26/10/24

EXPERIMENT - 12 A

AIM:

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

~~TCP SERVER~~

Client

```
import Socket
import time
def pingServer(host = '127.0.0.1', port = 12345):
    with Socket.socket(Socket.AF_NET, SOCKET_SOCK_DGRAM)as
        try:
            S.Sendto (b'Hello', (host, port))
        except .timeout:
            Print ("Request timed out");
    if name - = = "_main_":
        Ping.Server()
```

Server

```
import Socket
def start_Serverhost = '127.0.0.1', port = 12345):
    with Socket.Socket (Socket.AF_NET, SOCKET.SOCK_DGRAM)as!
        S.bind ((host, port))
        Print (f" UDP Server running on short")
while True:
        data .addr: S.reckform (1024)
    Printf ("Received message")

if -name_ = "_main":
        Start.Server()
```

o/p :

```
python receiver.py
UDP Server running on 127.0.0.1 : 12345
Received message from (127.0.0.1, 59290)
Python client.py
Received reply from server : Hello client.
```

Result :

Thus the program was successfully executed.

26/10/24

EXPERIMENT - 12 B

AIM:

Implement chat 'client' 'server' using TCP / UDP sockets.

chat server.py

```
import Socket
def receiver():
    Port = 12345
    host = '127.0.0.1'
    with socket.socket('Socket.AF-INET, SOCK-PROGRAM) as S:
        S.bind ((host, port))
while (True):
    d.add-S.recv.form ('1024')
    print ("client")
    a= input ("Enter Reply:")
    S.Sendto (a.encoded.add)
    if (a== 'end')
        break
        esut
    rev()
```

receiver.py

```
import Sochet
import time
def receiver (a):
    host = '127.0.0.1'.
while (True):
    a = input ("Enter message").
    if (a== 'end').
        receiver (o)
        break
    else:
        receiver (a)
```

O/P

Python .\chat . serv.py
client {'hi'}
client {'How are you'}
Enter reply I'm fine

Python .\rear.py
Enter message : hi
{'Hello'}
Enter message How are you
{"I am fine"}
Enter message

RESULT:

Thus the client server using TCP UDP is executed & output verified successfully.