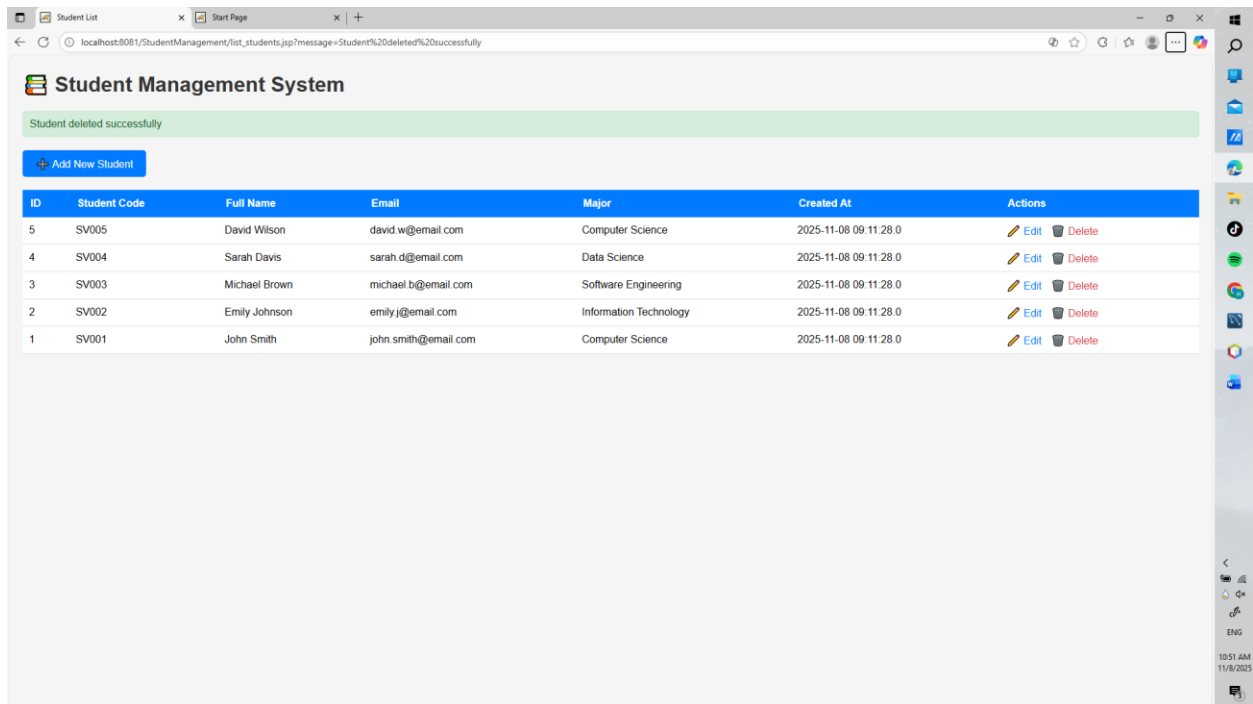


## EXERCISE 1: SETUP AND DISPLAY



### Functionality Overview

#### 1. Database Connection

The page uses `DriverManager.getConnection()` to connect to the `student_management` MySQL database through the JDBC driver.

#### 2. Data Retrieval

A SQL query (`SELECT * FROM students ORDER BY id DESC`) is executed to retrieve all student records. The results are stored in a `ResultSet`.

#### 3. Dynamic Table Generation

The JSP iterates through the `ResultSet` and dynamically generates HTML table rows to display student details such as ID, code, name, email, major, and creation date.

#### 4. User Actions

Each record includes:

- An **Edit** link to modify student information.
- A **Delete** link (with confirmation) to remove a student record.
- A button to **Add New Student**, linking to `add_student.jsp`.

#### 5. Message Display

The page checks for URL parameters `message` and `error` to display success or error notifications after user actions (e.g., after adding or deleting a student).

