Name: Gaurav Kumar Singh

Registration Number: 19BCE2119

Course: CSE 1004 Network and Communication

Digital Assignment 2

TOPIC: Code Snippet for Unicast

**CODE**

import os

import datetime

import shutil

import csv

import pandas as pd

from pandas.core.frame import DataFrame

#x=datetime.datetime.now()

path=os.getcwd()

path=path+'\Chats'

username=''

password=''

admin\_username='Gaurav'

admin\_password='19BCE2119'

def display\_chats(path\_user\_logs, To\_user, From\_user):

    data=pd.read\_csv(path\_user\_logs)

    data['Time']= pd.to\_datetime(data['Time'], infer\_datetime\_format=True)

    data.sort\_values(by = 'Time', ascending = True, inplace = True)

    data\_user= data.loc[data['From']==From\_user]

    print(data\_user)

def unique(list1):

    unique\_list = []

    for x in list1:

        if x not in unique\_list:

            unique\_list.append(x)

    return unique\_list

def display\_log(path\_user\_logs,username):

    data=pd.read\_csv(path\_user\_logs)

    data['Time']= pd.to\_datetime(data['Time'], infer\_datetime\_format=True)

    data.sort\_values(by = 'Time', ascending = False, inplace = True)

    data= data.loc[data['From']!=username]

    print(unique(data['From']))

def create\_log(path):

    with open(path+"\\"+'log.csv', 'w', newline='') as file:

        writer = csv.writer(file)

        writer.writerow(["Time", "From", "Message"])

        file.close()

def add\_log(to\_user, from\_user, message):

    path\_log=path+"\\"+to\_user.split(' ')[0]+"\\"+to\_user.split(' ')[1]+"\\"+'log.csv'

    with open(path\_log, 'a', newline='') as file:

        writer = csv.writer(file)

        x=datetime.datetime.now()

        writer.writerow([x, from\_user, message])

        file.close()

def session\_init():

    print("\nSession Initialization.....\n")

    subnets = int(input("Enter number of subnets: "))

    for i in range(subnets):

        path\_subnet=path+"\\"+chr(65+i)

        os.mkdir(path\_subnet)

        users=int(input("Enter number of users in subnet "+chr(65+i)+":"))

        for j in range(users):

            path\_user=path\_subnet+"\\"+str(j+1)

            os.mkdir(path\_user)

            create\_log(path\_user)

def add\_or\_delete\_user(delU,choice):

    flag=int(1)

    choice=int(choice)

    while(flag==1):

        if (choice==1):

            flag=0

            loc=input("Enter the subnet in which u want to add the user: ")

            if(os.path.exists(path+"\\"+loc)):

                dirlist=os.listdir(path+"\\"+loc)

                for i in range(len(dirlist)):

                    dirlist[i]=int(dirlist[i])

                username = max(dirlist)+1

                os.mkdir(path+"\\"+loc+"\\"+str(username))

                print("User Added... The username for the user is: "+loc+" "+str(username))

            else:

                print("Subnet not found")

        elif(choice==0):

            flag=0

            delU=delU.split(' ')

            shutil.rmtree(path+"\\"+str(delU[0])+"\\"+str(delU[1]), ignore\_errors=True)

            print("User removed")

        else:

            print("Enter valid option")

            flag=1

def add\_subnet(admin\_user, admin\_pass):

    if(admin\_user==admin\_username and admin\_pass==admin\_password):

        dirlist=os.listdir(path)

        for i in range(len(dirlist)):

            dirlist[i]=int(ord(dirlist[i]))

        subnet\_ascii=max(dirlist)+1

        os.mkdir(path+"\\"+str(chr(subnet\_ascii)))

        nusers=int(input("Enter number of users in the subnet: "))

        for j in range(nusers):

            path\_user=path+"\\"+str(chr(subnet\_ascii))+"\\"+str(j+1)

            os.mkdir(path\_user)

    else:

        print("Unauthorized user")

def del\_subnet(admin\_user, admin\_pass):

