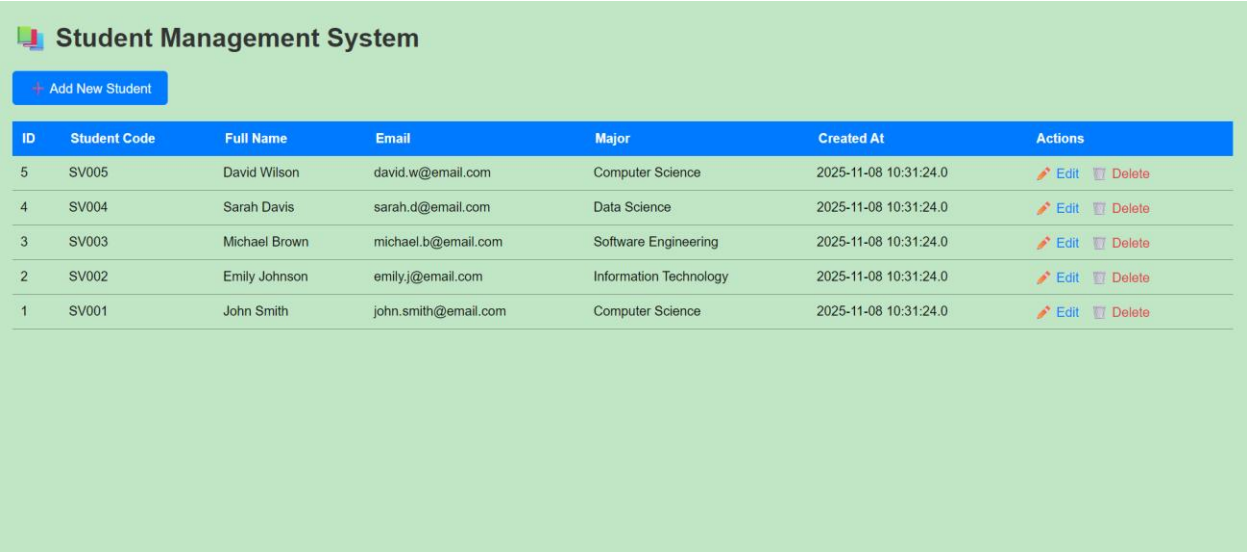
Exercise 1:




```
<% if (request.getParameter("message") != null) { %>
    <div class="message success">
        <%= request.getParameter("message") %>
    </div>
<% } %>

<% if (request.getParameter("error") != null) { %>
    <div class="message error">
        <%= request.getParameter("error") %>
    </div>
<% } %>
```

If the message is empty and the error isn't empty then the user will get an error notification

```
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");

        conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/student_management",
            "root",
            "root"
        );

        stmt = conn.createStatement();
        String sql = "SELECT * FROM students ORDER BY id DESC";
        rs = stmt.executeQuery(sql);

        while (rs.next()) {
            int id = rs.getInt("id");
            String studentCode = rs.getString("student_code");
            String fullName = rs.getString("full_name");
            String email = rs.getString("email");
            String major = rs.getString("major");
            Timestamp createdAt = rs.getTimestamp("created_at");
%>
```

