**NAME : PRAVEEN V**

**ROLL NO.:241901082**

# Exercise 3

**UDP CLIENT-SERVER COMMUNICATION USING SOCKET PROGRAMMING IN PYTHON**

**INTRODUCTION:**

UDP (User Datagram Protocol) is a connectionless protocol that allows sending messages (datagrams) between client and server without establishing a connection. This experiment demonstrates a simple UDP client-server communication in Python.

**AIM:**

Develop a simple UDP server and client using Python’s socket module to exchange messages.

**ALGORITHM:**

**SERVER:**

* Create a UDP socket and bind it to an IP and port.
* Wait for incoming messages.
* Receive data, print it, and optionally send a reply.

**CLIENT:**

* Create a UDP socket.
* Send message to server IP and port.
* Receive reply from server and display it.

**CODE:**

**SERVER:**

import socket

sockfd = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

print('UDP Socket Created')

sockfd.bind(('localhost', 55555))

print('Waiting for messages')

while True:

    data, addr = sockfd.recvfrom(1024)

    receivedMsg = data.decode()

    print("Received message from", addr)

    print("Message:", receivedMsg)

       # Send the same message back to client

    sockfd.sendto(data, addr)

    print("Message reply sent to Client!")

    choice = input("Do you want to continue (type y or n): ")

    if choice == 'n':

        break

sockfd.close()

**CLIENT:**

import socket

clientfd = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_address = ('localhost', 55555)

name = input("Enter your message: ")

clientfd.sendto(name.encode(), server\_address)

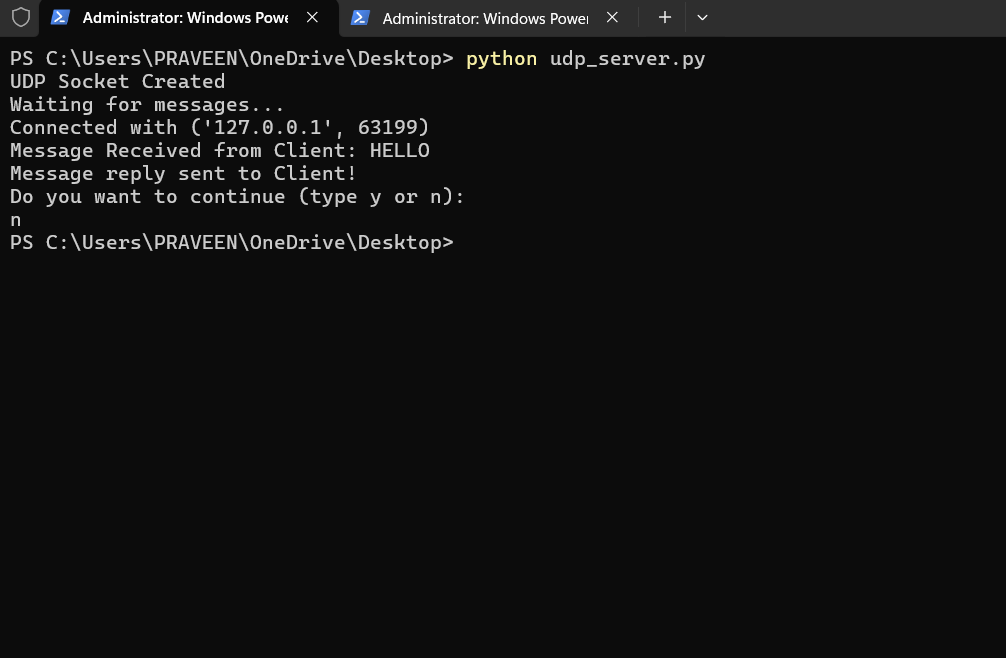
data, \_ = clientfd.recvfrom(1024)

print("Message Received from Server:", data.decode())

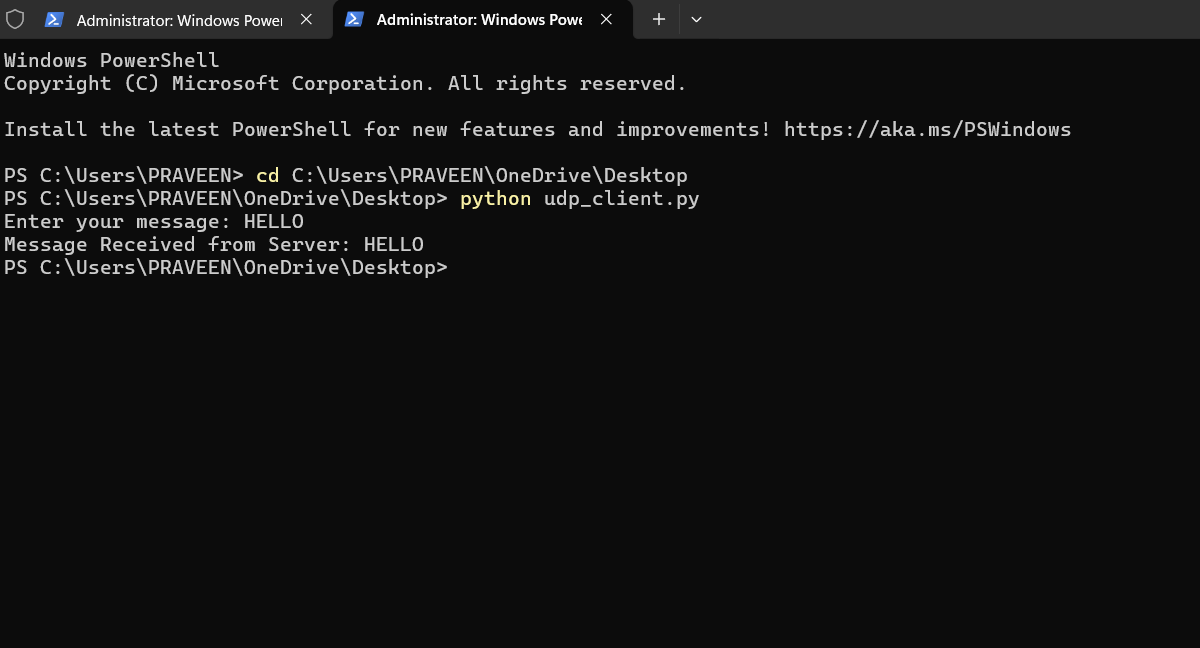
clientfd.close()

**OUTPUT:**

**SERVER:**

****

**CLIENT:**



**RESULT:**

The UDP client successfully sent messages to the server, and the server received and replied to each message.