## 6. Error Handling & Resource Cleanup

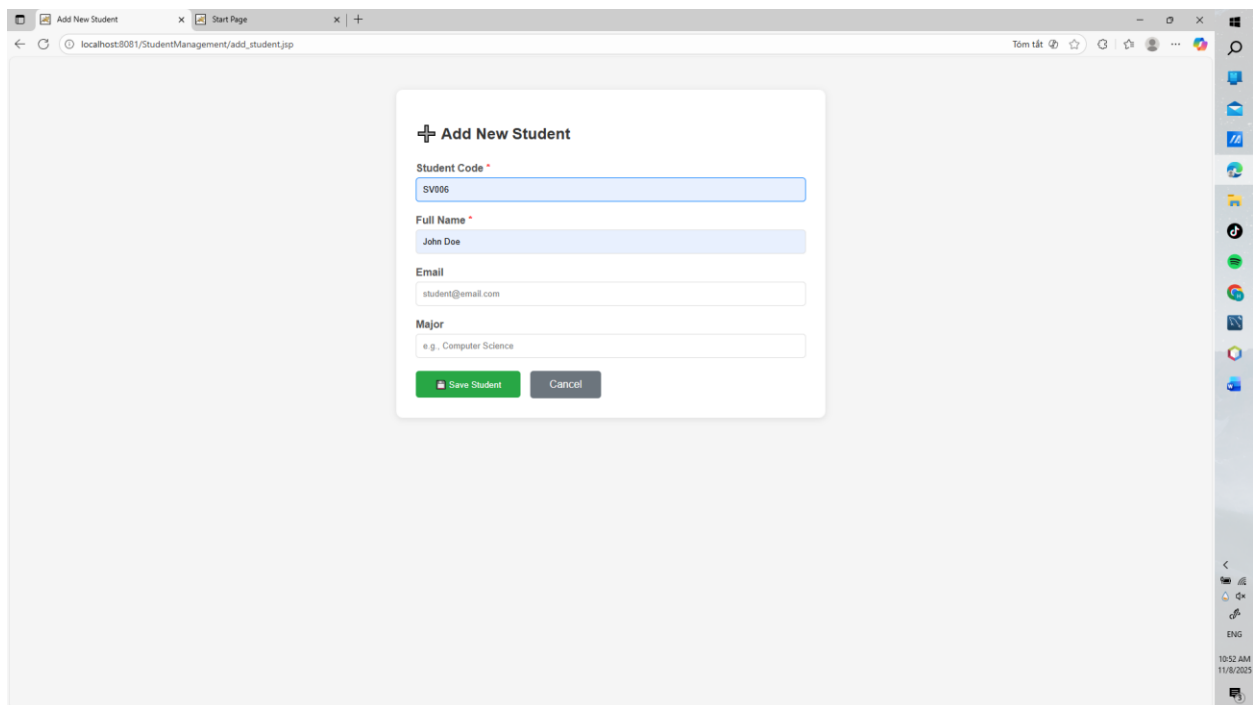
Exceptions like `ClassNotFoundException` or `SQLException` are caught and displayed.

The finally block ensures that the `ResultSet`, `Statement`, and `Connection` are properly closed to avoid resource leaks.

## Workflow

1. The user opens `student_list.jsp`.
2. The server executes embedded Java code to fetch data from MySQL.
3. The page is rendered as an HTML table containing all students.
4. The user can perform **Add**, **Edit**, or **Delete** operations through linked JSP pages.

## EXERCISE 2: CREATE OPERATION

A screenshot of a web browser window displaying a form titled "Add New Student". The form is centered on a light gray background. It contains four input fields: "Student Code" with the value "SV006", "Full Name" with the value "John Doe", "Email" with the value "student@email.com", and "Major" with the value "e.g., Computer Science". Below the fields are two buttons: a green "Save Student" button and a gray "Cancel" button. The browser's address bar shows the URL "localhost:8081/StudentManagement/add\_student.jsp". The Windows taskbar is visible on the right side of the screen.

