


Exercise 1:

 **Student Management System**

[+ Add New Student](#)

ID	Student Code	Full Name	Email	Major	Created At	Actions
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 10:31:24.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 10:31:24.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 10:31:24.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 10:31:24.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 10:31:24.0	Edit Delete

```
<% 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

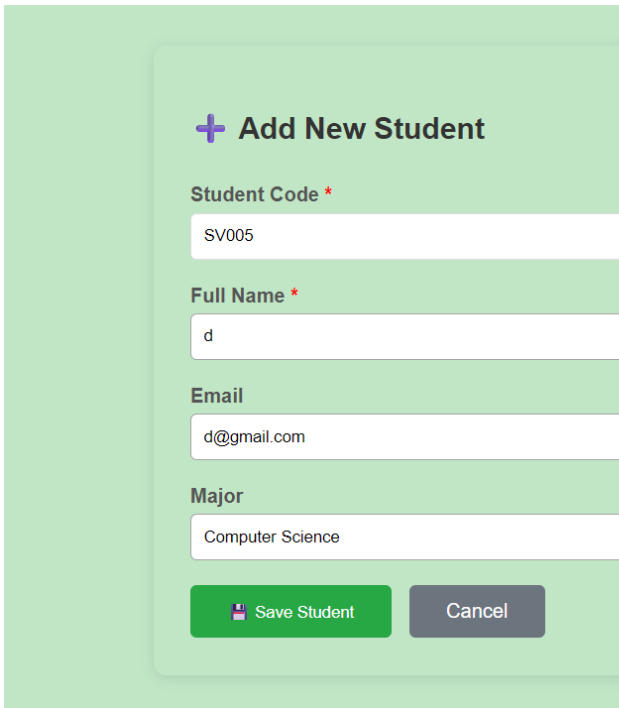
catch (SQLException) Handles database-related errors

finally

Closes database resources safely
(ResultSet, Statement, Connection)

Exercise 2:

On pressing Add New student:

A modal form titled "+ Add New Student" with a purple plus icon. It contains four input fields: "Student Code *" with value "SV005", "Full Name *" with value "d", "Email" with value "d@gmail.com", and "Major" with value "Computer Science". At the bottom are two buttons: a green "Save Student" button with a floppy disk icon and a grey "Cancel" button.

+ Add New Student

Student Code *

SV005

Full Name *


d

Email

d@gmail.com

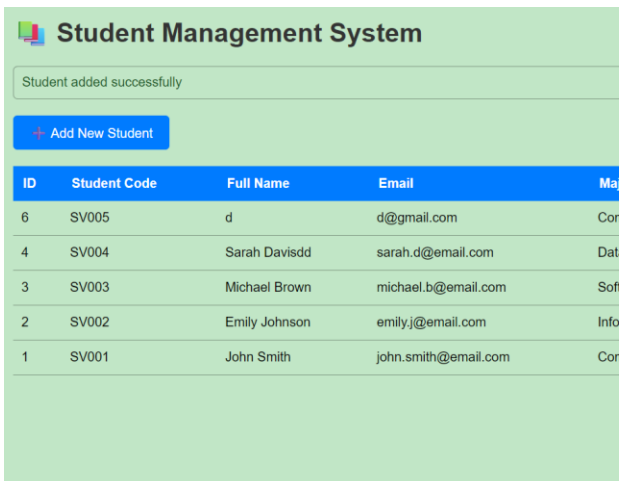
Major


Computer Science

 Save Student


Cancel

On pressing save student:

The interface shows a green header with a monitor icon and the title "Student Management System". Below it is a green box with the message "Student added successfully". A blue button with a plus icon and the text "Add New Student" is present. Below the button is a table with 5 columns: ID, Student Code, Full Name, Email, and Major. The table contains 6 rows of student data.

 **Student Management System**

Student added successfully

 Add New Student

ID	Student Code	Full Name	Email	Major
6	SV005	d	d@gmail.com	Computer Science
4	SV004	Sarah Davisdd	sarah.d@email.com	Data Science
3	SV003	Michael Brown	michael.b@email.com	Software Engineering
2	SV002	Emily Johnson	emily.j@email.com	Information Systems
1	SV001	John Smith	john.smith@email.com	Computer Science

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="POST">
  <div class="form-group">
    <label for="student_code">Student Code</label>
    <input type="text" id="student_code" value="" placeholder="e.g., SV001" pattern="[A-Z]{2}[0-9]{3,}" title="Format: 2 uppercase letters followed by 3 or more digits." />
  </div>

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

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

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

  <button type="submit" class="btn btn-primary">Add Student</button>

