

```
1 package com.onyimbo.sufeeds;
2
3 import java.sql.*;
4 import java.util.logging.Level;
5 import java.util.logging.Logger;
6
7 public class Main {
8     static final String DRIVER = "com.mysql.cj.jdbc.
Driver";
9     static final String DATABASE_URL = "jdbc:mysql://
localhost:3306/db_daisy_onyimbo_189599";
10    static final String USER = "root";
11    static final String PASSWORD = "burgundy";
12
13    public static void main(String[] args) {
14        new loginreg();
15    }
16    /**
17     * Provides a reusable database connection.
18     *
19     * @return a Connection object
20     */
21    public static Connection getDatabaseConnection
() {
22        try {
23            // Load the driver
24            Class.forName(DRIVER);
25            // Establish connection
26            return DriverManager.getConnection(
DATABASE_URL, USER, PASSWORD);
27        } catch (ClassNotFoundException e) {
28            Logger.getLogger(Main.class.getName()).
log(Level.SEVERE, "Database driver not found", e);
29        } catch (SQLException e) {
30            Logger.getLogger(Main.class.getName()).
log(Level.SEVERE, "Database connection error", e);
31        }
32        return null;
33    }
34 }
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form"
/ version="1" bind-to-class="com.onymbo.sufeeds.
LOGIN">
3   <grid id="27dc6" binding="loginpanel" layout-
manager="GridLayoutManager" row-count="6" column-
count="5" same-size-horizontally="false" same-size-
vertically="false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties/>
9     <border type="none"/>
10    <children>
11      <component id="6bbd" class="javax.swing.JLabel"
binding="lblid">
12        <constraints>
13          <grid row="2" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
14        </constraints>
15        <properties>
16          <text value="Student ID"/>
17        </properties>
18      </component>
19      <component id="b478f" class="javax.swing.JLabel"
" binding="lblpassword">
20        <constraints>
21          <grid row="3" column="1" row-span="1" col-
span="2" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
22        </constraints>
23        <properties>
24          <text value="Password"/>
25        </properties>
26      </component>
27      <component id="f05b7" class="javax.swing.
JTextField" binding="textField1" default-binding=
true">
28        <constraints>
```

```
29      <grid row="2" column="3" row-span="1" col-
  span="1" vsize-policy="0" hsize-policy="6" anchor="8"
    fill="1" indent="0" use-parent-layout="false">
30          <preferred-size width="150" height="-1"/>
31      </grid>
32  </constraints>
33  <properties/>
34 </component>
35 <component id="9a389" class="javax.swing.
  JPasswordField" binding="passwordField1" default-
  binding="true">
36      <constraints>
37          <grid row="3" column="3" row-span="1" col-
  span="1" vsize-policy="0" hsize-policy="6" anchor="8"
    fill="1" indent="0" use-parent-layout="false">
38          <preferred-size width="150" height="-1"/>
39      </grid>
40  </constraints>
41  <properties/>
42 </component>
43 <vspacer id="d1f65">
44   <constraints>
45       <grid row="0" column="3" row-span="1" col-
  span="1" vsize-policy="6" hsize-policy="1" anchor="0"
    fill="2" indent="0" use-parent-layout="false"/>
46   </constraints>
47 </vspacer>
48 <vspacer id="93375">
49   <constraints>
50       <grid row="5" column="3" row-span="1" col-
  span="1" vsize-policy="6" hsize-policy="1" anchor="0"
    fill="2" indent="0" use-parent-layout="false"/>
51   </constraints>
52 </vspacer>
53 <hspacer id="94970">
54   <constraints>
55       <grid row="2" column="0" row-span="1" col-
  span="1" vsize-policy="1" hsize-policy="6" anchor="0"
    fill="1" indent="0" use-parent-layout="false"/>
56   </constraints>
57 </hspacer>
```

```
58      <hspacer id="d678a">
59          <constraints>
60              <grid row="3" column="4" row-span="1" col-
61                  span="1" vsize-policy="1" hsize-policy="6" anchor="0"
62                  fill="1" indent="0" use-parent-layout="false"/>
63          </constraints>
64      </hspacer>
65      <component id="89860" class="javax.swing.JLabel"
66          " binding="lblLogin">
67          <constraints>
68              <grid row="1" column="3" row-span="1" col-
69                  span="1" vsize-policy="0" hsize-policy="0" anchor="0"
70                  fill="0" indent="0" use-parent-layout="false"/>
71          </constraints>
72          <properties>
73              <text value="LOGIN"/>
74          </properties>
75      </component>
76      <component id="2c466" class="javax.swing.
