SECURE CHAT ROOM SERVER

MINOR PROJECT REPORT

By

Sanyog Dani [Reg No.: RA2211031010087]
Arush Sirotiya [Reg No.: RA2211031010092]

Under the guidance of

Dr. M. Manickam

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in Department of Networking and Communications

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Secure Chat Room server**" is the bonafide work of **Sanyog Dani (RA2211031010087)** and **Arush Sirotiya (RA2211031010092)** who carried out the work under my supervision.

SIGNATURE

Dr. Manickam. M

Assistant Professor,

Department of Networking and Communications, School of Computing,

SRM Institute of Science and Technology

Kattankulathur

ABSTRACT

Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers.

The technology has been available for years but the acceptance was quite recent. Our project is an example of a chat server.

It is made up of two applications -the client application, which runs on the user's web browser and server application, runs on any hosting servers on the network.

To start chatting client should get connected to server where they can do private and group chat.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We are highly thankful to our my Course project Faculty **Dr. M. Manickam (Assistant Professor , Department of Networking and Communications)** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD Dr. Annapurani Panaiyappan .K** (**Networking and Communication**) and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Motivation

- To connect people and make the world a smaller place. Chat apps allow people to communicate instantly with each other, regardless of distance.
- This can be especially helpful for staying in touch with friends and family who live far away, or for collaborating with colleagues in different time zones.
- To provide a platform for people to express themselves and share their thoughts and ideas. Chat apps can be used to create communities where people can discuss their interests, share their experiences, and support each other.
- To innovate and create new ways for people to communicate. Chat apps are constantly evolving, with new features and functionality being added all the time.
- There is always room for new and innovative chat apps to enter the market.
- To build a successful business. Chat apps can be very profitable businesses. The global chat app market is expected to reach $1.6 trillion by 2026.

1.2 OBJECTIVE

- **GUI:**

  Easy to use GUI (Graphical User Interface), hence any user with minimal knowledge of operating a system can use the software.

- **Platform independence:**

  The messenger operates on any system irrelevant of the underlying operating system.

- **Unlimited clients:**

  "N" number of users can be connected without any performance degradation of the server.

Objective is to develop an instant messaging solution to enable users to seamlessly communicate with each other. Furthermore, it needs to be user-friendly, i.e., the project should be very easy to use enabling even a novice person to use it

1.3 Problem Statement

- This project is to create a chat application with a server and users to enable the users to chat with each other's.
- To develop an instant messaging solution to enable users to seamlessly communicate with each other
- The project should be very easy to use enabling even a novice person to use it.
- This project can play an important role in organizational field where employees can connect through LAN.

1.4 Challenges

- Scalability: As the number of users of a chat application grows, it can be difficult to scale the system to meet the increased demand. This can lead to performance problems, such as slow loading times and dropped messages.

- Security: Chat applications need to be secure to protect user data from being accessed by unauthorized individuals. This can be a challenge, especially as chat applications become increasingly sophisticated and offer new features, such as file sharing and video calling.

- Real-time communication: Chat applications need to provide real-time communication between users. This can be challenging to achieve, especially across long distances and with different network conditions.

- User experience: Chat applications need to be easy to use and navigate. This can be difficult to achieve, especially as chat applications become increasingly feature-rich.

- Competition: There are a number of popular chat applications already on the market. This can make it difficult for new chat applications to gain traction.

2. LITERATURE SURVEY

1. A considerable research effort has been devoted to Chatting Application in the last few years. Many new researches have been proposed in recent times. This paper proposed a mechanism for creating professional chat application that will not permit the user to send inappropriate or improper messages to the participants by using natural language processing. The message typed by user is evaluated to find any inappropriate terms in it that may include foul/negative words using NLP. If the context of the message is negative, then the user not permitted to send the message .

2. In this paper, a model for developing a secure chatting application was proposed. In proposed model End to End Encryption was achieved by involving ECDH key exchange algorithm to provide the key pair between the two parties to generate the secure shared key that will be used as a key for the encryption. The algorithm used for encrypting text messages was AES standard which is slower than other block cipher. The algorithm used for encrypting voice and image messages was the RC4 which is suitable for the mobile device when encrypting vast amounts of data .