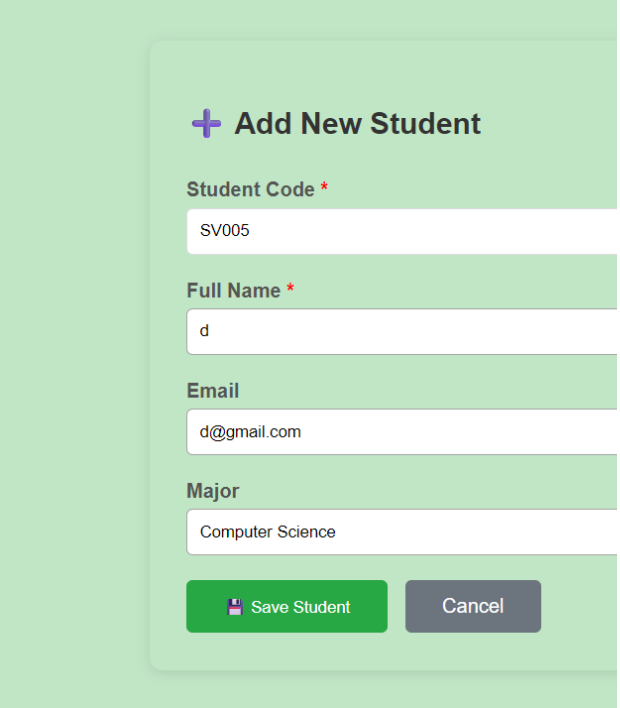
This line of code attempts to access the student_management sql file with the username of root and password of root, then assigns the corresponding values to a string or timestamp.

```
<%
        }
    } catch (ClassNotFoundException e) {
        out.println("<tr><td colspan='7'>Error: JDBC Driver not found!</td></tr>");
        e.printStackTrace();
    } catch (SQLException e) {
        out.println("<tr><td colspan='7'>Database Error: " + e.getMessage() + "</td></tr>");
        e.printStackTrace();
    } finally {
        try {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
%>
```

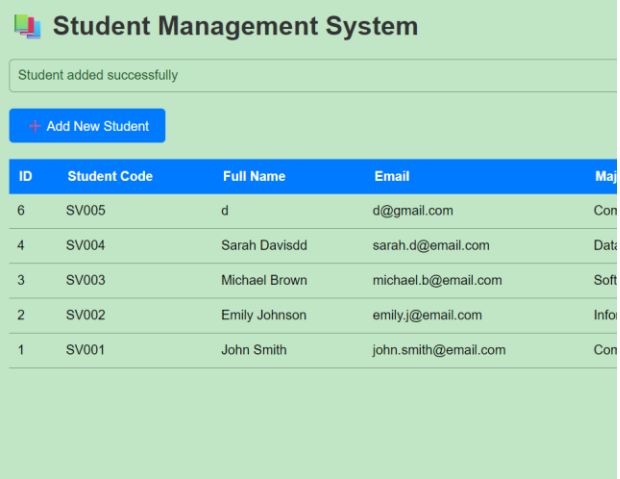
catch (ClassNotFoundException)	Handles missing JDBC driver errors
catch (SQLException)	Handles database-related errors
finally	Closes database resources safely (ResultSet, Statement, Connection)

Exercise 2:

On pressing Add New student:



On pressing save student:



A new student was added

Pressing the cancel button would just return the user to the main page again

```
<form action="process_add.jsp" method="POS
    <div class="form-group">
        <label for="student_code">Student
        <input type="text" id="student_cod
               placeholder="e.g., SV001" r
               pattern="[A-Z]{2}[0-9]{3,}"
               title="Format: 2 uppercase
    </div>

    <div class="form-group">
        <label for="full_name">Full Name <
        <input type="text" id="full_name"
               placeholder="Enter full nam
    </div>

    <div class="form-group">
        <label for="email">Email</label>
        <input type="email" id="email" nam
               placeholder="student@email.
    </div>

    <div class="form-group">
        <label for="major">Major</label>
        <input type="text" id="major" name
               placeholder="e.g., Computer
    </div>

    <button type="submit" class="btn-submi
```