77          JButton" binding="btnsubmit">
78          <constraints>
79              <grid row="4" column="3" row-span="1" col-
80                  span="1" vsize-policy="0" hsize-policy="3" anchor="0"
81                  fill="1" indent="0" use-parent-layout="false"/>
82          </constraints>
83          <properties>
84              <text value="Submit"/>
85          </properties>
86      </component>
87      </children>
88  </grid>
89 </form>
90
91
```

```
1 package com.onyimbo.sufeeds;
2
3 import javax.swing.*;
4 import java.sql.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 public class LOGIN extends JFrame{
9     private JTextField textField1;
10    private JPasswordField passwordField1;
11    private JLabel lblLogin;
12    private JLabel lblid;
13    private JLabel lblpassword;
14    private JPanel loginpanel;
15    private JButton btnsubmit;
16
17    public LOGIN(){
18        super("Login Page");
19
20        setContentPane(loginpanel);
21        setSize(500, 500);
22        setVisible(true);
23        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
);
24
25
26        btnsubmit.addActionListener(new
ActionListener() {
27            @Override
28            public void actionPerformed(ActionEvent e
) {
29                handleLogin();
30
31            }
32        });
33    }
34    /**
35     * Handles login logic when the user clicks the
Submit button.
36     */
37    private void handleLogin() {
```

```

38         String userId = textField1.getText();
39         String password = new String(passwordField1.
40             getPassword());
41         if (userId.isEmpty() || password.isEmpty()) {
42             JOptionPane.showMessageDialog(this, "
Please enter both ID and password.", "Input Error",
JOptionPane.ERROR_MESSAGE);
43             return;
44         }
45
46         // Check if the user exists in the database
47         if (checkUserExists(userId, password)) {
48             // Redirect to mainTab
49             JOptionPane.showMessageDialog(this, "
Login successful!");
50             new mainTab(userId).setVisible(true);
51             dispose(); // Close the login page
52         } else {
53             // Redirect to register page
54             JOptionPane.showMessageDialog(this, "User
not found. Redirecting to registration page.");
55             new REGISTER().setVisible(true);
56             dispose(); // Close the login page
57         }
58     }
59
60     /**
61      * Checks if the user exists in the database.
62      *
63      * @param userId The user ID entered.
64      * @param password The password entered.
65      * @return true if the user exists, false
otherwise.
66      */
67     private boolean checkUserExists(String userId,
String password) {
68         String query = "SELECT * FROM tbl_students
WHERE student_id = ? AND password_ = ?";
69         try (Connection connection = Main.
getDatabaseConnection());

```

```
70             PreparedStatement preparedStatement =
    connection.prepareStatement(query)) {
71
72                 // Set parameters for the query
73                 preparedStatement.setString(1, userId);
74                 preparedStatement.setString(2, password
    );
75
76                 // Execute the query
77                 ResultSet resultSet = preparedStatement.
executeQuery();
78                 return resultSet.next(); // Returns true
    if a matching record is found
79             } catch (Exception ex) {
80                 JOptionPane.showMessageDialog(this, "
Database error: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
81                 ex.printStackTrace();
82             }
83             return false;
84         }
85     }
86
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form"
/ version="1" bind-to-class="com.onymbo.sufeeds.
mainTab">
3   <grid id="27dc6" binding="maintabpanel" layout-
manager="GridLayoutManager" row-count="9" column-
count="6" same-size-horizontally="false" same-size-
vertically="false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties/>
9     <border type="none"/>
10    <children>
11      <component id="aeeb8" class="javax.swing.JLabel"
" binding="lblfeedback">
12        <constraints>
13          <grid row="6" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
14        </constraints>
15        <properties>
16          <text value="Feedback"/>
17        </properties>
18      </component>
19      <component id="eb627" class="javax.swing.JLabel"
" binding="lbltopic">
20        <constraints>
21          <grid row="4" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
22        </constraints>
23        <properties>
24          <text value="Topic"/>
25        </properties>
26      </component>
27      <component id="dabc6" class="javax.swing.
JTextField" binding="textField2" default-binding=
true">
28        <constraints>
```

```
29      <grid row="4" column="2" row-span="1" col-
  span="3" vsize-policy="0" hsize-policy="6" anchor="8"
    fill="1" indent="0" use-parent-layout="false">
30          <preferred-size width="150" height="-1"/>
31      </grid>
32  </constraints>
33  <properties/>
34  </component>
35  <component id="4a92e" class="javax.swing.
