Using TCP/IP sockets write a client-server prog. to make client sends the file name & the server to send back the contents of requested file if present.

Client.py

```
from socket import *

serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
Sentence = input("\n Enter File name: "):

clientSocket.send(sentence.encode())
file contents = clientSocket.recv(1024).decode()
print("\n From Server: \n")
print(file contents)
client
clientSocket.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()
    Sentence = ConnectionSocket.recv(1024).decode()
    file = open(sentence, 'r');
```

l = file.read(1024)

connectionSocket.send(l.encode())
print("\n Sent contents of" + sentence).
file.close()
connectionSocket.close()

bind() → bind() method binds a server to a specific IP & port so that it can listen to incoming requests on that IP & port.

& close()
accept() => initiates a connection with client & the close method closes the connection with client.

Client Output:

>>> The server is ready to receive

Server Output:

✦ Enter the file name: Server TCP.py
from server!

/*
    Contents of the file will be
    displayed here
*/

servertcp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/servertcp.py (3...

File Edit Format Run Options Window Help

```python
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()
    sentence = connectionSocket.recv(1024).decode()

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

    connectionSocket.send(l.encode())
    print ('\nsent contents or ' + sentence)
    file.close()
    connectionSocket.close()
```

clienttcp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/clienttcp.py (3.1...

File Edit Format Run Options Window Help

```python
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

IDLE Shell 3.10.8

File Edit Shell Debug Options Window Help

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/servertcp.py =
The server is ready to receive

Sent contents of servertcp.py
The server is ready to receive
```

IDLE Shell 3.10.8

File Edit Shell Debug Options Window Help

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clienttcp.py =

Enter file name: servertcp.py

From Server:

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()
    sentence = connectionSocket.recv(1024).decode()

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

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
>>>
```
Ln: 11 Col: 0