ABSTARCT

LAN chat application is one of easiest way to chat with a your friends through LAN. No internet connection is needed. The only thing which requires is server IP address and you will be able to connect to others members through LAN . It can help you to talk to your friends even you both do not have internet connection. As it is based on LAN. LAN i.e. local area network which connect different client to each other and also client to main server. So we have used the same concept here we are connecting two client or client and server with each other and by providing the IP address we can talk with each other.

On the other hand file sharing application is also implemented where a user can upload a file or download a file. We are providing the path to the directory where we have stored our file to the user who wants to download particular file.

INTORDUCTION

The growth of the Internet has led to new and faster forms of communication. There are now programs that allow users to communicate in real-time with one or more people. These instant messaging tools are commonly referred to as IRC (Internet Relay Chat). Some of these programs can be downloaded for free.

LAN chat is one the same way can be used to talk to your friend in your circle .All you have to do is to provide his IP address and then you both will get connect to each other and then you can talk.It is helpful because if you have any problem while solving something you do not have to go to his room and to meet him and you can talk to him on chat if you both are connected to each other and can get solution of your problem .Chat application is one of the most useful software which is used by every business to communicate with his employers if he is out of station.

File sharing on the other hand is good tool to share those file which your friend wants from you or any other file which can be useful for other. The user can see this file after login to system and can download this file .It is one of the easiest way as it doesn’t require much knowledge for sharing and downloading.

OVEVIEW

The LAN chat application which we have implemented contain following things :-

1. LOGIN
2. CHAT
3. JOIN CHAT ROOM
4. UPLOAD FILE
5. DOWNLOAD FILE

LOGIN :

- In computer security, a login or logon is the process by which individual access to a computer system is controlled by identification of the user using credentials provided by the user.
- A user can log in to a system to obtain access and can then log out or log off (perform a logout / logoff) when the access is no longer needed.
- To log out is to close off one's access to a computer system after having previously logged in.
- The registered user needs to provide his username and password to authenticate his account.
- After successful login user will allow to chat and upload and download file .

THERE ARE TWO MODULES IN IT:-

1.ADMIN LOGIN

2.USER LOGIN

ADMIN MODULE :-

1.Add new user :: Admin can add a new user.Whether to allow any one to create a account or not depends on user to this chat application .

2.View user : Admin has got the right to check the personal details of the user .He can see all user details .

3.Delete user : Admin can also delete user who is abusing other member and who is not following the rules .

4.Update chatroom: Admin can update chat room and can also limit a chat room a fixed number if he finds that there is too much crowd in chat room .

5.Delete files: If admin finds that particular file is bad or containing virus that he can delete that file.

USER MODULE

1.Profile update : User can update the information given by him such as name and about me .he can change it as many times as he want to there is no restriction in it.

2.Chat room : User can join any chat room he wish to join and there is no restriction in it.

3.File upload : User can upload file from the client Machine to server machine.

4.File download User can download file from the server Machine to client machine.

CHAT

After logged in user is allowed to chat with each other in a group or in private. This help user to get any information or solution of their problem and also to share his knowledge with others.

You can also connect with your friends only if you want to talk to him privately .

Same is in case of file sharing a user can download and upload file easily by just clicking the file .This is very easy process and any one can easily upload and download file.

JOIN CHAT ROOM

In this you can connect to particular room which is currently running or have at least 1 user in it .This type of room are made to share particular information. You can join any chat room if room is available and can talk to other people who are logged in right now .

UPLOAD FILE

This will help you to upload a particular file from your computer to a server which other can easily download.To upload a file a user must be logged in only then he can upload a file .A file should not be bad other wise admin has the right to delete it .

DOWNLOAD FILE

In this a user can download a particular file he wants to. For this a user just have to log in and then go to path where all files are available and then he has to click on that file and it will prompt a message to save a file to particular location and in this way you can download and save a file.

TECHNOLOGIES USED

Front end as: Eclipse helios

Back end as: Mysql

Database: Mysql

Querying language: Sql

Forms: Jsp , Java servlet

External library : Apache common.fileupload , Apache.commons.io

REQUIREMENTS

System Requirements:

Operating System: Microsoft® Windows® XP/Vista/window7/linux

Processor: 1 Ghz

Memory: 512 MB RAM

Hard Disk Space: 45 MB Available HDD Space

Video Card: 3D graphics accelerator equivalent to GF6200 or higher

Sound Card: 16-bit Sound Card

DirectX® Version: DirectX® 9.0c

TABLES USED :

LOGIN

loginid VARCHAR2(30)	name VARCHAR2(30)	password VARCHAR2(30)	email VARCHAR2(30)	type VARCHAR2(30)

CHAT ROOMS:

roomname VARCHAR2(30)	roomdesc VARCHAR2(30)

SHARED FILES:

fileName VARCHAR2(30)	contenttype VARCHAR(30)	loginid VARCHAR2(30)

CODE:

Login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-
8859-1"
     pageEncoding="ISO-8859-1"
import="java.util.*,java.text.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
<title>Insert title here</title>
</head>
<body>
<table width=700>
     <tr>
          <td width=200><%@ include file="menu.jsp"%></td>

          <td>&nbsp;&nbsp;&nbsp;
          <div><font color=000033
size=20>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <%
if("admin".equals(request.getParameter("type")))
{
     out.print("Admin Console</font>");
}
else
{
     out.print("User Console</font>");
}
DateFormat df=new SimpleDateFormat("EEEE, dd MMMM, yyyy ");
String date=df.format(new Date());
out.print("<br>");
for(int i=0;i<37;i++)
     out.print("&nbsp;");
out.print("<font size=3 color=3300FF>"+date+"</font>");
%> </font></div>
          </td>
     </tr>
     <tr>
          <td></td>
          <td>
          <form action=loginhandler.jsp method=post>
          <table align=center cellpadding=5>
               <tr>
                    <td>UserName</td>
                    <td><input type=text name=userName></td>
               </tr>
               <tr>
                    <td>Password</td>
                    <td><input type=password name=password></td>
               </tr>
               <tr>
                    <td>Type</td>
                    <td><label><%= request.getParameter("type")
%></label></td>
               </tr>
               <tr>
                    <td><input type=hidden name=type
                         value=<%= request.getParameter("type")
%>></td>
               </tr>
               <tr>
                    <td></td>
                    <td align=center><input type=submit
value=Submit></td>
               </tr>
          </table>
          </form>
          </td>
     </tr>

</table>
</body>
</html>
```

Adminchatservlet

```
package chat;


import java.io.*;

import javax.servlet.*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.*;

import java.sql.*;


/**

 * Servlet implementation class for Servlet: AdminChatServlet

 *

 */

 public class AdminChatServlet extends javax.servlet.http.HttpServlet implements
javax.servlet.Servlet {

   static final long serialVersionUID = 1L;


    /* (non-Java-doc)

             * @see javax.servlet.http.HttpServlet#HttpServlet()

             */

            public AdminChatServlet() {

                        super();

            }


            /* (non-Java-doc)
```

```
	 * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doPost(request,response);
	}


	/* (non-Java-doc)
	 * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub


		//adding rooms
		String roomname=request.getParameter("roomname");
		String roomdesc=request.getParameter("roomdesc");


		if(roomname!=null&&roomname.length()>0)
		{	try
			{
				Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
				Connection con=DriverManager.getConnection("jdbc:odbc:abc");
				String query="insert into chatrooms values(?,?)";
				PreparedStatement stmt=con.prepareStatement(query);
				stmt.setString(1, roomname);
```

