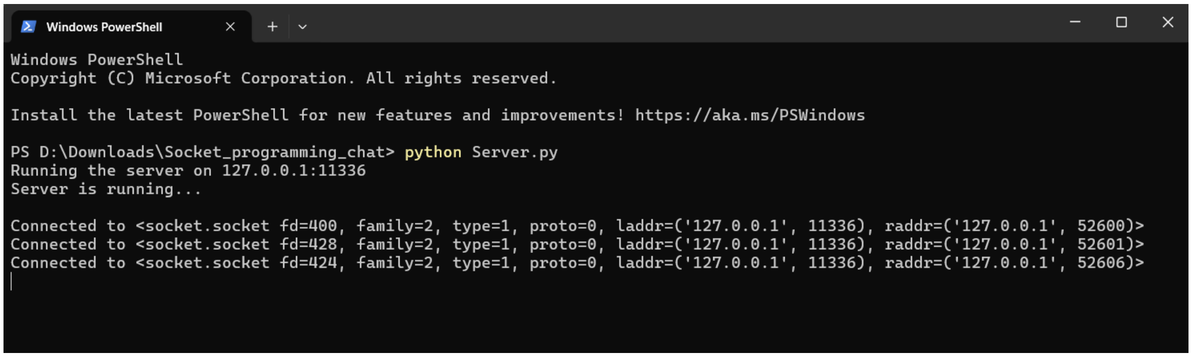
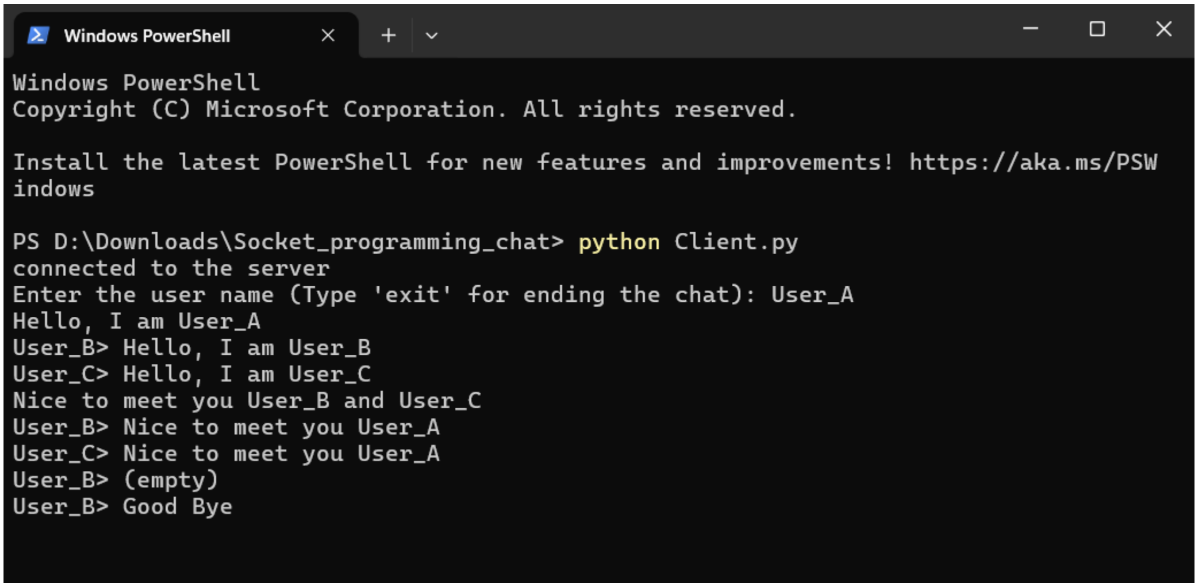
EE 514 Rodrigo P.R.P.