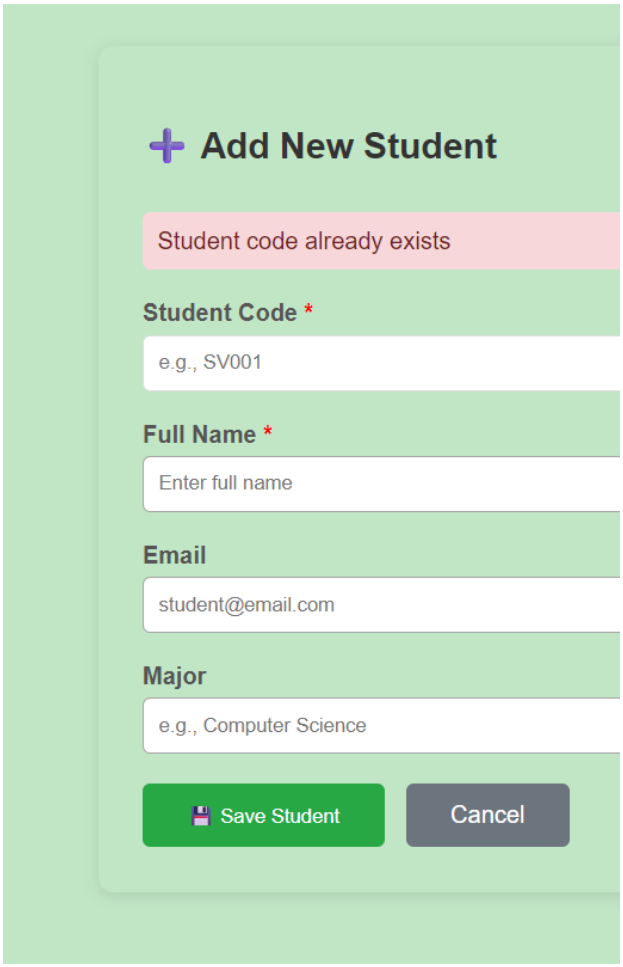
Add_student.jsp just has a form to send input information to process_add.jsp

```
String studentCode = request.getParameter("student_code"
String fullName = request.getParameter("full_name");
String email = request.getParameter("email");
String major = request.getParameter("major");

if (studentCode == null || studentCode.trim().isEmpty()
    fullName == null || fullName.trim().isEmpty()) {
    response.sendRedirect("add_student.jsp?error=Require
    return;
}

Connection conn = null;
PreparedStatement pstmt = null;

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/student_management",
        "root",
        "root"
    );

    String sql = "INSERT INTO students (student_code, fu
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, studentCode);
    pstmt.setString(2, fullName);
    pstmt.setString(3, email);
    pstmt.setString(4, major);
```

