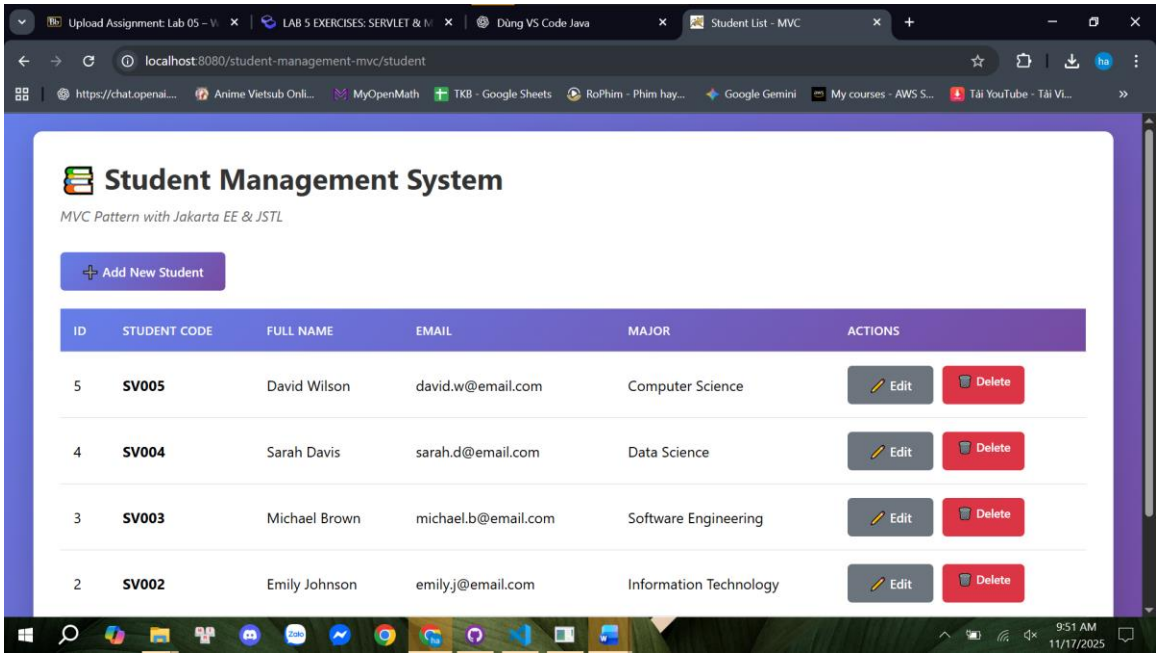
Lab 05 – Nguyễn Hà Thanh – ITITIU21144

1. LIST – Navigate to /student

When the user navigates to /student without providing an action parameter, the controller defaults the action to 'list':



The controller then calls listStudents(), which retrieves all students from the database:

List<Student> students = studentDAO.getAllStudents();

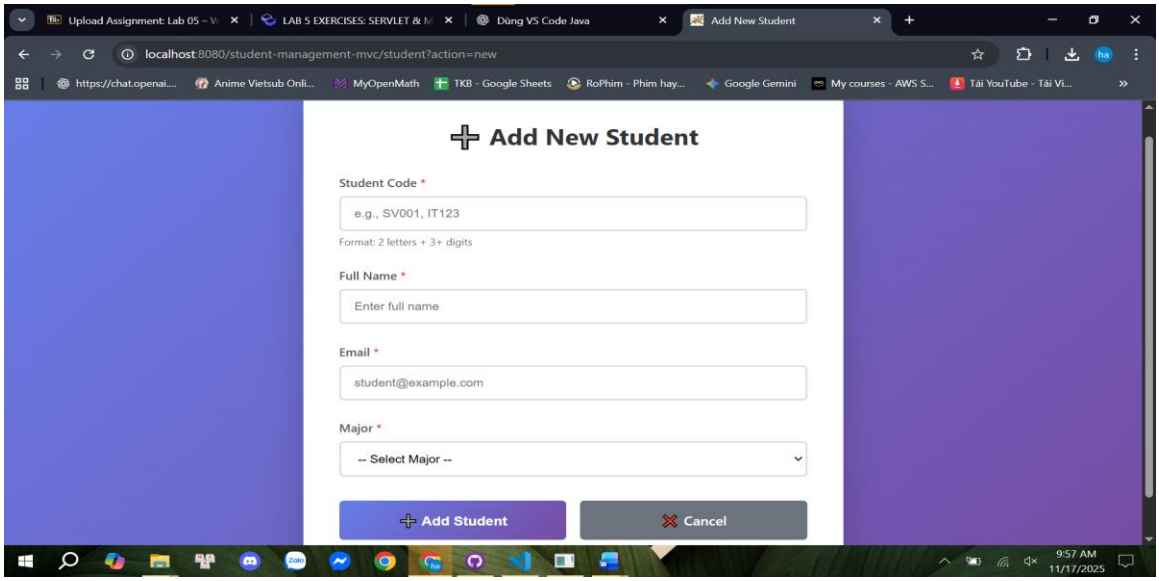
“request.setAttribute("students", students);”

Finally, it forwards the request to student-list.jsp:

student-list.jsp loops through the list and displays each student in a table. This shows all existing students retrieved from the database.