    if(admin\_user==admin\_username and admin\_pass==admin\_password):

        sub\_del=input("Enter the subnet to be deleted: ")

        dirlist=os.listdir(path)

        if (sub\_del in dirlist):

            shutil.rmtree(path+"\\"+str(sub\_del), ignore\_errors=True)

            print("Subnet removed")

        else:

            print("Subnet not found")

    else:

        print("Unauthorized user")

if (os.path.exists(path)):

    flag=1

    choice=input("Do you want to continue from last session? (y/n): ")

    if (choice=='y'):

        while(flag==1):

            option=input("Do you want to make any changes in the structure of the network? (y/n): ")

            if (option=='y'):

                choice=int(input("Enter 0 to delete a user\nEnter 1 to add a user\nEnter 2 to add a subnet with admin privileges\nEnter 3 to delete a subnet with admin privileges\nEnter 4 to exit:"))

                if (choice==1):

                    add\_or\_delete\_user('none', 1)

                elif (choice==0):

                    print("Login to account to delete.")

                    username=input("Username: ")

                    password=input("Password: ")

                    if (os.path.exists(path+"\\"+username.split(' ')[0]+"\\"+username.split(' ')[1]) and username==password):

                        add\_or\_delete\_user(username,0)

                    else:

                        print("User not found")

                elif (choice==2):

                    admin\_u=input("Enter username: ")

                    admin\_p=input("Enter password: ")

                    add\_subnet(admin\_u, admin\_p)

                elif(choice==3):

                    admin\_u=input("Enter username: ")

                    admin\_p=input("Enter password: ")

                    del\_subnet(admin\_u, admin\_p)

                else:

                    pass

                loop=input("Do you want to make more changes? (y/n) ")

                if(loop=='y'):

                    flag=1

                else:

                    flag=0

            elif(option=='n'):

                flag=0

                pass

            else:

                print("Enter valid option.")

                flag=1

        pass

    elif (choice=='n'):

        shutil.rmtree(path, ignore\_errors=True)

        print("Previous session deleted...")

        os.mkdir(path)

        print("New session created...")

        session\_init()

else:

    os.mkdir(path)

    print("No previous session found... New session created")

    session\_init()

print("\n\n\n\n\n------------------------LOGIN------------------------")

username='X1'

password='X2'

login='y'

flag='y'

while(login=='y'):

    flag='y'

    while(username!=password):

        username=input("Enter Username: ")

        password=input("Enter Password: ")

        if(username!=password):

            print("Incorrect username or password")

        else:

            print("Logged in to user "+ username)

    flag\_log='y'

    path\_user=path+"\\"+username.split(' ')[0]+"\\"+username.split(' ')[1]

    path\_user\_log=path\_user+"\\"+'log.csv'

    while(flag=='y'):

        choice=int(input("Enter 0 to get list of users who interacted most recently.\nEnter 1 to send message to another peer\nEnter 2 to logout: "))

        if (choice==0):

            display\_log(path\_user\_log, username)

            read\_chat='y'

            while(read\_chat=='y'):

                read\_chat=input("Do you want to read a chat from any user? (y/n):")

                if(read\_chat=='y'):

                    unicasted\_chat=input("Enter username of the person whose messages you want to see: ")

                    display\_chats(path\_user\_log, username, unicasted\_chat)

                else:

                    read\_chat='n'

        elif(choice==1):

            to\_user=input("Enter username of recepient: ")

            message=input("Enter your message: ")

            add\_log(to\_user,username,message)

            add\_log(username, username,message)

        elif(choice==2):

            print("Logged out")

            flag\_log='n'

            flag='n'

            username='X1'

            password='X2'

        else:

            print("Enter valid option")

        if (flag\_log=='y'):

            flag=input("Do you want to perform any other function? (y/n) ")

            if (flag=='n'):

                login='n'