Socket programming submission E/18/306



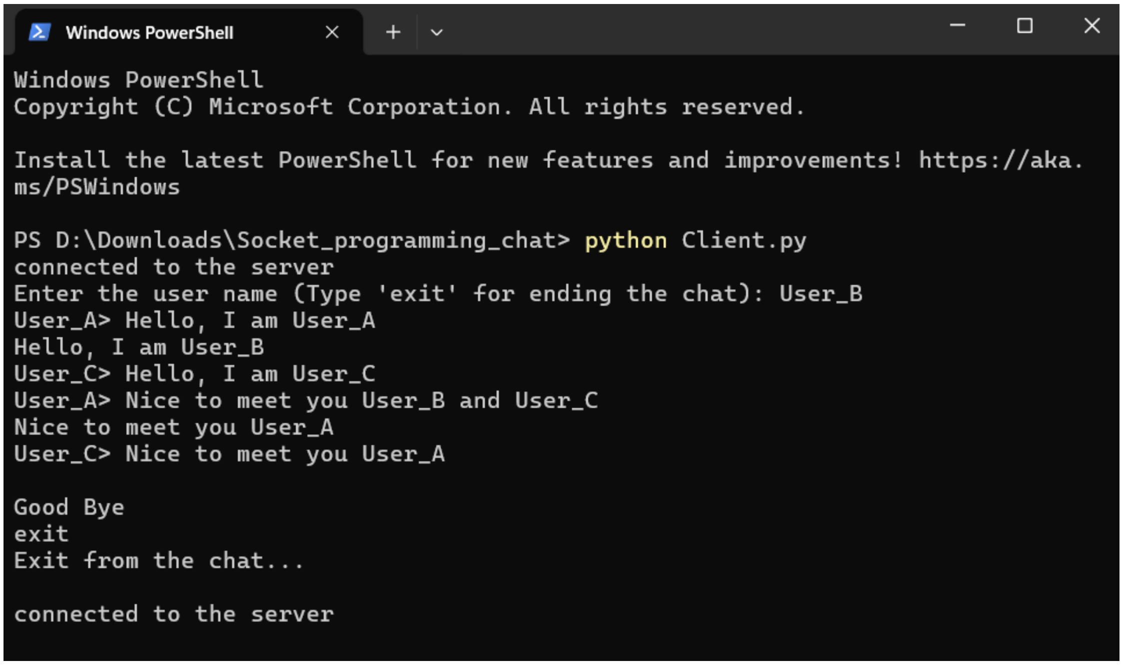
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Downloads\Socket_programming_chat> python Server.py
Running the server on 127.0.0.1:11336
Server is running...

Connected to <socket.socket fd=400, family=2, type=1, proto=0, laddr=('127.0.0.1', 11336), raddr=('127.0.0.1', 52600)>
Connected to <socket.socket fd=428, family=2, type=1, proto=0, laddr=('127.0.0.1', 11336), raddr=('127.0.0.1', 52601)>
Connected to <socket.socket fd=424, family=2, type=1, proto=0, laddr=('127.0.0.1', 11336), raddr=('127.0.0.1', 52606)>
```

Figure 01: Server



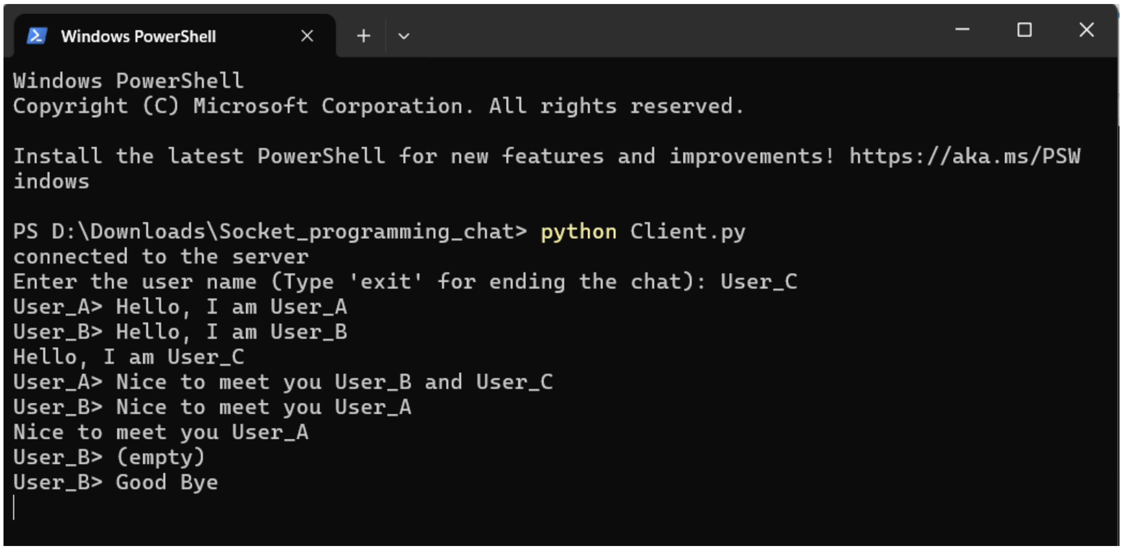
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSW
indows

PS D:\Downloads\Socket_programming_chat> python Client.py
connected to the server
Enter the user name (Type 'exit' for ending the chat): User_A
Hello, I am User_A
User_B> Hello, I am User_B
User_C> Hello, I am User_C
Nice to meet you User_B and User_C
User_B> Nice to meet you User_A
User_C> Nice to meet you User_A
User_B> (empty)
User_B> Good Bye
```

