**OPEN LAB – III**

**JAVA Project**

CB.SC.I5DAS21065

UTHAYARAHAVAN M S

* **IDE VERSION:**

Apache Netbeans IDE 22

* **SERVER WITH VERSION:**

PostgreSQL 15

* **JAVA VERSION:**

Java 22.0.1 (JDK 22)

* **CONNECTOR:**

PostgreSQL 42.5.4.jar

* **DESIGN:** A screenshot of a login page

  Description automatically generatedA screenshot of a login page

  Description automatically generated

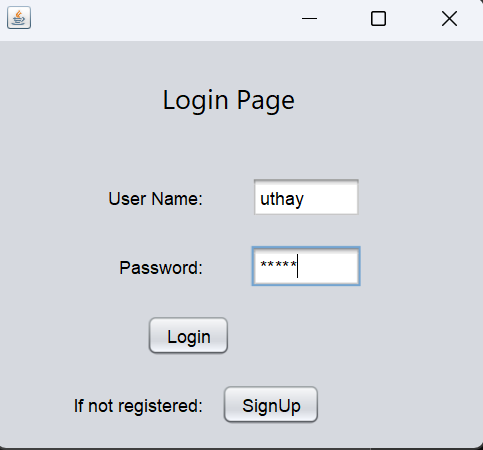
A screenshot of a computer

Description automatically generated

* **OUTPUT:**

A screenshot of a login page

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA computer screen shot of a message

Description automatically generatedA screenshot of a computer

Description automatically generated

**DATABASE DESIGN:**

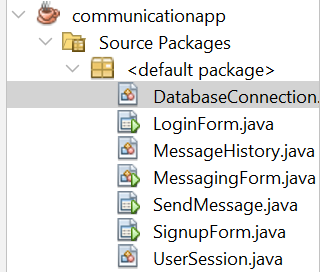
**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

**CODE:**

****

**DatabaseConnection.java:**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

private static final String URL = "jdbc:postgresql://localhost:5432/communication\_db"; // Replace with your database URL

private static final String USER = "postgres"; // Replace with your PostgreSQL username

private static final String PASSWORD = "0720"; // Replace with your PostgreSQL password

public static Connection getConnection() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

}

**LoginForm.java:**

import java.awt.Dimension;

import java.awt.Toolkit;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

/\*\*

\*

\* @author uthay

\*/

public class LoginForm extends javax.swing.JFrame {

/\*\*

\* Creates new form LoginForm

\*/

public LoginForm() {

initComponents();

centerFrame();

}

private void centerFrame() {

// Get the screen size

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

// Calculate the x and y coordinates for the frame to be centered

int x = (screenSize.width - getWidth()) / 2;

int y = (screenSize.height - getHeight()) / 2;

setLocation(x, y); // Set the location

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

Username = new javax.swing.JTextField();

Password = new javax.swing.JPasswordField();

Login = new javax.swing.JButton();

SignUp = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

jLabel1.setText("Login Page");

jLabel2.setText("User Name:");

jLabel3.setText("Password:");

Login.setText("Login");

Login.addActionListener(new java.awt.event.ActionListener() {

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

LoginActionPerformed(evt);

}

});

SignUp.setText("SignUp");

SignUp.addActionListener(new java.awt.event.ActionListener() {

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

SignUpActionPerformed(evt);

}

});

jLabel4.setText("If not registered:");

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(100, 100, 100)

.addComponent(Login))

.addGroup(layout.createSequentialGroup()

.addGap(52, 52, 52)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

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

.addComponent(SignUp)

.addGap(27, 27, 27))

.addGroup(layout.createSequentialGroup()

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

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(32, 32, 32)

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

.addComponent(Password, javax.swing.GroupLayout.DEFAULT\_SIZE, 74, Short.MAX\_VALUE)

.addComponent(Username)))))

.addGroup(layout.createSequentialGroup()

.addGap(111, 111, 111)

.addComponent(jLabel1)))

.addContainerGap(82, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jLabel1)

.addGap(40, 40, 40)

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

.addComponent(jLabel2)

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

.addGap(18, 18, 18)

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

.addComponent(jLabel3)

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

.addGap(18, 18, 18)

.addComponent(Login)

.addGap(18, 18, 18)

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

.addComponent(SignUp)

.addComponent(jLabel4))

.addContainerGap(16, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void SignUpActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

new SignupForm().setVisible(true);

}

private void LoginActionPerformed(java.awt.event.ActionEvent evt) {

String username = Username.getText(); // Assuming Username is your JTextField for the username

String password = new String(Password.getPassword()); // Assuming Password is your JPasswordField for the password

