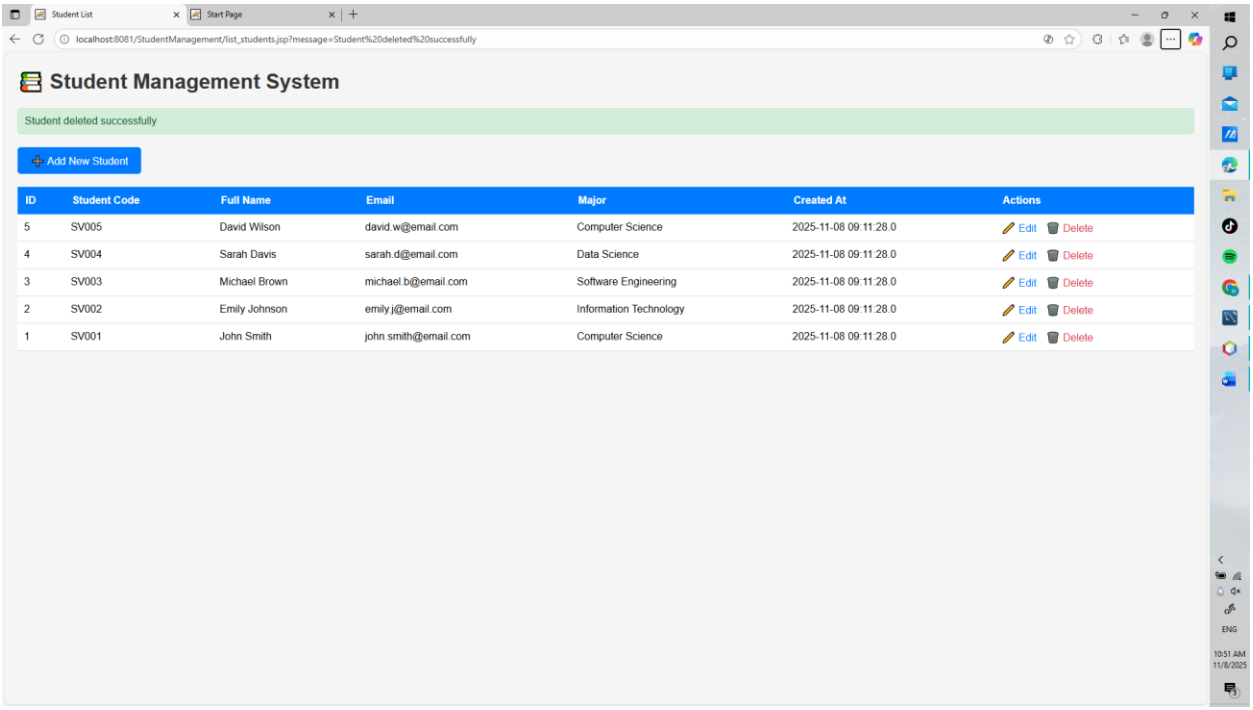
EXERCISE 1: SETUP AND DISPLAY



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.

