11. TCP Socket

Q) Using TCP/IP sockets, write a client server program to make client sending file name and server to send back the contents

→ A server has find() method which binds specific IP and it can listen to incomeing requests.

→ Passing an empty string means that the server can listen to incoming connections from other computer as well.

→ Server is in listening mode.

→ At last, we make a while loop and start to accept all incoming connections and close those connections

Client.py

```
from socket import *
server Name = '127.0.0.1'
server Port = 12000
client Socket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, ServerPort))
sentance = input("\n Enter file Name:")
clientsocket.send(sentance.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n From Server: \n")
point (file Contents)
client Socket.close()
```

Server.py

```
from socket import *
serverName= "127.0.0.1"
serverPort = 12000
serverSocket= socket (AF_INET, SOCK_STREAM)
serverSocket.bind (serverName, serverPort))
serversocket.listen(1)
while(1)
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentance = connectionSocket.recv(1024).decode()
    file=open(sentance,"r")
    l=file.read(1024)
    connectionsocket.send(l.encode())
    print('\n Sent contents of ' + sentance)
    file.close()
    connection Socket.close()
```

Output

For server.py

```
Server is Ready to Receive
```

For Client.py

```
Enter File name : server TCP.py
From Server:
    connectionsocket, addr = serverSocket.acept()
    sentance = connectionSocket.recv(1024).decode()
```

```
file= open (sentance, "r")
l = file.read (1024)

Connection Socket. send (l.encode())
print ('\n Sent contents of ' + sentance)
file.close()
connection Socket.close()
```