2. ADD – Add New Student

When the user clicks “Add New Student”, the browser goes to: “/student?action=new”



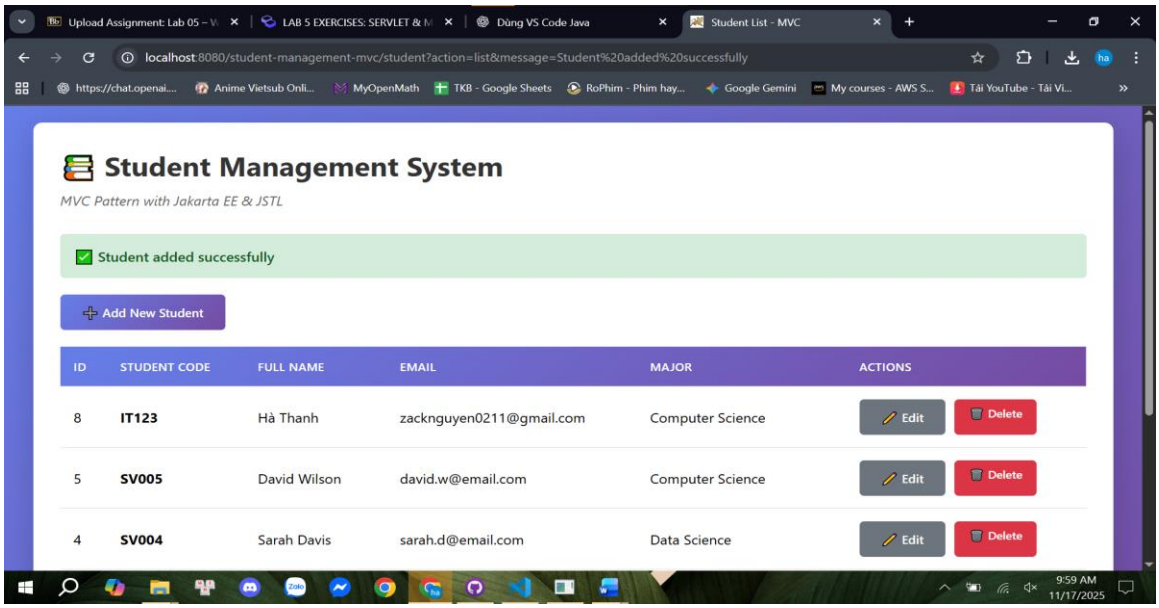
The controller forwards the request to student-form.jsp, showing an empty form.

When the user submits the form, a POST request is sent to: “/student?action=insert”

The controller extracts form data: “String studentCode = request.getParameter("studentCode");”

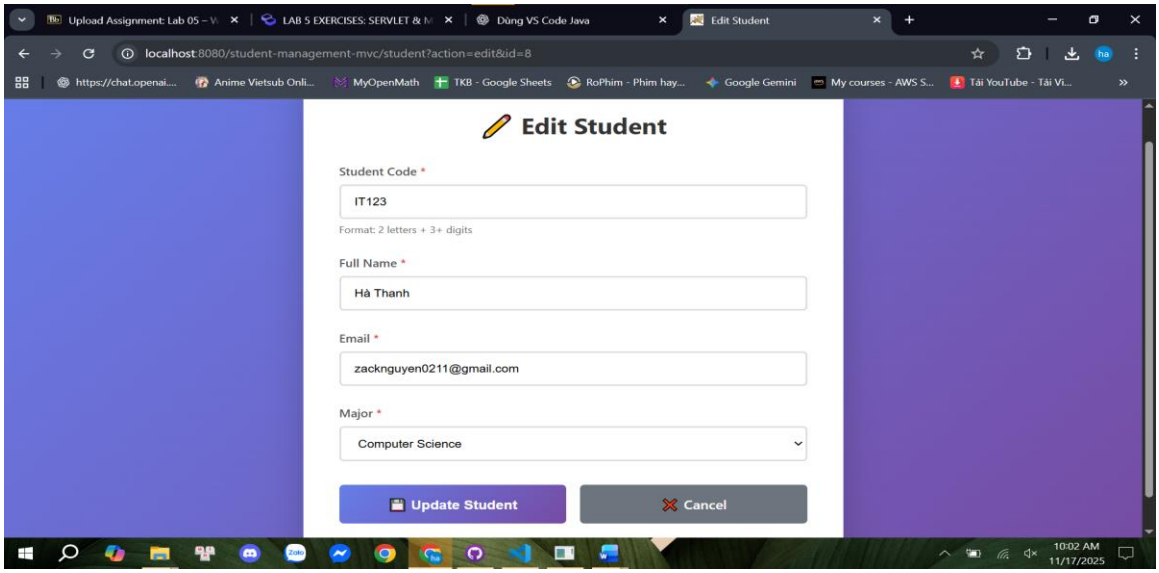
Creates a new Student object and calls DAO: “studentDAO.addStudent(newStudent);”