```
                stmt.setString(2, roomdesc);
                stmt.execute();
        }catch(Exception e)
        {
                System.out.print(e);
        }
}
//removing rooms
String[] remove=request.getParameterValues("remove");
if(remove!=null)
{
        try
        {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                Connection con=DriverManager.getConnection("jdbc:odbc:abc");
                String query="delete from chatrooms where roomname=?";
                PreparedStatement stmt=con.prepareStatement(query);
                for(int i=0;i<remove.length;i++)
                {
                        stmt.setString(1, remove[i]);
                        stmt.execute();
                }
        }catch(Exception e)
        {
                System.out.print(e);
        }
}
```

```
                RequestDispatcher rd=request.getRequestDispatcher("adminchatservlet.jsp");
                rd.forward(request,response);
        }
}
```

Chatroomservlet

```
package chat;


import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.text.*;
import java.util.*;
import java.sql.*;


/**
 * Servlet implementation class for Servlet: ChRoomServlet
 *
 */
 public class ChRoomServlet extends javax.servlet.http.HttpServlet implements
javax.servlet.Servlet {
   static final long serialVersionUID = 1L;


    /* (non-Java-doc)
         * @see javax.servlet.http.HttpServlet#HttpServlet()
         */
   public ChRoomServlet() {
                    super();
          }
          /* (non-Java-doc)
           * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
           */
          protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                    // TODO Auto-generated method stub
                    response.setContentType("text/html");
                    PrintWriter out=response.getWriter();
                    ChatRooms chatroom=getRoom(request,response);
                    if(chatroom==null)
                    {
                              return;
                    }
                    String s=request.getParameter("list");
                    if(s!=null&s.equals("true"))
                              writeMessage(out,chatroom,getProfileName(request));
                    else
                    {
                              out.println("<html><head>");
                              out.print("<script language=javascript>");
                 out.print("function fun1(e){");


                 out.print("if(e.keyCode==13)");
                 out.print("document.form1.submit();}");
```

```
       out.print("</script></head>");


            out.println("<body>");
            out.println("<form method=post action=ChRoomServlet
target=_parent>");
            out.println("<table align=center width=100%>");
            out.println("<tr><td><font color=#476BC0>Your
message</font></td></tr>");
            out.println("<tr><td><textarea name=msg cols=50 rows=3
class=smalltext></textarea></td></tr>");
            out.println("<tr><td><input type=submit value=Send class=smalltext
onkeydown=fun1(event) ></td></tr>");
            out.println("</table></form>");
            out.println("<form action="+response.encodeURL("/chat/roomlist.jsp")+"
target=_parent>");
            out.println("<center><input type=submit value='Exit from Chat Room'
class=smalltext>");
            out.println("</center></form>");
            out.println("<form action=logout.jsp method=get target=_parent>");
            out.println("<center><input type=submit value='Logout'
class=smalltext>");
            out.println("</center></form>");
            out.println("</body></html>");
        }
        out.close();
    }


    /* (non-Java-doc)
```

```
     * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.setContentType("text/html");
        ChatRooms chatroom=getRoom(request,response);
        if(chatroom==null)
        {
            return;
        }

        String s=getProfileName(request);
        String s1=request.getParameter("msg");

        if(s1!=null&&s1.length()!=0)
        {
            DateFormat d=new SimpleDateFormat("hh:mm");
            String time=d.format(new java.util.Date());
            s1="[ "+time+" ] "+s1;
            chatroom.joinChatEntry(new ChatRoomEntry(s,s1));
        }

        messageFrame(response,chatroom);
    }

    private String getProfileName(HttpServletRequest request)
```

```
{
        HttpSession httpsession=request.getSession(true);
        String s=(String)httpsession.getAttribute("profileName");

        if(s==null)
        {
                s=request.getParameter("profileName");
                if(s==null||s.length()==0)
                {
                        s="Anonymous";
                }
                httpsession.setAttribute("profileName", s);
        }
        else
        {
                String s1=request.getParameter("profileName");
                if(s1!=null&&s1.length()>0&& !s.equals(s1))
                {
                        s=s1;
                        httpsession.setAttribute("profileName", s);
                }
        }

        return s;
}
```

```
	private ChatRooms getRoom(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException

	{

		HttpSession httpsession=request.getSession(true);

		PrintWriter out=response.getWriter();

		String s=(String)httpsession.getAttribute("roomName");


		if(s==null)

		{

			s=request.getParameter("roomName");

			if(s==null||s.length()==0)

			{

				error(request,response);

			}

			httpsession.setAttribute("roomName", s);

		}

		else

		{

			String s1=request.getParameter("roomName");

			if(s1!=null&&s1.length()>0&& !s.equals(s1))

			{

				s=s1;

				httpsession.setAttribute("roomName", s);

			}

		}


		HashMap hashmap=(HashMap)getServletContext().getAttribute("roomList");
```

```
                ChatRooms chatroom=(ChatRooms)hashmap.get(s);


                if(chatroom==null)
                {
                        error(request,response);
                        return null;
                }
                else
                {
                        return chatroom;
                }
        }


        private void error(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
        {


        }


        private void messageFrame(HttpServletResponse response,ChatRooms chatroom) throws
ServletException, IOException
        {
                PrintWriter out=response.getWriter();
                out.println("<html>");
                out.println("<head><title>"+chatroom.getName()+"</title></head>");
                out.println("<frameset rows='320,180'>");
                out.println("<frame src=ChRoomServlet?list=true name=list
SCROLLING=AUTO>");
```

```
                out.println("<frame src=ChRoomServlet?list=false name=form
SCROLLING=no>");


                out.println("</frameset>");
                out.println("</html>");
        }


        private void writeMessage(PrintWriter out,ChatRooms chatroom,String s)
        {
                StringBuffer stringbuffer=new StringBuffer();
                out.println("<html>");
                out.println("<head><meta http-equiv=refresh content=1></head>");


                out.println("<body marginheight=0 marginwidth=0>");
                out.println("<table align=center bgcolor=#476BC0 width=100%
height=100%>");
                out.println("<tr><td valign=top>");
                out.println("Chat-Room Name: "+chatroom.getName()+"<br>You are:
"+s+"<br><br><br>");


                if(chatroom.size()==0)
                {
                        out.print("<font color=red>No messages available in this room</font>");
                }
                else
                {
                        for(Iterator iterator=chatroom.iterator();iterator.hasNext();)
                        {
```

```
                ChatRoomEntry chatentry=(ChatRoomEntry)iterator.next();
                if(chatentry!=null)
                {
                    String s1=chatentry.getProfileName();

                    if(s1.equals(s))
                        out.println("<font color=yellow>");
                    out.println(s1+" : "+chatentry.getMsg()+"<br>");

                    if(s1.equals(s))
                        out.println("</font>");

                }
            }
        }
        out.println("</td></tr></table>");
        out.print("</body></html>");
    }

    public void destroy()
    {
        System.out.print("chroomservlet");
    }

}
```

UPLOAD SERVLET

```
package chat;


import java.io.*;
import java.util.*;


import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.*;


import org.apache.commons.fileupload.*;
import org.apache.commons.fileupload.disk.*;
import org.apache.commons.fileupload.servlet.*;
import org.apache.commons.io.*;
import java.sql.*;



public class UploadServlet extends HttpServlet {


  //private boolean isMultipart;


  private int maxMemorySize = 50 * 1024;
  private int maxRequestSize = 300 * 1024;
  private File tempDirectory;


  public void init( ){
    }
```

```
public void doPost(HttpServletRequest request,HttpServletResponse response)
        throws ServletException, java.io.IOException {
         //check for file upload request
        // boolean isMultipart=ServletFileUpload.isMultipartContent(request);
         //response.getWriter().print(isMultipart);



