Internship Task - 3: Multithreaded Chat Application

Student Submission (Using VS Code)

This is a basic student-level implementation of a multithreaded chat application using Java.

The project was developed using Visual Studio Code with minimal styling and structure to reflect a beginner's effort.

Instructions Followed:

- Built a simple client-server chat application using Java Sockets.

- Used multithreading on the server side to handle multiple client connections.

- Ensured that all clients can send and receive messages in real time.

Tools Used:

- Java

- VS Code

- Command line for compiling and running

Code Structure:

1. Server.java

2. Client.java

The code is functional and demonstrates a basic understanding of Java Sockets and Threads.

Server.java

```
import java.io.*;
import java.net.*;
import java.util.*;

public class Server {
    private static Vector<ClientHandler> clients = new Vector<>();
    public static void main(String[] args) throws IOException {
        ServerSocket ss = new ServerSocket(1234);
        System.out.println("Server started...");
        while (true) {
            Socket s = ss.accept();
            System.out.println("Client connected.");
            ClientHandler client = new ClientHandler(s, clients);
            clients.add(client);
            new Thread(client).start();
        }
    }
}
class ClientHandler implements Runnable {
    Socket socket;
    Vector<ClientHandler> clients;
    BufferedReader in;
    PrintWriter out;

         public  ClientHandler(Socket  socket,  Vector<ClientHandler>  clients)  throws
IOException {
        this.socket = socket;
        this.clients = clients;
        in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        out = new PrintWriter(socket.getOutputStream(), true);
    }

    public void run() {
        try {
            String msg;
            while ((msg = in.readLine()) != null) {
                for (ClientHandler c : clients) {
                    c.out.println(msg);
                }
            }
        } catch (IOException e) {
            System.out.println("Client disconnected.");
        }
    }
}
```

Client.java

```
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        Socket s = new Socket("localhost", 1234);
        BufferedReader in = new BufferedReader(new
InputStreamReader(s.getInputStream()));
        PrintWriter out = new PrintWriter(s.getOutputStream(), true);
        BufferedReader console = new BufferedReader(new InputStreamReader(System.in));

        new Thread(() -> {
            try {
                String msg;
                while ((msg = in.readLine()) != null) {
                    System.out.println("Server: " + msg);
                }
            } catch (IOException e) {
                System.out.println("Connection closed.");
            }
        }).start();

        String input;
        while ((input = console.readLine()) != null) {
            out.println(input);
        }
    }
}
```