18/10/24

EXERCISE 12a

Aim:

To implement echo client server using TCP/UDP sockets

Algorithm

Server.py:

* create a UDP socket
* Bind the socket to specific IP address (127.0.0.1) & port (12345)
* continuous listen for incoming message
* when message received - decode it.
* Display message along with sender address.
* Repeat infinitely.

client.py:

* create UDP socket
* set a timeout for socket to avoid waiting.
* send a predefined message hello for server IP address & port 12345.
* If no response received in timeout period, print timeout message.
* close socket after sending message.

Code:

server.py:

```
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}:
                                {port}")

    while True:
        data, addr = s.recvfrom(1024)
        print(f"received message from
                        {addr}: {data.decode()}")

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

client.py

```
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.
                       SOCK_DGRAM) as s:
        s.settimeout(5)
        try:
            s.sendto(b'Hello', (host, port))
            print("message sent to server")
        except socket.timeout:
            print("Request timed out")

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

output:

server.py

Terminal

>python server.py

>>

UDP server running on 127.0.0.1 : 12345

client.py

Terminal

>python client.py

>>

message sent to server

server terminal:

Received message from ('127.0.0.1', 56003) : Hello

Result:

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