// Input validation

if (username.isEmpty() || password.isEmpty()) {

JOptionPane.showMessageDialog(this, "Username or password cannot be empty", "Error", JOptionPane.ERROR\_MESSAGE);

return; // Stop further execution if either field is empty

}

// SQL query to check if the user exists in the database

String query = "SELECT \* FROM users WHERE username = ? AND password = ?";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement pstmt = conn.prepareStatement(query)) {

// Setting the parameters in the prepared statement

pstmt.setString(1, username);

pstmt.setString(2, password);

// Executing the query

ResultSet rs = pstmt.executeQuery();

// If a matching record is found, login is successful

if (rs.next()) {

JOptionPane.showMessageDialog(this, "Login successful!");

// Retrieve the user\_id of the logged-in user

int userId = rs.getInt("user\_id");

UserSession.currentUserId = userId; // Set currentUserId to maintain session

// Close the login window and open the messaging form

this.dispose();

new MessagingForm().setVisible(true); // Open the messaging form

} else {

// Show an error message if username or password is incorrect

JOptionPane.showMessageDialog(this, "Invalid username or password", "Error", JOptionPane.ERROR\_MESSAGE);

}

} catch (SQLException ex) {

ex.printStackTrace(); // Print exception details in the console

JOptionPane.showMessageDialog(this, "Database error: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new LoginForm().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Login;

private javax.swing.JPasswordField Password;

private javax.swing.JButton SignUp;

private javax.swing.JTextField Username;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

// End of variables declaration

}

**SignupForm.java:**

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import java.awt.Dimension;

import java.awt.Toolkit;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

/\*\*

\*

\* @author uthay

\*/

public class SignupForm extends javax.swing.JFrame {

/\*\*

\* Creates new form SignupForm

\*/

public SignupForm() {

initComponents();

centerFrame();

}

private void centerFrame() {

// Get the screen size

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

// Calculate the x and y coordinates for the frame to be centered

int x = (screenSize.width - getWidth()) / 2;

int y = (screenSize.height - getHeight()) / 2;

setLocation(x, y);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

usernameField = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

passwordField = new javax.swing.JPasswordField();

Loginbutton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("User Name:");

jLabel2.setText("Password:");

jLabel3.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

jLabel3.setText("Signup Page");

jButton1.setText("Register");

jButton1.addActionListener(new java.awt.event.ActionListener() {

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

jButton1ActionPerformed(evt);

}

});

passwordField.addActionListener(new java.awt.event.ActionListener() {

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

passwordFieldActionPerformed(evt);

}

});

Loginbutton.setText("Login");

Loginbutton.addActionListener(new java.awt.event.ActionListener() {

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

LoginbuttonActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(31, 31, 31)

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

.addGroup(layout.createSequentialGroup()

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

.addComponent(jLabel1)

.addComponent(jLabel2))

.addGap(29, 29, 29)

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

.addComponent(usernameField)

.addComponent(passwordField, javax.swing.GroupLayout.DEFAULT\_SIZE, 94, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(8, 8, 8)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 32, Short.MAX\_VALUE)

.addComponent(Loginbutton))))

.addGroup(layout.createSequentialGroup()

.addGap(97, 97, 97)

.addComponent(jLabel3)))

.addContainerGap(76, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(34, 34, 34)

.addComponent(jLabel3)

.addGap(41, 41, 41)

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

.addComponent(jLabel1)

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

.addGap(18, 18, 18)

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

.addComponent(jLabel2)

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

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

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

.addComponent(jButton1)

.addComponent(Loginbutton))

.addContainerGap(44, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (username.isEmpty() || password.isEmpty()) {

JOptionPane.showMessageDialog(this, "Username or password cannot be empty", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

String query = "INSERT INTO users (username, password) VALUES (?, ?)";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setString(1, username);

pstmt.setString(2, password);

pstmt.executeUpdate();

JOptionPane.showMessageDialog(this, "User registered successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error registering user", "Error", JOptionPane.ERROR\_MESSAGE);

}

// TODO add your handling code here:

}

private void passwordFieldActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void LoginbuttonActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

new LoginForm().setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(SignupForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(SignupForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(SignupForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SignupForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new SignupForm().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Loginbutton;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPasswordField passwordField;

private javax.swing.JTextField usernameField;

// End of variables declaration

}

**MessagingForm.java:**

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

// Custom TableModel to prevent editing

class NonEditableTableModel extends DefaultTableModel {

// Constructor that takes column names and row count

public NonEditableTableModel(Object[] columnNames, int rowCount) {

super(columnNames, rowCount);

