Ex. 12

Date : 28/10/24 Practical-12

Aim :-

a) Implement echo client server using TCP / UDP sockets.

client :

```
import socket
import time
def ping_server(host='127.0.0.1',
                port=12345):
    with socket.socket(socket.AF_INET,
        socket.SOC_dgram) as s:
    try:
        s.sendto(6"hello", (host, port))
    except s.timeout:
        print("Request timed out")
if __name__ == "main":
    ping_server();
```

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" UDP server running on {host}")
while True:
    data, add = s.recvform(1024):
    print(f"Received message from {addr}:
                {data.decode()}")
if __name__ == "__main__":
    start_server()
```

o/p python Server.py

VDP Server running on 127.0.0.1 : 12345
Received message from ('127.0.0.1', 59290), hello
python client.py
Received reply from server : Hello, client

b) Implement chat client server using tcp/ucp sockets:

chat Serv.py

```
import socket
    def sent():
        port = 12345
        host = '127.0.0.1'
        with socket.socket(socket.AF_INET, socket)
            Sock_DGram) as s:
        s.bind((host, port))
        while (True):
            d, add = s.recvfrom(1024)
            print("client", {d.decode()})
            a = input("Enter reply")
            S.Sendto(a.encoded(), add)
            if (a == "end")
                break
                exit.
```

reciever 2.py:

```
import socket
import time
    def reciever2(a):
        host = '127.0.0.1'
        port = 12345
with socket.socket(socket.AF_INET, socket.SOCK
                   DGRAM) as s:
    s.sendto(a.encode(), (host, port))
    d, addr = s.recvfrom(1024)
    print(f{d.decode()})
    while (true):
        a=input("enter message")
        if (a == "end"):
            recvr2(a)
            break
        else:
            recev2(a)
```

O/P:

```
python /chart_serv.py
client {'hi'}
Enter reply hello
client {'how are you'}
Enter reply : Imfine
```

```
Python /recvr.py
Enter message hi
    {: hello'}
Enter messages how a
you {'Imfine'}
Enter message
```

Result:

Thus the program is executed successfully & the output is verified.