EX.NO:2

TCP CLIENT-SERVER USING SOCKET PROGRAMMING IN PYTHON

AIM:

To implement TCP client-server communication using socket programming in python algorithm.

SERVER:

1.Create a socket using socket.socket()

2.Bind the socket to an IP and port using bind().

3.Listen for client connections using listen().

4.accept for client connection using accept().

5.Receive data using rev().

6.Send response using send().

7.Close connection.

CLIENT:

1.Create a socket using socket.socket().

2.Connect to server using connect().

3.Send data using send().

4.Receive response using rev().

5.Close connection.

SERVER PROGRAM:

```
import socket



sockfd=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print('Socket Created')
```

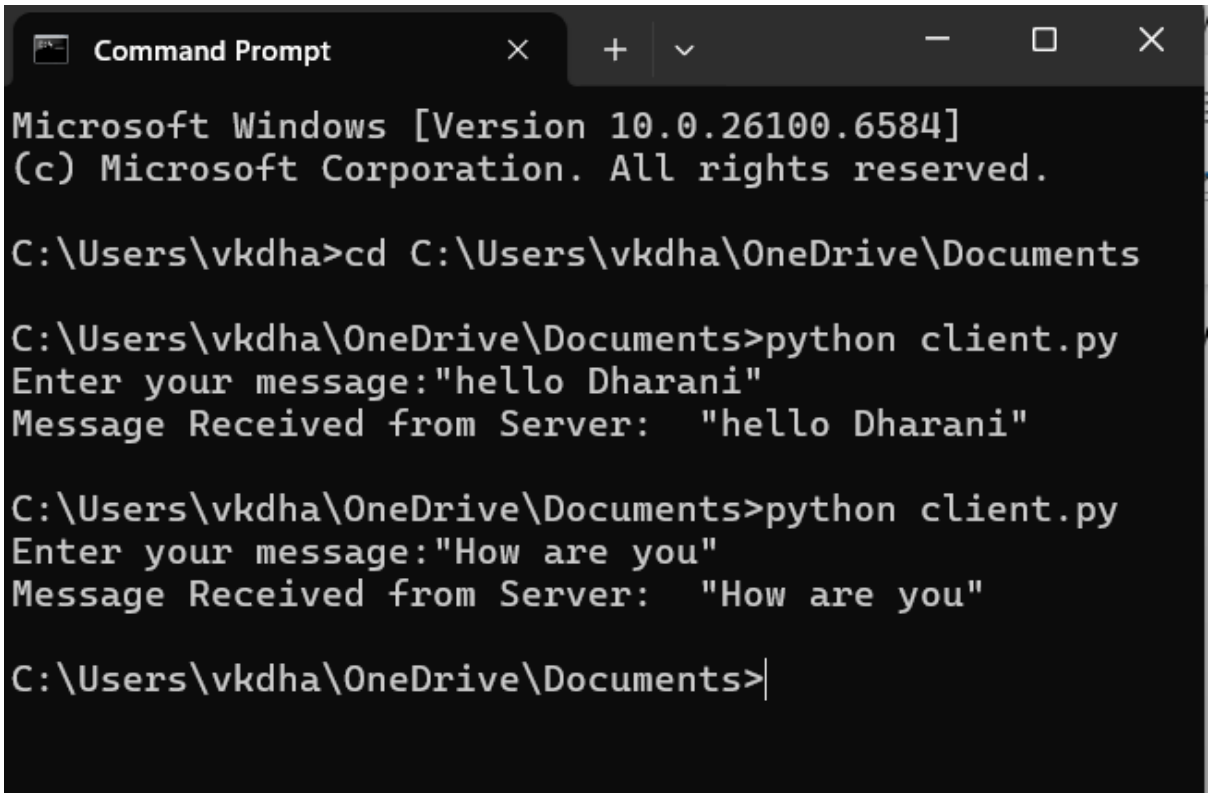
```
sockfd.bind(('localhost',55555))


sockfd.listen(3)
print('Waiting for connections')


while True:
    clientfd,addr=sockfd.accept()
    receivedMsg=clientfd.recv(1024).decode()
    print("Connected with ",addr)
    print("Message Received from Client: ",receivedMsg)
    clientfd.send(bytes(receivedMsg,'utf-8'))
    print("Message reply sent to Client!")
    print("Do you want to continue(type y or n):")
    choice=input()
    if choice=='n':
        break
```

CLIENT PROGRAM:

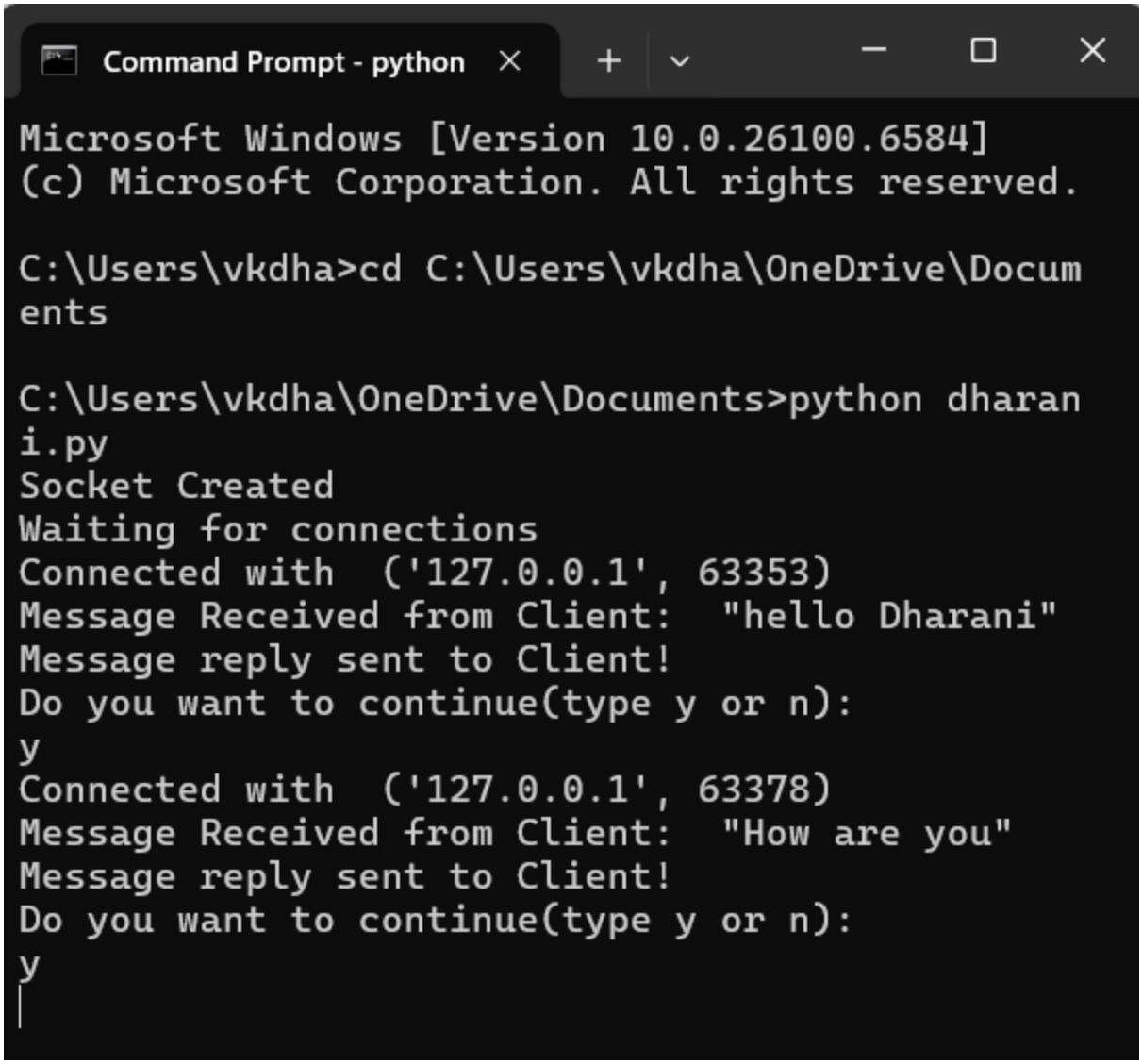
```
import socket
clientfd=socket.socket(socket.AF_INET, socket.SOCK_STREAM)


clientfd.connect(('localhost',55555))
```

```
name=input("Enter your message:")
clientfd.send(bytes(name,'utf-8'))
print("Message Received from Server: ",clientfd.recv(1024).decode())
```

INPUT:



OUTPUT:



RESULT:

Thus,TCP client-sever communication was successfully implemented using Python.

NAME : DHARANI K

ROLL NO : 241901025

DEPARTMENT:CSE-CYBER SECURITY