client

import javax.swing.\*;  
import javax.swing.border.EmptyBorder;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.util.\*;  
import java.text.\*;  
import java.io.\*;  
import java.net.\*;  
import java.sql.\*;  
  
public class Client implements ActionListener {  
 JTextField text;  
 static JPanel *a1*;  
 static Box *vertical* = Box.*createVerticalBox*();  
 static JFrame *frame* = new JFrame();  
 static DataOutputStream *dout*;  
 private static Connection *connection*;  
 Client(){  
 // Initialize database connection  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver"); // Load the JDBC driver  
 *connection* = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/chatapp", "root", "root");  
 createTable(); // Create the table if not exists  
 } catch (ClassNotFoundException | SQLException e) {  
 e.printStackTrace();  
 throw new RuntimeException("Failed to connect to the database");  
 }  
  
 *frame*.setLayout(null);  
 JPanel p1 = new JPanel();  
 p1.setBackground(new Color(7, 94,84));  
 p1.setBounds(0,0,900,70);  
 p1.setLayout(null);  
 *frame*.add(p1);  
 //back icon  
 ImageIcon i1 = new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/3.png"));  
 Image i2 = i1.getImage().getScaledInstance(25,25,Image.*SCALE\_DEFAULT*);  
 ImageIcon i3 = new ImageIcon(i2);  
 JLabel back = new JLabel(i3);  
 back.setBounds(5, 20, 25,25);  
 p1.add(back);  
  
 back.addMouseListener(new MouseAdapter(){  
 public void mouseClicked(MouseEvent ae){  
 *frame*.setVisible(false);  
 // System.exit(0);  
 }  
 });  
 //profile picture icon  
 ImageIcon i4 = new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/kaleen.png"));  
 Image i5 = i4.getImage().getScaledInstance(50,50,Image.*SCALE\_AREA\_AVERAGING*);  
 ImageIcon i6 = new ImageIcon(i5);  
 JLabel profile = new JLabel(i6);  
 profile.setBounds(40, 10, 50,50);  
 p1.add(profile);  
 //video call icon  
 ImageIcon i7= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/video.png"));  
 Image i8 = i7.getImage().getScaledInstance(30,30,Image.*SCALE\_AREA\_AVERAGING*);  
 ImageIcon i9 = new ImageIcon(i8);  
  
 JLabel video = new JLabel(i9);  
 video.setBounds(300, 20, 30,30);  
 p1.add(video);  
 //phone  
 ImageIcon i10= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/phone.png"));  
 Image i11 = i10.getImage().getScaledInstance(35,30,Image.*SCALE\_DEFAULT*);  
 ImageIcon i12 = new ImageIcon(i11);  
 JLabel phone = new JLabel(i12);  
 phone.setBounds(360, 20, 35,30);  
 p1.add(phone);  
 //more option  
 ImageIcon i13= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/3icon.png"));  
 Image i14 = i13.getImage().getScaledInstance(10,25,Image.*SCALE\_DEFAULT*);  
 ImageIcon i15 = new ImageIcon(i14);  
 JLabel option = new JLabel(i15);  
 option.setBounds(420, 20, 10,25);  
 p1.add(option);  
 //display name  
 JLabel name = new JLabel("kaleen");  
 name.setBounds(110,20,120,15);  
 name.setForeground(Color.*WHITE*);  
 name.setFont(new Font("SAN\_SERIF",Font.*BOLD*,19));  
 p1.add(name);  
 // status online or offline with a green dot  
 JPanel statusPanel = new JPanel(new FlowLayout(FlowLayout.*LEFT*));  
 statusPanel.setBounds(100, 35, 120, 20);  
 statusPanel.setOpaque(false);  
  