### add\_student.jsp

input type

Field	Required	Validation
-------	----------	------------

Student Code	Yes	Must follow the pattern: 2 uppercase letters + at least 3 digits (e.g., SV001).
--------------	-----	---

Full Name	Yes	Plain text, cannot be empty.
-----------	-----	------------------------------

Email	No	Must follow email format (validated via <code>type="email"</code> ).
-------	----	--

Field	Required Validation	
Major	No	Free text input.

#### Form Submission

Submits user input to process\_add.jsp using the POST method.

Includes “Save” and “Cancel” buttons:

“Save” → triggers data submission.

“Cancel” → redirects back to list\_students.jsp.

#### **process\_add.jsp**

Get Form Data using request.getParameter

Validation:

- Checks that studentCode and fullName are not empty.
- If missing, redirects back with: add\_student.jsp?error=Required fields are missing

Database Connection:

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

```
conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/student_management", "root", "nghia123");
```

Loads the MySQL JDBC driver.

Connects to the student\_management database.

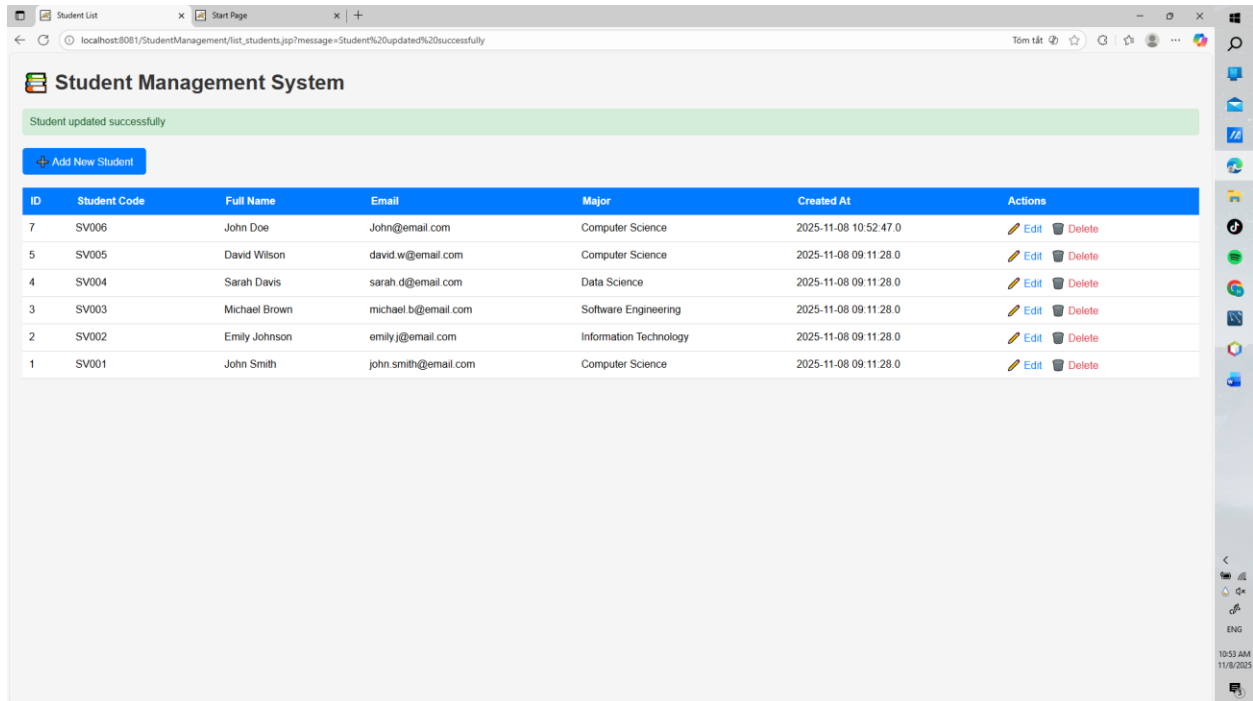
Insert Operation

Redirect Based on Result:

- Success → redirects to:
- list\_students.jsp?message=Student added successfully
- Failure → redirects to:

add\_student.jsp?error=Failed to add student






## edit\_student.jsp

When you click on Edit button:

1. User clicks "Edit" from the student list.
2. The page loads the student's current information from the database.
3. User updates the information and submits the form.
4. The data is sent to process\_edit.jsp for updating in the database.