Figure 02: Client



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.
ms/PSWindows

PS D:\Downloads\Socket_programming_chat> python Client.py
connected to the server
Enter the user name (Type 'exit' for ending the chat): User_B
User_A> Hello, I am User_A
Hello, I am User_B
User_C> Hello, I am User_C
User_A> Nice to meet you User_B and User_C
Nice to meet you User_A
User_C> Nice to meet you User_A

Good Bye
exit
Exit from the chat...

connected to the server
```

Figure 03: Client



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSW
indows

PS D:\Downloads\Socket_programming_chat> python Client.py
connected to the server
Enter the user name (Type 'exit' for ending the chat): User_C
User_A> Hello, I am User_A
User_B> Hello, I am User_B
Hello, I am User_C
User_A> Nice to meet you User_B and User_C
User_B> Nice to meet you User_A
Nice to meet you User_A
User_B> (empty)
User_B> Good Bye
```

Figure 04: Client

Figure 01 show the client server running. Figure 02, 03 and 04 show three client connected with the server same time. They can chat with each other through the interface.

APPENDIX

```
#Sever side
import socket, threading
# coanstant
host_ip = '127.0.0.1'
host_port = 11336
encoder = "utf-8"
byte_size = 1024
active_clients = []

def listen_to_messege(client, username,s):
    while True:
        messege = client.recv(2048).decode(encoder)
        if messege != "exit" and messege !="":
            final_mg = username + '>' + messege
            send_mg_to_all(final_mg)
        elif messege == "exit":
            client.close()
            break
        else:
            print(f"The messege sent by the client {username} is empty.")
def send_mg_to_client(client, messege):
    client.sendall(messege.encode())
def send_mg_to_all(messege):
    for user in active_clients:
        send_mg_to_client(user[1], messege)
def client_handler(client,s):
    while True:
        username = client.recv(2048).decode(encoder)
        if username != "":
            active_clients.append((username, client))
            break
        else:
            print("client user name is empty.")
    threading.Thread(target=listen_to_messege, args=(client, username, s,)).start()
# socket with ipv4 with TCP
def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((host_ip, host_port))
    print(f"Running the server on {host_ip}:{host_port}")
    s.listen()
    print("Server is running...\n")
    # send and receive
    while True:
        client, addr = s.accept()
        print(f"Connected to {client}")
        threading.Thread(target=client_handler, args=(client, s,)).start()
if __name__ == '__main__':
    main()
```

```
#client side
import socket, threading
des_ip='127.0.0.1'
des_port=11336
encoder="utf-8"
byte_size=1024

def listen_to_server_mg(client, myname):
    while True:
        messege=client.recv(2048).decode(encoder)
        if messege !="":
            username=messege.split('>')[0]
            content=messege.split('>')[1]
            if username != myname:
                print(f"{username}> {content}")
            else:
                pass
        else:
            print("Exit from the chat...\n")
            main()
def send_mg_to_server(client):
    while True:
        messege = input("")
        if messege !="":
            client.sendall((messege).encode())
        else:
            client.sendall(("(empty)").encode())
def communicate_to_server(client):
    username = input("Enter the user name (Type 'exit' for ending the chat): ")
    if username !="":
        client.sendall(username.encode())
    else:
        print("User name can not be empty! Enter valid user name again")
    threading.Thread(target=listen_to_server_mg, args=(client, username,)).start()
    send_mg_to_server(client)
def main():
    client=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    client.connect((des_ip,des_port))
    print("connected to the server")
    communicate_to_server(client)
if __name__ == '__main__':
    main()
```