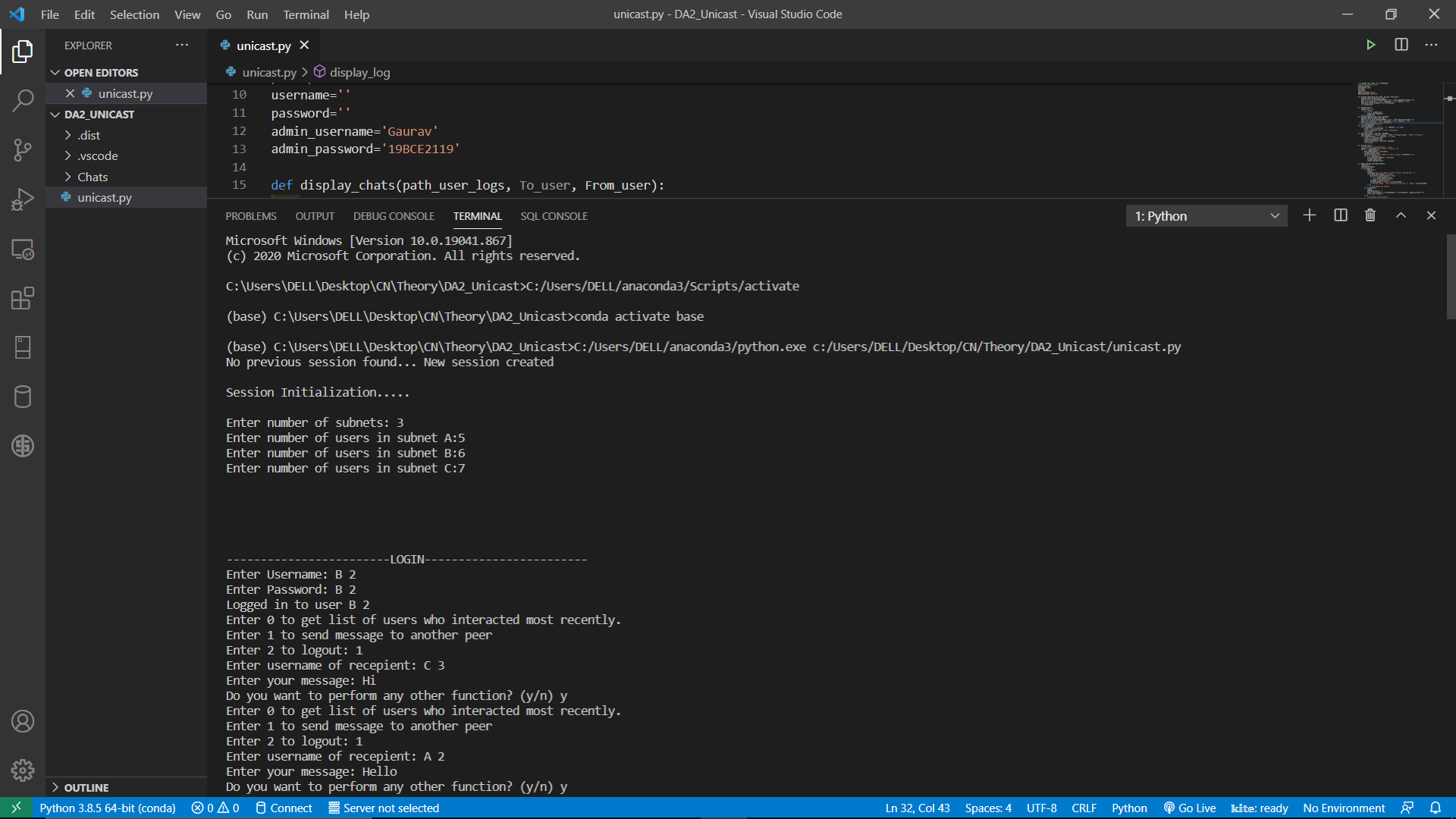
        else:

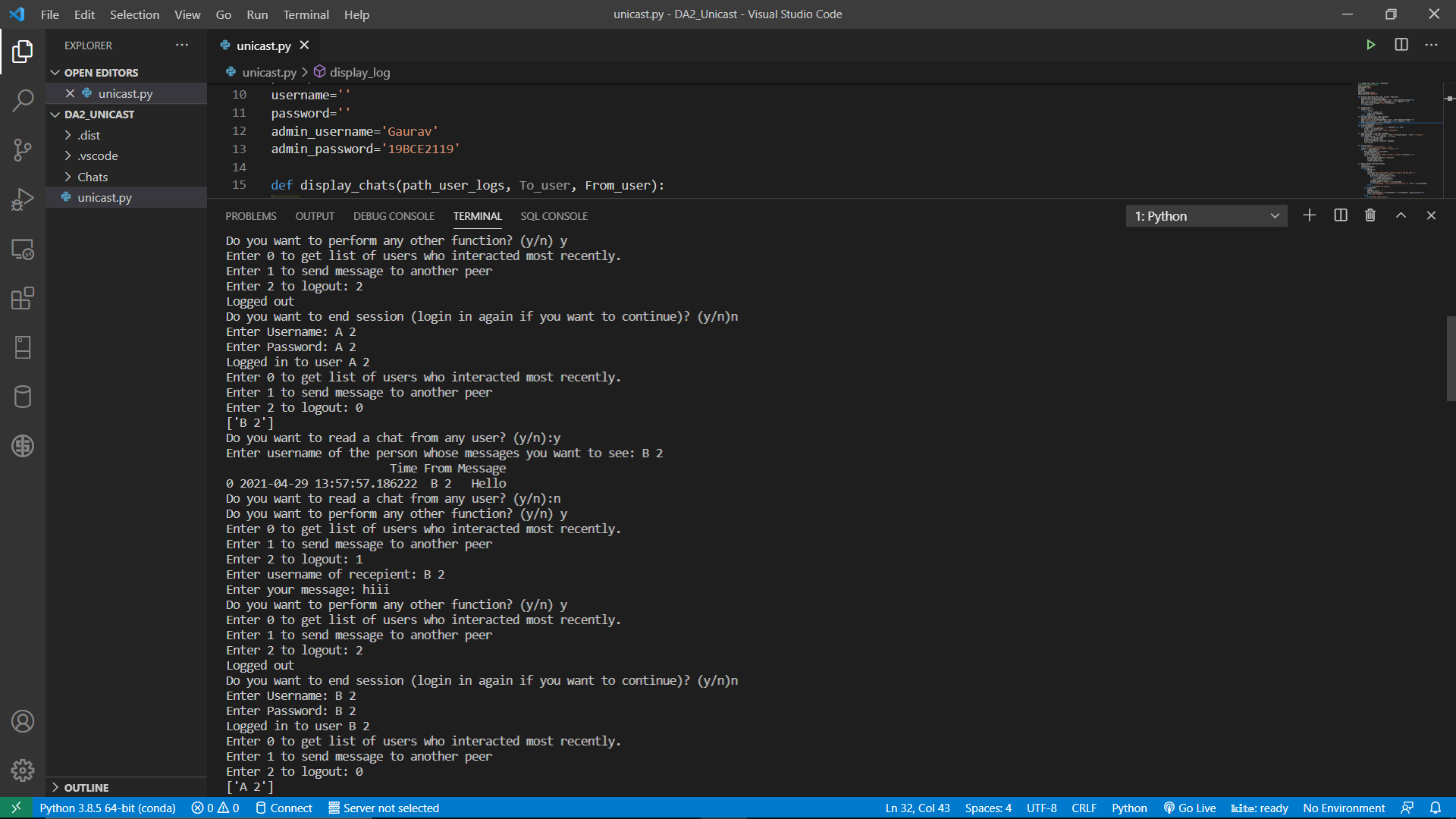
            another\_login=input("Do you want to end session (login in again if you want to continue)? (y/n)")