Includes button controls:

-  **Update** → Submits form to process\_edit.jsp.
- **Cancel** → Returns to list\_students.jsp.

Error messages are displayed in a styled alert box at the top of the form.

## process\_edit.jsp

Receives parameters: id, full\_name, email, and major.

Validates required fields (id and full\_name).



Uses JDBC to connect to MySQL and execute an UPDATE statement:

UPDATE students

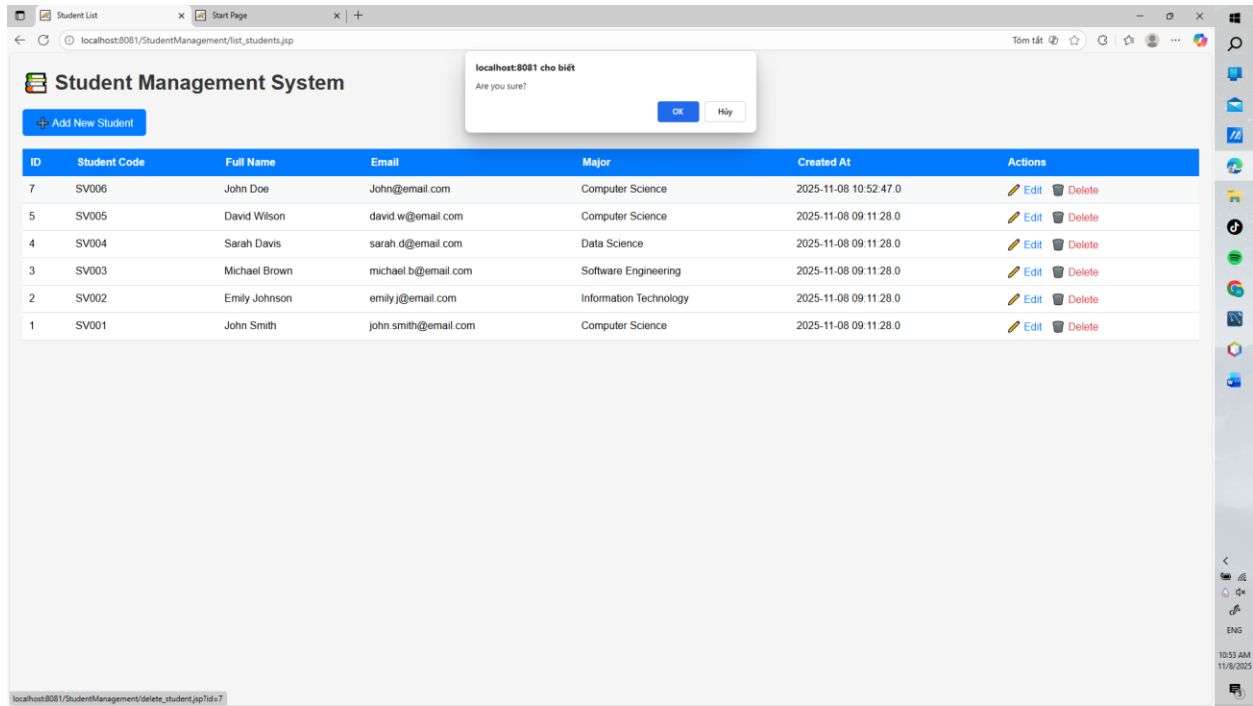
SET full\_name = ?, email = ?, major = ?

WHERE id = ?