3. This paper introduced the working and implementation of encrypted chat application. It describes how the messages are encrypted on device before moving to server using various cryptographic algorithms. Computer vision techniques can be used for encrypting images. The performance of the application over hundreds of users is not tested yet . Performance testing and image encrytion are in the list of future tasks .

4. This paper proposed a model for building a real time chatting application which provides the users to communicate to each other with an ease. The application will have a login page through which the user can register and login themselves. Home page of the application contains the previous messages if any. The user can search for the other user. User can send and receive text messages. New chat rooms can be created by users and they can search for the content or information. Users can exchange views and information about various topics inside these chat rooms. The identity of the user can also be made hidden in these public chat rooms .

3. REQUIREMENTS

The hardware and software requirements for making a chat applications are :-

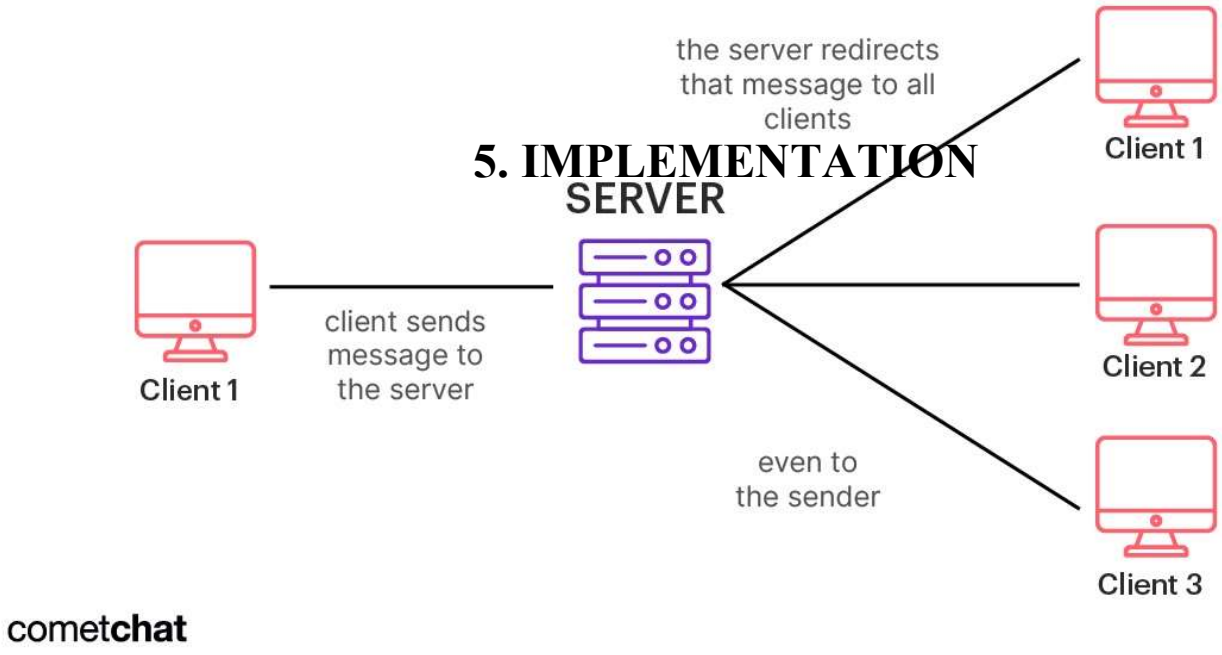
Hardware

- Server: A server with at least 256 MB of RAM and 300 MB of disk space.
- Database server: A database server with enough disk space to store your chat messages and user information.
- Network connection: A high-speed network connection to allow users to connect to your chat application.

Software

- Programming language: A programming language such as Python, Java, JavaScript, or Go.
- Web framework (optional): A web framework such as Django, Flask, or Express.js (if you are building a web-based chat application).
- Database: A database such as PostgreSQL, MySQL, or MongoDB.
- Messaging protocol: A messaging protocol such as WebSocket, XMPP, or MQTT.
- Operating system: A server-grade operating system such as Linux or macOS.

