



IT7412

WEB TECHNOLOGY LABORATORY

MULTI-CLIENT CHAT ROOM SYSTEM

***BY***

***HARIHARAN.S (2018506037)***

***SANTHOSH.Y (2018506107)***

**OBJECTIVE**

The aim of this project is to develop a multi-user chat system to facilitate communication between users within the multiple user clients. Chat application is to allow users for private chat, group chat (conference chat).

In our chat system, the thread will be used to control the flow of messages sent and received by users, because imagine if all they were stored in a queue and processed only by a thread: the service would be precarious and probably no one would use it. It is a multithreaded client/server ChatServer based on JAVA SWING which uses Java Socket programming. A server listens for connection requests from clients across the network or even from the same machine. Clients know how to connect to the server via an IP address and port number. After connecting to the server, the client sends a message, the message is sent to the server using ObjectOutputStream in java. After receiving the message from the client, the server broadcasts the message if it is not a private message.

**PLATFORMS USED**

* **ECLIPSE IDE**

**CONCEPTS USED**

**SOCKETS**

A socket in Java is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

**MULTI-THREADING**

MULTITHREADING in Java is a process of executing two or more threads simultaneously to maximum utilization of CPU. Multithreaded applications execute two or more threads concurrently. Each thread runs parallel to each other. Multiple threads don't allocate separate memory area, hence they save memory.

**SWING**

**EXCEPTION HANDLING**

**ARRAY LIST**

**STRING HANDLING**

**JDBC (Java Database Connectivity)**

**SERVER SIDE MODULES**

We have used JAVA SWING for server side connection, the module specification for the server side is java swing is given below:

**Start**

**The start button is used to initiate the connection of the server.**

**Online users**

**This button is used to show the users who are currently available online.**

**Clear**

**This button is used to clear the contents in the JFRAME.**

**End**

**This button is used to end the server connection.**

CLIENT SIDE MODULES

We have used JAVA SWING for client side connection, the module specification for the client side is java swing is given below :

**Address**

The address textarea is used for entering the IP address of the computer system, by default it will be LOCALHOST.

**Port**

The port textarea is used for entering the port number of the netbeans application.

**Connect**

This button is used for starting the connection between the client and the server.

**Disconnect**

This button is used for ending the connection between the client and the server.

**Username**

This text area is used for entering the username of the client for group chat.

**Password**

This text area is used for entering the password of the client for group chat.

**Unknown user**

This button is used to make an anonymous group chat without any username and password. After the disconnection of the unknown chat, the private group chat is initiated or vice versa.

**Send**

This button is used for sending messages in the group chat room.

**SOURCE CODE**

**Server\_frame.java**

package chat\_server;

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

public class server\_frame extends javax.swing.JFrame

{

ArrayList clientOutputStreams; ArrayList<String> users;

public class ClientHandler implements Runnable

{

BufferedReader reader; Socket sock; PrintWriter client;

public ClientHandler(Socket clientSocket, PrintWriter user)

{

client = user; try

{

sock = clientSocket;

InputStreamReader isReader = new InputStreamReader(sock.getInputStream()); reader = new BufferedReader(isReader);

}

catch (Exception ex)

{

ta\_chat.append("Unexpected error... \n");

}

}

@Override public void run()

{

String message, connect = "Connect", disconnect = "Disconnect", chat = "Chat" ; String[] data;

try

{

while ((message = reader.readLine()) != null)

{

ta\_chat.append("Received: " + message + "\n"); data = message.split(":");

for (String token:data)

{

ta\_chat.append(token + "\n");

}

if (data[2].equals(connect))

{

tellEveryone((data[0] + ":" + data[1] + ":" + chat)); userAdd(data[0]);

}

else if (data[2].equals(disconnect))

{

tellEveryone((data[0] + ":has disconnected." + ":" + chat)); userRemove(data[0]);

}

else if (data[2].equals(chat))

{

tellEveryone(message);

}

else

{

ta\_chat.append("No Conditions were met. \n");

}

}

}

catch (Exception ex)

{

ta\_chat.append("Lost a connection. \n"); ex.printStackTrace(); clientOutputStreams.remove(client);

}

}

}

public server\_frame()

{

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane(); ta\_chat = new javax.swing.JTextArea(); b\_start = new javax.swing.JButton();

b\_end = new javax.swing.JButton();

b\_users = new javax.swing.JButton(); b\_clear = new javax.swing.JButton(); lb\_name = new javax.swing.JLabel(); jLabel1 = new javax.swing.JLabel(); jLabel2 = new javax.swing.JLabel(); jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE); setTitle("Chat - Server's frame");

setName("server"); // NOI18N setResizable(false);

ta\_chat.setColumns(20); ta\_chat.setRows(5); jScrollPane1.setViewportView(ta\_chat);

b\_start.setText("START");

b\_start.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_startActionPerformed(evt);

}

});

b\_end.setText("END");

b\_end.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_endActionPerformed(evt);

}

});

b\_users.setText("Online Users");

b\_users.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_usersActionPerformed(evt);

}

});

b\_clear.setText("Clear");

b\_clear.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_clearActionPerformed(evt);

}

});

lb\_name.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0,

0)));

jLabel1.setText("ANNAUNIVERSITY SERVER");

jLabel2.setText("PROJECT DONE BY ");

jLabel3.setText(" HARIHARAN SANTHOSH");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane()); getContentPane().setLayout(layout);

layout.setHorizontalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(b\_end, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(b\_start, javax.swing.GroupLayout.DEFAULT\_SIZE, 75, Short.MAX\_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(47, 47, 47)

.addComponent(jLabel3))

.addGroup(layout.createSequentialGroup()

.addGap(130, 130, 130)

.addComponent(jLabel2)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 82, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(b\_clear, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(b\_users, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, Short.MAX\_VALUE))))

.addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lb\_name)

.addGap(209, 209, 209))

.addGroup(layout.createSequentialGroup()

.addGap(170, 170, 170)

.addComponent(jLabel1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 247, Short.MAX\_VALUE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(b\_start)

.addComponent(b\_users)

.addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(b\_clear)

.addComponent(b\_end))

.addGap(18, 18, 18))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(37, 37, 37)))

.addComponent(lb\_name))

);

pack();

}// </editor-fold>

private void b\_endActionPerformed(java.awt.event.ActionEvent evt) { try

{

Thread.sleep(5000); //5000 milliseconds is five second.

