

## JDBC Mini project

**Aim:** To present a attendance monitoring system and attendance calculator using database management system and java .

### Source code:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

class AttendanceCalculatorSwing extends JFrame {

    private static final String DB_URL =
"jdbc:mysql://localhost:3306/attendance_db";
    private static final String USER = "root";
    private static final String PASS = "12345678";

    private JTextField searchField;
    private JButton searchButton;
    private JTable resultsTable;
    private DefaultTableModel tableModel;

    public AttendanceCalculatorSwing() {

        setTitle("Attendance Calculator");
```

```

setSize(600, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);

// Layout and panel
JPanel panel = new JPanel(new BorderLayout());
JPanel inputPanel = new JPanel(new FlowLayout());

// Search field
searchField = new JTextField(20);
searchButton = new JButton("Search");

// Table for displaying results
String[] columns = {"ID", "Name", "Total Classes", "Classes
Attended", "Attendance %", "Status"};
tableModel = new DefaultTableModel(columns, 0);
resultsTable = new JTable(tableModel);

// Add search field and button to input panel
inputPanel.add(new JLabel("Enter student name:"));
inputPanel.add(searchField);
inputPanel.add(searchButton);

// Add components to main panel
panel.add(inputPanel, BorderLayout.NORTH);
panel.add(new JScrollPane(resultsTable), BorderLayout.CENTER);

// Add main panel to the frame
add(panel);

// Action listener for search button
searchButton.addActionListener(this::performSearch);
}

```

```

// Method to perform the search and update table
private void performSearch(ActionEvent e) {
    String studentNameInput = searchField.getText().trim();

    // Clear previous results
    tableModel.setRowCount(0);

    // Database query with optional name filter
    String sqlQuery = "SELECT student_id, student_name,
total_classes, classes_attended FROM students";
    if (!studentNameInput.isBlank()) {
        sqlQuery += " WHERE student_name LIKE ?";
    }

    try (Connection connection =
DriverManager.getConnection(DB_URL, USER, PASS);
        PreparedStatement preparedStatement =
connection.prepareStatement(sqlQuery)) {

        // Set the search parameter if a name was provided
        if (!studentNameInput.isBlank()) {
            preparedStatement.setString(1, "%" + studentNameInput +
"%");
        }

        ResultSet resultSet = preparedStatement.executeQuery();

        // Check if any results were found
        boolean hasResults = false;
        while (resultSet.next()) {
            hasResults = true;
            int studentId = resultSet.getInt("student_id");

```

```

String studentName = resultSet.getString("student_name");
int totalClasses = resultSet.getInt("total_classes");
int classesAttended = resultSet.getInt("classes_attended");

// Calculate attendance percentage
double attendancePercentage = (totalClasses > 0) ?
    ((double) classesAttended / totalClasses) * 100 : 0.0;

// Determine if there is a shortage of attendance
String status = (attendancePercentage < 75) ? "Shortage of
Attendance" : "Sufficient Attendance";

// Add row to table model
tableModel.addRow(new Object[]{studentId, studentName,
totalClasses, classesAttended, String.format("%.2f",
attendancePercentage), status});
}

// If no results were found, show a message
if (!hasResults) {
    JOptionPane.showMessageDialog(this, "No records found
for student name: " + studentNameInput, "No Results",
JOptionPane.INFORMATION_MESSAGE);
}

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this, "Database error: " +
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    ex.printStackTrace();
}
}

public static void main(String[] args) {

```

```

// Run the GUI application
SwingUtilities.invokeLater(() -> {
    AttendanceCalculatorSwing app = new
AttendanceCalculatorSwing();
    app.setVisible(true);
});
}

}


```

## Output:

- Attendance Monitor

Attendance Calculator				
ID	Name	Total Classes	Classes Attended	Attendance %
1	Alice	30	28	93.33%
2	Bob	30	25	83.33%
3	Charlie	30	27	90.00%
4	Diana	30	30	100.00%
5	Eve	30	20	66.67%
6	Alice	30	28	93.33%
7	Bob	30	25	83.33%
8	Charlie	30	27	90.00%
9	Diana	30	30	100.00%
10	Eve	30	20	66.67%
11	keerthana	73	45	61.64%
12	madhan	70	55	78.57%
13	davis	70	67	95.71%
14	hema	70	30	42.86%
15	siva	70	60	85.71%
16	cynthia	93	50	53.76%
17	riya	90	55	61.11%
18	tawfiq	80	47	58.75%
19	das	84	53	63.10%
20	leo	70	10	14.29%


- Attendance Calculator

 Attendance Calculator

Enter student name:

Search

ID	Name	Total Classes	Classes Attended	Attendance %	Status
----	------	---------------	------------------	--------------	--------

 Attendance Calculator

Enter student name:

Search

ID	Name	Total Classes	Classes Attended	Attendance %	Status
15	siva	70	60	85.71	Sufficient Atten...

**Result:**

The given project has been completed and performed successfully.

**Team members:** Sivarangini.Y

Hemanth kumar . A