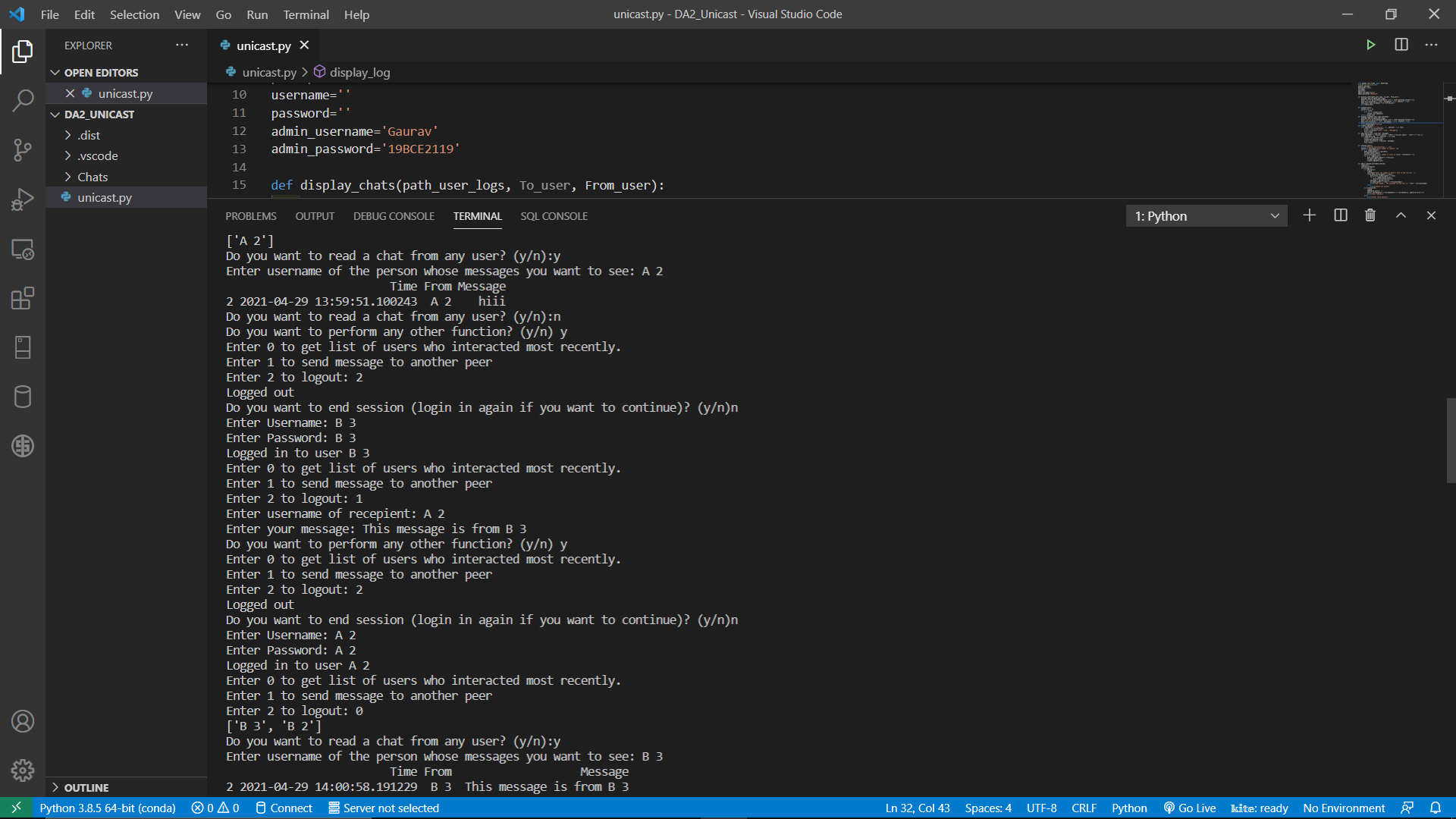
            if(another\_login=='y'):