}

catch(InterruptedException ex) {Thread.currentThread().interrupt();}

tellEveryone("Server:is stopping and all users will be disconnected.\n:Chat"); ta\_chat.append("Server stopping... \n");

ta\_chat.setText("");

}

private void b\_startActionPerformed(java.awt.event.ActionEvent evt) { Thread starter = new Thread(new ServerStart());

starter.start();

ta\_chat.append("Server started...\n");

}

private void b\_usersActionPerformed(java.awt.event.ActionEvent evt) {

ta\_chat.append("\n Online users : \n"); for (String current\_user : users)

{

ta\_chat.append(current\_user); ta\_chat.append("\n");

}

}

private void b\_clearActionPerformed(java.awt.event.ActionEvent evt) { ta\_chat.setText("");

}

public static void main(String args[])

{

java.awt.EventQueue.invokeLater(new Runnable()

{

@Override public void run() {

new server\_frame().setVisible(true);

}

});

}

public class ServerStart implements Runnable

{

@Override public void run()

{

clientOutputStreams = new ArrayList(); users = new ArrayList();

try

{

ServerSocket serverSock = new ServerSocket(2222);

while (true)

{

writer));

}

}

Socket clientSock = serverSock.accept();

PrintWriter writer = new PrintWriter(clientSock.getOutputStream()); clientOutputStreams.add(writer);

Thread listener = new Thread(new ClientHandler(clientSock, listener.start();

ta\_chat.append("Got a connection. \n");

catch (Exception ex)

{

ta\_chat.append("Error making a connection. \n");

}

}

}

public void userAdd (String data)

{

String message, add = ": :Connect", done = "Server: :Done", name = data; ta\_chat.append("Before " + name + " added. \n");

users.add(name);

ta\_chat.append("After " + name + " added. \n"); String[] tempList = new String[(users.size())]; users.toArray(tempList);

for (String token:tempList)

{

message = (token + add); tellEveryone(message);

}

tellEveryone(done);

}

public void userRemove (String data)

{

String message, add = ": :Connect", done = "Server: :Done", name = data; users.remove(name);

String[] tempList = new String[(users.size())]; users.toArray(tempList);

for (String token:tempList)

{

message = (token + add); tellEveryone(message);

}

tellEveryone(done);

}

public void tellEveryone(String message)

{

Iterator it = clientOutputStreams.iterator();

while (it.hasNext())

{

try

{

PrintWriter writer = (PrintWriter) it.next();

writer.println(message); ta\_chat.append("Sending: " + message + "\n");

writer.flush(); ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

}

catch (Exception ex)

{

ta\_chat.append("Error telling everyone. \n");

}

}

}

private javax.swing.JButton b\_clear; private javax.swing.JButton b\_end; private javax.swing.JButton b\_start; private javax.swing.JButton b\_users; private javax.swing.JLabel jLabel1; private javax.swing.JLabel jLabel2; private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1; private javax.swing.JLabel lb\_name;

private javax.swing.JTextArea ta\_chat;

}

# Client\_frame.java

package chat\_client;

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

public class client\_frame extends javax.swing.JFrame

{

String username, address = "localhost"; ArrayList<String> users = new ArrayList(); int port = 2222;

Boolean isConnected = false;

Socket sock; BufferedReader reader; PrintWriter writer;

//--------------------------//

public void ListenThread()

{

Thread IncomingReader = new Thread(new IncomingReader());

IncomingReader.start();

}

//--------------------------//

public void userAdd(String data)

{

users.add(data);

}

//--------------------------//

public void userRemove(String data)

{

ta\_chat.append(data + " is now offline.\n");

}

//--------------------------//

public void writeUsers()

{

String[] tempList = new String[(users.size())]; users.toArray(tempList);

for (String token:tempList)

{

//users.append(token + "\n");

}

}

//--------------------------//

public void sendDisconnect()

{

String bye = (username + ": :Disconnect"); try

{

writer.println(bye); writer.flush();

} catch (Exception e)

{

ta\_chat.append("Could not send Disconnect message.\n");

}

}

//--------------------------//

public void Disconnect()

{

try

{

ta\_chat.append("Disconnected.\n"); sock.close();

} catch(Exception ex) { ta\_chat.append("Failed to disconnect. \n");

}

isConnected = false; tf\_username.setEditable(true);

}

public client\_frame()

{

initComponents();

}

//--------------------------//

public class IncomingReader implements Runnable

{

@Override public void run()

{

"Chat";

String[] data;

String stream, done = "Done", connect = "Connect", disconnect = "Disconnect", chat =

try

{

while ((stream = reader.readLine()) != null)

{

data = stream.split(":");

if (data[2].equals(chat))

{

ta\_chat.append(data[0] + ": " + data[1] + "\n"); ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

}

else if (data[2].equals(connect))

{

ta\_chat.removeAll(); userAdd(data[0]);

}

else if (data[2].equals(disconnect))

{

userRemove(data[0]);

}

else if (data[2].equals(done))

