Chat server.java

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ChatServer {

private static final int PORT = 12345;

private static HashSet<PrintWriter> writers = new HashSet<>();

public static void main(String[] args) {

System.out.println("Chat Server is running...");

try (ServerSocket serverSocket = new ServerSocket(PORT)) {

while (true) {

new ClientHandler(serverSocket.accept()).start();

}

} catch (IOException e) {

System.err.println("Could not listen on port " + PORT);

System.exit(1);

}

}

// ClientHandler class to manage each client connection

private static class ClientHandler extends Thread {

private Socket socket;

private PrintWriter out;

private BufferedReader in;

private String name;

public ClientHandler(Socket socket) {

this.socket = socket;

}

public void run() {

try {

// Set up input and output streams

in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

out = new PrintWriter(socket.getOutputStream(), true);

// Request and store client name

while (true) {

out.println("SUBMITNAME");

name = in.readLine();

if (name == null) {

return;

}

synchronized (writers) {

if (!name.isEmpty()) {

break;

}

}

}

// Welcome message

out.println("NAMEACCEPTED " + name);

broadcast(name + " has joined the chat!");

// Add client to writers set

synchronized (writers) {

writers.add(out);

}

// Handle client messages

String message;

while ((message = in.readLine()) != null) {

if (!message.isEmpty()) {

broadcast(name + ": " + message);

}

}

} catch (IOException e) {

System.out.println(e);

} finally {

// Client is leaving

if (name != null) {

broadcast(name + " has left the chat.");

}

if (out != null) {

synchronized (writers) {

writers.remove(out);

}

}

try {

socket.close();

} catch (IOException e) {

System.out.println(e);

}

}

}

// Broadcast message to all clients

private void broadcast(String message) {

synchronized (writers) {

for (PrintWriter writer : writers) {

writer.println("MESSAGE " + message);

}

}

}

}

}

Chat client.java

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class ChatClient {

private static final String SERVER\_ADDRESS = "localhost";

private static final int SERVER\_PORT = 12345;

public static void main(String[] args) {

try {

Socket socket = new Socket(SERVER\_ADDRESS, SERVER\_PORT);

System.out.println("Connected to chat server");

// Create reader for server messages

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

// Create writer for sending messages

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

// Scanner for user input

Scanner scanner = new Scanner(System.in);

// Get username

System.out.print("Enter your name: ");

String name = scanner.nextLine();

out.println(name);

// Start a thread to handle server messages

new Thread(() -> {

try {

String message;

while ((message = in.readLine()) != null) {

if (message.startsWith("SUBMITNAME")) {

out.println(name);

} else if (message.startsWith("NAMEACCEPTED")) {

System.out.println("Name accepted by server");

} else if (message.startsWith("MESSAGE")) {

System.out.println(message.substring(8));

}

}

} catch (IOException e) {

System.out.println("Error reading from server: " + e.getMessage());

}

}).start();

// Main thread handles user input

String message;

while (true) {

message = scanner.nextLine();

if (message.equalsIgnoreCase("/quit")) {

break;

}

out.println(message);

}

// Clean up

socket.close();

scanner.close();

} catch (IOException e) {

System.out.println("Error connecting to server: " + e.getMessage());

}

}

}