 // green dot  
 JLabel statusDot;  
 statusDot = new JLabel(" ●");  
 statusDot.setForeground(new Color(37, 211, 102));  
 statusDot.setFont(new Font("SAN\_SERIF", Font.*BOLD*, 13));  
 statusPanel.add(statusDot);  
 JLabel status = new JLabel("Active Now");  
 status.setForeground(Color.*WHITE*);  
 status.setFont(new Font("SAN\_SERIF", Font.*BOLD*, 13));  
 statusPanel.add(status);  
 p1.add(statusPanel);  
 // writing boundry  
 *a1* = new JPanel();  
 *a1*.setBounds(5,75, 440, 570);  
 *frame*.add(*a1*);  
 //footer  
 //textfield  
 text= new JTextField();  
 text.setBounds(5, 655, 310, 40);  
 text.setFont(new Font ("SAN\_SERIF", Font.*PLAIN*, 16));  
 *frame*.add(text);  
 //send button  
 JButton send = new JButton("Send");  
 send.setBounds(320,655,123,40);  
 send.setBackground(new Color(7,94,84));  
 send.setForeground(Color.*WHITE*);  
 send.addActionListener(this);  
 send.setFont(new Font ("SAN\_SERIF", Font.*PLAIN*, 16));  
 *frame*.add(send);  
  
  
 *frame*.setSize(450, 700);  
 *frame*.setLocation(800,50);  
 *frame*.setUndecorated(true);  
 *frame*.getContentPane().setBackground(Color.*WHITE*);  
  
 *frame*.setVisible(true);  
  
 }  
 private void createTable() throws SQLException {  
 String createTableSQL = "CREATE TABLE IF NOT EXISTS messages(" +  
 "id INT AUTO\_INCREMENT PRIMARY KEY," +  
 " Sender VARCHAR(30)," +  
 "message VARCHAR(255) NOT NULL," +  
 "timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP)";  
 try (Statement statement = *connection*.createStatement()) {  
 statement.executeUpdate(createTableSQL);  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 throw new RuntimeException("Failed to create table");  
 }  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent ae) {  
 try{  
 String sender= "kaleen";  
 String out= text.getText();  
 *saveMessageToDatabase*(sender,out);  
 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*(15));  
 *a1*.add(*vertical*,BorderLayout.*PAGE\_START*);  
 *dout*.writeUTF(out);  
  
 text.setText("");  
  
 *frame*.repaint();  
 *frame*.invalidate();  
 *frame*.validate();  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
 private static void saveMessageToDatabase(String sender,String message) {  
 String sql = "INSERT INTO messages(sender,message, timestamp) VALUES (?,?, NOW())";  
 try (PreparedStatement statement = *connection*.prepareStatement(sql)) {  
 statement.setString(1,sender);  
 statement.setString(2, message);  
 // Set autocommit to false  
 *connection*.setAutoCommit(false);  
 statement.executeUpdate();  
 // Commit the transaction  
 *connection*.commit();  
 // Set autocommit back to true  
 *connection*.setAutoCommit(true);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 // Rollback the transaction on error  
 *rollback*();  
 throw new RuntimeException("Failed to save message to the database");  
 }  
 }  
  
 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: 150 px\">"+ out + "</p></html>");  
 output.setFont(new Font("Tahoma", Font.*PLAIN*,16));  
 output.setBackground(new Color(37,211,102));  
 output.setOpaque(true);  
 output.setBorder(new EmptyBorder(15,15,15,50));  
  
  
 panel.add(output);  
  
 Calendar cal = Calendar.*getInstance*();  
 SimpleDateFormat sdf= new SimpleDateFormat("HH:mm");  
  
 JLabel time = new JLabel();  
 time.setText(sdf.format(cal.getTime()));  
 panel.add(time);  
 return panel;  
 }  
 private static void rollback() {  
 try {  
 if (*connection* != null) {  
 *connection*.rollback();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 //main class  
 public static void main(String[] args){  
  
 new Client();  
  
 try {  
 Socket s = new Socket("127.0.0.1",6001);  
 System.*out*.println("connected to the server");  
 DataInputStream din = new DataInputStream(s.getInputStream());  
 *dout* = new DataOutputStream(s.getOutputStream());  
 while(true){  
 *a1*.setLayout(new BorderLayout());  
  
 String msg = din.readUTF();  
 // saveMessageToDatabase(msg);  
 JPanel panel = *formatLabel*(msg);  
  
 JPanel left = new JPanel(new BorderLayout());  
 left.add(panel,BorderLayout.*LINE\_START*);  
 *vertical*.add(left);  
 *vertical*.add(Box.*createVerticalStrut*(15));  
 *a1*.add(*vertical*,BorderLayout.*PAGE\_START*);  
 *frame*.validate();  
  