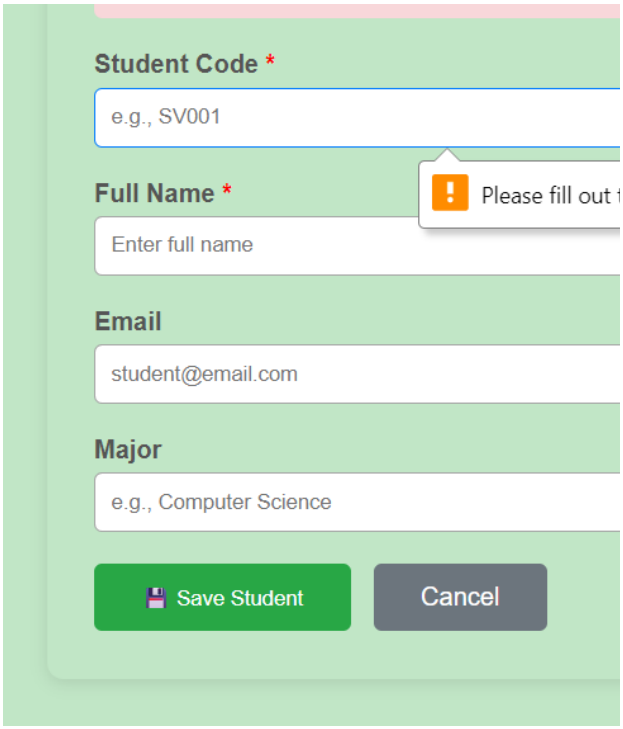
```
        if (rowsAffected > 0) {
            response.sendRedirect("list_students.jsp
        } else {
            response.sendRedirect("add_student.jsp?e
        }

    } catch (ClassNotFoundException e) {
        response.sendRedirect("add_student.jsp?error
        e.printStackTrace();
    } catch (SQLException e) {
        String errorMsg = e.getMessage();
        if (errorMsg.contains("Duplicate entry")) {
            response.sendRedirect("add_student.jsp?e
        } else {
            response.sendRedirect("add_student.jsp?e
        }