  JTextField" binding="textField3" default-binding="
  true">
36      <constraints>
37          <grid row="6" column="2" row-span="1" col-
  span="3" vsize-policy="0" hsize-policy="6" anchor="8"
    fill="1" indent="0" use-parent-layout="false">
38          <preferred-size width="150" height="-1"/>
39      </grid>
40  </constraints>
41  <properties/>
42  </component>
43  <hspacer id="99d0c">
44      <constraints>
45          <grid row="6" column="0" row-span="1" col-
  span="1" vsize-policy="1" hsize-policy="6" anchor="0"
    fill="1" indent="0" use-parent-layout="false"/>
46      </constraints>
47  </hspacer>
48  <component id="8f953" class="javax.swing.JLabel
  " binding="lblclass">
49      <constraints>
50          <grid row="2" column="1" row-span="1" col-
  span="1" vsize-policy="0" hsize-policy="0" anchor="8"
    fill="0" indent="0" use-parent-layout="false"/>
51      </constraints>
52      <properties>
53          <text value="Class"/>
54      </properties>
55  </component>
56  <vspacer id="263da">
57      <constraints>
58          <grid row="0" column="2" row-span="1" col-
```

```
58 span="1" vsize-policy="6" hsize-policy="1" anchor="0"
      fill="2" indent="0" use-parent-layout="false"/>
59         </constraints>
60     </vspacer>
61     <vspacer id="b9f4e">
62         <constraints>
63             <grid row="8" column="2" row-span="1" col-
64             span="1" vsize-policy="6" hsize-policy="1" anchor="0"
65             fill="2" indent="0" use-parent-layout="false"/>
66         </constraints>
67     </vspacer>
68     <component id="c4f22" class="javax.swing.
69         JButton" binding="btnupdate">
70         <constraints>
71             <grid row="7" column="3" row-span="1" col-
72             span="1" vsize-policy="0" hsize-policy="3" anchor="0"
73             fill="1" indent="0" use-parent-layout="false"/>
74         </constraints>
75         <properties>
76             <text value="Update"/>
77         </properties>
78     </component>
79     <component id="bf3a7" class="javax.swing.
80         JButton" binding="btndelete">
81         <constraints>
82             <grid row="7" column="4" row-span="1" col-
83             span="1" vsize-policy="0" hsize-policy="3" anchor="0"
84             fill="1" indent="0" use-parent-layout="false"/>
85         </constraints>
86         <properties>
87             <text value="Delete"/>
88         </properties>
89     </component>
90     <component id="db4c8" class="javax.swing.
91         JTextField" binding="textField1" default-binding="
92             true">
93         <constraints>
94             <grid row="2" column="2" row-span="1" col-
95             span="3" vsize-policy="0" hsize-policy="6" anchor="8"
96             fill="1" indent="0" use-parent-layout="false">
97                 <preferred-size width="150" height="-1"/>
```

```
86          </grid>
87      </constraints>
88      <properties/>
89  </component>
90  <component id="3dd6e" class="javax.swing.
JButton" binding="btnread">
91      <constraints>
92          <grid row="7" column="2" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="3" anchor="0
" fill="1" indent="0" use-parent-layout="false"/>
93      </constraints>
94      <properties>
95          <text value="Read"/>
96      </properties>
97  </component>
98  <component id="a0780" class="javax.swing.
JButton" binding="btncreate">
99      <constraints>
100         <grid row="7" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="3" anchor="0
" fill="1" indent="0" use-parent-layout="false"/>
101     </constraints>
102     <properties>
103         <text value="Create"/>
104     </properties>
105  </component>
106  <component id="373fa" class="javax.swing.
JLabel" binding="lblmaintab">
107      <constraints>
108          <grid row="1" column="2" row-span="1" col-
span="2" vsize-policy="0" hsize-policy="0" anchor="0
" fill="0" indent="0" use-parent-layout="false"/>
109     </constraints>
110     <properties>
111         <text value="MAIN TAB"/>
112     </properties>
113  </component>
114  <component id="274f4" class="javax.swing.
JLabel" binding="lblclassid">
115      <constraints>
116          <grid row="3" column="1" row-span="1" col-
```

```
116 span="1" vsize-policy="0" hsize-policy="0" anchor="8
    " fill="0" indent="0" use-parent-layout="false"/>
117     </constraints>
118     <properties>
119         <text value="Class ID"/>
120     </properties>
121     </component>
122     <component id="9e987" class="javax.swing.
    JTextField" binding="textField4" default-binding="
true">
123         <constraints>
124             <grid row="3" column="2" row-span="1" col-
span="3" vsize-policy="0" hsize-policy="6" anchor="8
    " fill="1" indent="0" use-parent-layout="false">
125                 <preferred-size width="150" height="-1"
    />
126             </grid>
127         </constraints>
128         <properties/>
129     </component>
130     <component id="fe722" class="javax.swing.
    JLabel" binding="lbltopicid">
131         <constraints>
132             <grid row="5" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="8
    " fill="0" indent="0" use-parent-layout="false"/>
133         </constraints>
134         <properties>
135             <text value="Topic ID"/>
136         </properties>
137     </component>
138     <component id="17fa4" class="javax.swing.