{

//users.setText(""); writeUsers(); users.clear();

}

}

}catch(Exception ex) { }

}

}

//--------------------------//

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() {

lb\_address = new javax.swing.JLabel(); tf\_address = new javax.swing.JTextField(); lb\_port = new javax.swing.JLabel(); tf\_port = new javax.swing.JTextField(); lb\_username = new javax.swing.JLabel();

tf\_username = new javax.swing.JTextField(); lb\_password = new javax.swing.JLabel();

tf\_password = new javax.swing.JTextField(); b\_connect = new javax.swing.JButton(); b\_disconnect = new javax.swing.JButton(); b\_anonymous = new javax.swing.JButton(); jScrollPane1 = new javax.swing.JScrollPane(); ta\_chat = new javax.swing.JTextArea(); tf\_chat = new javax.swing.JTextField(); b\_send = new javax.swing.JButton();

lb\_name = new javax.swing.JLabel(); jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE); setTitle("Chat - Client's frame");

setName("client"); // NOI18N setResizable(false);

lb\_address.setText("Address : ");

tf\_address.setText("localhost");

tf\_address.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_addressActionPerformed(evt);

}

});

lb\_port.setText("Port :");

tf\_port.setText("2222");

tf\_port.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_portActionPerformed(evt);

}

});

lb\_username.setText("Username :");

tf\_username.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_usernameActionPerformed(evt);

}

});

lb\_password.setText("Password : ");

b\_connect.setText("Connect");

b\_connect.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_connectActionPerformed(evt);

}

});

b\_disconnect.setText("Disconnect"); b\_disconnect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) { b\_disconnectActionPerformed(evt);

}

});

b\_anonymous.setText("UNKNOWN USER"); b\_anonymous.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) { b\_anonymousActionPerformed(evt);

}

});

ta\_chat.setColumns(20); ta\_chat.setRows(5); jScrollPane1.setViewportView(ta\_chat);

b\_send.setText("SEND");

b\_send.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt) {

}

});

b\_sendActionPerformed(evt);

lb\_name.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0,

0)));

jLabel1.setText("ANNAUNIVERSITY CHAT ROOM");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane()); getContentPane().setLayout(layout);

layout.setHorizontalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 4, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(tf\_chat, javax.swing.GroupLayout.PREFERRED\_SIZE, 347, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lb\_name))

.addGap(16, 16, 16)

.addComponent(b\_send, javax.swing.GroupLayout.PREFERRED\_SIZE, 152, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33))

.addGroup(layout.createSequentialGroup()

.addGap(4, 4, 4)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 528, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(lb\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, Short.MAX\_VALUE)

.addComponent(lb\_address, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(tf\_address, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(tf\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lb\_password, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lb\_port, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(tf\_password)

.addComponent(tf\_port, javax.swing.GroupLayout.DEFAULT\_SIZE, 99,

Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(165, 165, 165)

.addComponent(jLabel1)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(b\_connect)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(b\_disconnect))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(b\_anonymous)

.addGap(31, 31, 31))))))

);

layout.setVerticalGroup( layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 16, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lb\_address)

.addComponent(tf\_address, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lb\_port)

.addComponent(tf\_port, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_anonymous))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,

false)

.addComponent(tf\_password)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lb\_username)

.addComponent(lb\_password)

.addComponent(b\_connect)

.addComponent(b\_disconnect)

.addComponent(tf\_username)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 231, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_chat, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_send, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, Short.MAX\_VALUE)

.addComponent(lb\_name))

);

pack();

}// </editor-fold>

private void tf\_addressActionPerformed(java.awt.event.ActionEvent evt) {

}

private void tf\_portActionPerformed(java.awt.event.ActionEvent evt) {

}

private void tf\_usernameActionPerformed(java.awt.event.ActionEvent evt) {

}

private void b\_connectActionPerformed(java.awt.event.ActionEvent evt) { if (isConnected == false)

{

username = tf\_username.getText(); tf\_username.setEditable(false);

try

{

sock = new Socket(address, port);

InputStreamReader streamreader = new InputStreamReader(sock.getInputStream()); reader = new BufferedReader(streamreader);

writer = new PrintWriter(sock.getOutputStream()); writer.println(username + ":has connected.:Connect"); writer.flush();

isConnected = true;

}

catch (Exception ex)

{

ta\_chat.append("Cannot Connect! Try Again. \n");

tf\_username.setEditable(true);

}

ListenThread();

} else if (isConnected == true)

{

ta\_chat.append("You are already connected. \n");

}

}

private void b\_disconnectActionPerformed(java.awt.event.ActionEvent evt) { sendDisconnect();

Disconnect();

}

private void b\_anonymousActionPerformed(java.awt.event.ActionEvent evt) { tf\_username.setText("");

if (isConnected == false)

{

String anon="anon";

Random generator = new Random(); int i = generator.nextInt(999) + 1; String is=String.valueOf(i);

anon=anon.concat(is); username=anon;

tf\_username.setText(anon); tf\_username.setEditable(false);

try

{

sock = new Socket(address, port);

InputStreamReader streamreader = new InputStreamReader(sock.getInputStream()); reader = new BufferedReader(streamreader);

writer = new PrintWriter(sock.getOutputStream()); writer.println(anon + ":has connected.:Connect"); writer.flush();

isConnected = true;

}

catch (Exception ex)

{

ta\_chat.append("Cannot Connect! Try Again. \n"); tf\_username.setEditable(true);

}

ListenThread();

} else if (isConnected == true)

{

ta\_chat.append("You are already connected. \n");

}

}

private void b\_sendActionPerformed(java.awt.event.ActionEvent evt) { String nothing = "";

if ((tf\_chat.getText()).equals(nothing)) { tf\_chat.setText(""); tf\_chat.requestFocus();