}

@Override

public boolean isCellEditable(int row, int column) {

return false; // All cells are non-editable

}

}

public class MessagingForm extends JFrame {

private JLabel loggedInUserLabel;

private JButton composeMessageButton;

private JButton logoutButton;

private JTable previousMessagesTable;

private NonEditableTableModel tableModel; // Use custom model

public MessagingForm() {

initComponents();

displayLoggedInUser();

loadPreviousMessages();

}

private void displayLoggedInUser() {

String username = UserSession.getCurrentUsername();

loggedInUserLabel.setText("Logged in as: " + (username != null ? username : "Unknown User"));

}

private void initComponents() {

setTitle("Messaging Application");

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

// User Label

loggedInUserLabel = new JLabel("Logged in as: ");

add(loggedInUserLabel);

// Compose Message Button

composeMessageButton = new JButton("Compose Message");

composeMessageButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

openSendMessageForm();

}

});

add(composeMessageButton);

// Previous Messages Label

JLabel previousMessagesLabel = new JLabel("Previous Messages:");

add(previousMessagesLabel);

// Table to display previous messages

tableModel = new NonEditableTableModel(new String[]{"Previous Users"}, 0);

previousMessagesTable = new JTable(tableModel);

previousMessagesTable.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

previousMessagesTable.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

openMessageHistory();

}

});

JScrollPane scrollPane = new JScrollPane(previousMessagesTable);

scrollPane.setPreferredSize(new Dimension(250, 150));

add(scrollPane);

// Logout Button

logoutButton = new JButton("Logout");

logoutButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

logout();

}

});

add(logoutButton);

// Set the frame size and make it visible

setSize(300, 300);

setLocationRelativeTo(null); // Center the frame

setVisible(true);

}

private void loadPreviousMessages() {

try (Connection conn = DatabaseConnection.getConnection()) {

String query = "SELECT DISTINCT u.username FROM messages m " +

"JOIN users u ON m.receiver\_id = u.user\_id " +

"WHERE m.sender\_id = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, UserSession.currentUserId);

ResultSet rs = pstmt.executeQuery();

// Clear existing data in the table model

tableModel.setRowCount(0);

// Populate the table with previous usernames

while (rs.next()) {

String username = rs.getString("username");

tableModel.addRow(new Object[]{username});

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error loading previous messages.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void openSendMessageForm() {

SwingUtilities.invokeLater(() -> {

new SendMessage().setVisible(true); // Open SendMessage form

// Uncomment this line if you want to close the MessagingForm

// dispose(); // Close MessagingForm if desired

});

}

private void openMessageHistory() {

int selectedRow = previousMessagesTable.getSelectedRow();

if (selectedRow != -1) { // Check if a row is selected

String selectedUsername = (String) tableModel.getValueAt(selectedRow, 0);

MessageHistory messageHistory = new MessageHistory(selectedUsername); // Create MessageHistory instance

messageHistory.setVisible(true);// Open MessageHistory for selected user

// Do not dispose of MessagingForm to keep it open

}

}

private void logout() {

UserSession.reset(); // Reset user session

new LoginForm().setVisible(true); // Open login page

dispose(); // Close MessagingForm

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new MessagingForm());

}

}

**MessageHistory.java:**

import javax.swing.\*;

import java.sql.\*;

public class MessageHistory extends JFrame {

private int userId; // This should store the current user's ID

private String selectedUser; // The username of the user whose messages are being viewed

private JTextArea messageArea;

public MessageHistory(String selectedUser) {

this.userId = UserSession.currentUserId; // Assuming UserSession holds the logged-in user's ID

this.selectedUser = selectedUser; // Set the selected username

initComponents();

loadMessages();

}

private void initComponents() {

messageArea = new JTextArea();

messageArea.setEditable(false); // Make the text area non-editable

JScrollPane scrollPane = new JScrollPane(messageArea); // Add a scroll pane for better usability

add(scrollPane);

setTitle("Message History with " + selectedUser);

setSize(400, 400);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the frame

}

private void loadMessages() {

try (Connection conn = DatabaseConnection.getConnection()) {

// Query to fetch messages between the current user and the selected user

String query = "SELECT u.username AS sender\_username, m.message, m.timestamp " +

"FROM messages m " +

"JOIN users u ON u.user\_id = m.sender\_id " +

"WHERE (m.sender\_id = ? AND m.receiver\_id = (SELECT user\_id FROM users WHERE username = ?)) " +

"OR (m.receiver\_id = ? AND m.sender\_id = (SELECT user\_id FROM users WHERE username = ?)) " +

