**5.4. SAMPLE CODE**

**🡪Register.java**

package BFC;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.net.Socket;

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import net.miginfocom.swing.MigLayout;

import java.awt.Color;

import javax.swing.JFrame;

public class Register extends JFrame{

JPanel p1;

Font f1;

JLabel l1,l2,l3,l4,l5;

JTextField tf1,tf2,tf3,tf4,tf5;

JButton b1,b2;

Dimension d;

int w,h;

int center;

public Register(){

super("New User Registration Screen");

d = Toolkit.getDefaultToolkit().getScreenSize();

w = (int)d.getWidth();

h = (int)d.getHeight();

center = w/30;

setBackground(Color.white);

p1 = new JPanel();

p1.setBackground(Color.white);

p1.setLayout(new MigLayout("wrap 2","","[]15[]"));

f1 = new Font("Verdana",Font.PLAIN,14);

p1.add(new JLabel(),"cell 0 30");

p1.add(new JLabel(),"cell 1 30");

l1 = new JLabel("Username");

l1.setFont(f1);

p1.add(l1,"gap left "+center);

tf1 = new JTextField(12);

tf1.setFont(f1);

p1.add(tf1,"gap left 30");

l2 = new JLabel("Password");

l2.setFont(f1);

p1.add(l2,"gap left "+center);

tf2 = new JPasswordField(12);

tf2.setFont(f1);

p1.add(tf2,"gap left 30");

l3 = new JLabel("Contact No");

l3.setFont(f1);

p1.add(l3,"gap left "+center);

tf3 = new JTextField(12);

tf3.setFont(f1);

p1.add(tf3,"gap left 30");

l4 = new JLabel("Email ID");

l4.setFont(f1);

p1.add(l4,"gap left "+center);

tf4 = new JTextField(12);

tf4.setFont(f1);

p1.add(tf4,"gap left 30");

l5 = new JLabel("Address");

l5.setFont(f1);

p1.add(l5,"gap left "+center);

tf5 = new JTextField(30);

tf5.setFont(f1);

p1.add(tf5,"gap left 30");

p1.add(new JLabel(""));

b1 = new JButton("Register");

b1.setFont(f1);

p1.add(b1,"split 2");

b1.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent ae){

process();

}

});

b2 = new JButton("Clear");

b2.setFont(f1);

p1.add(b2);

b2.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent ae){

clearFields();

}

});

add(p1);

}

public void process(){

try{

String uname=tf1.getText();

String pass=tf2.getText();

String contact=tf3.getText();

String email=tf4.getText();

String address=tf5.getText();

if(uname.length() <=0 || uname == null){

JOptionPane.showMessageDialog(this,"Username must be enter");

tf1.requestFocus();

return;

}

if(pass.length() <=0 || pass == null){

JOptionPane.showMessageDialog(this,"Password must be enter");

tf2.requestFocus();

return;

}

if(contact.length() <=0 || contact == null){

JOptionPane.showMessageDialog(this,"Contact No must be enter");

tf3.requestFocus();

return;

}

if(contact.length() <=0 || contact == null){

JOptionPane.showMessageDialog(this,"Contact no must be enter");

tf3.requestFocus();

return; }

if(!validatePhoneNumber(contact.trim())){

JOptionPane.showMessageDialog(this,"Enter valid contact no");

tf3.requestFocus();

return;

}

if(email.length() <=0 || email == null){

JOptionPane.showMessageDialog(this,"Email id must be enter");

tf4.requestFocus();

return;

}

if(!CheckMail.checkMail(email)){

JOptionPane.showMessageDialog(this,"Enter valid mailid");

tf4.requestFocus();

return;

}

if(address.length() <=0 || address == null){

JOptionPane.showMessageDialog(this,"Address must be enter");

tf5.requestFocus();

return;

}

Socket socket=new Socket("localhost",1200);

ObjectOutputStream out=new ObjectOutputStream(socket.getOutputStream());

ObjectInputStream in=new ObjectInputStream(socket.getInputStream());

Object req[]={"register",uname,pass,contact,email,address};

out.writeObject(req);

out.flush();

Object res[]=(Object[])in.readObject();

String msg = res[0].toString();

if(msg.equals("Registration process completed")){

JOptionPane.showMessageDialog(this,msg);

setVisible(false);

}else{

JOptionPane.showMessageDialog(this,msg);

}

}catch(Exception e){

e.printStackTrace();

}

}

public void clearFields(){

tf1.setText("");

tf2.setText("");

tf3.setText("");

tf4.setText("");

tf5.setText("");

}

private static boolean validatePhoneNumber(String phoneNo){

//validate phone numbers of format "1234567890"

if(phoneNo.matches("\\d{10}"))

return true;

