

Assignment # 02

Name : Muhammad Saad Saeed
Reg# 4592-FOC/BSCS/F22-B

Java Code

```
import javax.swing.*;
import java.awt.*;
import java.sql.*;

public class StudentApp extends JFrame {
    JTextField nameField, marksField, idField;
    JTextArea outputArea;
    Connection conn;

    public StudentApp() {
        setTitle("Student Database App");
        setSize(500, 400);
        setLayout(new FlowLayout());
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        nameField = new JTextField(15);
        marksField = new JTextField(5);
        idField = new JTextField(5);
        outputArea = new JTextArea(10, 40);
        outputArea.setEditable(false);

        JButton addBtn = new JButton("Add");
        JButton showBtn = new JButton("Show All");
        JButton updateBtn = new JButton("Update");
        JButton deleteBtn = new JButton("Delete");
        JButton callSPBtn = new JButton("Find by ID");

        add(new JLabel("Name:"));
        add(nameField);
        add(new JLabel("Marks:"));
        add(marksField);
        add(addBtn);

        add(new JLabel("ID:"));
        add(idField);
        add(updateBtn);
        add(deleteBtn);
        add(callSPBtn);
        add(showBtn);
```

```
add(new JScrollPane(outputArea));
```

```
try {  
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/studentdb", "root",  
    "secretpassword");  
} catch (Exception e) {  
    outputArea.setText("Database connection error: " + e);  
}
```

```
addBtn.addActionListener(e -> insertStudent());  
showBtn.addActionListener(e -> showStudents());  
updateBtn.addActionListener(e -> updateStudent());  
deleteBtn.addActionListener(e -> deleteStudent());  
callSPBtn.addActionListener(e -> callProcedure());  
}
```

```
void insertStudent() {  
    try (PreparedStatement stmt = conn.prepareStatement("INSERT INTO students (name,  
    marks) VALUES (?, ?)")) {  
        stmt.setString(1, nameField.getText());  
        stmt.setInt(2, Integer.parseInt(marksField.getText()));  
        stmt.executeUpdate();  
        outputArea.setText("Student added.");  
    } catch (Exception ex) {  
        outputArea.setText("Error: " + ex.getMessage());  
    }  
}
```

```
void showStudents() {  
    try (Statement stmt = conn.createStatement()) {  
        ResultSet rs = stmt.executeQuery("SELECT * FROM students");  
        StringBuilder sb = new StringBuilder("Students:\n");  
        while (rs.next()) {  
            sb.append(rs.getInt("id")).append(" - ")  
            .append(rs.getString("name")).append(" - ")  
            .append(rs.getInt("marks")).append("\n");  
        }  
        outputArea.setText(sb.toString());  
    } catch (Exception ex) {  
        outputArea.setText("Error: " + ex.getMessage());  
    }  
}
```

```
void updateStudent() {  
    try (PreparedStatement stmt = conn.prepareStatement("UPDATE students SET name=?,  
    marks=? WHERE id=?")) {  
        stmt.setString(1, nameField.getText());  
        stmt.setInt(2, Integer.parseInt(marksField.getText()));  
    }
```

```

stmt.setInt(3, Integer.parseInt(idField.getText()));
int rows = stmt.executeUpdate();
outputArea.setText(rows > 0 ? "Student updated." : "ID not found.");
} catch (Exception ex) {
outputArea.setText("Error: " + ex.getMessage());
}
}

```

```

void deleteStudent() {
try (PreparedStatement stmt = conn.prepareStatement("DELETE FROM students WHERE
id=?")) {
stmt.setInt(1, Integer.parseInt(idField.getText()));
int rows = stmt.executeUpdate();
outputArea.setText(rows > 0 ? "Student deleted." : "ID not found.");
} catch (Exception ex) {
outputArea.setText("Error: " + ex.getMessage());
}
}

```

```

void callProcedure() {
try (CallableStatement cs = conn.prepareCall("{CALL getStudentById(?)}") {
cs.setInt(1, Integer.parseInt(idField.getText()));
ResultSet rs = cs.executeQuery();
if (rs.next()) {
outputArea.setText("ID: " + rs.getInt("id") +
"\nName: " + rs.getString("name") +
"\nMarks: " + rs.getInt("marks"));
} else {
outputArea.setText("Student not found.");
}
} catch (Exception ex) {
outputArea.setText("Error: " + ex.getMessage());
}
}

```

```

public static void main(String[] args) {
SwingUtilities.invokeLater(() -> new StudentApp().setVisible(true));
}
}

```

Output

The screenshot shows a window titled "Student Database App" with standard window controls (minimize, maximize, close). The interface includes a form with the following elements:

- Name:** A text input field containing "Raja Saad Saeed".
- Marks:** A text input field containing "76".
- ID:** A text input field containing "7".
- Buttons:** "Add", "Update", and "Delete" buttons are positioned to the right of the form fields. "Find by ID" and "Show All" buttons are located below the Name field.
- Output Area:** A text box on the right side of the window displays the following information:
ID: 7
Name: Raja Saad Saeed
Marks: 76