🔗 Redirects the user based on the result:

-  Success → `list_students.jsp?message=Student updated successfully`
-  Failure → `edit_student.jsp?id=...&error=Update failed`

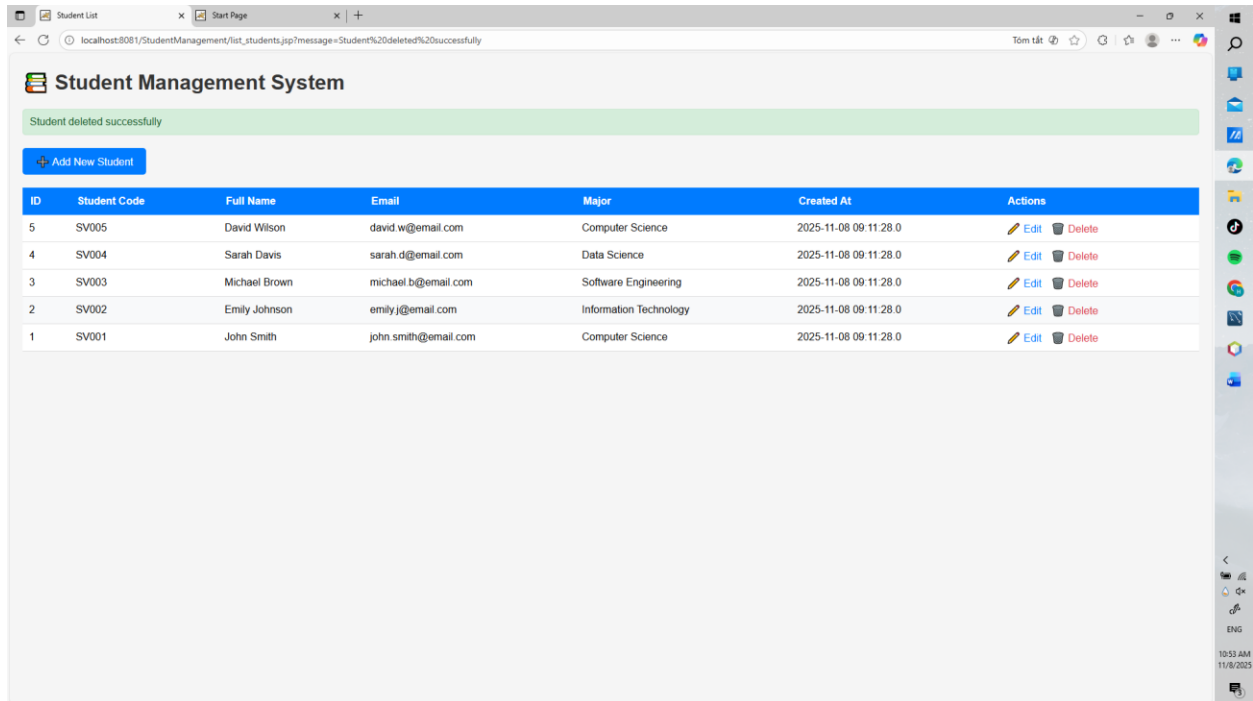
## EXERCISE 4: DELETE OPERATION



The screenshot shows a web browser window with the URL `localhost:8081/StudentManagement/list_students.jsp`. The page title is "Student Management System". A modal dialog box is displayed in the center, asking "Are you sure?" with "OK" and "Hủy" buttons. Below the dialog is a table with the following data:

ID	Student Code	Full Name	Email	Major	Created At	Actions
7	SV006	John Doe	Johni@email.com	Computer Science	2025-11-08 10:52:47.0	<a href="#">Edit</a> <a href="#">Delete</a>
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:11:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:11:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:11:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:11:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:11:28.0	<a href="#">Edit</a> <a href="#">Delete</a>

The status bar at the bottom of the browser shows the URL `localhost:8081/StudentManagement/delete_student.jsp?id=7`.



User clicks the “Delete” link/button in list\_students.jsp.

The system passes the student’s id to delete\_student.jsp.

The script:

- Checks that id is valid.
- Connects to the MySQL database.
- Executes the query:
- DELETE FROM students WHERE id = ?

Depending on the result:

- Success → Redirects to list\_students.jsp?message=Student deleted successfully
- Not found → Redirects with list\_students.jsp?error=Student not found