CN Assignment II

Ishank Nijhawan

1710110150

1. serverHttp.java

```
package com.ishank;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

public class serverHttp {
    //static ServerSocket variable
    private static ServerSocket serverSocket;
    //static socket variable
    private static Socket socket;
    //socket server port on which it will listen
    private static int port = 6666;

    public static void main(String[] args) throws IOException {
        try {
            //server waits for connection requests
            System.out.println("\nwaiting for incoming connection requests...\n");
            serverSocket = new ServerSocket(port);
            int counter = 0;

            //infinite loop for accepting socket requests
            while (true){
                counter++;
                socket = serverSocket.accept();
                System.out.println("Client " + counter + " started!");
                //passing the client request to a new thread
                ServerClientThread serverClientThread = new
ServerClientThread(socket,counter,serverSocket);
                serverClientThread.start();
            }
        }
        catch (Exception e){
            System.out.println("All the clients have disconnected, Socket closed");
        }
    }
}

//ServerClientThread class extending Thread class
```

```
class ServerClientThread extends Thread {
   Socket clientSocket;
   int clientCount;
   ServerSocket ss;
   //ServerClientThread constructor with socket and counter as the parameters
   ServerClientThread(Socket inSocket,int counter, ServerSocket serverSocket){
      clientSocket = inSocket;
      clientCount = counter;
      ss = serverSocket;
   }
   //calling run method
   public void run() {
      try {
         //declaring input and output streams
         DataInputStream dataInputStream = new
DataInputStream(clientSocket.getInputStream());
         DataOutputStream dataOutputStream = new
DataOutputStream(clientSocket.getOutputStream());
         String clientMessage="", serverMessage="server connected";
         dataOutputStream.writeUTF(serverMessage);

         //if clientMessage == Bye, terminate the connection
         while (!clientMessage.equals("Bye")) {
            //reading message from the client side
            clientMessage = dataInputStream.readUTF();

            if(!clientMessage.contains("GET"))
               break;

            String[] splitArray = clientMessage.split(" ");

            //reading clientMessage and sending file name to read the html file
            File file = new File("C:\\" + splitArray[1]);
            BufferedReader br = new BufferedReader(new FileReader(file));
            String initialString,finalString="";
            while ((initialString = br.readLine()) != null)
               finalString = finalString + initialString;

            //parsing the html using Jsoup
            Document html = Jsoup.parse(finalString);
            String body = html.body().text();

            //sending back message to the client
            dataOutputStream.writeUTF("200 OK\n" + body);
            dataOutputStream.flush();
         }

         dataInputStream.close();
         dataOutputStream.close();
         System.out.println("connection with client " + clientCount + " closed");
         //reducing client number when a client disconnects
         clientCount = clientCount -1;
```

```
            //System.out.println("Number of active connections with the server: " + clientCount);
            //closing the socket
            clientSocket.close();
            //if clientNo becomes 0, close the server
            if (clientCount ==0){
                ss.close();
            }

        }
        catch (Exception ex){
            System.out.println("");
        }
    }
}
```

2. clientHttp.java

```
package com.ishank;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class clientHttp {
    public static void main(String[] args) throws IOException, UnknownHostException {
        try {
            //InetAddress variable to get the host name
            InetAddress host = InetAddress.getLocalHost();
            //socket object for the client side
            Socket socket = new Socket(host.getHostName(),6666);
            //input and output streams
            System.out.println("server connected\n");

            DataInputStream dataInputStream = new DataInputStream(socket.getInputStream());
            DataOutputStream dataOutputStream = new
DataOutputStream(socket.getOutputStream());

            //server sends hello message to it's clients
            System.out.println("message from server: " + dataInputStream.readUTF());

            Scanner input = new Scanner(System.in);
            String clientMessage="",serverMessage="";

            //if client message is equal to bye, terminate the connection with the server
            while (!clientMessage.equals("Bye")){
```

```
            System.out.println("request the file from the server\n");
            clientMessage = input.nextLine();
            //sending the input to the server
            dataOutputStream.writeUTF(clientMessage);
            dataOutputStream.flush();

            //reading the reply from the server and printing on the client screen
            serverMessage = dataInputStream.readUTF();
            System.out.println(serverMessage);
        }
        //exit the loop once the client types Bye
        System.out.println("connection closed");
        dataOutputStream.close();
        //close the socket
        socket.close();
    }
    catch (Exception e){
        System.out.println("Server disconnected");
    }
  }
}
```