Name : Subham Subhasis Sahoo
Entry : 2020csb1317
Server: It is designed as a multithreaded application that can handle multiple clients simultaneously. It has an object-oriented implementation that spawns multiple client threads upon each connection.

Client: It is designed in a class-based fashion. The client helps to connect to the server. It helps to receive and send messages and commands.

ChatRooms: They are designed in an object-oriented manner. Chat history is stored in a file format for each chatroom.

```
# Class to represent a chat room
class ChatRoom:
    def __init__(self, name, creator):
        self.name = name
        self.users = set()
        self.users.add(creator)
        self.history = ""

    def add_user(self, user):
        self.users.add(user)

    def remove_user(self, user):
        self.users.remove(user)

    def send_message(self, message):

        with open(self.name+'.txt', 'a') as f:
            f.write(message + '\n')
```

Connecting multiple clients to the server :

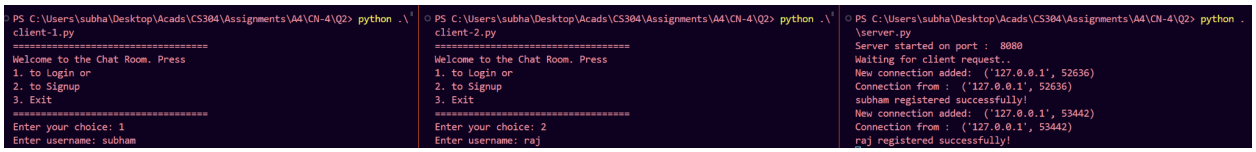
```
if __name__ == "__main__":

    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server.bind((HOST, PORT))


    print("Server started on port : ", PORT)
    print("Waiting for client request..")

    while True:
        server.listen(1)
        clientsock, clientAddress = server.accept()

        try:
            newthread = ClientThread(clientAddress, clientsock)
            newthread.start()
        except:
            print("Error in thread")
            clientsock.close()
```



Handle Authentication :

```python
def handleAuth(self):

    while True:

        method = self.connection.recv(1024).decode()
        if method == 'login':
            if (self.login() == 1):
                break

        elif method == 'signup':
            if (self.signup() == 1):
                break

    # Add the client to the dictionary of active clients
    ACTIVE_USERS[self.username] = self.connection

    # Send the list of active users to the client
    self.connection.send(
        f'[SYSTEM]: Active users: {list(ACTIVE_USERS.keys())} \n'.encode())
```

Signup :

```python
def signup(self):

    self.username = self.connection.recv(1024).decode()

    if len(USERS) != 0:
        if self.username in USERS:
            self.connection.send('[SYSTEM] :Username already exists!'.encode())
            return -1

    self.connection.send('[SYSTEM] :Enter Password : '.encode())

    password = self.connection.recv(1024).decode()
    USERS[self.username] = password
    self.connection.send(
        '[SYSTEM] :User registered successfully!'.encode())
    print(f'{self.username} registered successfully!')
    return 1
```

```
==================================
Welcome to the Chat Room. Press
1. to Login or
2. to Signup
3. Exit
==================================
Enter your choice: 2
Enter username: subham
Enter password: password
==================================
Welcome to the Chat Room. Press
1. Join a chat room
2. Create a chat room
3. Logout
==================================

[SYSTEM]: Active users: ['subham']
[SYSTEM]: Chat rooms: []


Enter Choice :
```

Login :

```python
def login(self):
    self.username = self.connection.recv(1024).decode()
    password = self.connection.recv(1024).decode()

    if len(USERS) != 0:

        if self.username in USERS:
            if password == USERS[self.username]:
                self.connection.send('[SYSTEM] :Login successful!'.encode())
                return 1
            else:
                self.connection.send(
                    '[SYSTEM] :Incorrect password!'.encode())
                return -1

    self.connection.send('User not registered!'.encode())
    return -1
```

```
====================================
Welcome to the Chat Room. Press
1. to Login or
2. to Signup
3. Exit
====================================
Enter your choice: 1
Enter username: raj
Enter password: password

[SYSTEM] :Login successful!
```

Join a chat room :

```python
def join_chat_room(self, chat_room_name):

    if(CHAT_ROOMS==set()):
        self.connection.send(
        f"[SYSTEM]: Chat room {chat_room_name} does not exist.".encode())

    else :
        for chat_room in CHAT_ROOMS:
            # Check if the chat room exists
            if chat_room.name == chat_room_name:

                # Add the user to the chat room
                chat_room.add_user(self.username)
                USERS_CHAT_ROOM[self.username] = chat_room_name

                # Send a message to the chat room
                chat_room_message = f"[SYSTEM][{datetime.datetime.now().strftime('%H:%M:%S')}]: {self.username} has joined the chat room."
                chat_room.send_message(
                    chat_room_message)

                # Send a message to the client
                self.connection.send(
                    f"[SYSTEM]: You have joined the chat room \"{chat_room_name}\"".encode())

            else:

                # Send a message to the client
                self.connection.send(
                    f"[SYSTEM]: Chat room {chat_room_name} does not exist.".encode())
```

```
==================================
Welcome to the Chat Room. Press
1. Join a chat room
2. Create a chat room
3. Logout
==================================

[SYSTEM]: Active users: ['subham', 'raj']
[SYSTEM]: Chat rooms: ['games']


Enter Choice : 1
Enter chat room name: games

[SYSTEM]: You have joined the chat room "games"

==================================
Welcome to the Chat Room. Press
1. to send a message
2. to receive messages
    ==========================
    While sending messages, You have the following command
      /leave : to leave the chatroom
 ==================================
```

