Practical - 12A

Aim:

a) Implement echo client server TCP / UDP Sockets

Algorithm:

TCP echo client Server Algorithm

Server (TCP):

- Create and bind a TCP socket to host and port
- Listen for connections and accept a client
- Receive data from the client
- Close the client and continue.

Client:

```
import socket
    import time

def ping-server (host= '127.0.0.1', port=
                                    12345
with socket.socket (socket.AF_INET, socket
                        Sock. Diagram) ass
try:

    S.sendto (b'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_DIGRAM):
    s.bind((host, port))
    print(f"UDP server running on {host}:
                                {port}")
if __name__ == "__main__":
    start_server()
```

Output:

Python server.py

UDP server running on 127.0.0.1:12345

Received message from ('127.0.0.1', 58009): Ping

Python client.py

Received Ping from ('127.0.0.1', 12345)
in 0.0 seconds.