    JTextField" binding="textField5" default-binding="
true">
139         <constraints>
140             <grid row="5" column="2" row-span="1" col-
span="3" vsize-policy="0" hsize-policy="6" anchor="8
    " fill="1" indent="0" use-parent-layout="false">
141                 <preferred-size width="150" height="-1"
    />
142         </grid>
```

```
143      </constraints>
144      <properties/>
145      </component>
146      <hspacer id="eda33">
147          <constraints>
148              <grid row="2" column="5" row-span="1" col-
span="1" vsize-policy="1" hsize-policy="6" anchor="0
" fill="1" indent="0" use-parent-layout="false"/>
149          </constraints>
150      </hspacer>
151      </children>
152  </grid>
153 </form>
154
```

```
1 package com.onyimbo.sufeeds;
2
3 import javax.swing.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.sql.*;
7
8 public class mainTab extends JFrame{
9     private JTextField textField2;
10    private JTextField textField3;
11    private JLabel lbltopic;
12    private JLabel lblfeedback;
13    private JLabel lblclass;
14    private JTextField textField1;
15    private JLabel lblmaintab;
16    private JButton btncreate;
17    private JPanel maintabpanel;
18    private JButton btnupdate;
19    private JButton btndelete;
20    private JButton btnread;
21    private JTextField textField4;
22    private JTextField textField5;
23    private JLabel lblclassid;
24    private JLabel lbltopicid;
25
26    private final String loggedInUser;
27
28    public mainTab(String userId) {
29        super("The Main Tab");
30
31        this.loggedInUser = userId;
32
33        setContentPane(maintabpanel);
34        setSize(500, 500);
35        setVisible(true);
36        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE
37    );
38        btncreate.addActionListener(new
39            ActionListener() {
39                @Override
```

```
40         public void actionPerformed(ActionEvent e
41     ) {
42             createFormData();
43         }
44     });
45
46     btnread.addActionListener(new ActionListener()
47     ) {
48         @Override
49         public void actionPerformed(ActionEvent e
50     ) {
51         readData();
52     });
53
54     btnupdate.addActionListener(new
55     ActionListener() {
56         @Override
57         public void actionPerformed(ActionEvent e
58     ) {
59         updateData();
60     });
61
62     btndelete.addActionListener(new
63     ActionListener() {
64         @Override
65         public void actionPerformed(ActionEvent e
66     ) {
67         deleteData();
68     });
69
70     /**
71      * Saves form data to the database.
72      */
73     private void createFormData() {
74         String className = textField1.getText();
75         String classId = textField4.getText();
76         String topicName = textField2.getText();
77         String topicId = textField5.getText();
```

```

74         String feedback = textField3.getText();
75
76         if (className.isEmpty() || classId.isEmpty()
77             () || topicName.isEmpty() || topicId.isEmpty() || feedback.isEmpty()) {
77             JOptionPane.showMessageDialog(this, "All
78             fields are required.", "Input Error", JOptionPane.
79             ERROR_MESSAGE);
80
81         String insertClassQuery = "INSERT INTO
82             tbl_classes (class_id ,class_name, student_id)
83             VALUES (?, ?, ?) " + "ON DUPLICATE KEY UPDATE
84             class_name = VALUES(class_name)";
85
85         try (Connection connection = Main.
86             getDatabaseConnection()) {
87             // Insert class
88             try (PreparedStatement ps = connection.
89                 prepareStatement(insertClassQuery, PreparedStatement
90                     .RETURN_GENERATED_KEYS)) {
91                 ps.setString(1, classId);
92                 ps.setString(2, className);
93                 ps.setString(3, loggedInUser);
94                 ps.executeUpdate();
95             }
96
97             // Insert topic
98             try (PreparedStatement ps = connection.
99                 prepareStatement(insertTopicQuery, PreparedStatement
100                     .RETURN_GENERATED_KEYS)) {
101                 ps.setString(1, topicId);
102                 ps.setString(2, topicName);
103             }
104         }
105     }
106 }
```

```
98                 ps.setString(3, classId);
99                 ps.executeUpdate();
100            }
101
102            // Insert feedback
103            try (PreparedStatement ps = connection.
104                 prepareStatement(insertFeedbackQuery)) {
105                 ps.setString(1, feedback);
106                 ps.setString(2, topicId);
107                 ps.executeUpdate();
108            }
109            JOptionPane.showMessageDialog(this, "
110                Data saved successfully!");
111            clearFields();
112        } catch (SQLException ex) {
113            JOptionPane.showMessageDialog(this, "
114                Error saving data: " + ex.getMessage(), "Database
115                Error", JOptionPane.ERROR_MESSAGE);
116            ex.printStackTrace();
117        }
118    }
119
120    /**
121     * Reads data from the database.