} else {

try {

writer.println(username + ":" + tf\_chat.getText() + ":" + "Chat"); writer.flush(); // flushes the buffer

} catch (Exception ex) { ta\_chat.append("Message was not sent. \n");

}

tf\_chat.setText(""); tf\_chat.requestFocus();

}

tf\_chat.setText(""); tf\_chat.requestFocus();

}

public static void main(String args[])

{

java.awt.EventQueue.invokeLater(new Runnable()

{

@Override public void run()

{

new client\_frame().setVisible(true);

}

});

}

private javax.swing.JButton b\_anonymous; private javax.swing.JButton b\_connect; private javax.swing.JButton b\_disconnect; private javax.swing.JButton b\_send; private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1; private javax.swing.JLabel lb\_address; private javax.swing.JLabel lb\_name;

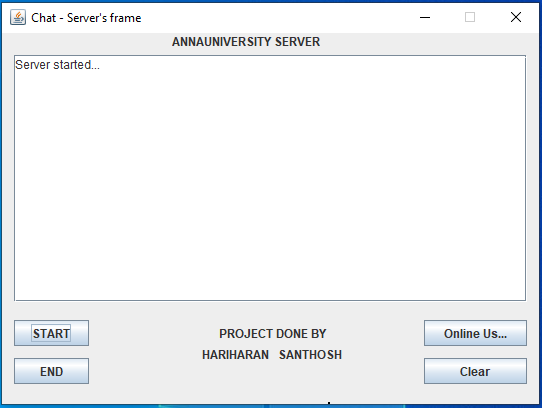
private javax.swing.JLabel lb\_password; private javax.swing.JLabel lb\_port;

private javax.swing.JLabel lb\_username; private javax.swing.JTextArea ta\_chat; private javax.swing.JTextField tf\_address; private javax.swing.JTextField tf\_chat; private javax.swing.JTextField tf\_password; private javax.swing.JTextField tf\_port; private javax.swing.JTextField tf\_username;