A SQL INSERT is executed, and the controller redirects back to the list with a success message.



3. EDIT – Edit Existing Student

When clicking Edit, the browser navigates to: ”/student?action=edit&id=<studentId>”



The controller loads the student: “Student existingStudent = studentDAO.getStudentById(id);”

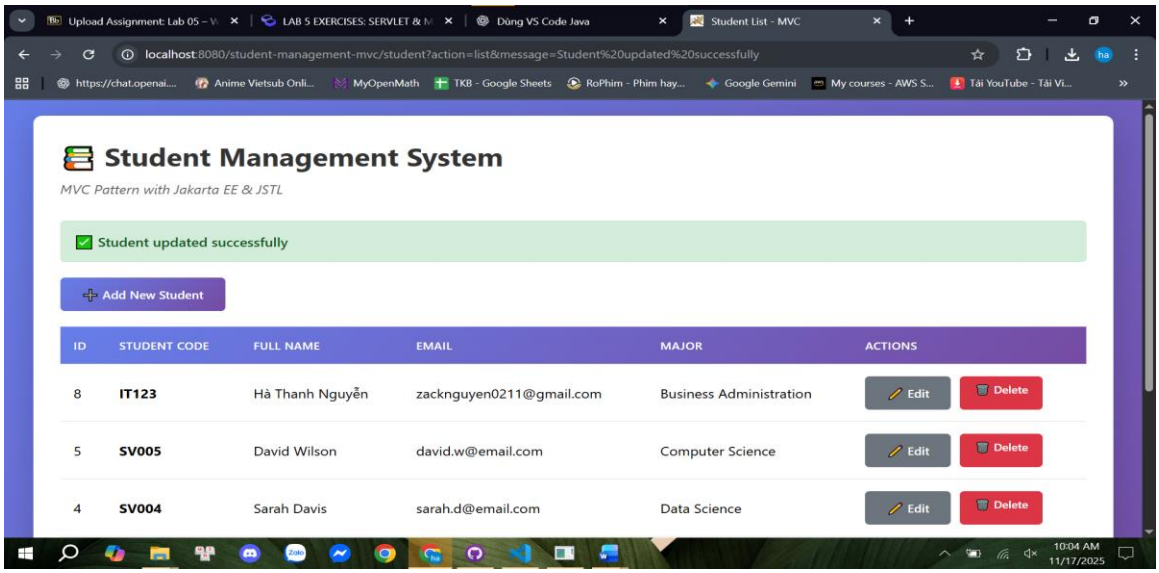
request.setAttribute("student", existingStudent);

Then forwards to student-form.jsp, where fields are pre-filled.

After submitting, a POST is sent to: “/student?action=update”

The controller reads updated fields, calls: “studentDAO.updateStudent(student);”

This executes an UPDATE SQL statement and redirects back to the list with an update message.