4. ARCHITECTURE AND DESIGN



5. IMPLEMENTATION

6. IMPLEMENTATION

6.1 Server File

```
package chatui;

import java.io.*;
import java.util.*;
import java.text.*;
import java.net.*;
import javax.swing.*;
import java.awt.*;
import javax.swing.border.*;
import java.awt.event.*;

public class Server implements ActionListener {

    JTextField text;
    JPanel a1;
    static Box vertical = Box.createVerticalBox();
    static JFrame f = new JFrame();
    static DataOutputStream dout;

    public void actionPerformed(ActionEvent ae) {
        try {
            String out = text.getText();
            JPanel p2 = formatLabel(out);
            a1.setLayout(new BorderLayout());
            JPanel right = new JPanel(new BorderLayout());
            right.add(p2, BorderLayout.LINE_END);
            vertical.add(right);
            vertical.add(Box.createVerticalStrut(height:15));
            a1.add(vertical, BorderLayout.PAGE_START);
            dout.writeUTF(out);
            text.setText(t:"");
            f.repaint();
            f.invalidate();
            f.validate();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static JPanel formatLabel(String out) {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        JLabel output = new JLabel("<html><p style=\"width: 150px\">" + out + "</p></html>");
        output.setFont(new Font(name:"Tahoma", Font.PLAIN, size:16));
        output.setBackground(new Color(r:0, g:156, b:255));
        output.setOpaque(isOpaque:true);
        output.setBorder(new EmptyBorder(top:15, left:15, bottom:15, right:50));
        panel.add(output);
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern:"HH:mm");
        JLabel time = new JLabel();
```

```
52          time.setText(sdf.format(cal.getTime()));
53          panel.add(time);
54          return panel;
55      }
        Run | Debug
56      public static void main(String[] args) {
57          new Server();
58          try {
59              ServerSocket skt = new ServerSocket(port:6001);
60              while (true) {
61                  Socket s = skt.accept();
62                  DataInputStream din = new DataInputStream(s.getInputStream());
63                  dout = new DataOutputStream(s.getOutputStream());
64                  while (true) {
65                      String msg = din.readUTF();
66                      JPanel panel = formatLabel(msg);
67                      JPanel left = new JPanel(new BorderLayout());
68                      left.add(panel, BorderLayout.LINE_START);
69                      vertical.add(left);
70                      f.validate();
71                  }
72              }
73          } catch (Exception e) {
74              e.printStackTrace();
75          }
76      }
77
78      Server() {
79          f.setLayout(manager:null);
80          JPanel p1 = new JPanel();
81          p1.setBackground(new Color(r:255, g:165, b:0));
82          p1.setBounds(x:0, y:0, width:450, height:70);
83          p1.setLayout(mgr:null);
84          f.add(p1);
85
86          ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3.png"));
87          Image i2 = i1.getImage().getScaledInstance(width:25, height:25, Image.SCALE_DEFAULT);
88          ImageIcon i3 = new ImageIcon(i2);
89          JLabel back = new JLabel(i3);
90          back.setBounds(x:5, y:20, width:25, height:25);
91          p1.add(back);
92
93          back.addMouseListener(new MouseAdapter() {
94              public void mouseClicked(MouseEvent ae) {
95                  System.exit(status:0);
96              }
97          });
98          ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource(name:""));
99          Image i5 = i4.getImage().getScaledInstance(width:50, height:50, Image.SCALE_DEFAULT);
100         ImageIcon i6 = new ImageIcon(i5);
101         JLabel profile = new JLabel(i6);
```

```
        profile.setBounds(x:40, y:10, width:50, height:50);
        p1.add(profile);
        ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource(name:"images/video.png"));
        Image i8 = i7.getImage().getScaledInstance(width:30, height:30, Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel video = new JLabel(i9);
        video.setBounds(x:300, y:20, width:30, height:30);
        p1.add(video);
        ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource(name:"images/phone.png"));
        Image i11 = i10.getImage().getScaledInstance(width:35, height:30, Image.SCALE_DEFAULT);
        ImageIcon i12 = new ImageIcon(i11);
        JLabel phone = new JLabel(i12);
        phone.setBounds(x:360, y:20, width:35, height:30);
        p1.add(phone);
        ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3icon.png"));
        Image i14 = i13.getImage().getScaledInstance(width:10, height:25, Image.SCALE_DEFAULT);
        ImageIcon i15 = new ImageIcon(i14);
        JLabel morevert = new JLabel(i15);
        morevert.setBounds(x:420, y:20, width:10, height:25);
        p1.add(morevert);
        JLabel name = new JLabel(text:"SANYOG");
        name.setBounds(x:110, y:15, width:100, height:18);
        name.setForeground(Color.WHITE);
        name.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:18));
        p1.add(name);
        JLabel status = new JLabel(text:"Active");
        status.setBounds(x:110, y:35, width:100, height:18);
        status.setForeground(Color.WHITE);
        status.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:14));
        p1.add(status);
        a1 = new JPanel();
        a1.setBounds(x:5, y:75, width:440, height:500);
        f.add(a1);
        text = new JTextField();
        text.setBounds(x:5, y:575, width:310, height:40);
        text.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(text);
        JButton send = new JButton(text:"Send");
        send.setBounds(x:320, y:575, width:123, height:40);
        send.setBackground(new Color(r:255, g:165, b:0));
        send.setForeground(Color.WHITE);
        send.addActionListener(this);
        send.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(send);
        f.setSize(width:450, height:700);
        f.setLocation(x:200, y:50);
        f.setUndecorated(undecorated:true);
        f.getContentPane().setBackground(Color.WHITE);
        f.setVisible(b:true);
    }
}
```