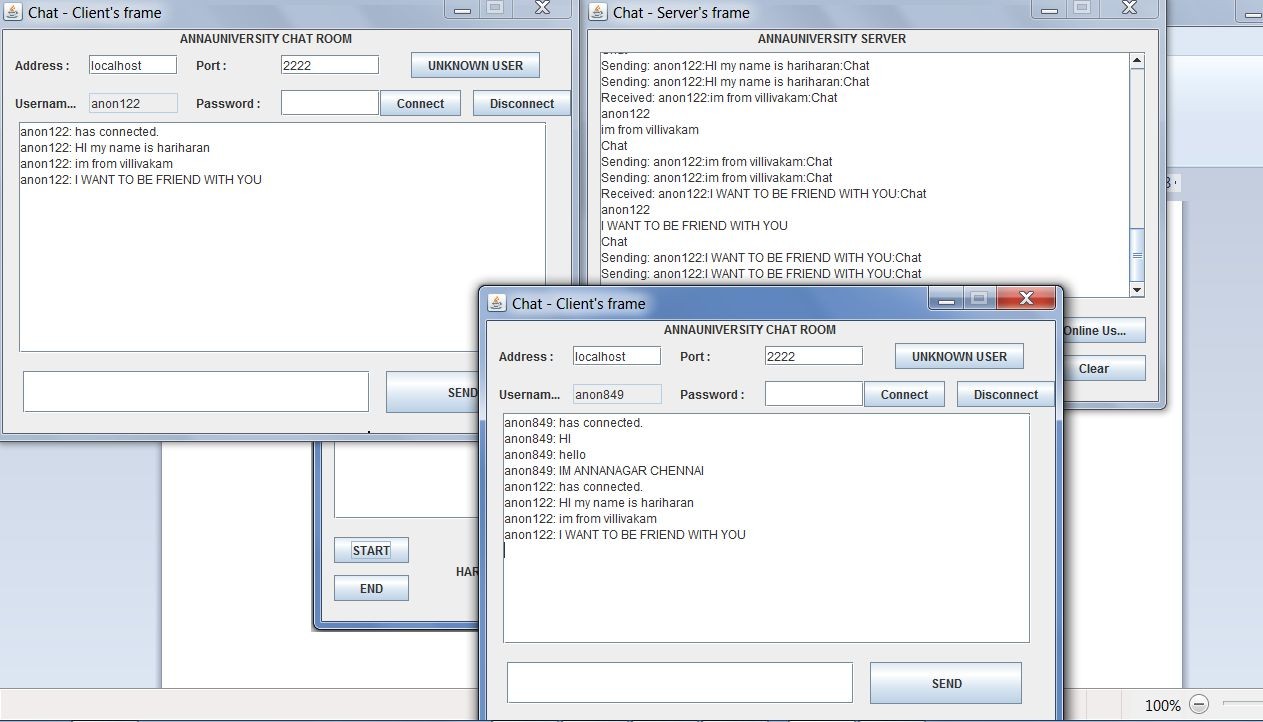
}

# OUTPUT SCREENSHOTS

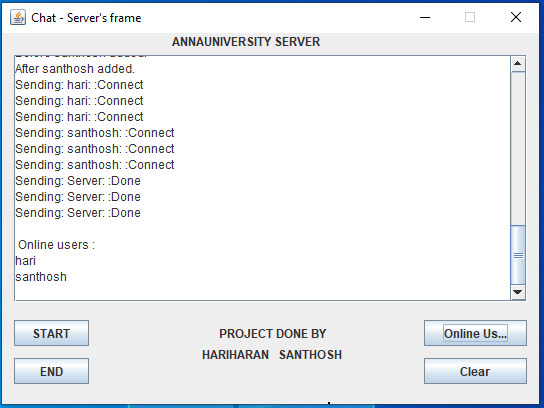
## STARTING OF THE SERVER

****

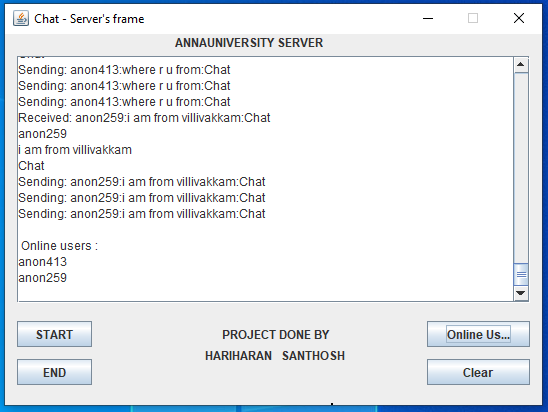
**TWO ANONYMOUS USER’S CHAT**



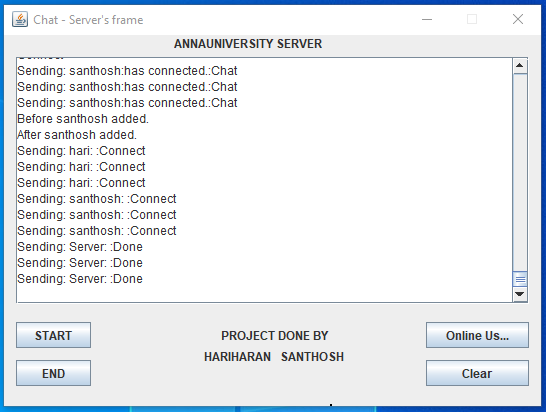
DISPLAYING ONLINE USERS



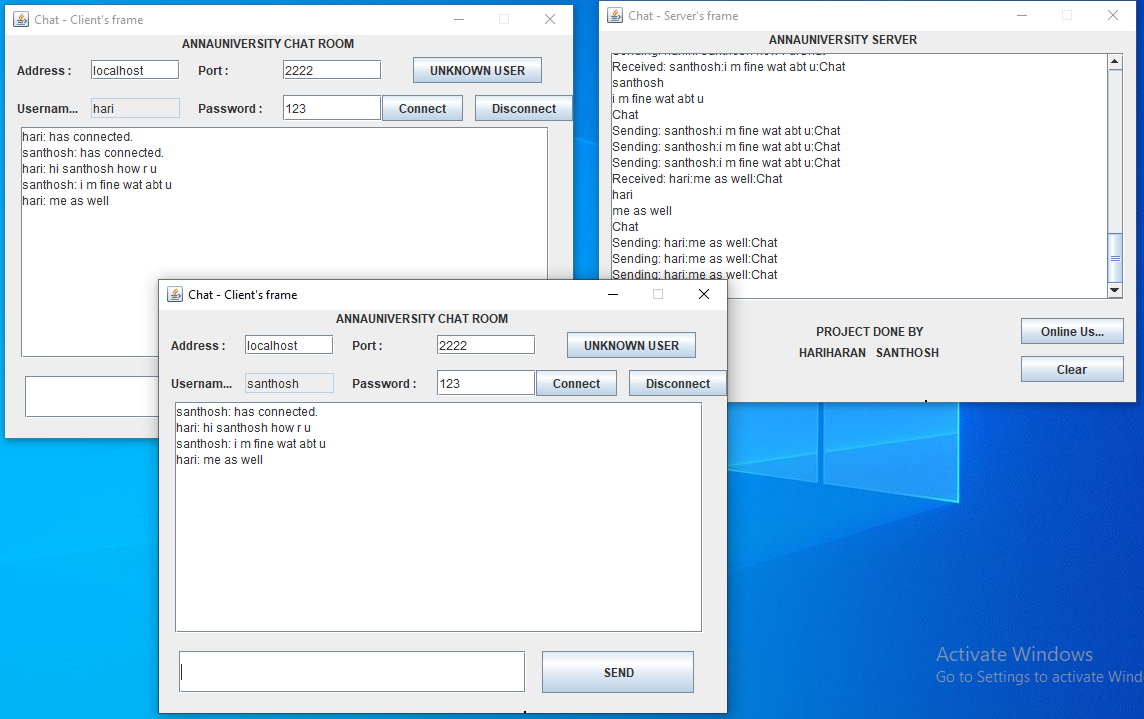
DISCONNECTION BETWEEN THE ANONYMOUS USERS