Create a chat room :

```python
def create_chat_room(self, chat_room_name):

    if CHAT_ROOMS != set():

        for chat_room in CHAT_ROOMS:

            # Check if the chat room exists
            if chat_room.name == chat_room_name:

                # Send a message to the client
                self.connection.send(
                    f"[SYSTEM]: Chat room {chat_room_name} already exists. Please join it.".encode())
                return

    # Create a new chat room
    chat_room = ChatRoom(
        chat_room_name, self.username)
    CHAT_ROOMS.add(chat_room)

    # Add the user to the chat room
    chat_room.add_user(self.username)
    USERS_CHAT_ROOM[self.username] = chat_room_name

    # Send a message to the chat room
    chat_room_message = f"[SYSTEM][{datetime.datetime.now().strftime('%H:%M:%S')}]: {self.username} has created the chat room."

    f = open(chat_room_name+'.txt', 'w')
    f.close()

    chat_room.send_message(
        chat_room_message)

    # Send a message to the client
    self.connection.send(
        f"[SYSTEM]: You have created the chat room \"{chat_room_name}\" ".encode())
```

```
==================================
Welcome to the Chat Room. Press
1. Join a chat room
2. Create a chat room
3. Logout
==================================

[SYSTEM]: Active users: ['subham']
[SYSTEM]: Chat rooms: []


Enter Choice : 2
Enter chat room name: games

[SYSTEM]: You have created the chat room "games"


[SYSTEM]: You have created the chat room "games"

==================================
Welcome to the Chat Room. Press
1. to send a message
2. to receive messages
    ==========================
    While sending messages, You have the following command
      /leave : to leave the chatroom
 ==================================


Enter choice :
```

Send and receive messages:

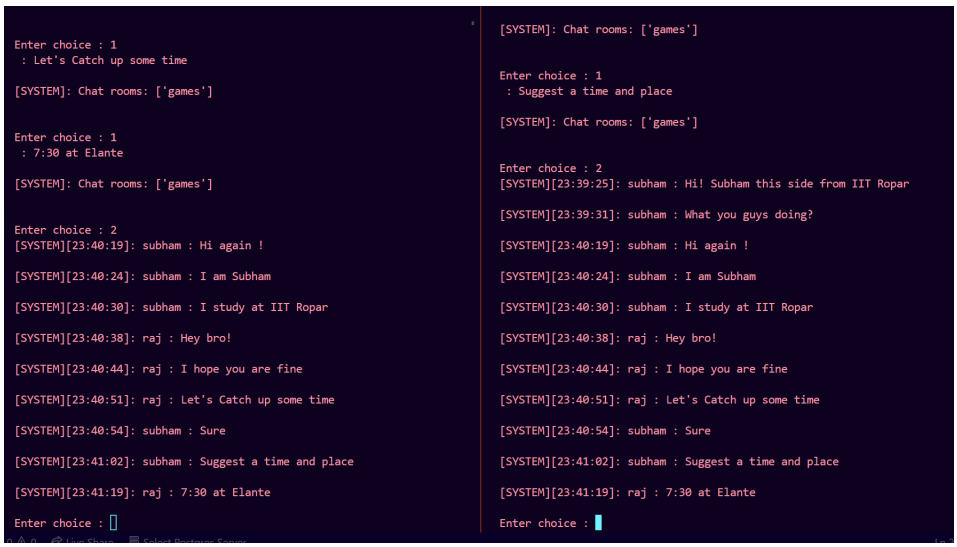
```python
    if (USERS_CHAT_ROOM[self.username] == None):
        self.connection.send(
            "[SYSTEM]: You are not in any chat room.".encode())

    else:
        # Get the chat room name
        chat_room_name = USERS_CHAT_ROOM[self.username]

        room_message = f"[SYSTEM][{datetime.datetime.now().strftime('%H:%M:%S')}]: {self.username} : {message} "

        # Find the chat room
        for chat_room in CHAT_ROOMS:
            if chat_room.name == chat_room_name:

                # Send the message to the chat room
                chat_room.send_message(
                    room_message)
```



Leave the chatroom :

```python
    def leave_chat_room(self, chat_room):

        # Remove the user from the chat room
        chat_room.remove_user(self.username)
        USERS_CHAT_ROOM.pop(self.username)

        # Send a message to the chat room
        chat_room_message = f"[SYSTEM][{datetime.datetime.now().strftime('%H:%M:%S')}]: {self.username} has left the chat room."
        chat_room.send_message(chat_room_message)

        # Send a message to the client
        self.connection.send(
            f"[SYSTEM]: You have left the chat room \"{chat_room.name}\".".encode())
```

First press 1 to send message and then type /leave to leave the chatroom

```
Enter choice : 1
 : /leave
====================================
Welcome to the Chat Room. Press
1. Join a chat room
2. Create a chat room
3. Logout
====================================

[SYSTEM]: Chat rooms: ['games']
[SYSTEM]: You have left the chat room "games".
[SYSTEM]: Chat rooms: ['games']
```

Logout :

```python
    if self.username in USERS_CHAT_ROOM:

        chat_room_name = USERS_CHAT_ROOM[self.username]

        # Find the chat room
        for chat_room in CHAT_ROOMS:
            if chat_room.name == chat_room_name:
                self.leave_chat_room(chat_room)

    # Remove the user from the active users
    ACTIVE_USERS.pop(self.username)

    # Send a message to the client
    self.connection.send(
        "[SYSTEM]: Logged out successfully!".encode())

    # Close the self.connection
    self.connection.close()
    return -1
```

```
3. Logout
==================================

[SYSTEM]: Chat rooms: ['games']
[SYSTEM]: You have left the chat room "games".[SYSTEM]: Chat rooms: ['games']


Enter Choice : 3

[SYSTEM]: Logged out successfully!

Logging out...
==================================
Welcome to the Chat Room. Press
1. to Login or
2. to Signup
3. Exit
==================================
Enter your choice:
```