Computer Networks
Lab_05

Submitted By: Muhammad Ali Raza
Roll No. : 2018-UET-NML-CS-31(1802031)

commonthread.py File:

```
from threading import Thread
import socket
import time
import pickle


SCHEME = "utf-8"
authorization = {}
user_resourses = {}
class CommonThread(Thread):
    def __init__(self,clientsocket):
        Thread.__init__(self)

        self.clientsocket = clientsocket

    def run(self):
        print("Client Thread Started ... ")

        # self.clientsocket.send("Assalam ulaikum".encode(SCHEME))
        # data = self.clientsocket.recv(1024)
        # time.sleep(2)
        # print("From Client : ", data.decode(SCHEME))
        choice = self.clientsocket.recv(1024)
        choice = choice.decode(SCHEME)
        if(choice == "1"):
            #recieving the username and password
            username = self.clientsocket.recv(1024)
            password = self.clientsocket.recv(1024)
            username = username.decode(SCHEME)
            password = password.decode(SCHEME)

            #setting username and password
            authorization[username] = password

            #printing the username and password from userlist
            print("Users List \n")
            for x , y in authorization.items():
                print(" Username : ",x ,"\n" ,"password : ", y , "\n")

            #assigning resousrses to the user
            resourses = self.clientsocket.recv(4096)
            resourses_list = pickle.loads(resourses)
```

```
            user_resourses[username] = resourses_list
            print(user_resourses)

            self.clientsocket.send("User Added successfully \n".encode(SCHEME))

        elif(choice == "2"):
            #getting the username and password from clients
            user_name = self.clientsocket.recv(1024)
            password = self.clientsocket.recv(1024)

            match = 0
            #decoding the username and password
            check_name = user_name.decode(SCHEME)
            check_pass = password.decode(SCHEME)

            #verifing from our dictionary
            for users , val in authorization.items():
                if(users == check_name and val == check_pass):
                    match = match + 1

            #if any match found then it will authenticate otherwise not.
            print("Verifying Please Wait ... \n")
            time.sleep(2)
            if(match != 0):
                self.clientsocket.send("Authorized User".encode(SCHEME))
            else:
                self.clientsocket.send("Unautherized User".encode(SCHEME))

        elif(choice == "3"):
            res = 0
            #reciving the username and resourse no.
            username = self.clientsocket.recv(1024)
            username = username.decode(SCHEME)
            resourse = self.clientsocket.recv(1024)
            resourse = resourse.decode(SCHEME)
            #assigning the resourse no
            if(resourse == "1"):
                res = 0
            elif(resourse == "2"):
                res = 1
            elif(resourse == "3"):
                res = 2
            else:
                print("Error")
```

```
            #checking resourse
            print("Searching Please Wait... \n")
            time.sleep(2)
            for user , val in user_resourses.items():
                print(user , val)
                if(user == username):
                    if(val[res] == 1):
                        self.clientsocket.send("User has access to the Resourse".
encode(SCHEME))
                    else:
                        match = 0
                        self.clientsocket.send("User does not have access to the
Resourse".encode(SCHEME))


            #if no matching user found
            self.clientsocket.send("No User Found".encode(SCHEME))
        elif(choice == "4"):

            data = pickle.dumps(authorization)
            self.clientsocket.send(data)
        else:
            print("Error Encountered")
        self.clientsocket.close()
```

Multithreadedserver.py File:

```
import socket
from threading import Thread
import time
from commonthread import CommonThread


ADDRESS = "127.0.0.1"

PORT = 2222
def main():


    s = socket.socket()
    s.bind((ADDRESS,PORT))
```

```
    s.listen(5)
    print("Listing for clients ...")
    while True:
        c , addr = s.accept()
        print("Client Connected :  "  , addr)
        clientThread = CommonThread(c)
        clientThread.start()

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

Client.py File:

```
from commonthread import SCHEME
import socket
import pickle

ADDRESS = "127.0.0.1"

PORT = 2222

c = socket.socket()

c.connect((ADDRESS,PORT))

print("Connected with server successfully \n")

print(" Press 1 to Add New User  \n Press 2 to verify User \n Press 3 to authenti
cate User  \n Press 4 to Display Users \n---------------------------------\n")
choice = input("Enter Choice :  ")

if(choice == "1"):
    resourses = []
    #sending choice
    c.send(choice.encode(SCHEME))

    #adding new username and password and sending it to the server
    username = input("Enter Username :  ")
    password = input("Enter Password :  ")

    c.send(username.encode(SCHEME))
```

```
    c.send(password.encode(SCHEME))
    #setting resourses for the user
    print("please enter 1 for yes and 0 for no for resourses to use for the User"
)
    for i in range(3):
        print("for resourse R",i+1)
        resourse = int(input())
        resourses.append(resourse)
    data = pickle.dumps(resourses)
    c.send(data)

    #receiving the output from server
    data = c.recv(1024)
    print("From Server : " , data.decode(SCHEME))

elif(choice == "2"):
    c.send(choice.encode(SCHEME))

    username = input("Enter Username :  ")
    password = input("Enter Password :  ")

    c.send(username.encode(SCHEME))
    c.send(password.encode(SCHEME))