122     */
123    private void readData() {
124        String className = textField1.getText();
125        if (className.isEmpty()) {
126            JOptionPane.showMessageDialog(this, "
127                Class name is required to read data.", "Input Error"
128                , JOptionPane.ERROR_MESSAGE);
129        }
130
131        String query = "SELECT c.class_id, c.
132            class_name, t.topic_id, t.topic_name, f.
133            feedback_text " +
134                "FROM tbl_classes c " +
135                "JOIN tbl_topics t ON c.class_id = t
136                .class_id " +
```

```

130          "JOIN tbl_feedback f ON t.topic_id
131      = f.topic_id " +
132          "WHERE c.class_name = ?";
133
134      try (Connection connection = Main.
135          getDatabaseConnection();
136          PreparedStatement ps = connection.
137          prepareStatement(query)) {
138          ps.setString(1, className);
139
140          try (ResultSet rs = ps.executeQuery()) {
141              StringBuilder result = new
142                  StringBuilder();
143
144              boolean dataFound = false;
145              while (rs.next()) {
146                  dataFound = true;
147                  String classId = rs.getString(
148                      "class_id");
149                  String topicId = rs.getString(
150                      "topic_id");
151                  String topicName = rs.getString(
152                      "topic_name");
153                  String feedback = rs.getString(
154                      "feedback_text");
155
156                  // Build a readable string for
157                  // the class, topic, and feedback
158                  result.append("Class ID: ").
159                  append(classId)
160                      .append(", Topic ID: ").
161                  append(topicId)
162                      .append(", Topic: ").
163                  append(topicName)
164                      .append(", Feedback: ").
165                  append(feedback)
166                      .append("\n");
167
168              }
169
170              if (dataFound) {
171                  // Display all results in a

```

```

157 dialog box
158         JOptionPane.showMessageDialog(
159             this, result.toString(), "Data Found", JOptionPane.
160             INFORMATION_MESSAGE);
161         } else {
162             JOptionPane.showMessageDialog(
163                 this, "No data found for the given class name.", "
164                 Not Found", JOptionPane.INFORMATION_MESSAGE);
165         }
166     }
167 }
168
169 /**
170 * Updates records in the database.
171 */
172 private void updateData() {
173     String classId = textField4.getText();
174     String topicId = textField5.getText();
175     String topicName = textField2.getText();
176     String feedback = textField3.getText();
177
178     if (classId.isEmpty() || topicId.isEmpty()
179         () || topicName.isEmpty() || feedback.isEmpty()) {
180         JOptionPane.showMessageDialog(this, "All
181             fields are required for update.", "Input Error",
182             JOptionPane.ERROR_MESSAGE);
183         return;
184     }
185
186     String updateTopicQuery = "UPDATE tbl_topics
187         SET topic_name = ? WHERE topic_id = ?";
188     String updateFeedbackQuery = "UPDATE
189         tbl_feedback SET feedback_text = ? WHERE topic_id
190         = ?";
191
192     try {
193         Statement statement = connection.createStatement();
194         statement.executeUpdate(updateTopicQuery);
195         statement.executeUpdate(updateFeedbackQuery);
196     } catch (SQLException ex) {
197         JOptionPane.showMessageDialog(this, "
198             Error reading data: " + ex.getMessage(), "Database
199             Error", JOptionPane.ERROR_MESSAGE);
200         ex.printStackTrace();
201     }
202 }

```

```
186     try (Connection connection = Main.  
187          getDatabaseConnection()) {  
188         // Update topic  
189         try (PreparedStatement ps = connection.  
190             prepareStatement(updateTopicQuery)) {  
191             ps.setString(1, topicName);  
192             ps.setString(2, topicId);  
193             ps.executeUpdate();  
194         }  
195         // Update feedback  
196         try (PreparedStatement ps = connection.  
197             prepareStatement(updateFeedbackQuery)) {  
198             ps.setString(1, feedback);  
199             ps.setString(2, topicId);  
200             ps.executeUpdate();  
201         }  
202         JOptionPane.showMessageDialog(this, "  
Data updated successfully!");  
203     } catch (SQLException ex) {  
204         JOptionPane.showMessageDialog(this, "  
Error updating data: " + ex.getMessage(), "Database  
Error", JOptionPane.ERROR_MESSAGE);  
205         ex.printStackTrace();  
206     }  
207  
208     /**  
209      * Deletes records from the database.  