         // Create a factory for disk-based file items
         DiskFileItemFactory factory = new DiskFileItemFactory();

         // Set factory constraints
         factory.setSizeThreshold(maxMemorySize);
         factory.setRepository(tempDirectory);

         // Create a new file upload handler
         ServletFileUpload upload = new ServletFileUpload(factory);

         // Set overall request size constraint
         upload.setSizeMax(maxRequestSize);

         // Parse the request
         try
         {
                 List items = upload.parseRequest(request);

         // Process the uploaded items
         Iterator iter = items.iterator();
```

```
while (iter.hasNext()) {
   FileItem item = (FileItem) iter.next();

   if (item.isFormField()) {

   } else {
                         //response.getWriter().println("a file is uploaded");
         String fieldName = item.getFieldName();
         String fileName=item.getName();
         //response.getWriter().println(fileName.lastIndexOf('/')+1);
        // fileName = fileName.substring(fileName.lastIndexOf('/')+1);
         fileName = fileName.substring(fileName.lastIndexOf('\\')+1);

          String contentType = item.getContentType();
          boolean isInMemory = item.isInMemory();
          long sizeInBytes = item.getSize();

          InputStream uploadedStream = item.getInputStream();

          item.write(new File("D:\\PROJECT\\chat\\WebContent\\"+fileName));
          uploadedStream.close();

          int x=contentType.indexOf('/');
          contentType=contentType.substring(0, x);

          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                Connection con=DriverManager.getConnection("jdbc:odbc:abc");
```

```
                    String query="insert into sharedfiles values(?,?,?)";
                    PreparedStatement pstmt=con.prepareStatement(query);

                    HttpSession httpsession=request.getSession();
                    String loginid=(String)httpsession.getAttribute("login");

                    pstmt.setString(1,fileName);
                    pstmt.setString(2,contentType);
                    pstmt.setString(3,loginid);

                    pstmt.execute();




              }
            }
            }catch(Exception e)
            {
                response.getWriter().println(e);
            }
            RequestDispatcher view=request.getRequestDispatcher("viewfiles.jsp");
               view.forward(request,response);

    }
    public void doGet(HttpServletRequest request,HttpServletResponse response)
      throws ServletException, java.io.IOException {
            doPost(request,response);


    }

}
```

IMAGE DOWNLOADER

```
package chat;


import java.io.BufferedInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.URL;

import java.net.URLConnection;




import javax.servlet.*;

import javax.servlet.ServletOutputStream;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


/**

 * Servlet implementation class for Servlet: imageDownloader2

 *

 */

 public class imageDownloader2 extends javax.servlet.http.HttpServlet implements
javax.servlet.Servlet {

   static final long serialVersionUID = 1L;


    /* (non-Java-doc)

             * @see javax.servlet.http.HttpServlet#HttpServlet()

             */

            public imageDownloader2() {
```

```
		super();
	}


	/* (non-Java-doc)
	 * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request, 
HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws 
ServletException, IOException {
		// TODO Auto-generated method stub
	}


	/* (non-Java-doc)
	 * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request, 
HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) 
throws ServletException, IOException {
		// TODO Auto-generated method stub
		try
		{
			String filename2=request.getParameter("filename");
			URL u=new URL("http://localhost:8080/chat/"+filename2);
			//URL u=new URL("http://www.aitpune.com/directorpage.asp");

			URLConnection uc=u.openConnection();
			String contentType=uc.getContentType();
			int contentLength=uc.getContentLength();
			System.out.print(contentLength);
```

```
            System.out.print(contentType);

            InputStream raw=uc.getInputStream();

            InputStream buffer=new BufferedInputStream(raw);

            String filename=u.getFile();

            filename=filename.substring(filename.lastIndexOf('/')+1);

            ServletOutputStream stream = null;

            stream = response.getOutputStream();

            response.setContentType(contentType);


          response.addHeader("Content-Disposition","attachment; filename="+filename
);


          response.setContentLength( contentLength );

          int readBytes = 0;


          while((readBytes = buffer.read()) != -1)

               stream.write(readBytes);

          RequestDispatcher view=request.getRequestDispatcher("viewfiles.jsp");

                view.forward(request,response);

        }catch(Exception e){

            response.getWriter().print(e);

        }

    }

}
```

DELETE USER:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" import="java.sql.*"%>
<%
String[] users=request.getParameterValues("loginid");
try
{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con=DriverManager.getConnection("jdbc:odbc:abc");
        String query="delete from login where loginid=?";
        PreparedStatement stmt=con.prepareStatement(query);
        for(int i=0;i<users.length;i++)
        {
                stmt.setString(1,users[i]);
                stmt.execute();
        }
        RequestDispatcher rd=request.getRequestDispatcher("viewuser.jsp");
        rd.forward(request,response);

}catch(Exception e)
{
        out.print(e);
}
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

</body>
</html>
```

ADD USER :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
          pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Java Chat-Admin Console-Add User</title>
</head>
<body>
<table width=700>
          <tr>
                    <td width=300><%@ include file="menu.jsp"%></td>
                    <td>
                    <div align=center>
                    <table>
                              <tr>
                                        <td align=center height=30 width=200 bgcolor=red><a
                                                  href=adduser.jsp>Add Users</a></td>
                                        <td align=center height=30 width=200 bgcolor=CCCCFF><a
                                                  href=viewuser.jsp>View Users</a></td>
                                        <td align=center height=30 width=200 bgcolor=CCFFFF><a
                                                  href=AdminChatServlet>Configure Rooms</a></td>
                                        <td align=center height=30 width=200 bgcolor=9966FF><a
                                                  href=logout.jsp>Logout</a></td>
                              </tr>
                    </table>
                    <br>
                    <br>
                    <font size=5 color=000033>Enter New User Information</font></div>
          </tr>

          <tr>
                    <td></td>
                    <td>
                    <form action=adduserhandler.jsp method=post>
                    <table align=center cellpadding=5>
                              <tr>
                                        <td>Name</td>
                                        <td><input type=text name=name></td>
                              </tr>
                              <tr>
                                        <td>Email</td>
                                        <td><input type=text name=email></td>
```