    #receiving the output from server
    data = c.recv(1024)
    print("From Server : " , data.decode(SCHEME))
elif(choice == "3"):
    c.send(choice.encode(SCHEME))
    username = input("Enter Username :  ")
    c.send(username.encode(SCHEME))
    resourse = input("Which resourse do you want to check \n  Enter R1 , R2 or R3
 : ")
    if(resourse == "R1" or resourse == "r1"):
        res = 1
        c.send(str(res).encode(SCHEME))
    elif(resourse == "R2" or resourse == "r2"):
        res = 2
        c.send(str(res).encode(SCHEME))
    elif(resourse == "R3" or resourse == "r3"):
        res = 3
        c.send(str(res).encode(SCHEME))


    data = c.recv(1024)
```

```
    print("From Server : " , data.decode(SCHEME))
elif(choice == "4"):
    c.send(choice.encode(SCHEME))

    data = c.recv(4096)
    data_list = pickle.loads(data)
    print(" ---- Users List --- \n")
    for users , val in data_list.items():
        print(" UserName = " , users , "\n" , "Password = " , val,  "\n")

else:
    print("Invalid choice")
```

----------Tasks----------

Task 01:-

Adding a user with given credentials and access level to resources

```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE                    1: powershell

PS E:\STUDY MATRIAL\semester 6\Computer Networks Lab\Lab_05> python client.py
Connected with server successfully

 Press 1 to Add New User
 Press 2 to verify User
 Press 3 to authenticate User
 Press 4 to Display Users
 ---------------------------------

Enter Choice :  1
Enter Username :  Ali
Enter Password :  123
please enter 1 for yes and 0 for no for resourses to use for the User
for resourse R 1
1
for resourse R 2
0
for resourse R 3
1
From Server :  User Added successfully

