**Task 3**

**Build a client server chat application using java sockets and multithreaded to handle multiple users**

**Server Code:**

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ChatServer {

private static final int PORT = 12345;

private static Set<PrintWriter> clientWriters = new HashSet<>();

public static void main(String[] args) {

System.out.println("Chat server started...");

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

while (true) {

Socket clientSocket = serverSocket.accept();

System.out.println("New client connected: " + clientSocket.getInetAddress());

new ClientHandler(clientSocket).start();

}

} catch (IOException e) {

System.err.println("Error in server: " + e.getMessage());

}

}

private static class ClientHandler extends Thread {

private Socket socket;

private PrintWriter out;

public ClientHandler(Socket socket) {

this.socket = socket;

}

@Override

public void run() {

try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

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

synchronized (clientWriters) {

clientWriters.add(out);

}

String message;

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

System.out.println("Received: " + message);

broadcastMessage(message);

}

} catch (IOException e) {

System.err.println("Client disconnected: " + e.getMessage());

} finally {

try {

socket.close();

} catch (IOException e) {

System.err.println("Error closing client socket: " + e.getMessage());

}

synchronized (clientWriters) {

clientWriters.remove(out);

}

}

}

private void broadcastMessage(String message) {

synchronized (clientWriters) {

for (PrintWriter writer : clientWriters) {

writer.println(message);

}

}

}

}

}

**Client Code:**

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 the chat server.");

new Thread(new IncomingMessagesHandler(socket)).start();

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

Scanner scanner = new Scanner(System.in);

System.out.println("Enter your messages (type 'exit' to quit):");

while (true) {

String message = scanner.nextLine();

if (message.equalsIgnoreCase("exit")) {

break;

}

out.println(message);

}

} catch (IOException e) {

System.err.println("Error in client: " + e.getMessage());

}

}

private static class IncomingMessagesHandler implements Runnable {

private Socket socket;

public IncomingMessagesHandler(Socket socket) {

this.socket = socket;

}

@Override

public void run() {

try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

String message;

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

System.out.println("Server: " + message);

}

} catch (IOException e) {

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

}

}

}

}

**Steps:**

1.Create ChatServer.java file and write the server code in it

2.Run ChatServer.java File

3.Create ChatClient.java file and write the client code in it

4.Run the ChatClient.java File

5.Now we can start chatting with multiple clients also

6.By taking instatance of netbeans or new window of netbeans by running both files we can connect client to server.

**Output:**