```
                        </tr>
                        <tr>
                                <td>Loginid</td>
                                <td><input type=text name=loginid></td>
                        </tr>
                        <tr>
                                <td>Password</td>
                                <td><input type=password name=password></td>
                        </tr>
                        <tr>
                                <td>Type</td>
                                <td><select name=type>
                                        <option value=admin>admin</option>
                                        <option value=user>user</option>
                                </select></td>
                        <tr>
                                <td></td>
                                <td align=center><input type=submit value=Submit></td>
                        </tr>
                </table>
                </form>
                </td>
        </tr>
</table>
</body>
</html>
```

CHAT CLIENT :

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

public class ChatClient extends Frame implements ActionListener,Runnable {

        private Button button1, button2;
   private TextArea textarea1, textarea2;
   private TextField textfield1;
   private Label label1, label2, label3;
   Socket socket;
   Thread thread;

   InputStream in;
   OutputStream out;
  // PrintWriter out;
```

```
public static void main(String[] args)
{
   new ChatClient();
}

public ChatClient()
{
   setLayout(null);

   label1 = new Label("Server IP address:");
   label1.setBounds(35, 80, 105, 20);
   add(label1);

   textfield1 = new TextField("127.0.0.1");
   textfield1.setBounds(145, 80, 100, 20);
   add(textfield1);

   button1 = new Button("Connect");
   button1.setBounds(255, 80, 80, 20);
   add(button1);
   button1.addActionListener(this);

   button2 = new Button("Send");
   button2.setBounds(160, 390, 60, 20);
   add(button2);
   button2.addActionListener(this);

   textarea1 = new TextArea("", 7, 45, TextArea.SCROLLBARS_VERTICAL_ONLY);
   textarea1.setBounds(20, 110, 340, 120);
   add(textarea1);

   label2 = new Label();
   label2.setBounds(20, 240, 100, 20);
   label2.setText("Type here:");
   add(label2);

   textarea2 = new TextArea("", 7, 45,
      TextArea.SCROLLBARS_VERTICAL_ONLY);
   textarea2.setBounds(20, 260, 340, 120);
   textarea2.setForeground(Color.RED);
   add(textarea2);

   label3 = new Label("Chat Client");
   label3.setFont(new Font("Times New Roman", Font.BOLD, 36));
   label3.setBounds(100, 35, 200, 30);
```

```
        add(label3);

        setSize(400, 430);

        setTitle("Chat Client");
        setVisible(true);
        //textarea2.requestFocus();

        this.addWindowListener(new WindowAdapter(){
            public void windowClosing(
                WindowEvent e){
                    System.exit(0);
                }
            }
        );


    }
    public void actionPerformed(ActionEvent event)
    {
            if(event.getSource()==button1)
            {
                    String hostname=textfield1.getText();
                    try
                    {
                            socket = new Socket(hostname, 13);

                            in=socket.getInputStream();
                            out=socket.getOutputStream();

                            textarea1.setText("you are now connected");

                            thread=new Thread(this);
                            thread.start();



                    }catch (UnknownHostException e)
                    {
                            textarea1.setText(e.getMessage());
                    }catch (IOException e)
                    {
                            textarea1.setText(e.getMessage());
                    }
            }

                    if(event.getSource()==button2)
                    {
                            try
                            {
                                String str = textarea2.getText() + "\n";
                    byte buffer[] = str.getBytes();
                    out.write(buffer);
                    textarea1.setForeground(Color.RED);
                    textarea1.append(str+"\n");
                    textarea2.setText("");
                    textarea2.requestFocus();
                    }catch(Exception e){}
                    }

    }

            public void run()
            {
                    String instring;
        try {

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

            textarea1.setForeground(Color.BLUE);
            while((instring = in.readLine()) != null){
                textarea1.append(instring + "\n");
            }

        }catch (Exception e)
        {
            textarea1.setText(e.getMessage());
        }

            }


        }
```

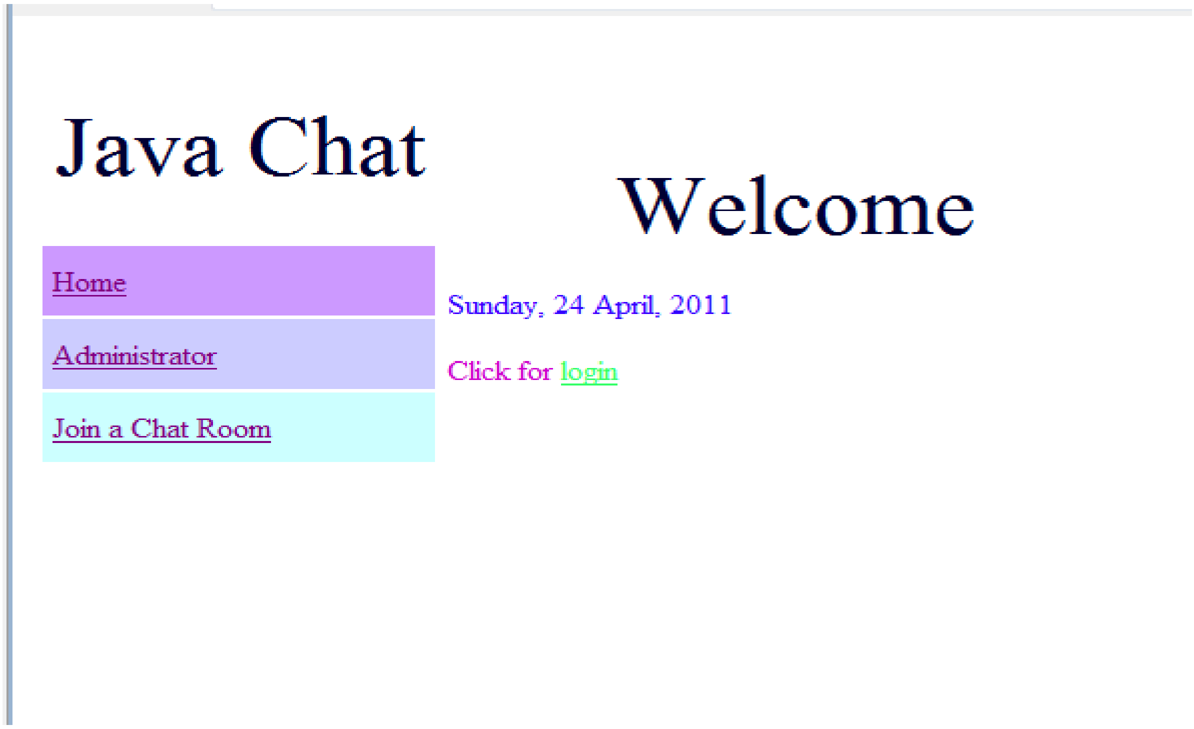
CHAT SERVER :

```
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.util.Date;
import java.io.*;

public class ChatServer2 extends Frame implements ActionListener,Runnable  {

        private Label label1,label2;
        private Button button1;
        private TextArea textarea1,textarea2;
        ServerSocket socket;
        Socket insocket;
        int port=13;
        Thread thread;

        ServerSocket server;
        Socket connection=null;

        InputStream in;
        OutputStream out;


        public static void main(String[] arg)
        {
                new ChatServer2();
        }
        public ChatServer2()
        {
                setLayout(null);

                label2=new Label("Chat Server");
                label2.setBounds(100,35,200,30);
                label2.setFont(new Font("Times New Roman",Font.BOLD,36));
      add(label2);

                setSize(400,400);
      setVisible(true);
      setTitle("Chat Server");

      button1 = new Button("Send");
      button1.setBounds(160, 360, 60, 20);
      add(button1);
      button1.addActionListener(this);
```

```
    textarea1 = new TextArea("", 7, 45, TextArea.SCROLLBARS_VERTICAL_ONLY);
    textarea1.setBounds(20, 80, 340, 100);
    //textarea1.setText("hi");
    add(textarea1);

    label1=new Label();
    label1.setBounds(20,210,100,20);
    label1.setText("Type Here");
    add(label1);

    textarea2 = new TextArea("", 7, 45, TextArea.SCROLLBARS_VERTICAL_ONLY);
    textarea2.setBounds(20, 230, 340, 120);
    textarea2.setForeground(Color.RED);
    add(textarea2);

    this.addWindowListener(new WindowAdapter()
    {
      public void windowClosing(WindowEvent e)
      {
            System.exit(0);
            try
            {
                  socket.close();
            }catch(Exception ex){}
      }
    });

    try
      {
            socket = new ServerSocket(port);
            insocket = null;

                  try {
                        insocket = socket.accept( );
                        //OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream( ));
                        in=insocket.getInputStream();
                        out=insocket.getOutputStream();

                        thread=new Thread(this);
                        thread.start();


                  }catch (IOException e) {}

      }catch (IOException e)
```

```
            {
                        System.err.println(e);
            }
            }

            public void actionPerformed(ActionEvent event)
    {
                        if(event.getSource()==button1)
                        {
                                    try
                                    {
                                                String str = textarea2.getText() + "\n";
                        byte buffer[] = str.getBytes();
                        out.write(buffer);
                        textarea1.setForeground(Color.RED);
                        textarea1.append(str+"\n");
                        textarea2.setText("");
                        textarea2.requestFocus();
                                    }catch(Exception e){}
                        }

    }

            public void run()
            {
                        String instring;
        try {

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

            textarea1.setForeground(Color.BLUE);

            while((instring = in.readLine()) != null){
               textarea1.append(instring + "\n");
            }

        }catch (Exception e)
        {
            textarea1.setText(e.getMessage());
        }

            }
}
```