//validating phone number with -, . or spaces

else if(phoneNo.matches("\\d{3}[-\\.\\s]\\d{3}[-\\.\\s]\\d{4}"))

return true;

//validating phone number with extension length from 3 to 5

else if(phoneNo.matches("\\d{3}-\\d{3}-\\d{4}\\s(x|(ext))\\d{3,5}"))

return true;

//validating phone number where area code is in braces ()

else if(phoneNo.matches("\\(\\d{3}\\)-\\d{3}-\\d{4}"))

return true;

//return false if nothing matches the input

else

return false;

}}

**Login.java**

package BFC;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.BorderLayout;

import java.awt.Color;

import javax.swing.JPasswordField;

import javax.swing.UIManager;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.net.Socket;

import javax.swing.JOptionPane;

public class Login extends JFrame

{

JLabel l1,l2,l3,l4;

JTextField tf1,tf2;

JButton b1,b2,b3;

Font f1,f2;

JPanel p1,p2,p3,p4,p5,p6;

ImageIcon icon;

public Login(){

super("Login Screen");

p1 = new JPanel();

p1.setBackground(Color.black);

f1 = new Font("Monospaced",Font.BOLD,22);

l1=new JLabel("<html><body><center>User Login Screen</center></body></html>");

l1.setForeground(new Color(125,54,2));

l1.setFont(f1);

p1.add(l1);

p1.setBackground(new Color(140,150,180));

p2 = new JPanel();

p2.setBackground(Color.black);

icon = new ImageIcon("img/vista.jpg");

JLabel label = new JLabel(icon);

p2.add(label);

p3 = new JPanel();

p3.setLayout(new BorderLayout());

p4 = new JPanel();

f2 = new Font("Verdana",Font.PLAIN,14);

l3 = new JLabel("Username");

l3.setFont(f2);

p4.add(l3);

tf1 = new JTextField(15);

tf1.setFont(f2);

p4.add(tf1);

p5 = new JPanel();

l4 = new JLabel("Password");

l4.setFont(f2);

p5.add(l4);

tf2 = new JPasswordField(15);

tf2.setFont(f2);

p5.add(tf2);

p6 = new JPanel();

b1 = new JButton("Login");

b1.setFont(f2);

b1.setBackground(new Color(51, 51, 51));

p6.add(b1);

b1.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

login();

}

});

b2 = new JButton("Reset");

b2.setFont(f2);

b2.setBackground(new java.awt.Color(51, 51, 51));

p6.add(b2);

b2.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

reset();

}

});

b3 = new JButton("New User");

b3.setFont(f2);

b3.setBackground(new java.awt.Color(51, 51, 51));

p6.add(b3);

b3.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

Register register = new Register();

register.pack();

register.setVisible(true);

register.setLocationRelativeTo(null);

}

});

p3.add(p4,BorderLayout.NORTH);

p3.add(p5,BorderLayout.CENTER);

p3.add(p6,BorderLayout.SOUTH);

getContentPane().add(p1,BorderLayout.NORTH);

getContentPane().add(p2,BorderLayout.CENTER);

getContentPane().add(p3,BorderLayout.SOUTH);

}

public static void main(String a[])throws Exception{

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

Login screen = new Login();

screen.setVisible(true);

screen.pack();

screen.setLocationRelativeTo(null);

}

public void reset(){

tf1.setText("");

tf2.setText("");

}

public void login(){

try{

String user = tf1.getText();

String pass = tf2.getText();

if(user == null || user.trim().length() <= 0){

JOptionPane.showMessageDialog(this,"Username must be enter");

tf1.requestFocus();

return;

}

if(pass == null || pass.trim().length() <= 0){

JOptionPane.showMessageDialog(this,"Password must be enter");

tf2.requestFocus();

return;

}

Socket socket=new Socket("localhost",1200);

ObjectOutputStream out=new ObjectOutputStream(socket.getOutputStream());

ObjectInputStream in=new ObjectInputStream(socket.getInputStream());

Object req[]={"login",user,pass};

out.writeObject(req);

out.flush();

Object res[]=(Object[])in.readObject();

String msg = res[0].toString();

if(msg.equals("success")){

setVisible(false);

UserScreen us = new UserScreen(this,user);

us.setVisible(true);

us.setExtendedState(JFrame.MAXIMIZED\_BOTH);

}

else

{

JOptionPane.showMessageDialog(this,"invalid login");

}

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**UserScreen.java**

package BFC;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.BorderLayout;

import java.awt.Color;

import javax.swing.JTextArea;

import javax.swing.JScrollPane;

import javax.swing.JOptionPane;

import javax.swing.JFileChooser;

import java.net.Socket;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.io.RandomAccessFile;

import java.util.Random;

import java.io.File;

import java.io.FileInputStream;

public class UserScreen extends JFrame

{

JButton b1,b2,b3;

JPanel p1,p2;