 }  
  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
  
  
 }  
  
  
}

Server

import javax.swing.\*;  
import javax.swing.border.EmptyBorder;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.text.\*;  
import java.util.\*;  
import java.net.\*;  
import java.io.\*;  
import java.sql.\*;  
  
public class Server implements ActionListener {  
 JTextField text;  
 static JPanel *a1*;  
 static Box *vertical* = Box.*createVerticalBox*();  
 static JFrame *frame* = new JFrame();  
 static DataOutputStream *dout*;  
 static Connection *connection*;  
  
 Server(){  
 // Initialize database connection  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver"); // Load the JDBC driver  
 *connection* = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/chatapp", "root", "root");  
 createTable(); // Create the table if not exists  
 } catch (ClassNotFoundException | SQLException e) {  
 e.printStackTrace();  
 throw new RuntimeException("Failed to connect to the database");  
 }  
  
 *frame*.setLayout(null);  
 JPanel p1 = new JPanel();  
 p1.setBackground(new Color(7, 94,84));  
 p1.setBounds(0,0,900,70);  
 p1.setLayout(null);  
 *frame*.add(p1);  
 //back icon  
 ImageIcon i1 = new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/3.png"));  
 Image i2 = i1.getImage().getScaledInstance(25,25,Image.*SCALE\_DEFAULT*);  
 ImageIcon i3 = new ImageIcon(i2);  
 JLabel back = new JLabel(i3);  
 back.setBounds(5, 20, 25,25);  
 p1.add(back);  
  
 back.addMouseListener(new MouseAdapter(){  
 public void mouseClicked(MouseEvent ae){  
 *frame*.setVisible(false);  
 // System.exit(0);  
 }  
 });  
 //profile picture icon  
 ImageIcon i4 = new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/munna.png"));  
 Image i5 = i4.getImage().getScaledInstance(50,50,Image.*SCALE\_AREA\_AVERAGING*);  
 ImageIcon i6 = new ImageIcon(i5);  
 JLabel profile = new JLabel(i6);  
 profile.setBounds(40, 10, 50,50);  
 p1.add(profile);  
 //video call icon  
 ImageIcon i7= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/video.png"));  
 Image i8 = i7.getImage().getScaledInstance(30,30,Image.*SCALE\_AREA\_AVERAGING*);  
 ImageIcon i9 = new ImageIcon(i8);  
  
 JLabel video = new JLabel(i9);  
 video.setBounds(300, 20, 30,30);  
 p1.add(video);  
 //phone  
 ImageIcon i10= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/phone.png"));  
 Image i11 = i10.getImage().getScaledInstance(35,30,Image.*SCALE\_DEFAULT*);  
 ImageIcon i12 = new ImageIcon(i11);  
 JLabel phone = new JLabel(i12);  
 phone.setBounds(360, 20, 35,30);  
 p1.add(phone);  
 //more option  
 ImageIcon i13= new ImageIcon(ClassLoader.*getSystemResource*("CHatapppicture/3icon.png"));  
 Image i14 = i13.getImage().getScaledInstance(10,25,Image.*SCALE\_DEFAULT*);  
 ImageIcon i15 = new ImageIcon(i14);  
 JLabel option = new JLabel(i15);  
 option.setBounds(420, 20, 10,25);  
 p1.add(option);  
 //display name  
 JLabel name = new JLabel("Munnabhai");  
 name.setBounds(110,20,120,15);  
 name.setForeground(Color.*WHITE*);  
 name.setFont(new Font("SAN\_SERIF",Font.*BOLD*,19));  
 p1.add(name);  
 // status online or offline with a green dot  
 JPanel statusPanel = new JPanel(new FlowLayout(FlowLayout.*LEFT*));  
 statusPanel.setBounds(100, 35, 120, 20);  
 statusPanel.setOpaque(false);  
  
 // green dot  
 JLabel statusDot;  
 statusDot = new JLabel(" ●");  
 statusDot.setForeground(new Color(37, 211, 102));  
 statusDot.setFont(new Font("SAN\_SERIF", Font.*BOLD*, 13));  
 statusPanel.add(statusDot);  
 JLabel status = new JLabel("Active Now");  
 status.setForeground(Color.*WHITE*);  
 status.setFont(new Font("SAN\_SERIF", Font.*BOLD*, 13));  
 statusPanel.add(status);  
 p1.add(statusPanel);  
  