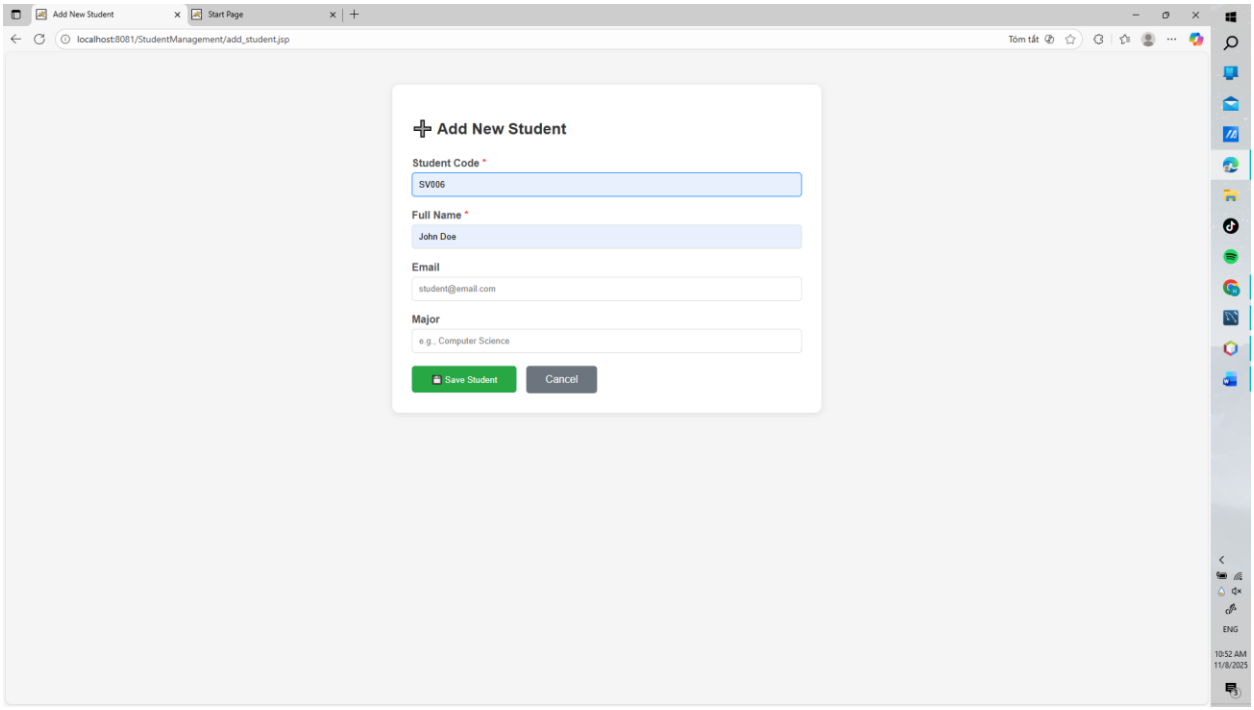
5. **Message Display**
   The page checks for URL parameters message and error to display success or error notifications after user actions (e.g., after adding or deleting a student).

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



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 type="email").

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.

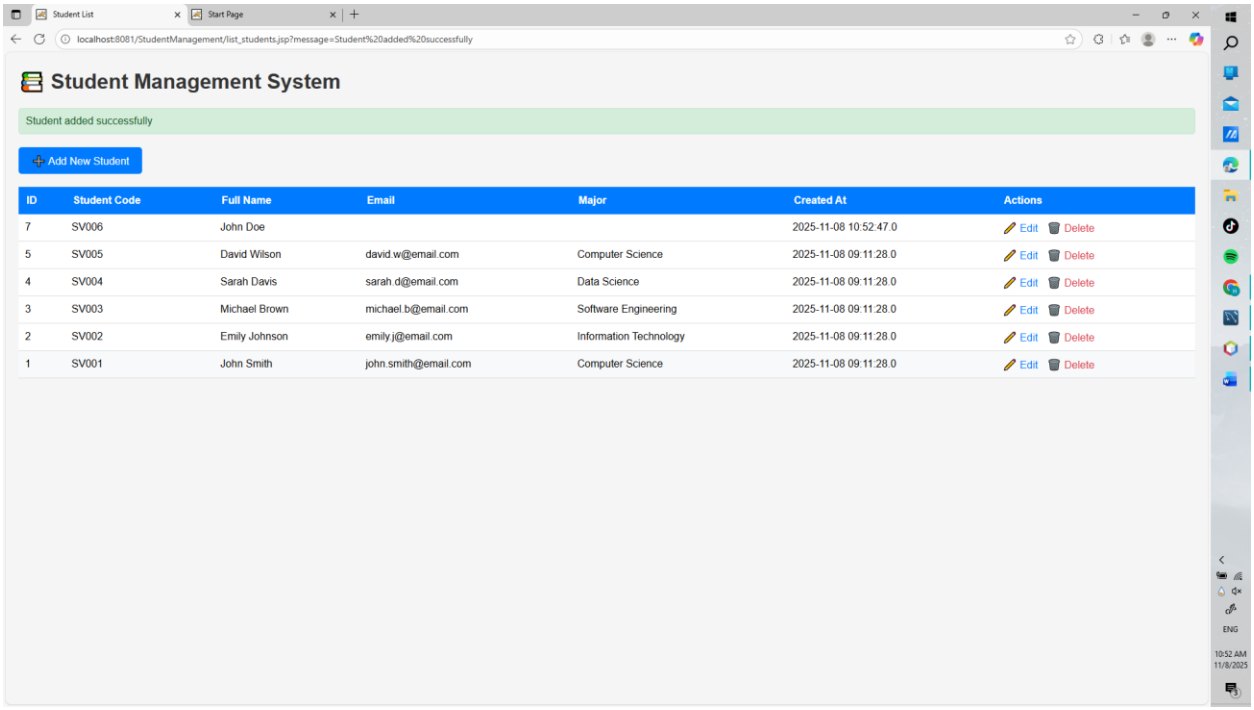
Connects to the student_management database.