210      */  
211     private void deleteData() {  
212         String classId = textField4.getText();  
213  
214         if (classId.isEmpty()) {  
215             JOptionPane.showMessageDialog(this, "  
Class ID is required for deletion.", "Input Error",  
JOptionPane.ERROR_MESSAGE);  
216             return;  
217         }  
218     }
```

```

219         String deleteQuery = "DELETE f, t, c " +
220                     "FROM tbl_classes c " +
221                     "JOIN tbl_topics t ON c.class_id = t
222                         .class_id " +
223                         "JOIN tbl_feedback f ON t.topic_id
224                             = f.topic_id " +
225                             "WHERE c.class_id = ?";
226
227         try (Connection connection = Main.
228             getDatabaseConnection();
229             PreparedStatement ps = connection.
230             prepareStatement(deleteQuery)) {
231             ps.setString(1, classId);
232
233             int rowsDeleted = ps.executeUpdate();
234             if (rowsDeleted > 0) {
235                 JOptionPane.showMessageDialog(this,
236                     "Data deleted successfully!");
237                 clearFields();
238             } else {
239                 JOptionPane.showMessageDialog(this,
240                     "No data found for the given Class ID.", "Not Found"
241                     , JOptionPane.INFORMATION_MESSAGE);
242             }
243         } catch (SQLException ex) {
244             JOptionPane.showMessageDialog(this, "
245                 Error deleting data: " + ex.getMessage(), "Database
246                 Error", JOptionPane.ERROR_MESSAGE);
247             ex.printStackTrace();
248         }
249     }
250
251     /**
252      * Clears all input fields.
253      */
254     private void clearFields() {
255         textField1.setText("");
256         textField2.setText("");
257         textField3.setText("");
258         textField4.setText("");
259         textField5.setText("");

```

```
251      }
252  }
253
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form"
/ version="1" bind-to-class="com.onymbo.sufeeds.
loginreg">
3   <grid id="27dc6" binding="loginregpanel" layout-
manager="GridLayoutManager" row-count="4" column-
count="3" same-size-horizontally="false" same-size-
vertically="false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties/>
9     <border type="none"/>
10    <children>
11      <component id="20d60" class="javax.swing.
JButton" binding="btnlogin">
12        <constraints>
13          <grid row="1" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="3" anchor="0"
fill="1" indent="0" use-parent-layout="false"/>
14        </constraints>
15        <properties>
16          <text value="LOGIN"/>
17        </properties>
18      </component>
19      <component id="e89c3" class="javax.swing.
JButton" binding="btnreg">
20        <constraints>
21          <grid row="2" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="3" anchor="0"
fill="1" indent="0" use-parent-layout="false"/>
22        </constraints>
23        <properties>
24          <text value="REGISTER"/>
25        </properties>
26      </component>
27      <vspacer id="8f1b6">
28        <constraints>
29          <grid row="0" column="1" row-span="1" col-
span="1" vsize-policy="6" hsize-policy="1" anchor="0"
```

```
29  fill="2" indent="0" use-parent-layout="false"/>/
30      </constraints>
31      </vspacer>
32      <vspacer id="14029">
33          <constraints>
34              <grid row="3" column="1" row-span="1" col-
span="1" vsize-policy="6" hsize-policy="1" anchor="0"
fill="2" indent="0" use-parent-layout="false"/>/
35          </constraints>
36      </vspacer>
37      <hspacer id="faa08">
38          <constraints>
39              <grid row="1" column="0" row-span="1" col-
span="1" vsize-policy="1" hsize-policy="6" anchor="0"
fill="1" indent="0" use-parent-layout="false"/>/
40          </constraints>
41      </hspacer>
42      <hspacer id="df755">
43          <constraints>
44              <grid row="2" column="2" row-span="1" col-
span="1" vsize-policy="1" hsize-policy="6" anchor="0"
fill="1" indent="0" use-parent-layout="false"/>/
45          </constraints>
46      </hspacer>
47      </children>
48  </grid>
49 </form>
50
```

```
1 package com.onyimbo.sufeeds;
2
3 import javax.swing.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 public class loginreg extends JFrame {
8     private JPanel loginregpanel;
9     private JButton btnlogin;
10    private JButton btnreg;
11
12    public loginreg() {
13
14        super("Login/Register");
15
16        setContentPane(loginregpanel);
17        setSize(500, 500);
18        setVisible(true);
19
20        btnlogin.addActionListener(new ActionListener
() {
21            @Override
22            public void actionPerformed(ActionEvent e
) {
23                LOGIN login = new LOGIN();
24
25                login.show();
26
27            }
28        });
29        btnreg.addActionListener(new ActionListener
() {
30            @Override
31            public void actionPerformed(ActionEvent e
) {
32                REGISTER register = new REGISTER();
33
34                register.show();
35            }
36        });
37    }
}
```

38 }

39

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form"
/ version="1" bind-to-class="com.onymbo.sufeeds.