        e.printStackTrace();
    } finally {
        try {
            if (pstmt != null) pstmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

Process_add.jsp enters the input into the database, sending errors if any input is missing, a driver isnt found or if information is duplicate

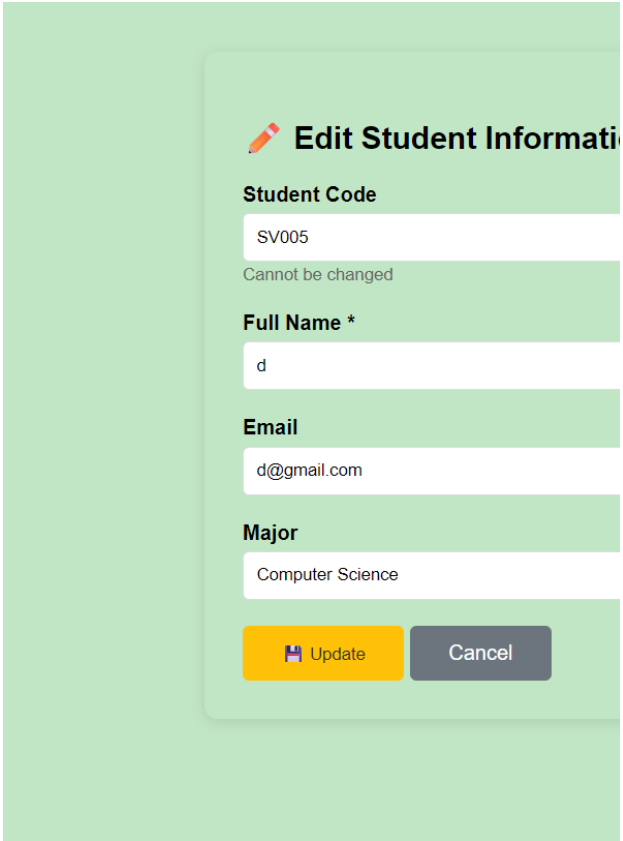
Duplicate error:


Add New Student
Student code already exists
Student Code *
e.g., SV001
Full Name *
Enter full name
Email
student@email.com
Major
e.g., Computer Science
Save Student
Cancel

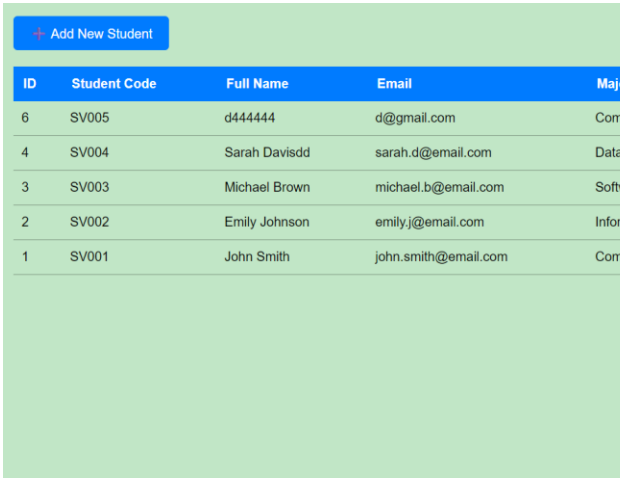
Empty:


Student Code *