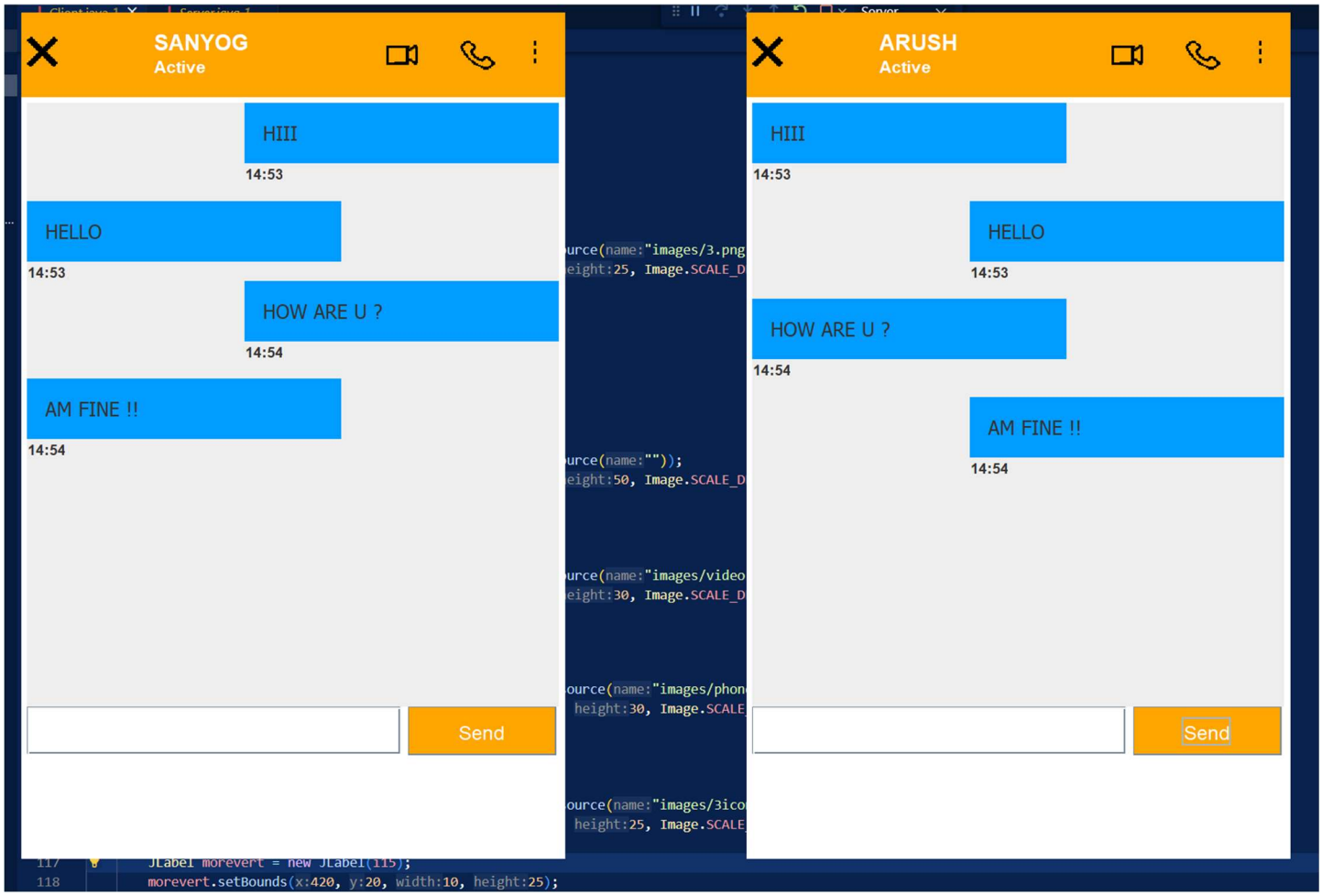
6.2 CLIENT FILE

```
package chatui;

import java.io.*;
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.text.*;
import java.net.*;
import javax.swing.border.*;

public class Client implements ActionListener {

    JTextField text;
    static JPanel a1;
    static Box vertical = Box.createVerticalBox();
    static JFrame f = new JFrame();
    static DataOutputStream dout;

    public static JPanel formatLabel(String out) {
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        JLabel output = new JLabel("<html><p style=\"width: 150px\">" + out + "</p></html>");
        output.setFont(new Font(name:"Tahoma", Font.PLAIN, size:16));
        output.setBackground(new Color(r:0,g:156,b:255));
        output.setOpaque(isOpaque:true);
        output.setBorder(new EmptyBorder(top:15, left:15, bottom:15, right:50));
        panel.add(output);
        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern:"HH:mm");
        JLabel time = new JLabel();
        time.setText(sdf.format(cal.getTime()));
        panel.add(time);
        return panel;
    }

    public void actionPerformed(ActionEvent ae) {
        try {
            String out = text.getText();
            JPanel p2 = formatLabel(out);
            a1.setLayout(new BorderLayout());
            JPanel right = new JPanel(new BorderLayout());
            right.add(p2, BorderLayout.LINE_END);
            vertical.add(right);
            vertical.add(Box.createVerticalStrut(height:15));
            a1.add(vertical, BorderLayout.PAGE_START);
            dout.writeUTF(out);
            text.setText(t:"");
            f.repaint();
            f.invalidate();
```

```
52              f.validate();
53          } catch (Exception e) {
54              e.printStackTrace();
55          }
56      }
        Run | Debug
57      public static void main(String[] args) {
58          new Client();
59          try {
60              Socket s = new Socket(host:"127.0.0.1", port:6001);
61              DataInputStream din = new DataInputStream(s.getInputStream());
62              dout = new DataOutputStream(s.getOutputStream());
63              while (true) {
64                  a1.setLayout(new BorderLayout());
65                  String msg = din.readUTF();
66                  JPanel panel = formatLabel(msg);