REGISTER">
3   <grid id="27dc6" binding="regpanel" layout-manager
="GridLayoutManager" row-count="9" column-count="7"
same-size-horizontally="false" same-size-vertically="
false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="500" height="400"/>
7     </constraints>
8     <properties/>
9     <border type="none"/>
10    <children>
11      <component id="ccb12" class="javax.swing.JLabel"
" binding="lblfirstname">
12        <constraints>
13          <grid row="2" column="1" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
14        </constraints>
15        <properties>
16          <text value="FirstName"/>
17        </properties>
18      </component>
19      <component id="7950c" class="javax.swing.JLabel"
" binding="lblLastname">
20        <constraints>
21          <grid row="3" column="1" row-span="1" col-
span="2" vsize-policy="0" hsize-policy="0" anchor="8"
fill="0" indent="0" use-parent-layout="false"/>
22        </constraints>
23        <properties>
24          <text value="LastName"/>
25        </properties>
26      </component>
27      <component id="66396" class="javax.swing.JLabel"
" binding="lbлемail">
28        <constraints>
29          <grid row="4" column="1" row-span="1" col-
```

```
29 span="3" vsize-policy="0" hsize-policy="0" anchor="8"
  fill="0" indent="0" use-parent-layout="false"/>
30   </constraints>
31   <properties>
32     <text value="Email"/>
33   </properties>
34   </component>
35   <component id="5bdae" class="javax.swing.JLabel"
  " binding="lblpassword">
36     <constraints>
37       <grid row="6" column="1" row-span="1" col-
  span="5" vsize-policy="0" hsize-policy="0" anchor="8"
  fill="0" indent="0" use-parent-layout="false"/>
38     </constraints>
39     <properties>
40       <text value="Create Password"/>
41     </properties>
42     </component>
43     <component id="54a43" class="javax.swing.JLabel"
  " binding="lblid">
44       <constraints>
45         <grid row="5" column="1" row-span="1" col-
  span="4" vsize-policy="0" hsize-policy="0" anchor="8"
  fill="0" indent="0" use-parent-layout="false"/>
46       </constraints>
47       <properties>
48         <text value="Student ID"/>
49       </properties>
50     </component>
51     <component id="f2e3c" class="javax.swing.
  JTextField" binding="textField1" default-binding="
  true">
52       <constraints>
53         <grid row="2" column="6" row-span="1" col-
  span="1" vsize-policy="0" hsize-policy="6" anchor="8"
  fill="1" indent="0" use-parent-layout="false">
54           <preferred-size width="150" height="-1"/>
55         </grid>
56       </constraints>
57       <properties/>
58     </component>
```

```
59      <component id="8382e" class="javax.swing.  
       JTextField" binding="textField2" default-binding="  
       true">  
60          <constraints>  
61              <grid row="3" column="6" row-span="1" col-  
               span="1" vsize-policy="0" hsize-policy="6" anchor="8"  
               fill="1" indent="0" use-parent-layout="false">  
62                  <preferred-size width="150" height="-1"/>  
63                  </grid>  
64          </constraints>  
65          <properties/>  
66      </component>  
67      <component id="acdfd" class="javax.swing.  
       JTextField" binding="textField3" default-binding="  
       true">  
68          <constraints>  
69              <grid row="4" column="6" row-span="1" col-  
               span="1" vsize-policy="0" hsize-policy="6" anchor="8"  
               fill="1" indent="0" use-parent-layout="false">  
70                  <preferred-size width="150" height="-1"/>  
71                  </grid>  
72          </constraints>  
73          <properties/>  
74      </component>  
75      <component id="ac823" class="javax.swing.  
       JTextField" binding="textField4" default-binding="  
       true">  
76          <constraints>  
77              <grid row="5" column="6" row-span="1" col-  
               span="1" vsize-policy="0" hsize-policy="6" anchor="8"  
               fill="1" indent="0" use-parent-layout="false">  
78                  <preferred-size width="150" height="-1"/>  
79                  </grid>  
80          </constraints>  
81          <properties/>  
82      </component>  
83      <component id="5c24b" class="javax.swing.  