SNAPSHOTS:

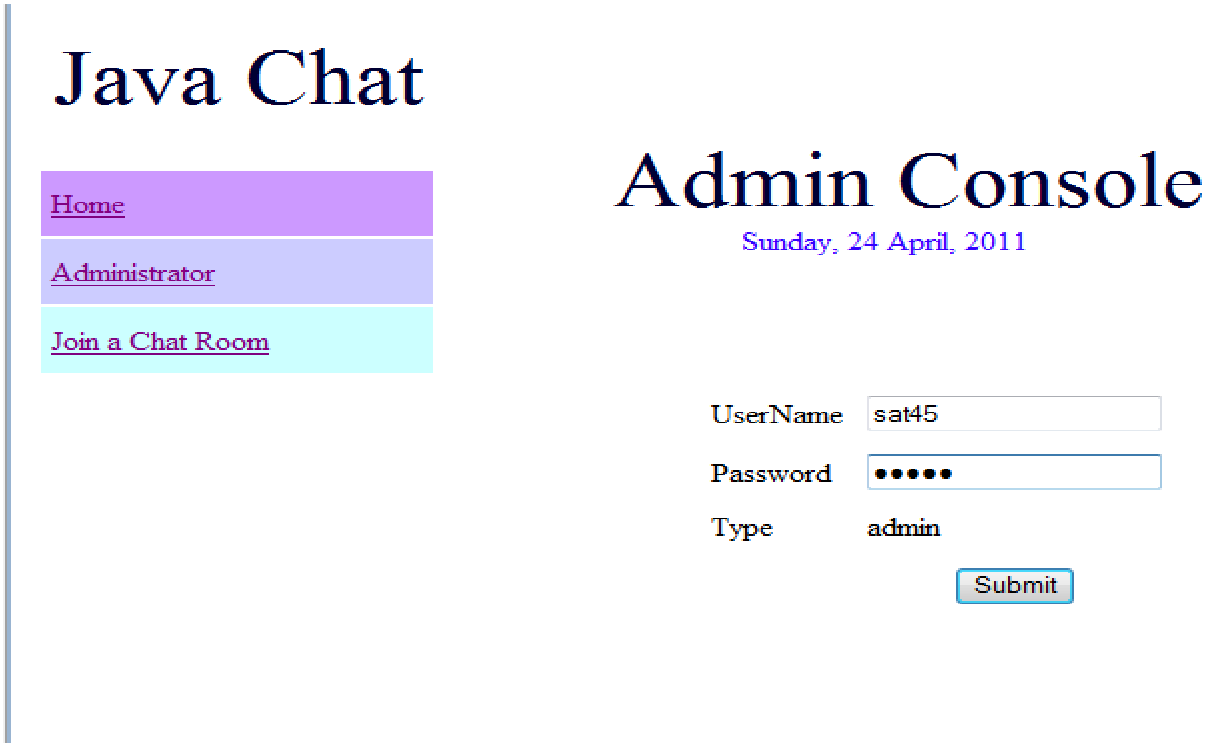


Java Chat

Welcome

Sunday, 24 April, 2011

Click for login



Java Chat

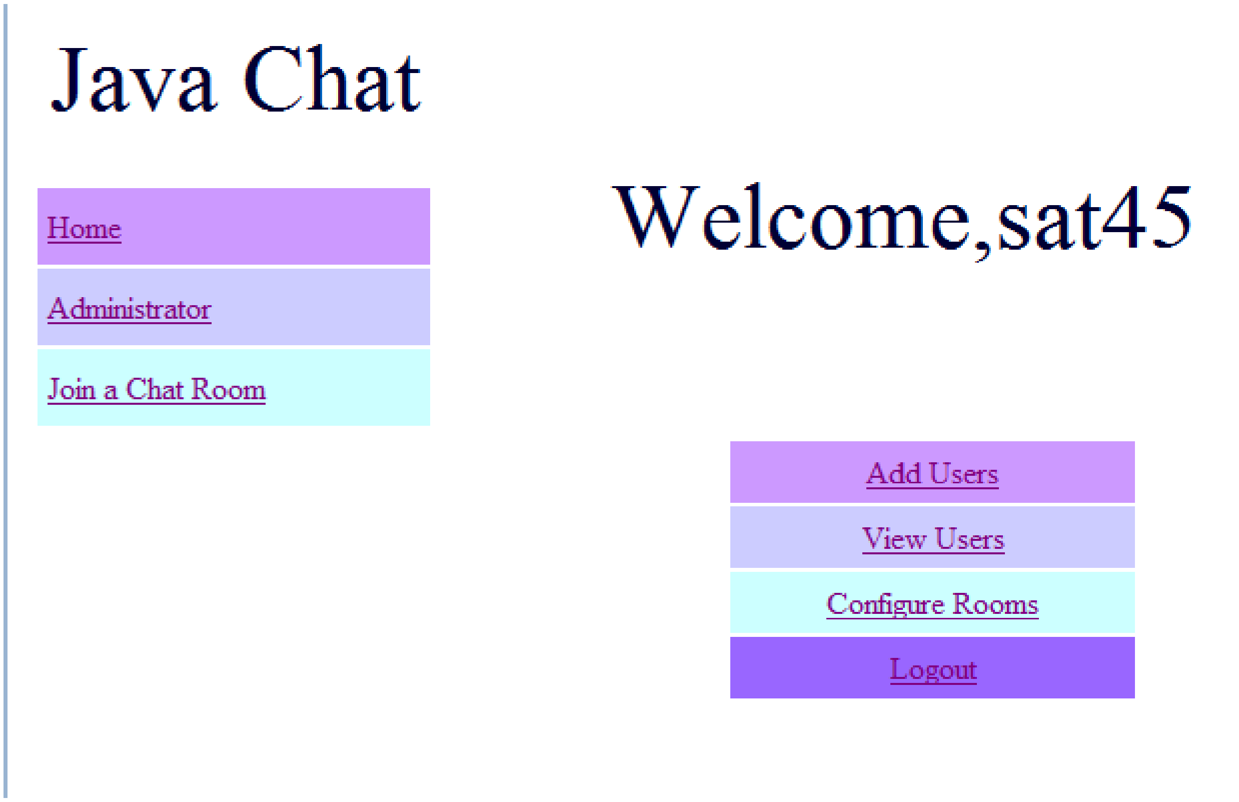
Admin Console

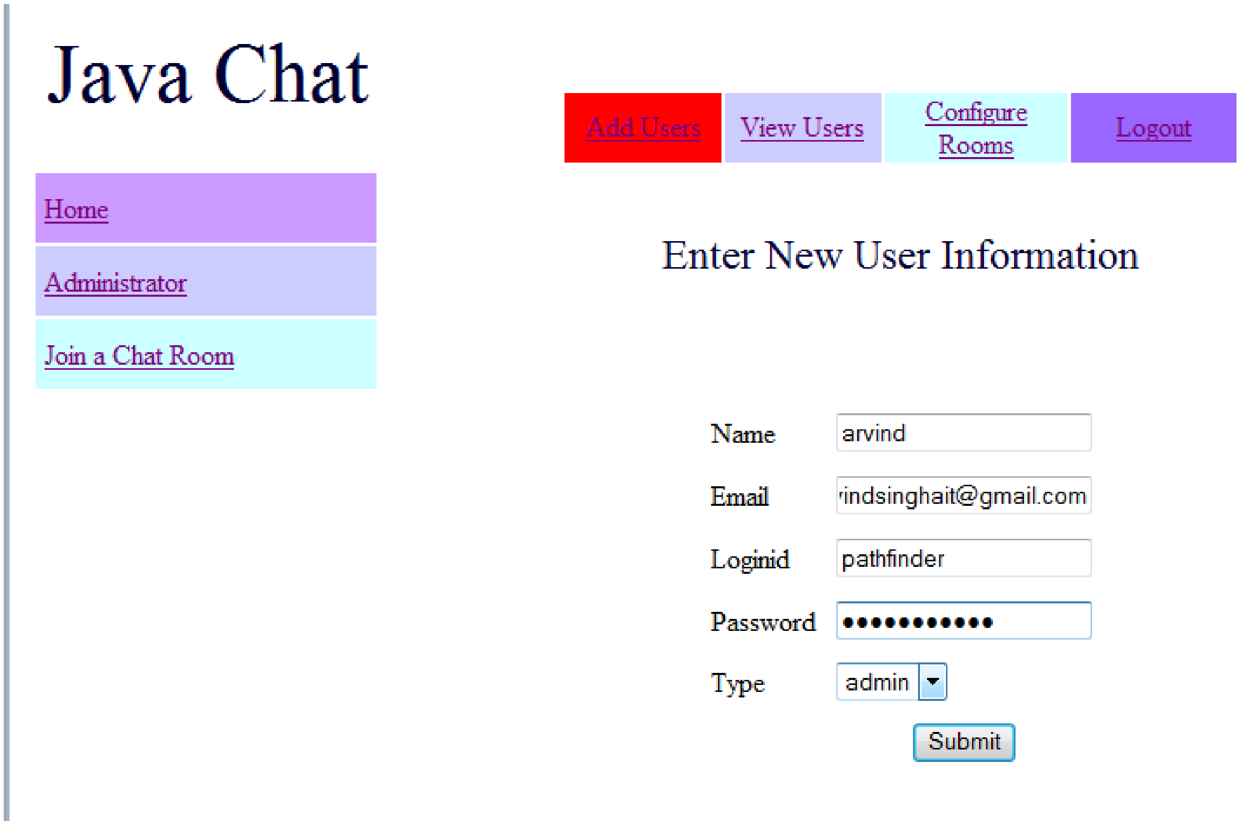
Sunday, 24 April, 2011

UserName sat45

Password •••••

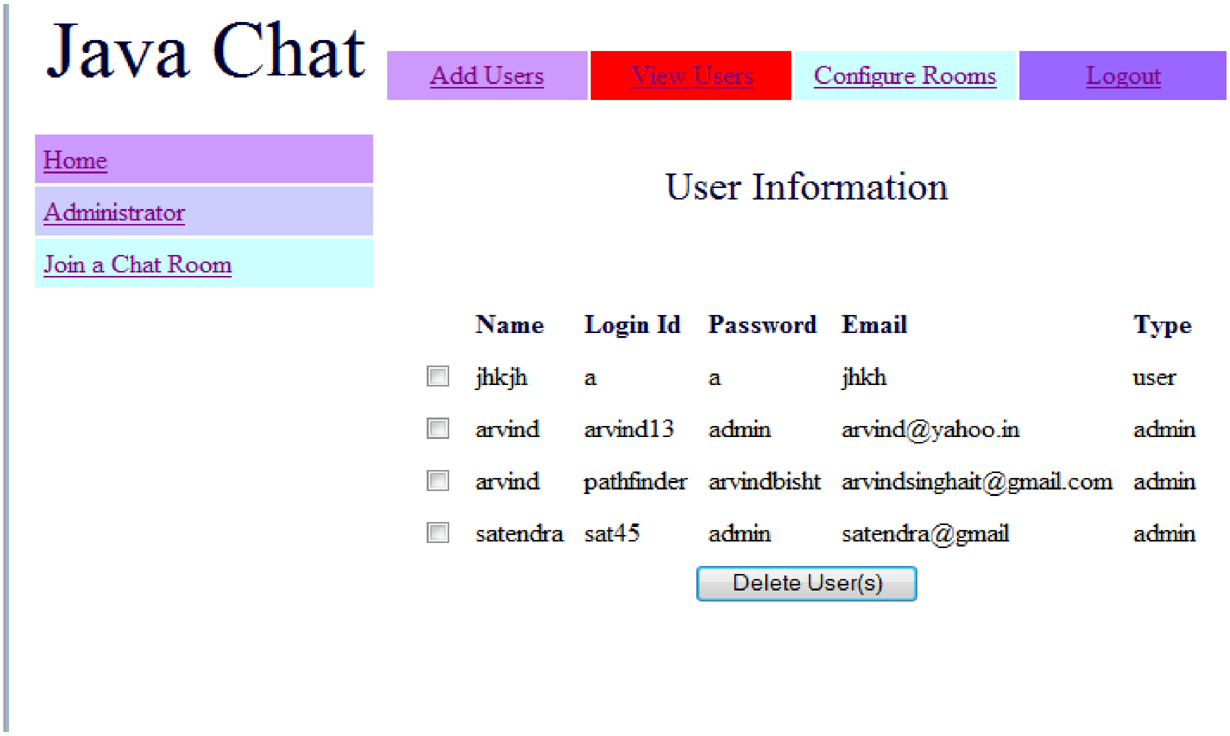