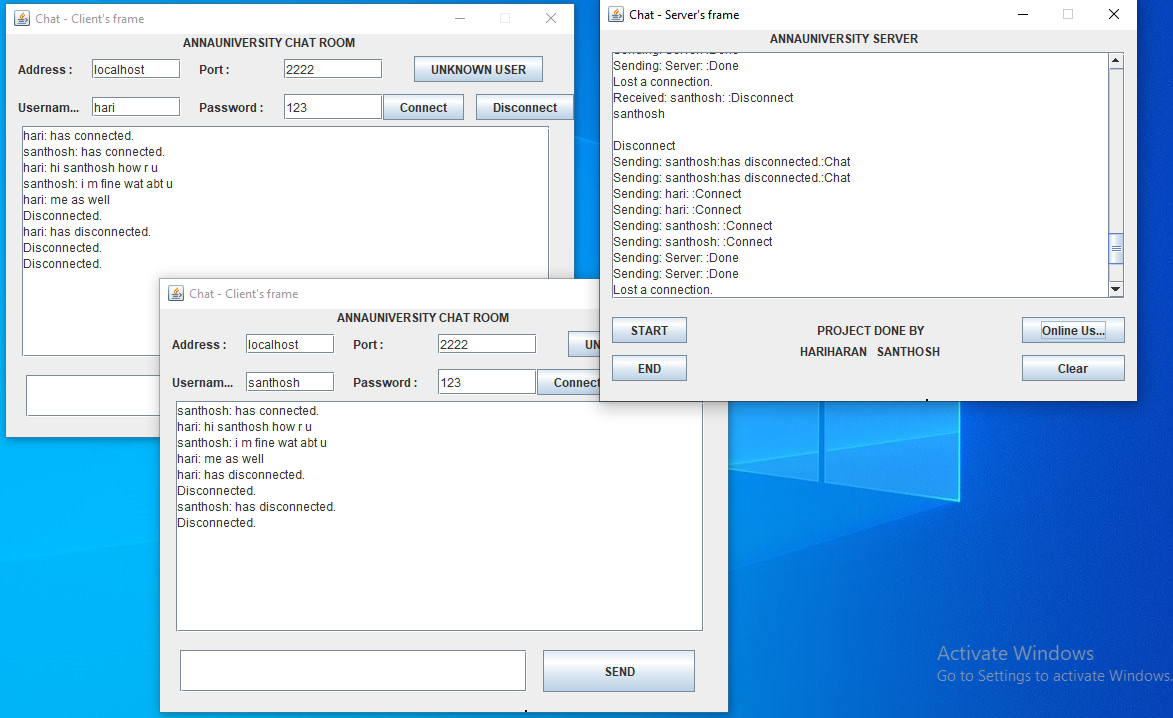
PRIVATE USER’S SERVER CONNECTION



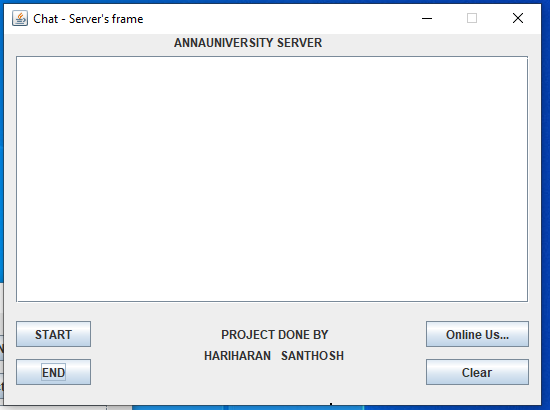
PRIVATE CHAT



TWO USER’S DISCONNECTION



DISCONNECTION OF SERVER



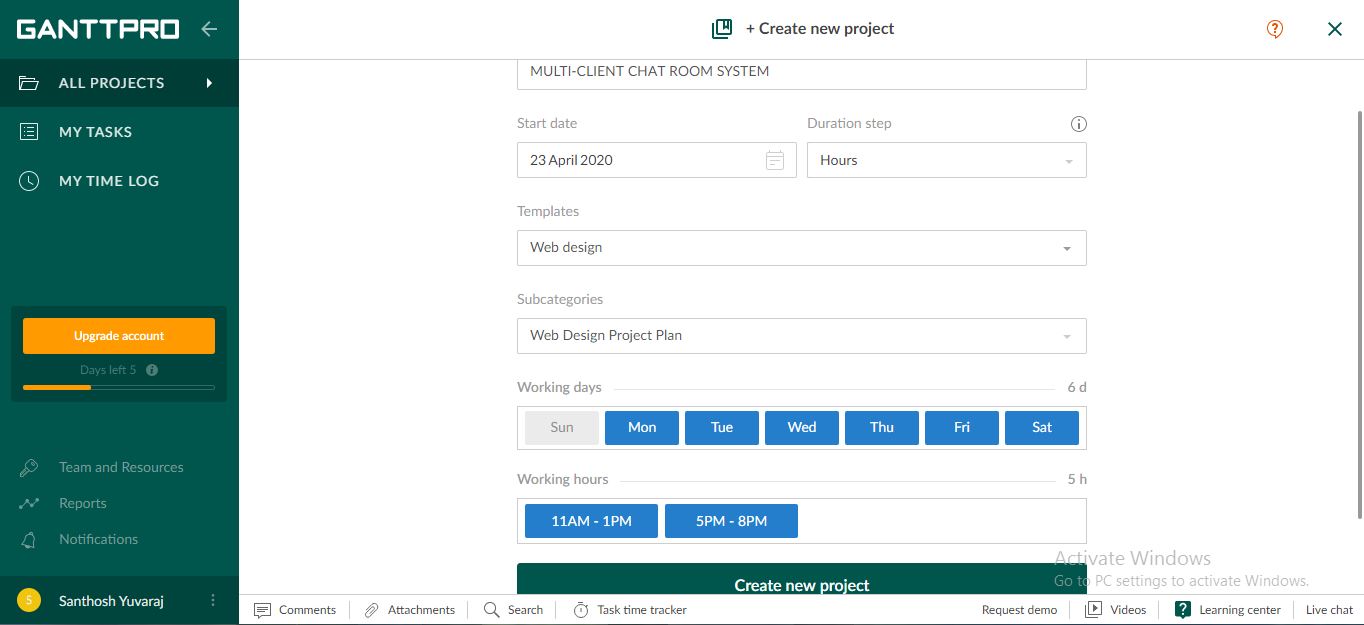