PS E:\STUDY MATRIAL\semester 6\Computer Networks Lab\Lab_05>
```

Figure 1 : Adding credentials and access to resources

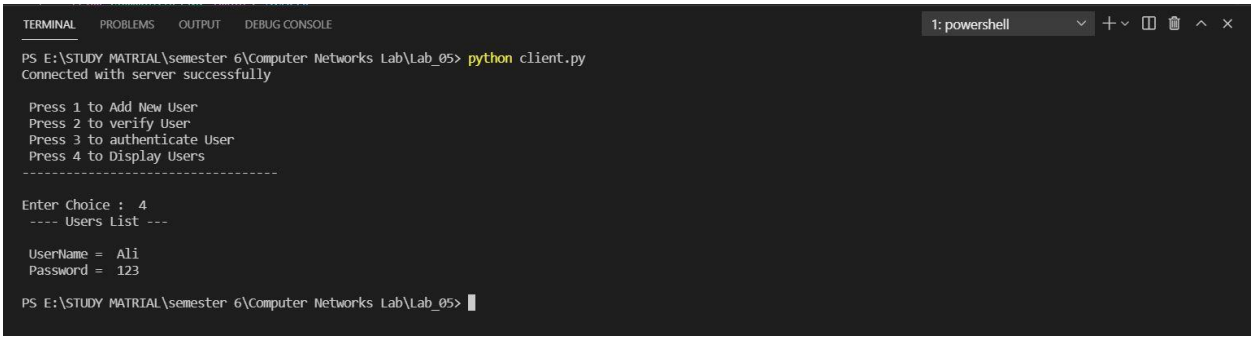

Figure 2 : Display of the users

Task 02:-

Validating a username and password

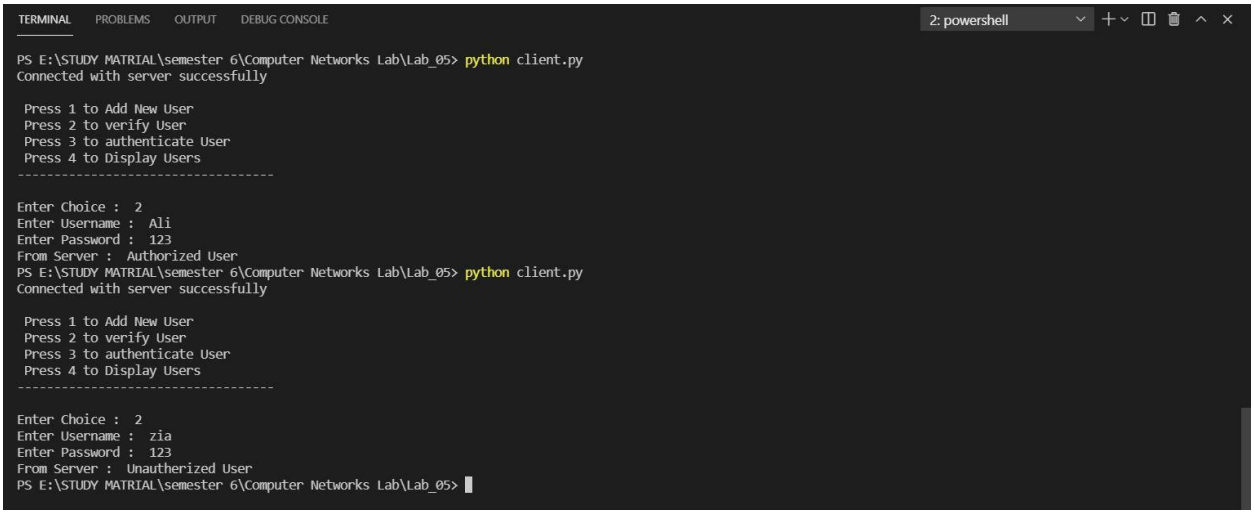

Figure 3 : Verification of a User

Task 03:-

Authorization of a user

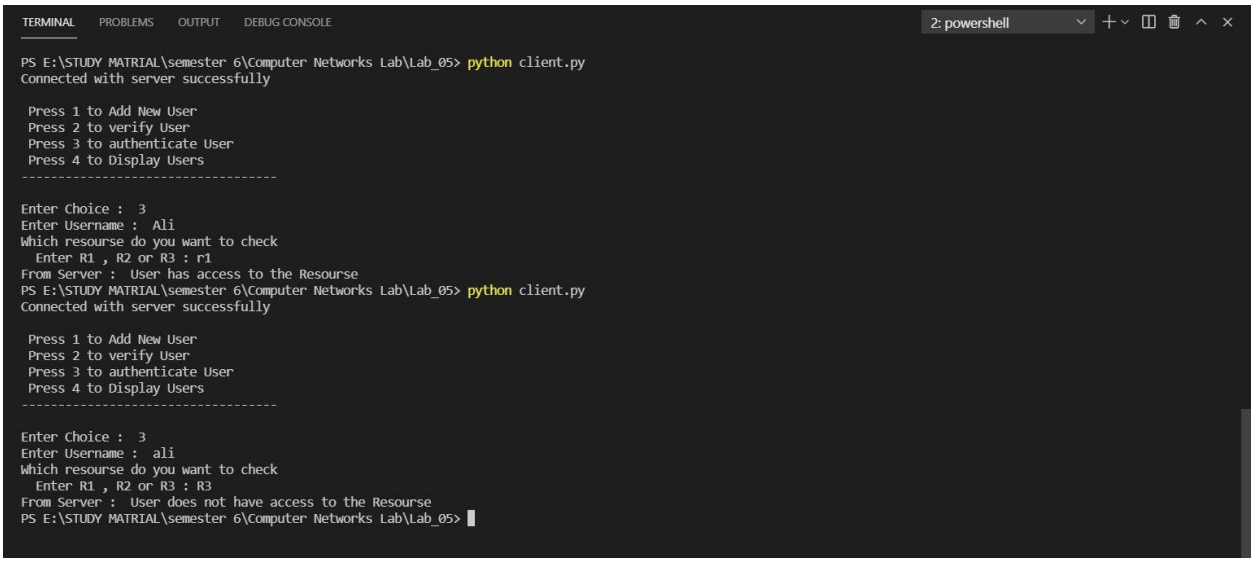

Figure 4 : Authorization of the users for a given resource

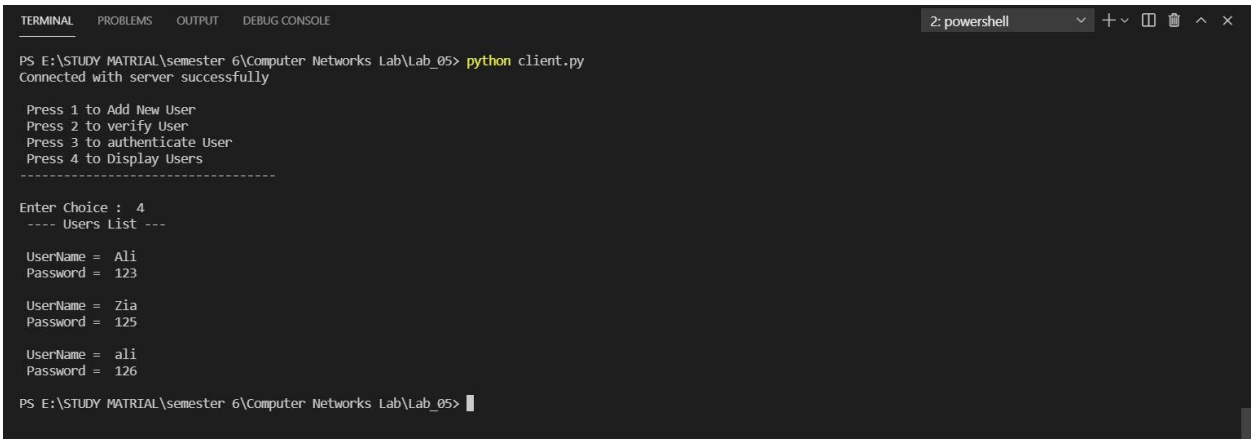

Figure 5 : Display of users