Type admin





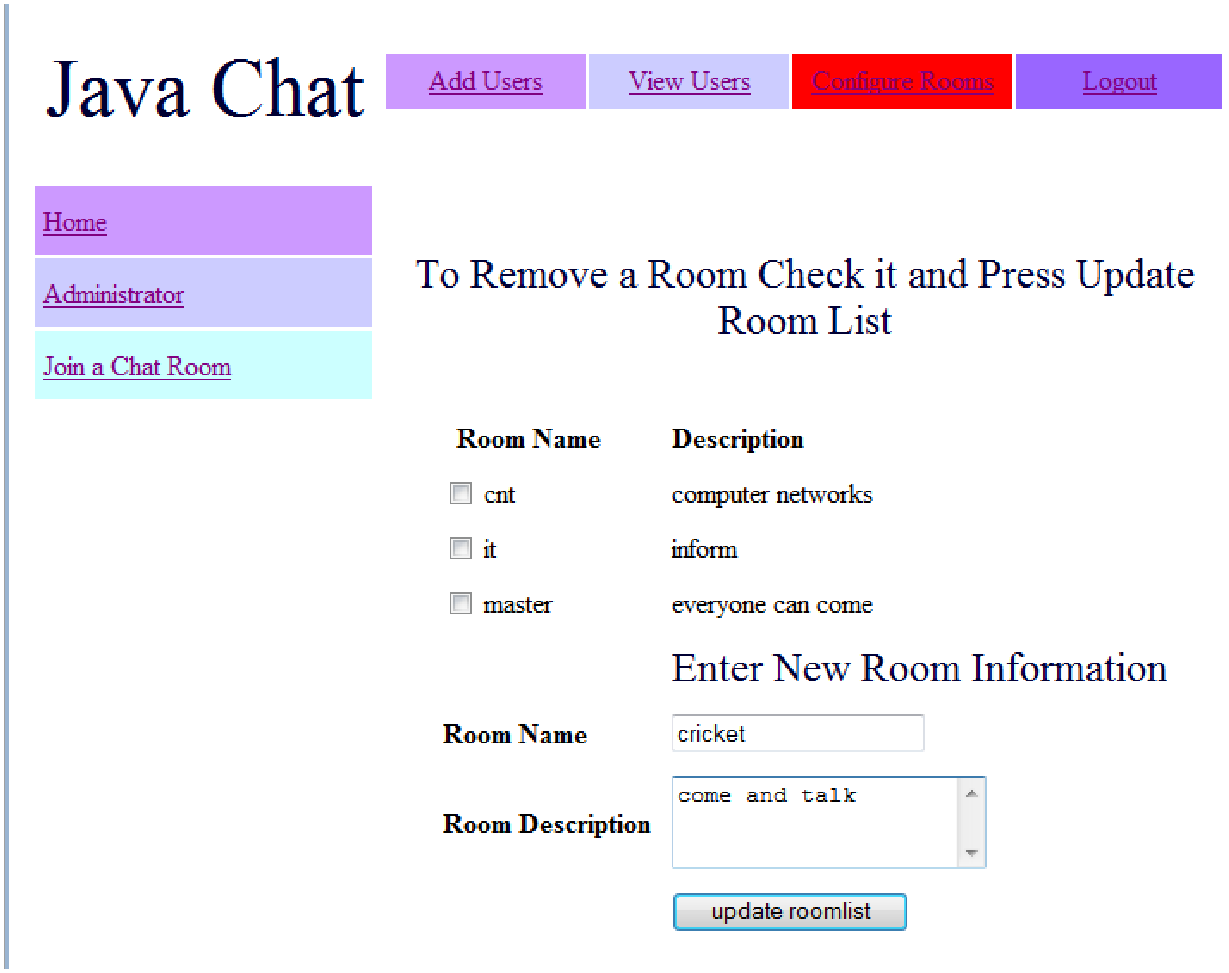
Enter New User Information

Name	arvind
Email	indsinghait@gmail.com
Loginid	pathfinder
Password	●●●●●●●●●●●
Type	admin



User Information

	Name	Login Id	Password	Email	Type
□	jhkjh	a	a	jhkh	user
□	arvind	arvind13	admin	arvind@yahoo.in	admin