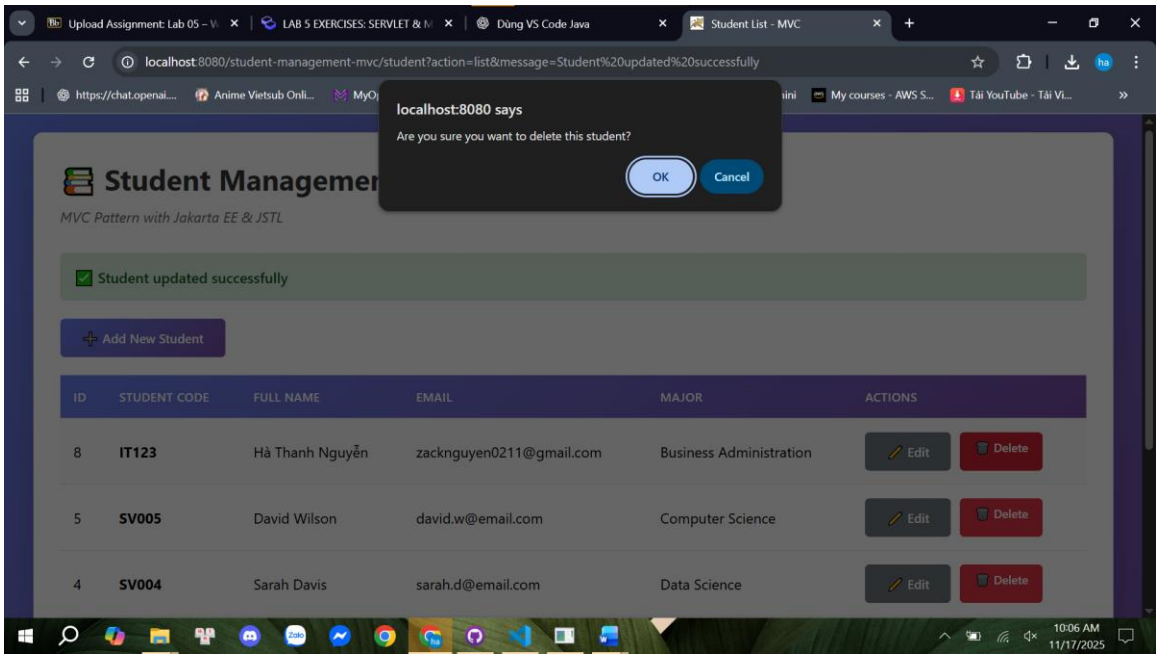
4. DELETE – Remove Student

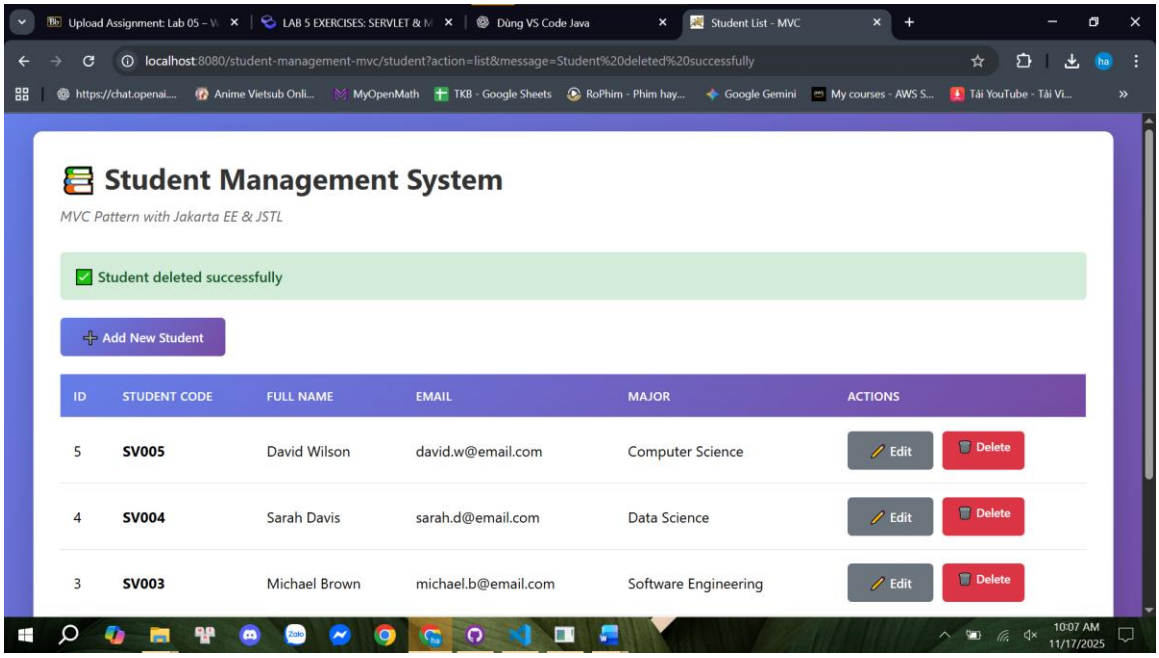
Clicking Delete sends: ”/student?action=delete&id=<studentId>”

The controller calls: ‘studentDAO.deleteStudent(id);”

The DAO executes a SQL DELETE statement, and the controller redirects back with a deletion message.

If all students are deleted, the list will be empty.





5. EMPTY STATE – No Students Found

When all students have been deleted, StudentDAO.getAllStudents() returns an empty list.

The JSP checks:

“<c:if test="${empty students}">

No students found

</c:if>”

The UI displays a friendly message indicating that the system contains no students.