Insert Operation

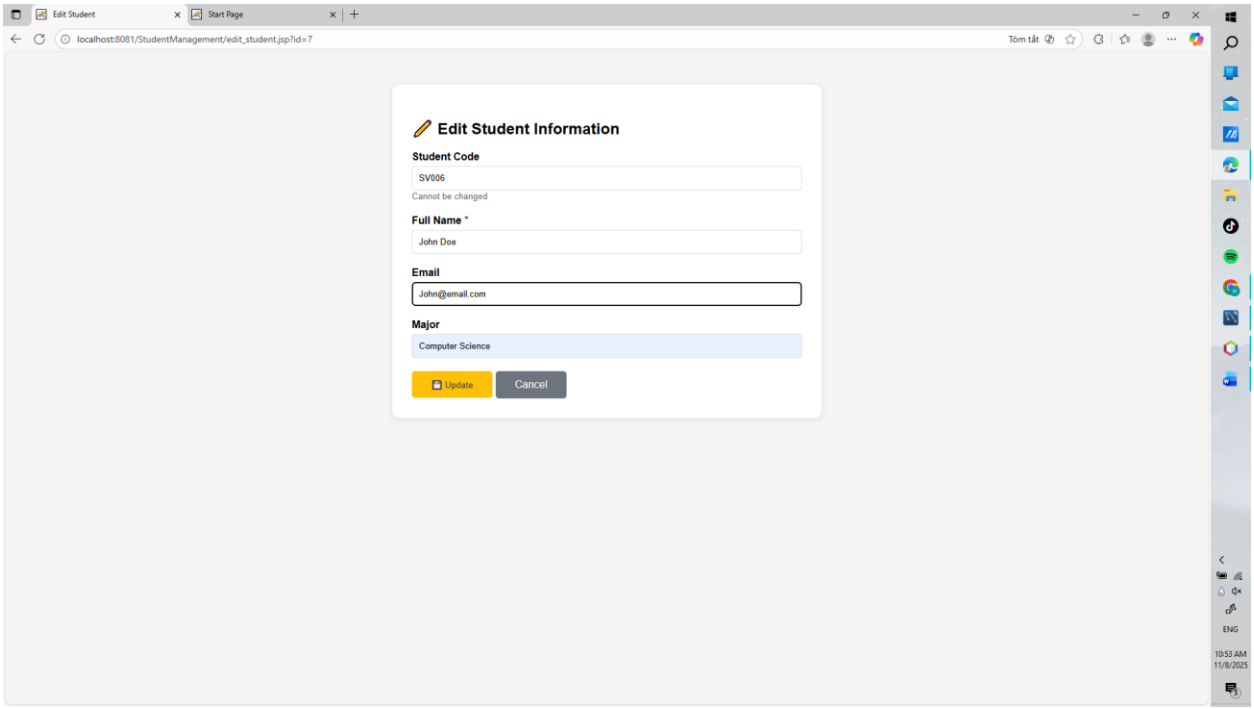
Redirect Based on Result:

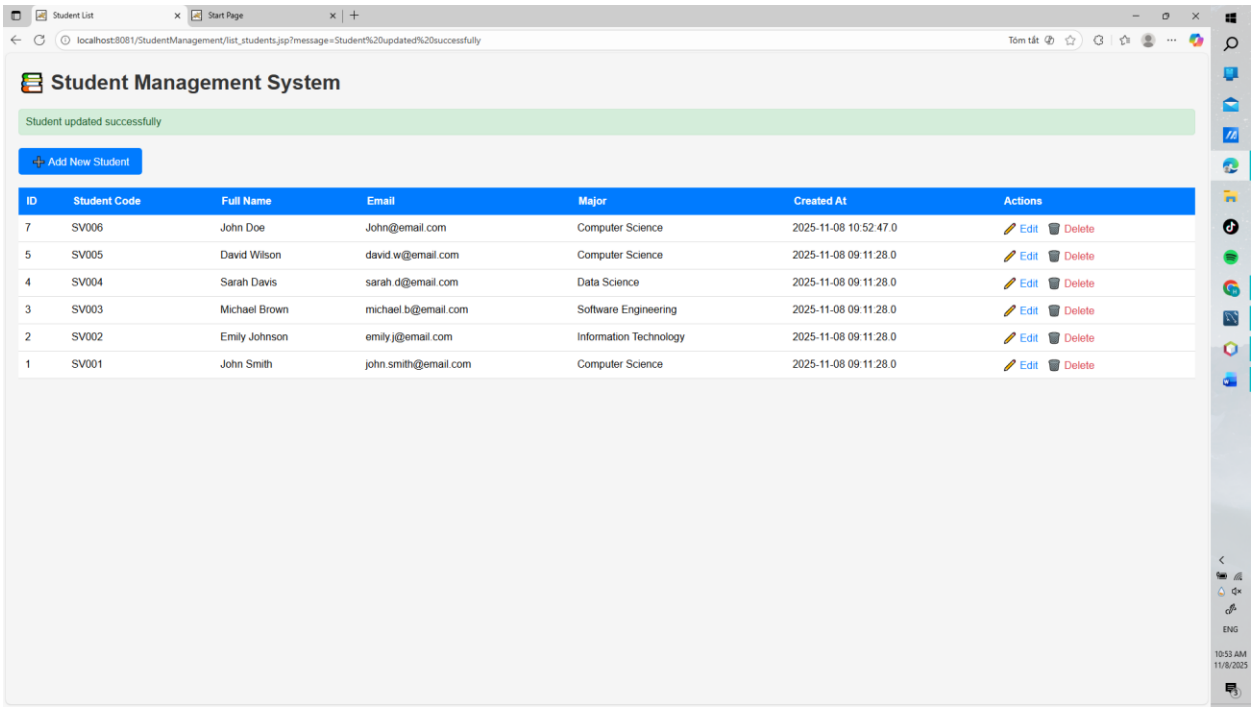
- Success → redirects to:
- list_students.jsp?message=Student added successfully
- Failure → redirects to:

add_student.jsp?error=Failed to add student



EXERCISE 3: UPDATE OPERATION





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

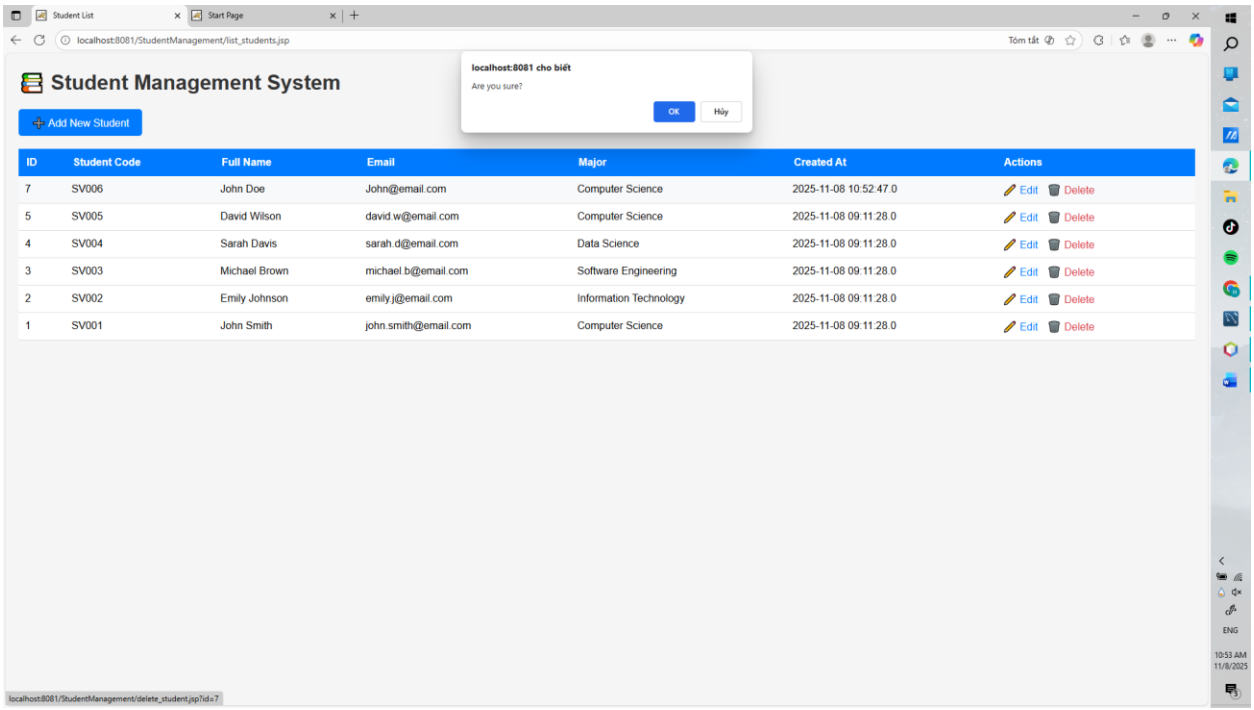
SET full_name = ?, email = ?, major = ?