□	arvind	pathfinder	arvindbisht	arvindsinghait@gmail.com	admin
□	satendra	sat45	admin	satendra@gmail	admin



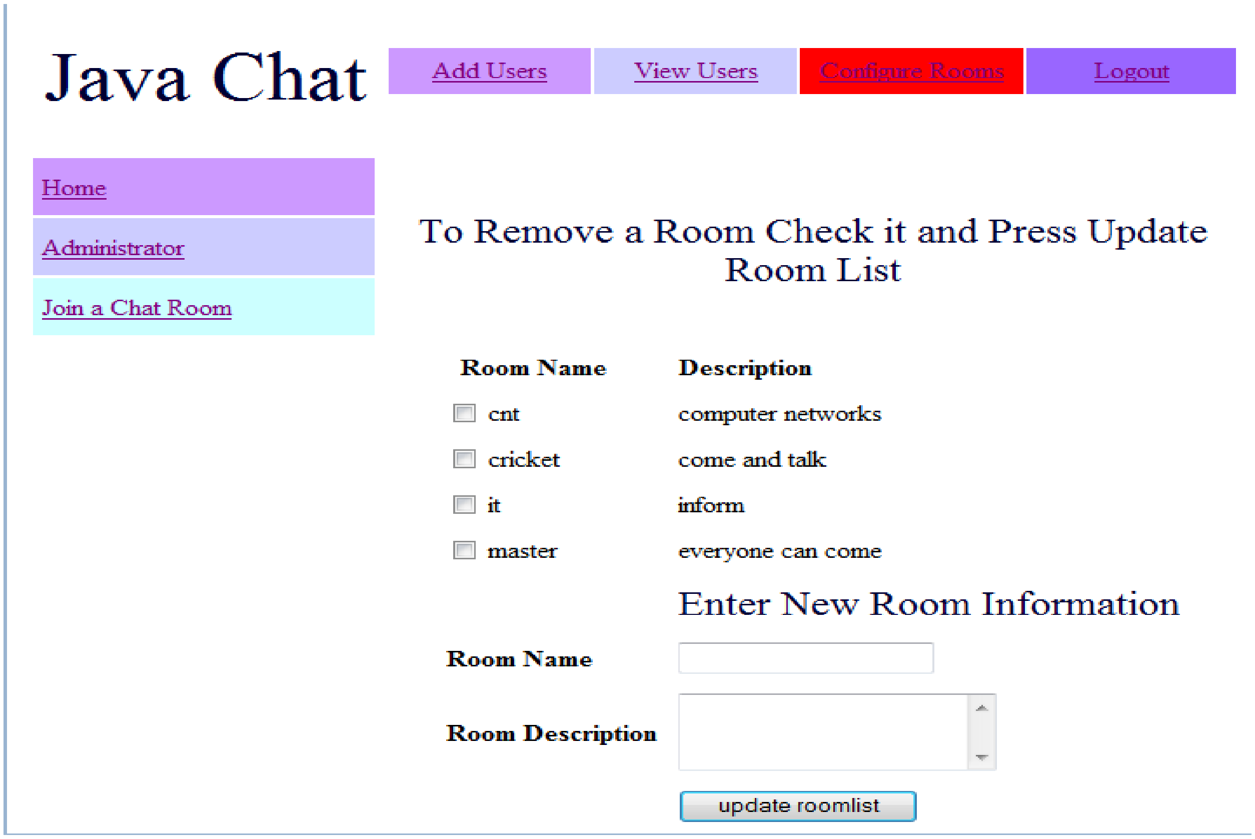
To Remove a Room Check it and Press Update Room List

Room Name	Description
□ cnt	computer networks
□ it	inform
□ master	everyone can come