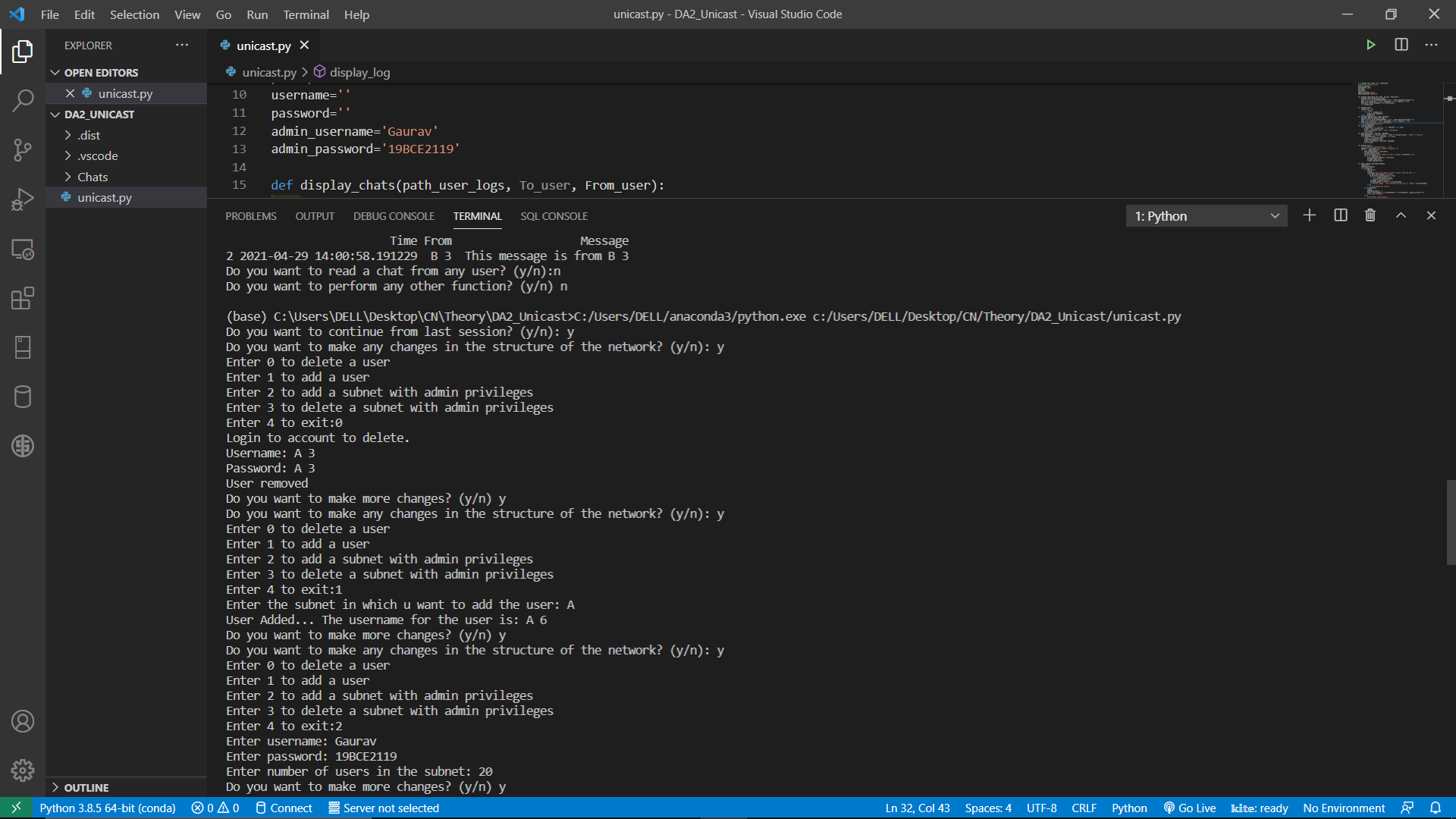
                login='n'

