: ")
            s.sendto(msg.encode(), addr)
Server()
```

~~UDP~~

Output:

Server:

```
Server listening on 127.0.0.1:12345
 Client: Hi
 You: How are you?
 Client: I am fine
```

You: Cool

Client-end:

You: Hi

Server: How are you?

Yo: I am fihe

Server: Cool

Result:

Hence, echo and chat client server was executed succesfully.

13/11/2