EX NO:12a

DATE: 23.09.25

ECHO CLIENT SERVER USING TCP/UDP SOCKETS

Aim:

To implement echo client server using TCP/UDP sockets.

Algorithm:

a) Client

```
import socket
def ping_server(host = '127.0.0.1', port = 12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    as s:
        s.settimeout(2)
        message = input("Enter message to send: ").
                                        encode()

        s.sendto(message, (host, port))
        try:
            data, addr = s.recvfrom(1024)
            print("Echo from server: ", data.decode())
        except socket.timeout:
            print("Request timed out")

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

b) Server

```
import socket
def start_server(host = '127.0.0.1', port = 12345):
    with socket.socket(socket.AF_INET, sock.SOCK_DGRAM) as s:
        s.bind((host, port))
        print(f" UDP server running on {host}:{port}")
        while TRUE:
            data, addr = s.recvfrom(1024)
            print(f"Received message from {addr}: {data.
                                                decode()}
            s.sendto(data, addr)
```

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

Input & Output:

Client:

Enter message to send: Hello
Echo from server: Hello

Server:

UDP server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 54321): Hello

Result:

The program has been executed successfully.

13/10/25

Ex NO: 12b CHAT CLIENT SERVER USING TCP/UDP SOCKETS

DATE: 27.09.25

Aim:

To implement chat client server using TCP/UDP sockets.

a) Server

```
import socket
def start_server(host='127.0.0.1', port=12345):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((host, port))
    s.listen(1)
    print("Server is waiting for connection...")
    conn, addr = s.accept()
    print("Connected by", addr)
    while True:
        data = conn.recv(1024).decode()
        if not data or data.lower() == 'exit':
            print("Client ended the chat.")
            break
        print("Client:", data)
        msg = input("You: ")
        conn.sendall(msg.encode())
        if msg.lower() == 'exit':
            print("Chat ended.")
            break
    conn.close()
    s.close()
start_server()
```

b) Client

```
import socket
def start_client(host = '127.0.0.1', port = 12345):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((host, port))
    print("Connected to server.")
    while True:
        msg = input("You:")
        s.sendall(msg.encode())
        if msg.lower() == 'exit':
            print("Chat ended.")
            break
        data = s.recv(1024).decode()
        if not data or data.lower() == 'exit':
            print("Server ended the chat.")
            break
        print("Server: ", data)
    s.close()
start_client()
```

Output:

Server:

Server is waiting for connection
Connected by ('127.0.0.1', 61773)
Client: Hello
You: Hi
Client ended the chat

Client:

connected to server
You: Hello
Server: Hi
You: exit
Chat ended.

Result: The program has been executed successfully.

13/01