67                  JPanel left = new JPanel(new BorderLayout());
68                  left.add(panel, BorderLayout.LINE_START);
69                  vertical.add(left);
70                  vertical.add(Box.createVerticalStrut(height:15));
71                  a1.add(vertical, BorderLayout.PAGE_START);
72                  f.validate();
73              }
74          } catch (Exception e) {
75              e.printStackTrace();
76          }
77      }
78      Client() {
79          f.setLayout(manager:null);
80          JPanel p1 = new JPanel();
81          p1.setBackground(new Color(r:255, g:165, b:0));
82          p1.setBounds(x:0, y:0, width:450, height:70);
83          p1.setLayout(mgr:null);
84          f.add(p1);
85          ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3.png"));
86          Image i2 = i1.getImage().getScaledInstance(width:25, height:25, Image.SCALE_DEFAULT);
87          ImageIcon i3 = new ImageIcon(i2);
88          JLabel back = new JLabel(i3);
89          back.setBounds(x:5, y:20, width:25, height:25);
90          p1.add(back);
91          back.addMouseListener(new MouseAdapter() {
92              public void mouseClicked(MouseEvent ae) {
93                  System.exit(status:0);
94              }
95          });
96          ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource(name:""));
97          Image i5 = i4.getImage().getScaledInstance(width:50, height:50, Image.SCALE_DEFAULT);
98          ImageIcon i6 = new ImageIcon(i5);
99          JLabel profile = new JLabel(i6);
100         profile.setBounds(x:40, y:10, width:50, height:50);
101         p1.add(profile);
```

```
        profile.setBounds(x:40, y:10, width:50, height:50);
        p1.add(profile);
        ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource(name:"images/video.png"));
        Image i8 = i7.getImage().getScaledInstance(width:30, height:30, Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel video = new JLabel(i9);
        video.setBounds(x:300, y:20, width:30, height:30);
        p1.add(video);
        ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource(name:"images/phone.png"));
        Image i11 = i10.getImage().getScaledInstance(width:35, height:30, Image.SCALE_DEFAULT);
        ImageIcon i12 = new ImageIcon(i11);
        JLabel phone = new JLabel(i12);
        phone.setBounds(x:360, y:20, width:35, height:30);