       JPasswordField" binding="passwordField1" default-  
       binding="true">  
84          <constraints>  
85              <grid row="6" column="6" row-span="1" col-
```

```
85 span="1" vsize-policy="0" hsize-policy="6" anchor="8
  " fill="1" indent="0" use-parent-layout="false">
86           <preferred-size width="150" height="-1"
/>>
87           </grid>
88           </constraints>
89           <properties/>
90           </component>
91           <hspacer id="8a309">
92             <constraints>
93               <grid row="2" column="0" row-span="1" col-
span="1" vsize-policy="1" hsize-policy="6" anchor="0
  " fill="1" indent="0" use-parent-layout="false"/>
94             </constraints>
95           </hspacer>
96           <vspacer id="c9659">
97             <constraints>
98               <grid row="0" column="5" row-span="1" col-
span="1" vsize-policy="6" hsize-policy="1" anchor="0
  " fill="2" indent="0" use-parent-layout="false"/>
99             </constraints>
100           </vspacer>
101           <vspacer id="771e0">
102             <constraints>
103               <grid row="8" column="5" row-span="1" col-
span="1" vsize-policy="6" hsize-policy="1" anchor="0
  " fill="2" indent="0" use-parent-layout="false"/>
104             </constraints>
105           </vspacer>
106           <hspacer id="bfd9a">
107             <constraints>
108               <grid row="6" column="0" row-span="1" col-
span="1" vsize-policy="1" hsize-policy="6" anchor="0
  " fill="1" indent="0" use-parent-layout="false"/>
109             </constraints>
110           </hspacer>
111           <component id="f8afed" class="javax.swing.
JLabel" binding="lblreg">
112             <constraints>
113               <grid row="1" column="5" row-span="1" col-
span="1" vsize-policy="0" hsize-policy="0" anchor="0
```

```
113 " fill="0" indent="0" use-parent-layout="false"/>
114         </constraints>
115         <properties>
116             <text value="REGISTER"/>
117         </properties>
118     </component>
119     <component id="79c95" class="javax.swing.
JButton" binding="btnsubmit">
120         <constraints>
121             <grid row="7" column="1" row-span="1" col-
span="5" vsize-policy="0" hsize-policy="3" anchor="0
" fill="1" indent="0" use-parent-layout="false"/>
122         </constraints>
123         <properties>
124             <text value="Submit"/>
125         </properties>
126     </component>
127     </children>
128 </grid>
129 </form>
130
```

```
1 package com.onyimbo.sufeeds;
2
3 import javax.swing.*;
4 import java.sql.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 public class REGISTER extends JFrame{
9     private JPanel regpanel;
10    private JLabel lblreg;
11    private JLabel lblfirstname;
12    private JLabel lblLastname;
13    private JLabel lblemail;
14    private JLabel lblid;
15    private JLabel lblpassword;
16    private JTextField textField1;
17    private JTextField textField2;
18    private JTextField textField3;
19    private JTextField textField4;
20    private JPasswordField passwordField1;
21    private JButton btnsubmit;
22
23    public REGISTER(){
24        super("Register Page");
25
26        setContentPane(regpanel);
27        setSize(500, 500);
28        setVisible(true);
29        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30
31        btnsubmit.addActionListener(new
32            ActionListener() {
33                @Override
34                public void actionPerformed(ActionEvent e
35                ) {
36                    handleRegistration();
37                }
38            }/**
```

```
39     * Handles the registration logic.
40     */
41     private void handleRegistration() {
42         // Gather input data
43         String firstName = textField1.getText();
44         String lastName = textField2.getText();
45         String email = textField3.getText();
46         String id = textField4.getText();
47         String password = new String(passwordField1.
48             getPassword());
49
50         // Validate inputs
51         if (firstName.isEmpty() || lastName.isEmpty()
52             () || email.isEmpty() || id.isEmpty() || password.
53             isEmpty()) {
54             JOptionPane.showMessageDialog(this, "All
55             fields are required!", "Input Error", JOptionPane.
56             ERROR_MESSAGE);
57             return;
58         }
59
60         // Save data to the database
61         if (saveToDatabase(firstName, lastName, email
62             , id, password)) {
63             JOptionPane.showMessageDialog(this, "
64             Registration successful!");
65         }
66
67         /**
68          * Saves user data to the database.
69          *
70          * @return true if the data was saved
71     }
```

```
70 successfully, false otherwise.
71     */
72     private boolean saveToDatabase(String firstName
73         , String lastName, String email, String id, String
74         password) {
75         String insertQuery = "INSERT INTO
76             tbl_students (first_name, last_name, email,
77             student_id, password_) VALUES (?, ?, ?, ?, ?)";
78         try (Connection connection = Main.
79             getDatabaseConnection();
80             PreparedStatement preparedStatement =
81             connection.prepareStatement(insertQuery)) {
82             // Set the parameters
83             preparedStatement.setString(1, firstName
84             );
85             preparedStatement.setString(2, lastName
86             );
87             preparedStatement.setString(3, email);
88             preparedStatement.setString(4, id);
89             preparedStatement.setString(5, password
90             );
91             // Execute the query
92             int rowsInserted = preparedStatement.
93                 executeUpdate();
94             return rowsInserted > 0;
95         } catch (Exception ex) {
96             ex.printStackTrace();
97         }
98         return false;
99     }
100 }
```