

EX NO: 66.
DATE: 1.11.24.

CHAT CLIENT SERVER

AIM:

→ Implement chat client server using TCP/UDP sockets

CODE:

server.py

import socket

def server():

port = 12345

host = '127.0.0.1'

s = 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.encode(), add)

if a == "end":

break

exit

server()

client.py

```
import socket
```

```
import time
```

```
def recv2(a):
```

```
    host = '127.0.0.1'
```

```
    port = 10345
```

```
    with socket.socket(socket.AF_INET,
```

```
                        socket.SOCK_DGRAM) as s:
```

```
        s.sendto(a.encode(), (host, port))
```

```
        d, addr = s.recvfrom(1024)
```

```
        print([d.decode()])
```

```
while True:
```

```
    a = input("Enter message")
```

```
    if (a == "end"):
```

```
        recv(a)
```

```
        break
```

```
    else:
```

```
        recv2(a)
```

Thus the client was successfully
connected to the server

pg. 10

Let's begin
and begin

(a) server file

1. 0.0.0.0 = host

2. 8080 = port

(1) 11.11.11.11 = host

2. 8080 = port

(1) 11.11.11.11 = host

(2) 8080 = port

(3) 8080 = port

(4) 8080 = port

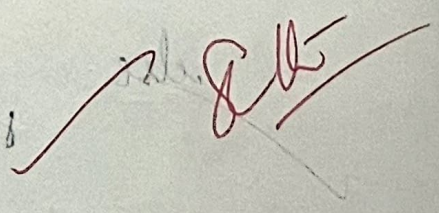
(5) 8080 = port

(6) 8080 = port

(7) 8080 = port

(8) 8080 = port

(9) 8080 = port



RESULT:

Thus chat client server was implemented
successfully using TCP/UDP sockets.