"ORDER BY m.timestamp";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, userId); // Set the current user's ID as sender

pstmt.setString(2, selectedUser); // Set the selected username as receiver

pstmt.setInt(3, userId); // Set the current user's ID as receiver

pstmt.setString(4, selectedUser); // Set the selected username as sender

ResultSet rs = pstmt.executeQuery();

messageArea.setText(""); // Clear existing text

while (rs.next()) {

String senderUsername = rs.getString("sender\_username"); // Retrieve sender's username

String message = rs.getString("message");

String timestamp = rs.getString("timestamp");

// Format message with sender's username and timestamp

messageArea.append("[" + timestamp + "] " + senderUsername + ": " + message + "\n");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error loading messages.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

**SendMessages.java:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import java.sql.ResultSet;

/\*\*

\*

\* @author uthay

\*/

public class SendMessage extends javax.swing.JFrame {

public SendMessage() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

touser = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

textarea = new javax.swing.JTextArea();

Send = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("To User:");

textarea.setColumns(20);

textarea.setRows(5);

jScrollPane1.setViewportView(textarea);

Send.setText("Send");

Send.addActionListener(new java.awt.event.ActionListener() {

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

SendActionPerformed(evt);

}

});

jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N

jLabel2.setText("Send Message");

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addComponent(jLabel1)

.addGap(18, 18, 18)

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

.addComponent(jLabel2)

.addComponent(touser, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(34, 34, 34)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 266, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(124, 124, 124)

.addComponent(Send)))

.addContainerGap(56, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(20, 20, 20)

.addComponent(jLabel2)

.addGap(26, 26, 26)

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

.addComponent(jLabel1)

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

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(Send)

.addContainerGap(21, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void SendActionPerformed(java.awt.event.ActionEvent evt) {

String recipientUsername = touser.getText().trim();

String messageContent = textarea.getText().trim();

int senderId = UserSession.currentUserId; // Get the current user's ID

// Validate that both fields are not empty

if (recipientUsername.isEmpty() || messageContent.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please fill in both fields.", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

// Insert message into the database

try (Connection conn = DatabaseConnection.getConnection()) {

// Retrieve the recipient's user\_id based on their username

String queryRecipient = "SELECT user\_id FROM users WHERE username = ?";

try (PreparedStatement pstmtRecipient = conn.prepareStatement(queryRecipient)) {

pstmtRecipient.setString(1, recipientUsername);

try (ResultSet rsRecipient = pstmtRecipient.executeQuery()) {

if (rsRecipient.next()) {

int recipientId = rsRecipient.getInt("user\_id");

// Insert the message into the 'messages' table

String insertMessage = "INSERT INTO messages (sender\_id, receiver\_id, message, timestamp) VALUES (?, ?, ?, NOW())";

try (PreparedStatement pstmtMessage = conn.prepareStatement(insertMessage)) {

pstmtMessage.setInt(1, senderId); // Use current user's ID

pstmtMessage.setInt(2, recipientId);

pstmtMessage.setString(3, messageContent);

int rowsInserted = pstmtMessage.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(this, "Message sent successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);

this.dispose(); // Close the SendMessage window

} else {

JOptionPane.showMessageDialog(this, "Failed to send the message.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

} else {

JOptionPane.showMessageDialog(this, "Recipient not found.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "An error occurred while sending the message.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SendMessage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

/\* Create and display the form \*/

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

public void run() {

new SendMessage().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Send;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea textarea;

private javax.swing.JTextField touser;

// End of variables declaration

}

**UserSession.java:**

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class UserSession {

public static int currentUserId = 0; // Store the logged-in user's ID

static String currentUsername = null; // Cache the username after it is retrieved

// Method to get the username of the currently logged-in user

public static String getCurrentUsername() {

// If username is already cached, return it

if (currentUsername != null) {

return currentUsername;

}

// If no currentUserId is set, return null

if (currentUserId == 0) {

return null;

}

// Query the database to retrieve the username based on currentUserId

String query = "SELECT username FROM users WHERE user\_id = ?";

try (Connection conn = DatabaseConnection.getConnection();

PreparedStatement pstmt = conn.prepareStatement(query)) {

// Set the user\_id in the query

pstmt.setInt(1, currentUserId);

// Execute the query and get the result

ResultSet rs = pstmt.executeQuery();

if (rs.next()) {

// Retrieve the username from the result set

currentUsername = rs.getString("username");

return currentUsername;

} else {

// If no user is found with the currentUserId, return null

return null;

}

} catch (SQLException ex) {

ex.printStackTrace();

return null;

}

}

// Method to reset the session (on logout, for example)

public static void reset() {

currentUserId = 0;

currentUsername = null;

}

}