e.g., SV001
Full Name *
Please fill out
Enter full name
Email
student@email.com
Major
e.g., Computer Science
Save Student
Cancel

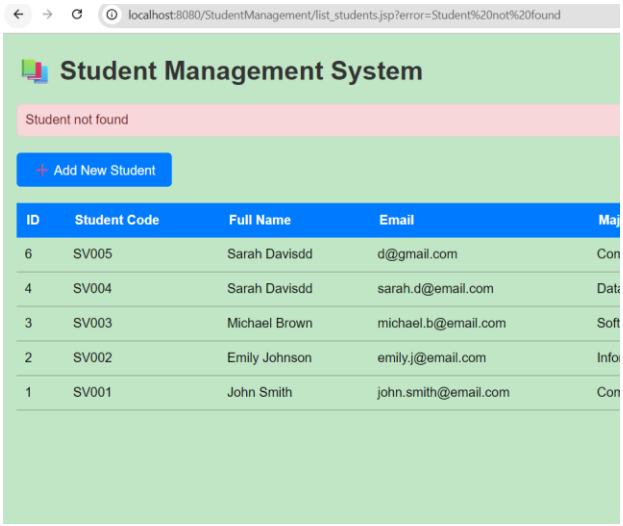
Exercise 3:


Edit Student Informati
Student Code
SV005
Cannot be changed
Full Name *
d
Email
d@gmail.com
Major
Computer Science
Update
Cancel

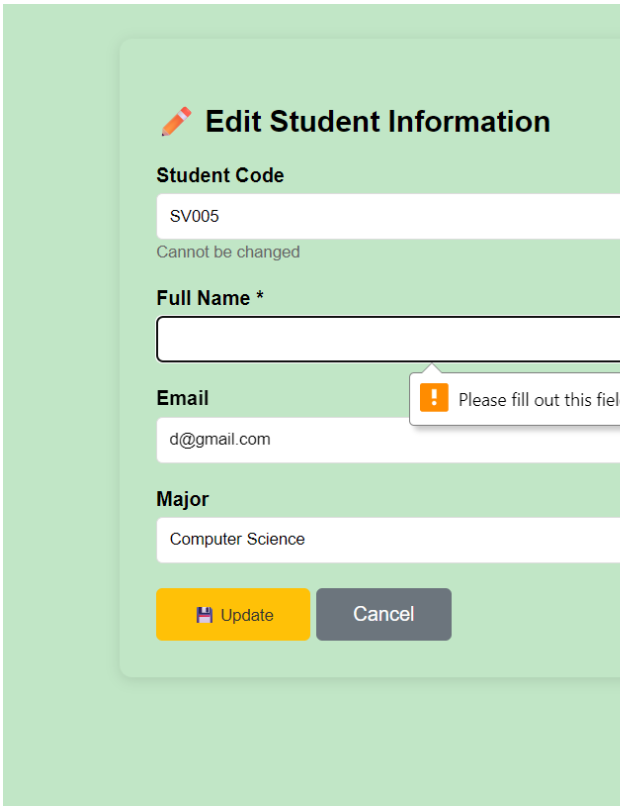
After edit:



Wrong id:

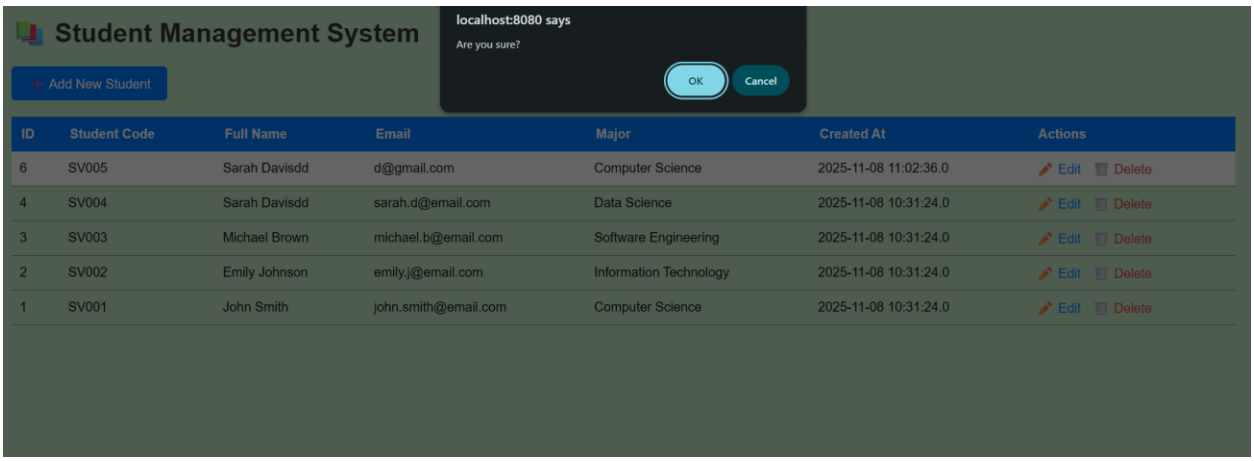


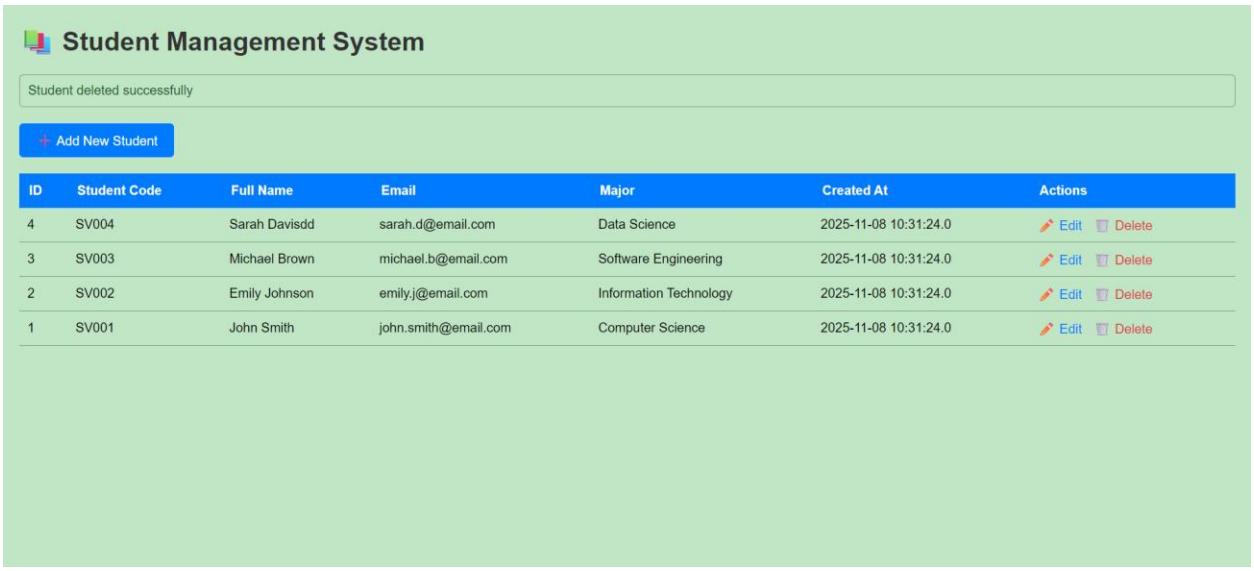
Blank name:



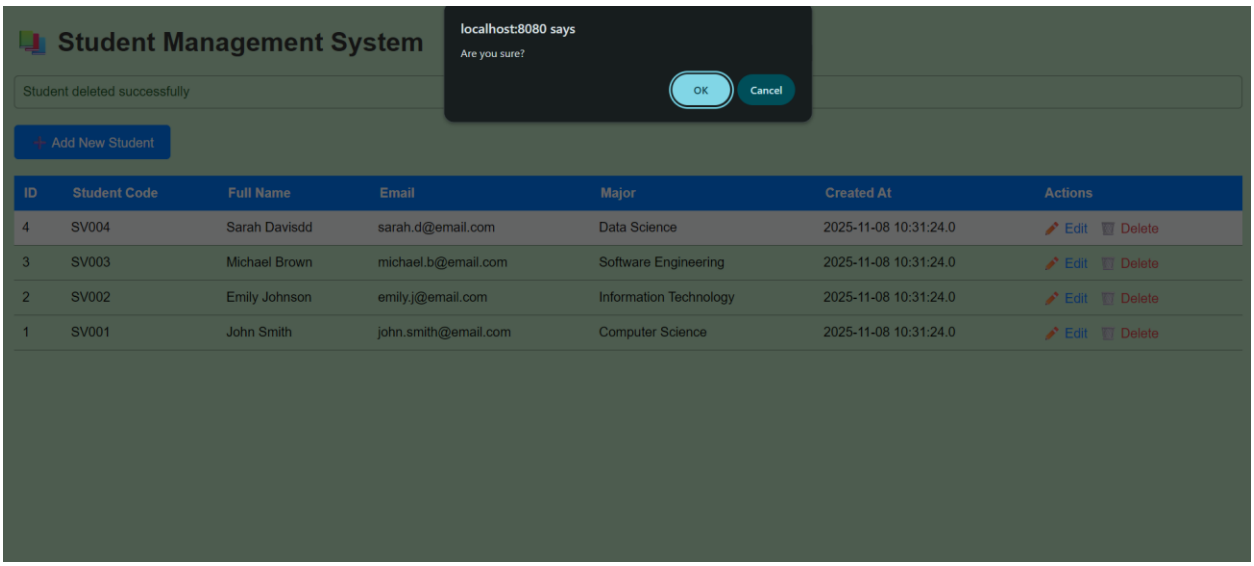
Process_edit.jsp and edit_student.jsp works similarly to adding, one is a form and one changes information in the databbase