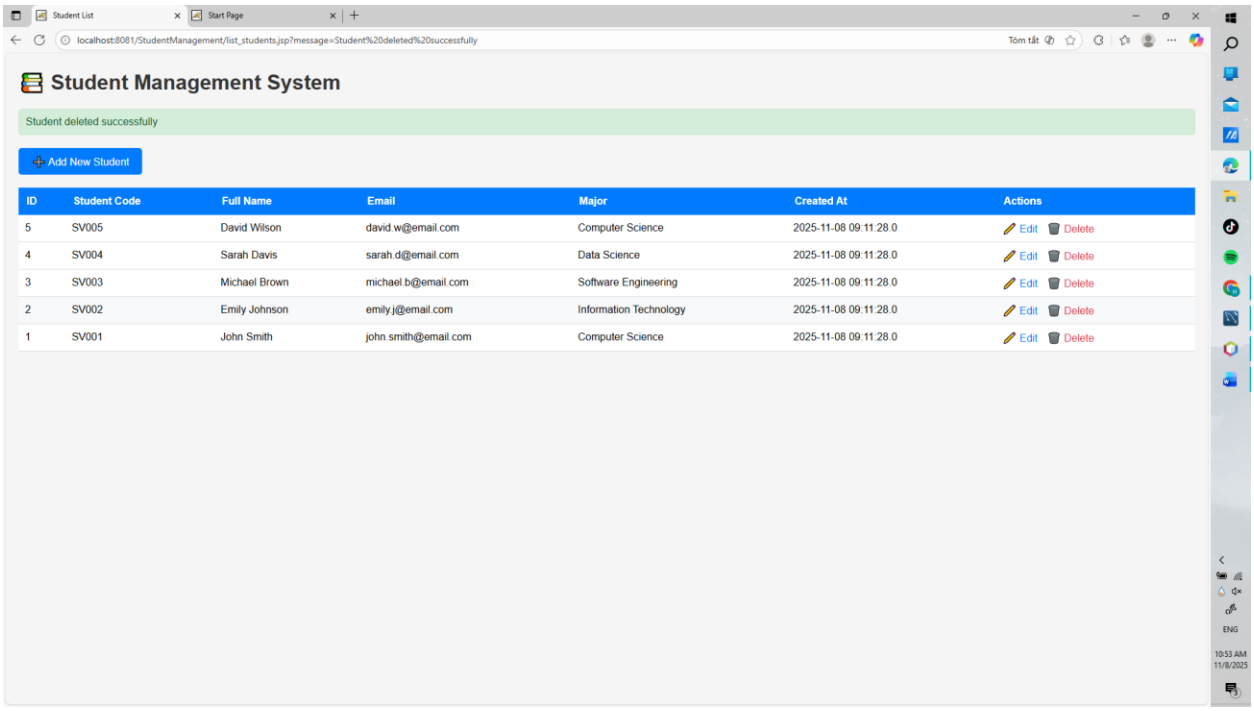
WHERE id = ?

 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





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