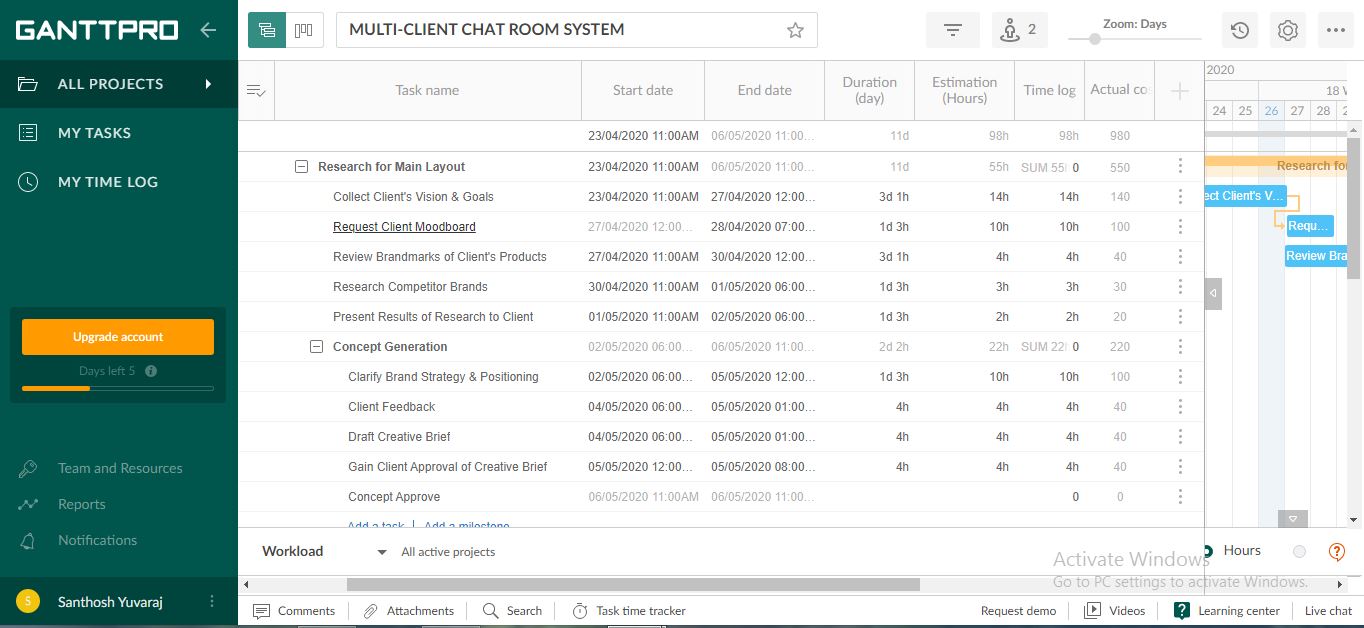
**COST ESTIMATION:**

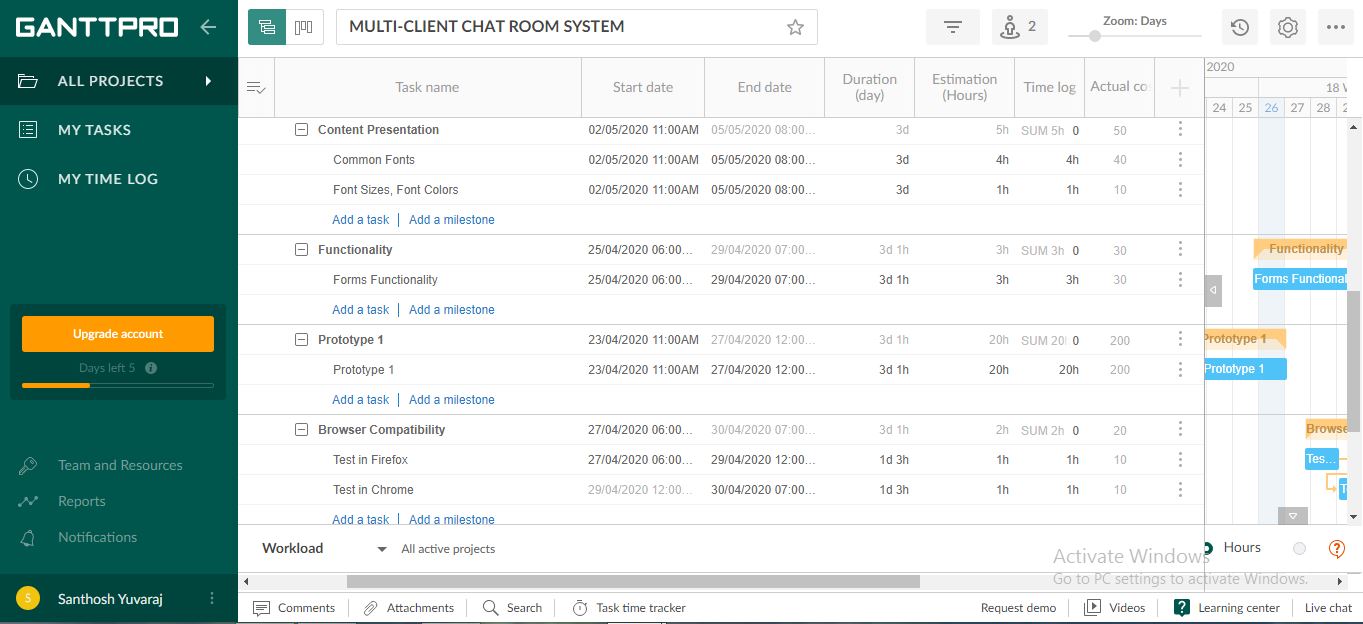
For cost estimation, we have used GANTTPRO tool. GANTTPRO has a wide range of domains for cost estimation.