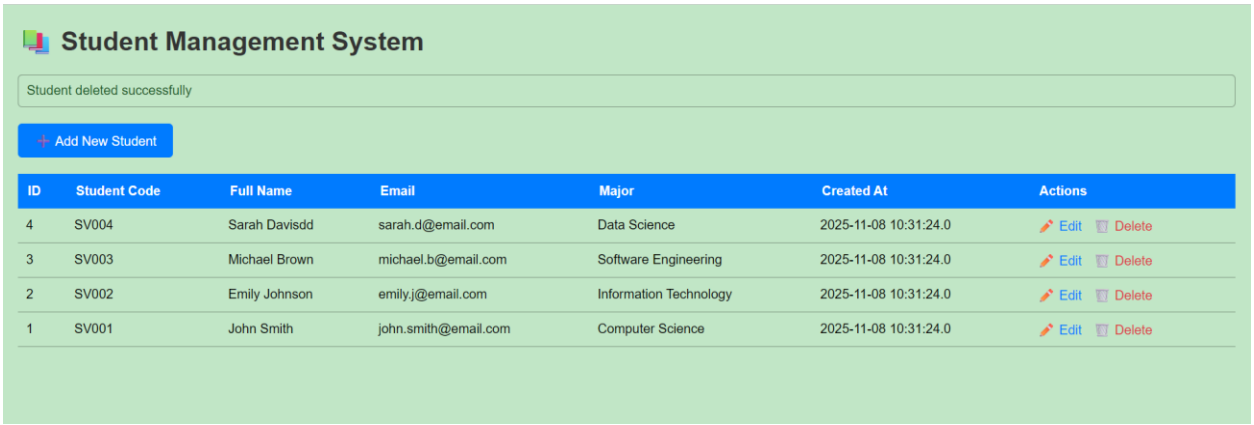
Exercise 4:



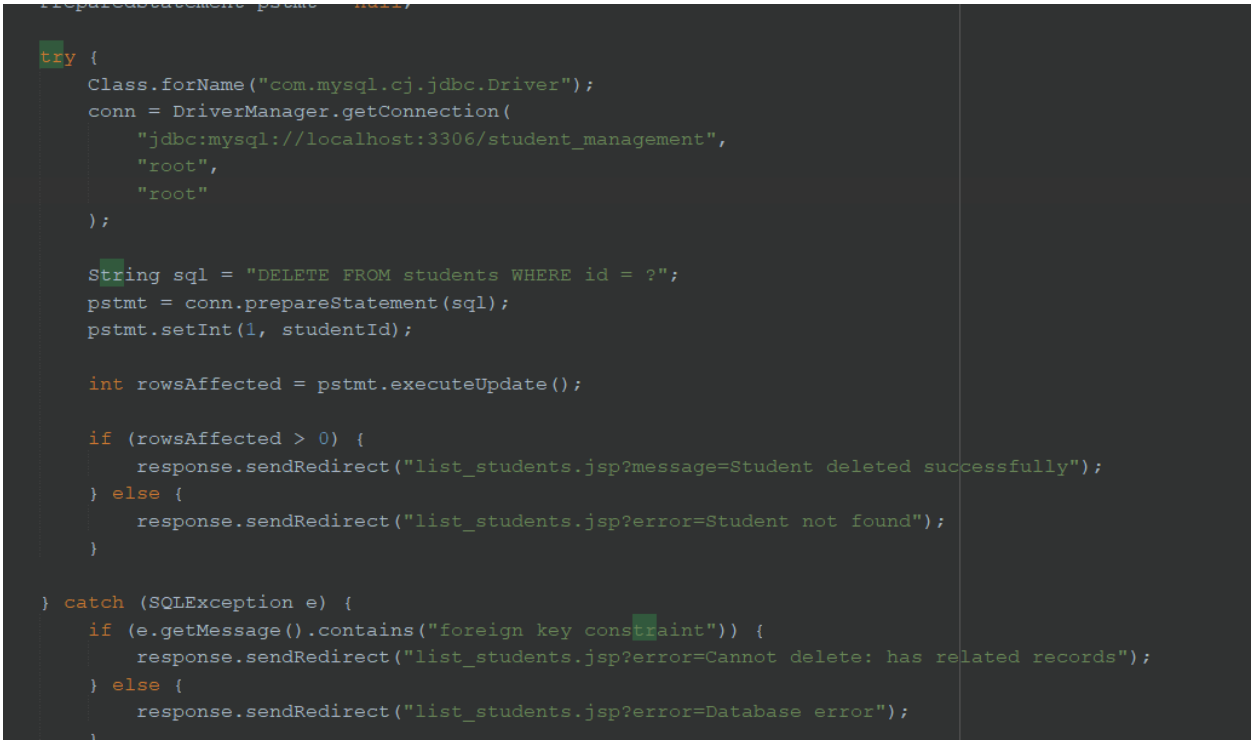


Cancel:





The student remains



```
PreparedStatement pstmt = null;

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/student_management",
        "root",
        "root"
    );

    String sql = "DELETE FROM students WHERE id = ?";
    pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, studentId);

    int rowsAffected = pstmt.executeUpdate();

    if (rowsAffected > 0) {
        response.sendRedirect("list_students.jsp?message=Student deleted successfully");
    } else {
        response.sendRedirect("list_students.jsp?error=Student not found");
    }

} catch (SQLException e) {
    if (e.getMessage().contains("foreign key constraint")) {
        response.sendRedirect("list_students.jsp?error=Cannot delete: has related records");
    } else {
        response.sendRedirect("list_students.jsp?error=Database error");
    }
```

The jsp file attempts to delete the correlating file from the sql database, report to the user if it encounters any errors while doing so such as if it has foreign key constraints