        p1.add(phone);
        ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource(name:"images/3icon.png"));
        Image i14 = i13.getImage().getScaledInstance(width:10, height:25, Image.SCALE_DEFAULT);
        ImageIcon i15 = new ImageIcon(i14);
        JLabel morevert = new JLabel(i15);
        morevert.setBounds(x:420, y:20, width:10, height:25);
        p1.add(morevert);
        JLabel name = new JLabel(text:"UNKNOWN");
        name.setBounds(x:110, y:15, width:100, height:18);
        name.setForeground(Color.WHITE);
        name.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:18));
        p1.add(name);
        JLabel status = new JLabel(text:"Active");
        status.setBounds(x:110, y:35, width:100, height:18);
        status.setForeground(Color.WHITE);
        status.setFont(new Font(name:"SAN_SERIF", Font.BOLD, size:14));
        p1.add(status);
        a1 = new JPanel();
        a1.setBounds(x:5, y:75, width:440, height:500);
        f.add(a1);
        text = new JTextField();
        text.setBounds(x:5, y:575, width:310, height:40);
        text.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(text);
        JButton send = new JButton(text:"Send");
        send.setBounds(x:320, y:575, width:123, height:40);
        send.setBackground(new Color(r:255, g:165, b:0));
        send.setForeground(Color.WHITE);
        send.addActionListener(this);
        send.setFont(new Font(name:"SAN_SERIF", Font.PLAIN, size:16));
        f.add(send);
        f.setSize(width:450, height:700);
        f.setLocation(x:800, y:50);
        f.setUndecorated(undecorated:true);
        f.getContentPane().setBackground(Color.WHITE);
        f.setVisible(b:true);
    }
}
```

7. RESULTS


SANYOG
Active
HIII
14:53
HELLO
14:53
HOW ARE U ?
14:54
AM FINE !!
14:54
Send
ARUSH
Active
HIII
14:53
HELLO
14:53
HOW ARE U ?
14:54
AM FINE !!
14:54
Send

8. CONCLUSION

- We studied about that how **SERVER , CLIENT** works in Java .
- We learned how to make GUI using java Swing .
- This Project helped us to know more about JAVA , JAVA SWING, SERVER, CLIENT in a simple manner.
- Thus a Working SECURE CHAT ROOM SERVER is created which enable people to chat with Privacy.

9. REFERENCES

- https://www.geeksforgeeks.org/
- https://www.javatpoint.com/
- https://www.wikipedia.org/
- https://www.youtube.com/