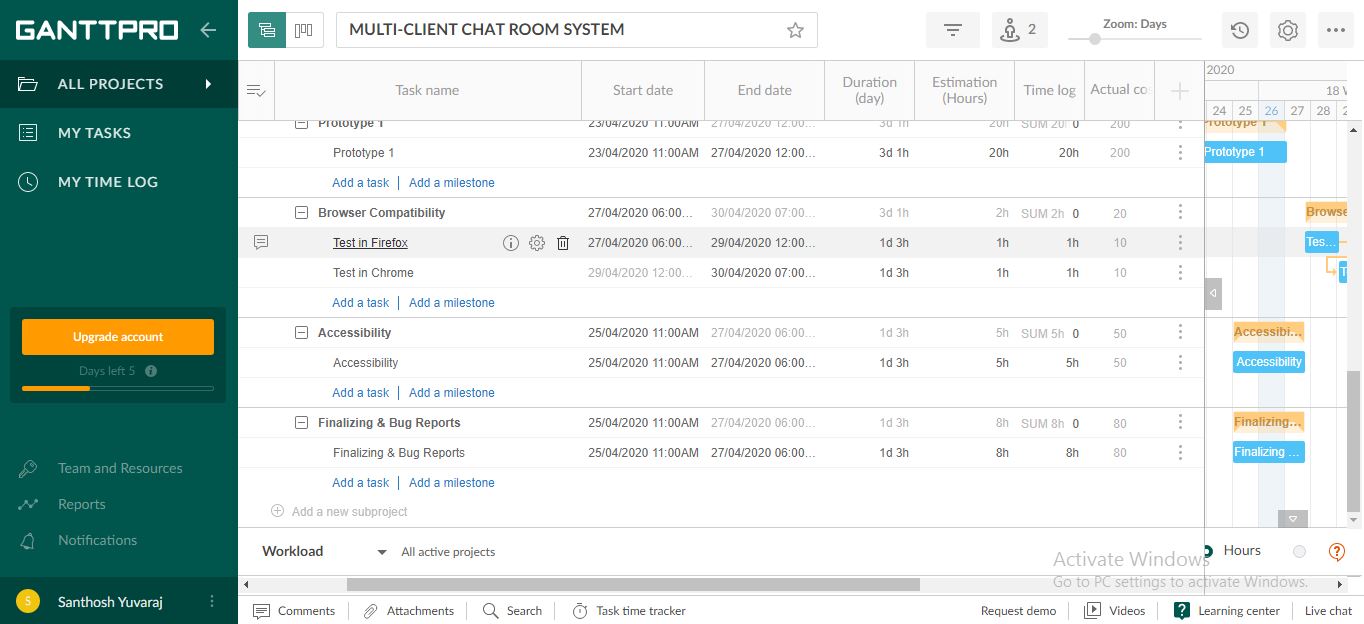


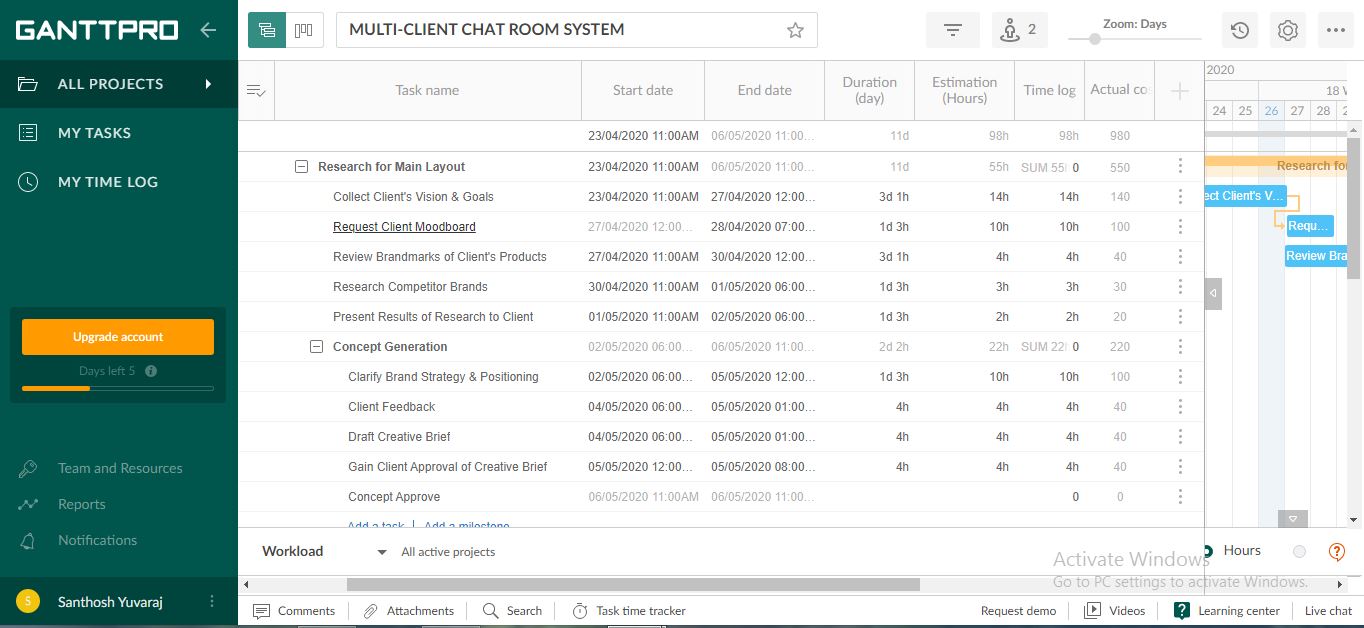
**OUR PROJECT CREATION:**











As boxed in the image, the estimated total time for the project is 98hrs, putting a cost of 10Rs per hour of work, our project amounts a total cost of 980Rs.