```

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() ||
    email == null || email.trim().isEmpty() ||
    major == null || major.trim().isEmpty()) {
    response.sendRedirect("add_student.jsp?error=RequiresValidInput");
    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, full_name, email, major) VALUES (?, ?, ?, ?)";
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, studentCode);
    pstmt.setString(2, fullName);
    pstmt.setString(3, email);
    pstmt.setString(4, major);

    int rowsAffected = pstmt.executeUpdate();

    if (rowsAffected > 0) {
        response.sendRedirect("add_student.jsp?success=StudentAdded");
    } else {
        response.sendRedirect("add_student.jsp?error=FailedToAddStudent");
    }
} catch (Exception e) {
    response.sendRedirect("add_student.jsp?error=" + e.getMessage());
} finally {
    if (pstmt != null) pstmt.close();
    if (conn != null) conn.close();
}

```

```

        if (rowsAffected > 0) {
            response.sendRedirect("list_students.jsp");
        } else {
            response.sendRedirect("add_student.jsp?error=1");
        }
    }

} catch (ClassNotFoundException e) {
    response.sendRedirect("add_student.jsp?error=2");
    e.printStackTrace();
} catch (SQLException e) {
    String errorMsg = e.getMessage();
    if (errorMsg.contains("Duplicate entry")) {
        response.sendRedirect("add_student.jsp?error=3");
    } else {
        response.sendRedirect("add_student.jsp?error=4");
    }
    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 isn't 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 *


Enter full name

Email

student@email.com

Major


e.g., Computer Science

 Save Student

Cancel

Please fill out t

Exercise 3:

 **Edit Student Information**

Student Code

SV005

Cannot be changed

Full Name *


d

Email

d@gmail.com

Major

Computer Science

 Update

Cancel


After edit:

+ Add New Student

ID	Student Code	Full Name	Email	Maj
6	SV005	d444444	d@gmail.com	Com
4	SV004	Sarah Davisdd	sarah.d@email.com	Data
3	SV003	Michael Brown	michael.b@email.com	Soft
2	SV002	Emily Johnson	emily.j@email.com	Infor
1	SV001	John Smith	john.smith@email.com	Com

Wrong id:

localhost:8080/StudentManagement/list_students.jsp?error=Student%20not%20found


 **Student Management System**

Student not found

+ Add New Student

ID	Student Code	Full Name	Email	Maj
6	SV005	Sarah Davisdd	d@gmail.com	Com
4	SV004	Sarah Davisdd	sarah.d@email.com	Data
3	SV003	Michael Brown	michael.b@email.com	Soft
2	SV002	Emily Johnson	emily.j@email.com	Infor
1	SV001	John Smith	john.smith@email.com	Com

Blank name:




Edit Student Information

Student Code

Cannot be changed

Full Name *


Email


Please fill out this field

Major











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


Exercise 4:



Student Management System









localhost:8080 says
Are you sure?

ID	Student Code	Full Name	Email	Major	Created At	Actions
6	SV005	Sarah Davisdd	d@gmail.com	Computer Science	2025-11-08 11:02:36.0	 Edit  Delete
4	SV004	Sarah Davisdd	sarah.d@email.com	Data Science	2025-11-08 10:31:24.0	 Edit  Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 10:31:24.0	 Edit  Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 10:31:24.0	 Edit  Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 10:31:24.0	 Edit  Delete


 **Student Management System**

Student deleted successfully

+ Add New Student

ID	Student Code	Full Name	Email	Major	Created At	Actions
4	SV004	Sarah Davisdd	sarah.d@email.com	Data Science	2025-11-08 10:31:24.0	 Edit  Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 10:31:24.0	 Edit  Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 10:31:24.0	 Edit  Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 10:31:24.0	 Edit  Delete

Cancel:









 **Student Management System**

Student deleted successfully

+ Add New Student

localhost:8080 says
Are you sure?

OK Cancel

ID	Student Code	Full Name	Email	Major	Created At	Actions
4	SV004	Sarah Davisdd	sarah.d@email.com	Data Science	2025-11-08 10:31:24.0	 Edit  Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 10:31:24.0	 Edit  Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 10:31:24.0	 Edit  Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 10:31:24.0	 Edit  Delete

 **Student Management System**

Student deleted successfully

+ Add New Student

ID	Student Code	Full Name	Email	Major	Created At	Actions
4	SV004	Sarah Davisdd	sarah.d@email.com	Data Science	2025-11-08 10:31:24.0	 Edit  Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 10:31:24.0	 Edit  Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 10:31:24.0	 Edit  Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 10:31:24.0	 Edit  Delete

remains