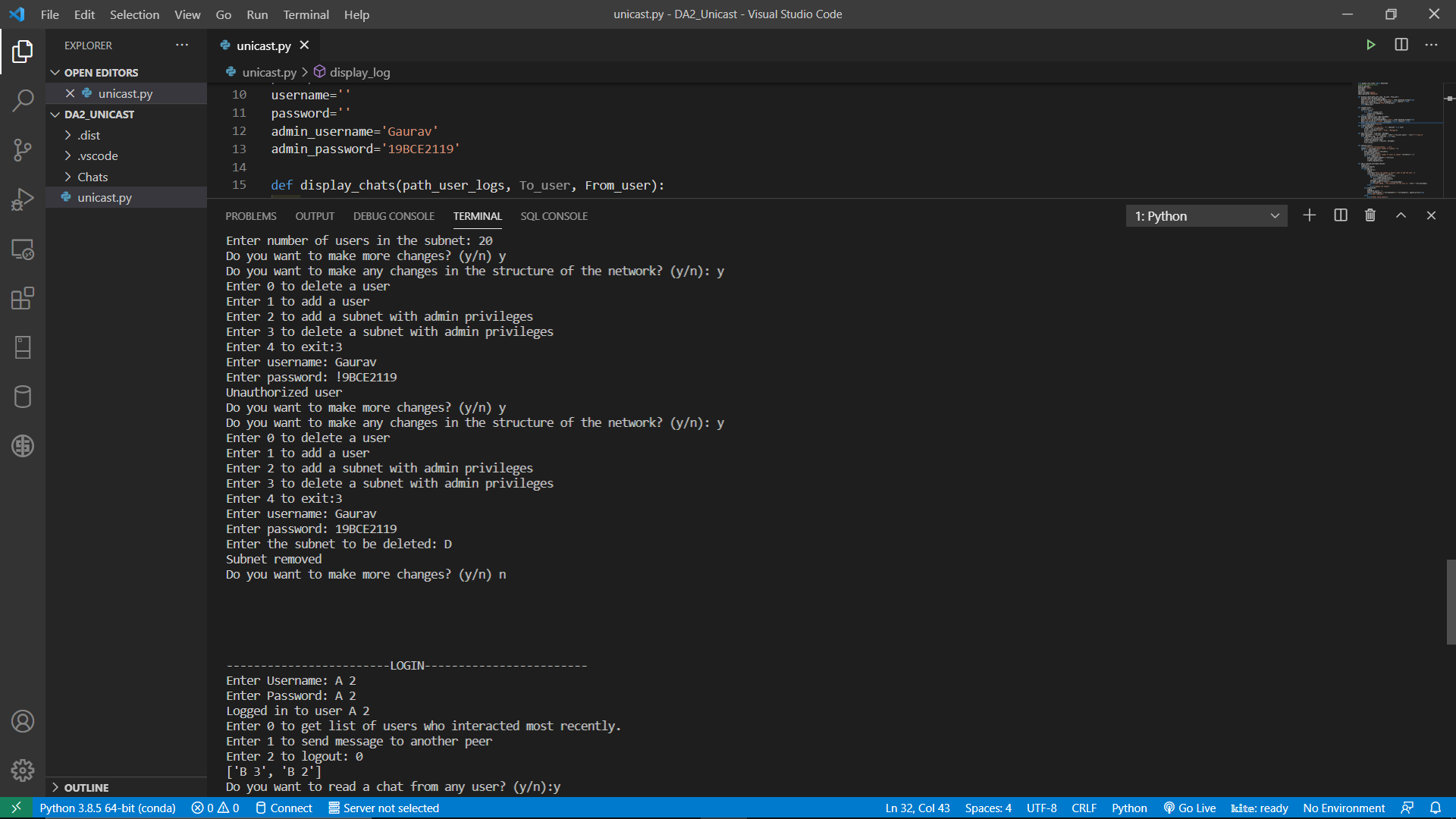
**OUTPUT**

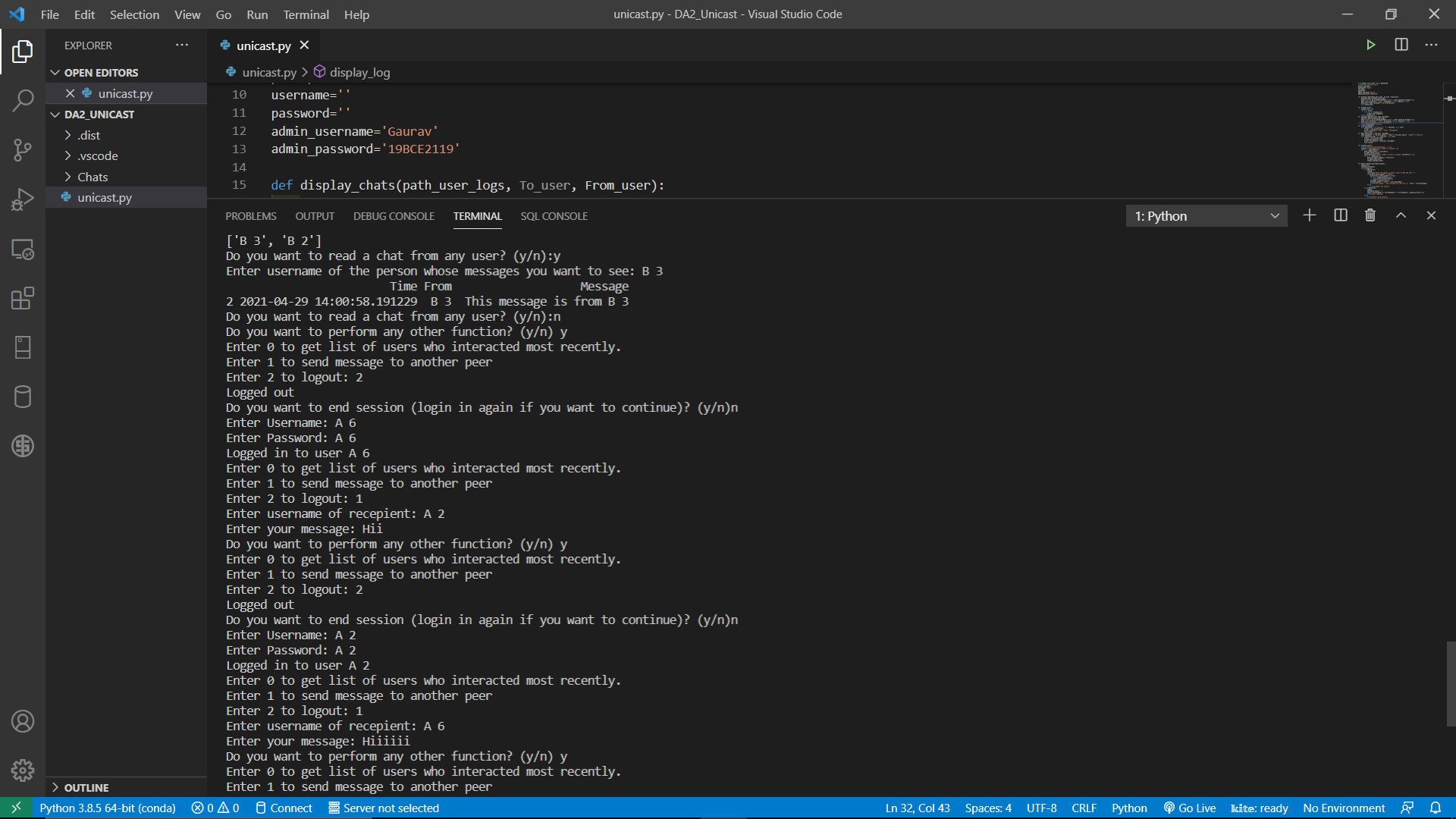












**VIDEO LINK**

<https://drive.google.com/file/d/1q1sZaBnCxwrFm-6zNN4UQCvxlyGcbO1s/view?usp=sharing>

Alternative link

https://youtu.be/iucCC1VXljY