Enter New Room Information

Room Name: cricket

Room Description: come and talk



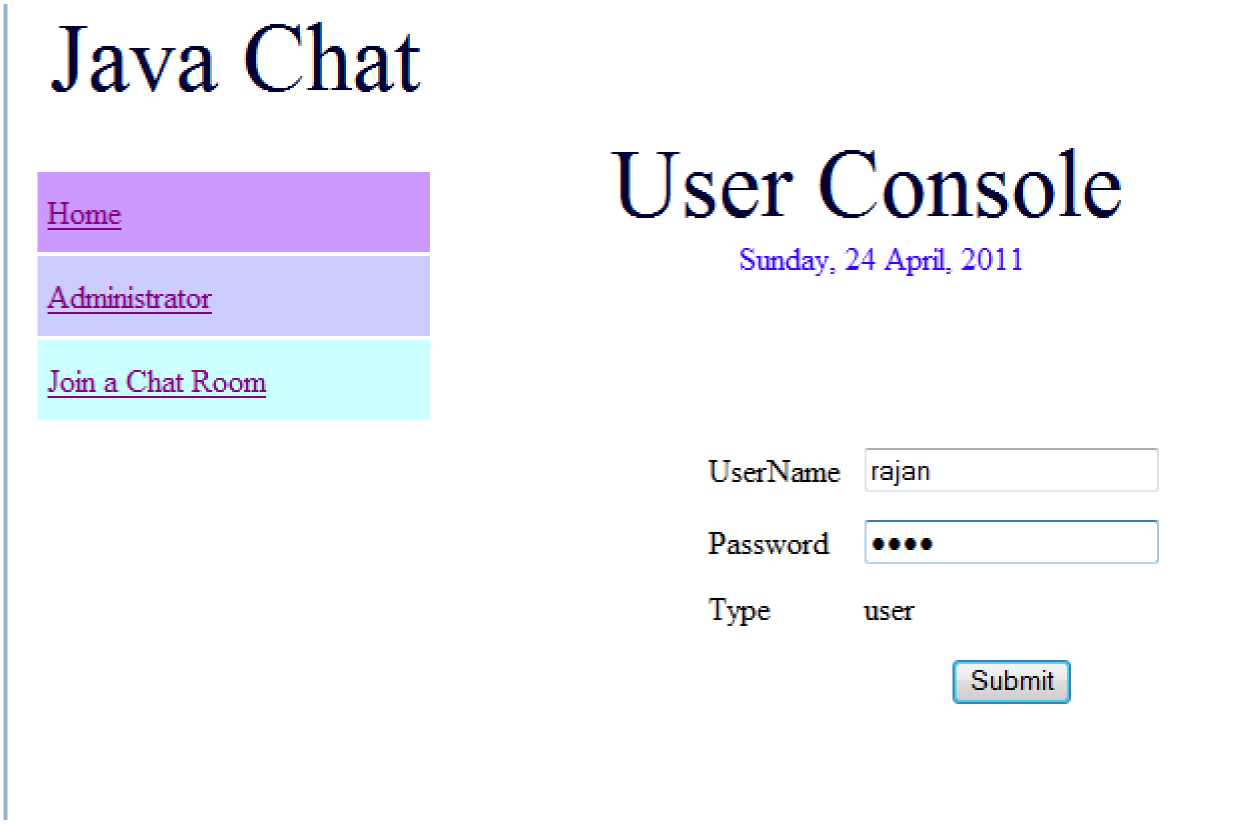
To Remove a Room Check it and Press Update Room List

Room Name	Description
□ cnt	computer networks
□ cricket	come and talk
□ it	inform
□ master	everyone can come

Enter New Room Information

Room Name

Room Description



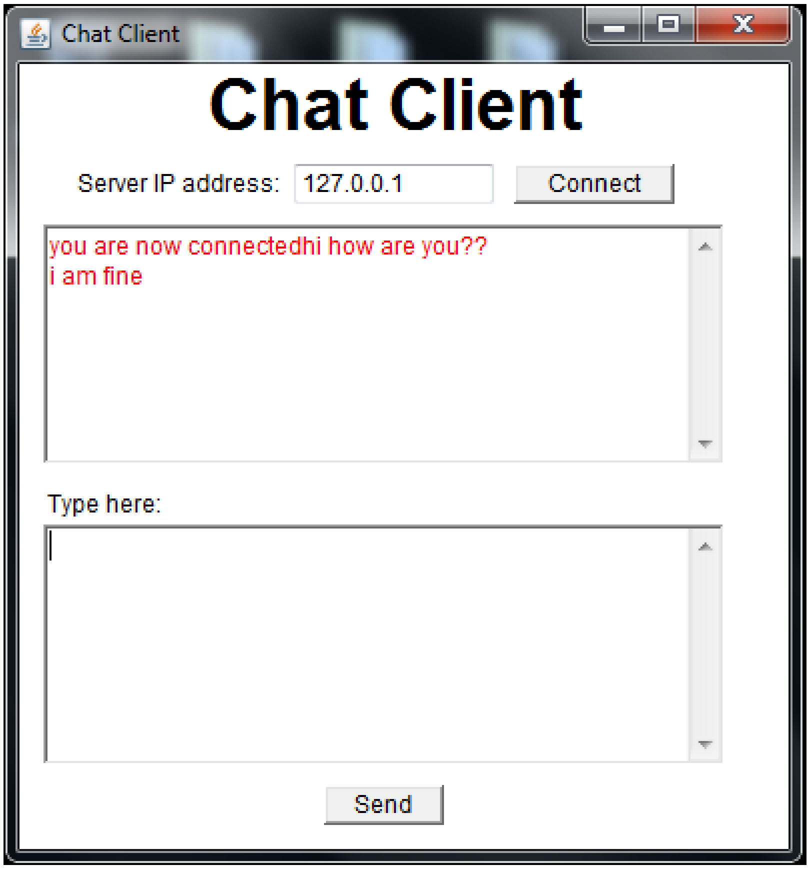
User Console

Sunday, 24 April, 2011

UserName rajan

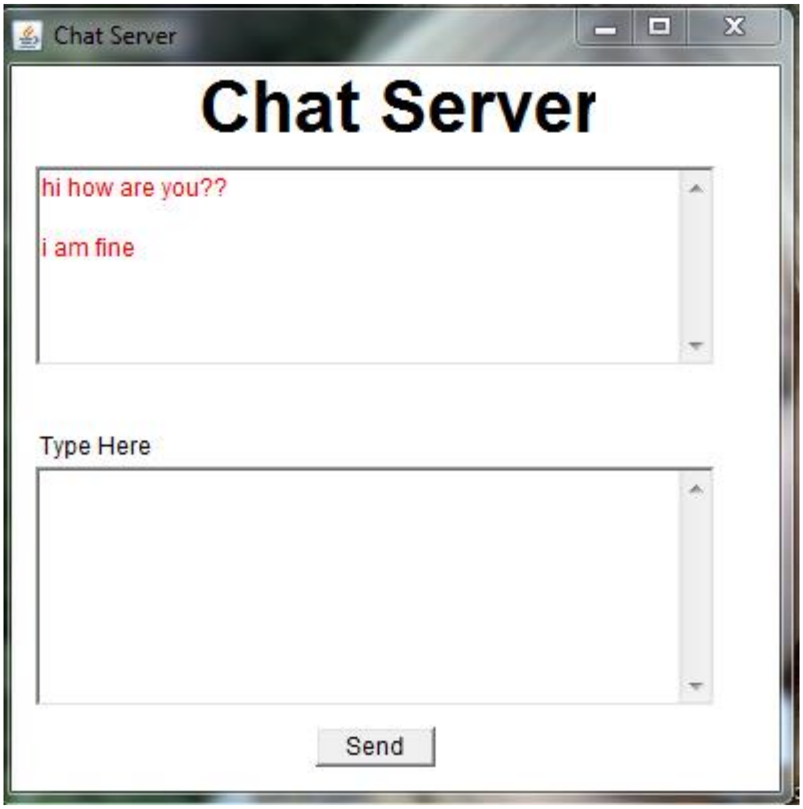
Password ●●●●

Type user



Server IP address: 127.0.0.1

```
you are now connectedhi how are you??
i am fine
```



```
hi how are you??

i am fine
```

FUTURE SCOPE

Jabber framework can be used for instant messaging.
Session initiation protocol [SIP] can help us to provide this TEXT chat special features such as voice and video chat

CONCLUSION

The LAN chat and file sharing is developed using java ,jsp ,servlet,tomcat library and mysql fully meets the objectives of the system for which it has been developed. We have tested it for 6 clients and it has worked .