 // writing boundry  
 *a1* = new JPanel();  
 *a1*.setBounds(5,75, 440, 570);  
 *frame*.add(*a1*);  
 //footer  
 //textfield  
 text= new JTextField();  
 text.setBounds(5, 655, 310, 40);  
 text.setFont(new Font ("SAN\_SERIF", Font.*PLAIN*, 16));  
 *frame*.add(text);  
 //send button  
 JButton send = new JButton("Send");  
 send.setBounds(320,655,123,40);  
 send.setBackground(new Color(7,94,84));  
 send.setForeground(Color.*WHITE*);  
 send.addActionListener(this);  
 send.setFont(new Font ("SAN\_SERIF", Font.*PLAIN*, 16));  
 *frame*.add(send);  
  
  
 *frame*.setSize(450, 700);  
 *frame*.setLocation(200,50);  
 *frame*.setUndecorated(true);  
 *frame*.getContentPane().setBackground(Color.*WHITE*);  
  
 *frame*.setVisible(true);  
 }  
 private void createTable() throws SQLException {  
 String createTableSQL = "CREATE TABLE IF NOT EXISTS messages(" +  
 "id INT AUTO\_INCREMENT PRIMARY KEY," +  
 " Sender VARCHAR(30)," +  
 "message VARCHAR(255) NOT NULL," +  
 "timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP)";  
 try (Statement statement = *connection*.createStatement()) {  
 statement.executeUpdate(createTableSQL);  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 throw new RuntimeException("Failed to create table");  
 }  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent ae) {  
  
 try{  
 String sender = "Munnabhai";  
 String out= text.getText();  
 JPanel p2 = *formatLabel*(out);  
 //saves messages to the databases  
 *saveMessageToDatabase*(sender,out);  
 *a1*.setLayout(new BorderLayout());  
  
 JPanel right = new JPanel(new BorderLayout());  
 right.add(p2, BorderLayout.*LINE\_END*);  
 *vertical*.add(right);  
 *vertical*.add(Box.*createVerticalStrut*(15));  
 *a1*.add(*vertical*,BorderLayout.*PAGE\_START*);  
 *dout*.writeUTF(out);  
 text.setText("");  
  
 *frame*.repaint();  
 *frame*.invalidate();  
 *frame*.validate();}  
 catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
 private static void saveMessageToDatabase(String sender,String message) {  
 String sql = "INSERT INTO messages(sender,message, timestamp) VALUES (?,?, NOW())";  
 try (PreparedStatement statement = *connection*.prepareStatement(sql)) {  
 statement.setString(1,sender);  
 statement.setString(2, message);  
 // Set autocommit to false  
 *connection*.setAutoCommit(false);  
 statement.executeUpdate();  
 *connection*.commit();  
 *connection*.setAutoCommit(true);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 *rollback*();  
 throw new RuntimeException("Failed to save message to the database");  
 }  
 }  
  
 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: 150 px\">"+ out + "</p></html>");  
 output.setFont(new Font("Tahoma", Font.*PLAIN*,16));  
 output.setBackground(new Color(37,211,102));  
 output.setOpaque(true);  
 output.setBorder(new EmptyBorder(15,15,15,50));  
  
  
 panel.add(output);  
  
 Calendar cal = Calendar.*getInstance*();  
 SimpleDateFormat sdf= new SimpleDateFormat("HH:mm");  
  
 JLabel time = new JLabel();  
 time.setText(sdf.format(cal.getTime()));  
 panel.add(time);  
 return panel;  
 }  
  
  
 private static void rollback() {  
 try {  
 if (*connection* != null) {  
 *connection*.rollback();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
  
  
 //main class  
 public static void main(String[] args){  
  
 new Server();  
 try {  
 ServerSocket skt = new ServerSocket(6001);  
 System.*out*.println("connected to the server");  
 while(true){  
 Socket s = skt.accept();  
 DataInputStream din = new DataInputStream(s.getInputStream());  
 *dout* = new DataOutputStream(s.getOutputStream());  
  
 while(true){  
 String msg = din.readUTF();  
 //saveMessageToDatabase(msg);  
  
 JPanel panel = *formatLabel*(msg);  
 JPanel left = new JPanel(new BorderLayout());  
 left.add(panel,BorderLayout.*LINE\_START*);  
 *vertical*.add(left);  
 *frame*.validate();  
  
 }  
 }  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
  
 }  
  
}