Font f1;

JTextArea area;

JScrollPane jsp;

Login login;

String user;

JFileChooser chooser;

RandomAccessFile random;

int tot\_blocks;

public UserScreen(Login log,String usr){

super("User Screen");

login = log;

user = usr;

p1 = new JPanel();

f1 = new Font("Monospaced",Font.BOLD,16);

chooser = new JFileChooser();

b1 = new JButton("Upload File");

b1.setFont(f1);

p1.add(b1);

b1.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

int option = chooser.showOpenDialog(UserScreen.this);

if(option == JFileChooser.APPROVE\_OPTION){

File file = chooser.getSelectedFile();

upload(file);

}}});

b2 = new JButton("Download File");

b2.setFont(f1);

p1.add(b2);

b2.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

DownloadFile df = new DownloadFile(area);

df.setUser(user);

df.setSize(300,100);

df.setVisible(true);

df.setLocationRelativeTo(null);

df.getFileName();

}});

b3 = new JButton("Logout");

b3.setFont(f1);

p1.add(b3);

b3.addActionListener(new ActionListener(){

public void actionPerformed(java.awt.event.ActionEvent evt) {

setVisible(false);

login.setVisible(true);

}});

p2 = new JPanel();

p2.setLayout(new BorderLayout());

area = new JTextArea();

area.setFont(f1);

area.setEditable(false);

jsp = new JScrollPane(area);

p2.add(jsp,BorderLayout.CENTER);

getContentPane().add(p1,BorderLayout.NORTH);

getContentPane().add(p2,BorderLayout.CENTER);

}

public long getChunkSize(File file){

long length = file.length();

tot\_blocks=0;

long size = 0;

if(length >= 1000){

size = length/10;

tot\_blocks = 10;

}

if(length < 1000 && length > 500){

size = length/5;

tot\_blocks = 5;

}

if(length < 500 && length > 1){

size = length/3;

tot\_blocks = 3;

}

return size;

}

public void upload(File file){

try{

FileInputStream fin = new FileInputStream(file);

byte file\_data[] = new byte[fin.available()];

fin.read(file\_data,0,file\_data.length);

fin.close();

byte encrypt[] = AES.encrypt(file\_data);

String sha = SHA.ShaSignature(encrypt);

long chunk\_size = getChunkSize(file);

random = new RandomAccessFile(file,"r");

Object row[][] = createChunks(chunk\_size,file.getName());

random.close();

Socket socket=new Socket("localhost",1200);

ObjectOutputStream out=new ObjectOutputStream(socket.getOutputStream());

ObjectInputStream in=new ObjectInputStream(socket.getInputStream());

Object req[]={"upload",row,user,file.getName(),sha,Long.toString(file.length())};

out.writeObject(req);

out.flush();

Object res[]=(Object[])in.readObject();

String server\_res = res[0].toString();

area.append(server\_res+"\n");

out.close();

in.close();

socket.close();

}catch(Exception e){

e.printStackTrace();

}

}

public Object[][] createChunks(long chunks\_size,String name)

{

area.append("Total chunks "+tot\_blocks+" With chunk size "+chunks\_size+"\n");

Object row[][]=new Object[tot\_blocks][2];

Try

{

String ext = name.substring(name.lastIndexOf(".")+1,name.length());

for(int i=0;i<tot\_blocks;i++){

byte b[]=new byte[(int)chunks\_size];

random.read(b);

random.seek(random.getFilePointer());

row[i][0]=b;

row[i][1]=name+"\_chunk"+i+"."+ext;

}

}

catch(Exception e)

{

e.printStackTrace();

}

return row;

}

}

}

**DBCon.java**

package BFC;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

public class DBCon

{

private static Connection con;

public static Connection getCon()throws Exception

{

if(con == null){

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://localhost/BFC","root","root");

}

return con;

}

public static String register(String[] input)throws Exception

{

String msg="Error in registration";

boolean flag=false;

con = getCon();

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select username from bfc\_users where

username='"+input[0]+"'");

if(rs.next())

{

flag=true;

msg = "Username already exist";

}

if(!flag)

{

PreparedStatement stat=con.prepareStatement("insert into bfc\_users

values(?,?,?,?,?)");

stat.setString(1,input[0]);

stat.setString(2,input[1]);

stat.setString(3,input[2]);

stat.setString(4,input[3]);

stat.setString(5,input[4]);

int i=stat.executeUpdate();

if(i > 0)

{

msg = "Registration process completed";

}

stat.close();

}

rs.close();stmt.close();

return msg;

}

public static String login(String input[])throws Exception

{

String msg="fail";

con = getCon();

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select username from bfc\_users where

username='"+input[0]+"' && password='"+input[1]+"'");

if(rs.next())

{

msg = "success";

}

rs.close();stmt.close();

return msg;

}

}