**Name:** Phạm Gia Phúc
**ID:** ITCSIU22178

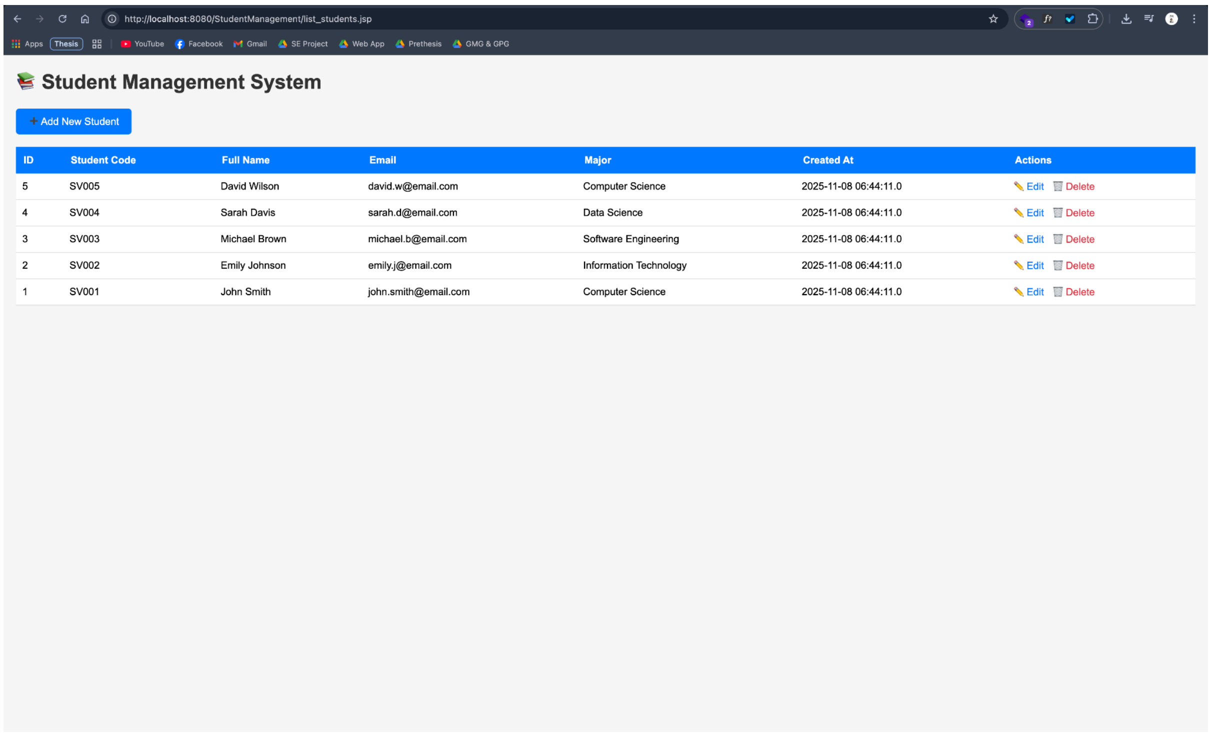
# Web App Development Laboratory - Lab 04

**GitHub Link:** [https://github.com/phamgiaphuc/student-management-jsp](https://github.com/phamgiaphuc/student-management-jsp)

## Exercise 1: Setup and display

### Task 1.1: Project Setup

- After setting up the project, the view when run the project:

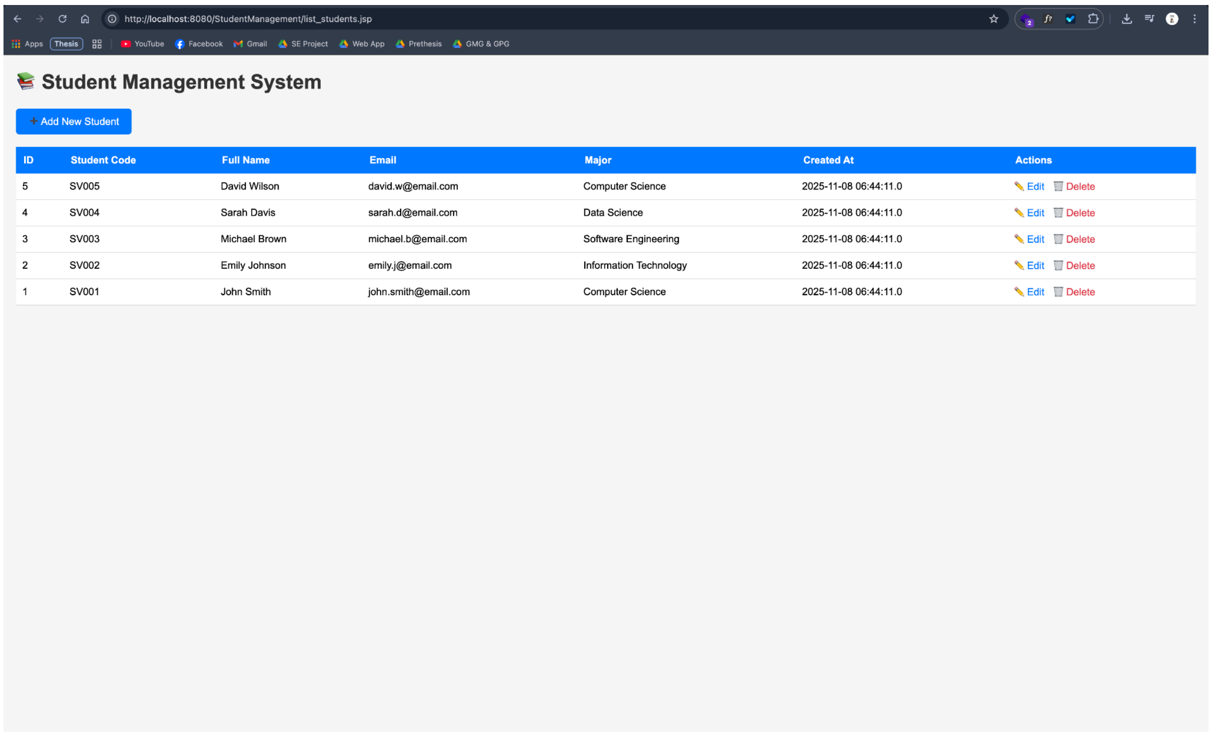


Student Management System

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |

### Task 1.2: Display Student List

- 5 sample students:


http://localhost:8080/StudentManagement/list_students.jsp

Student Management System

+ Add New Student

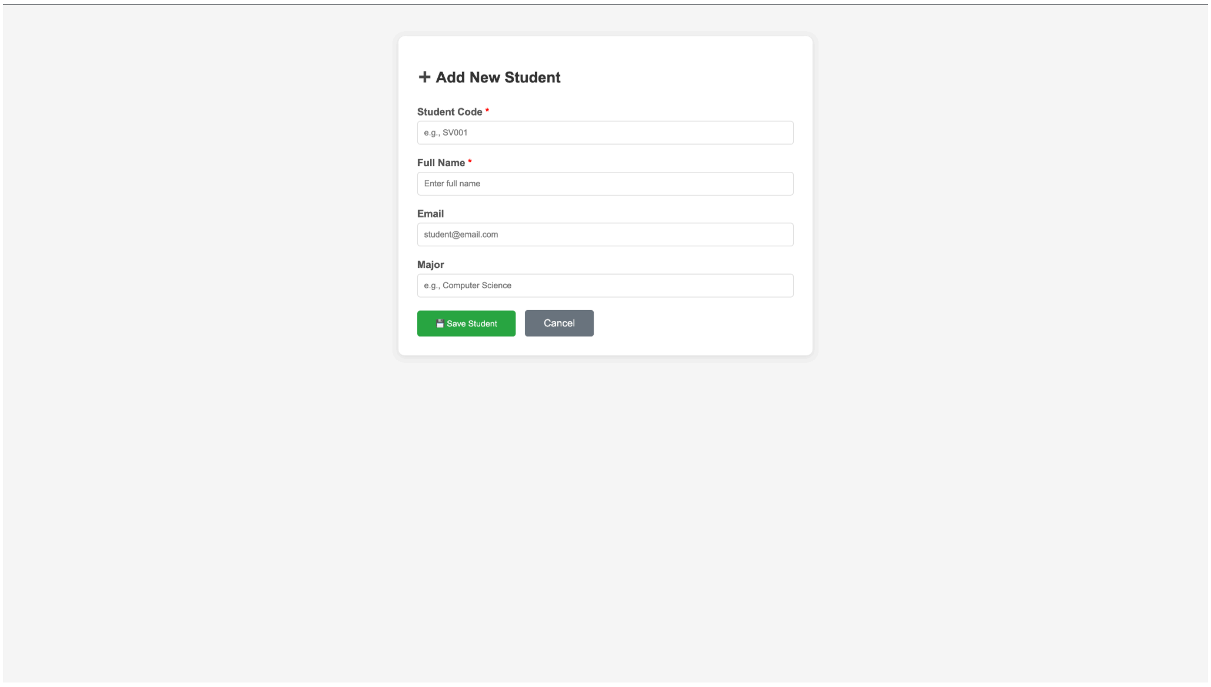
| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | Edit Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | Edit Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | Edit Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | Edit Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | Edit Delete |

→ Explanation: The database connection is established and a prepared statement to get the list of students is executed. A list of students will be rendered into an HTML file and displayed in the browser.

### Exercise 2: Create operation

### Task 2.1: Create Add Student Form

- The create student form:


+ Add New Student

Student Code *
e.g., SV001

Full Name *
Enter full name

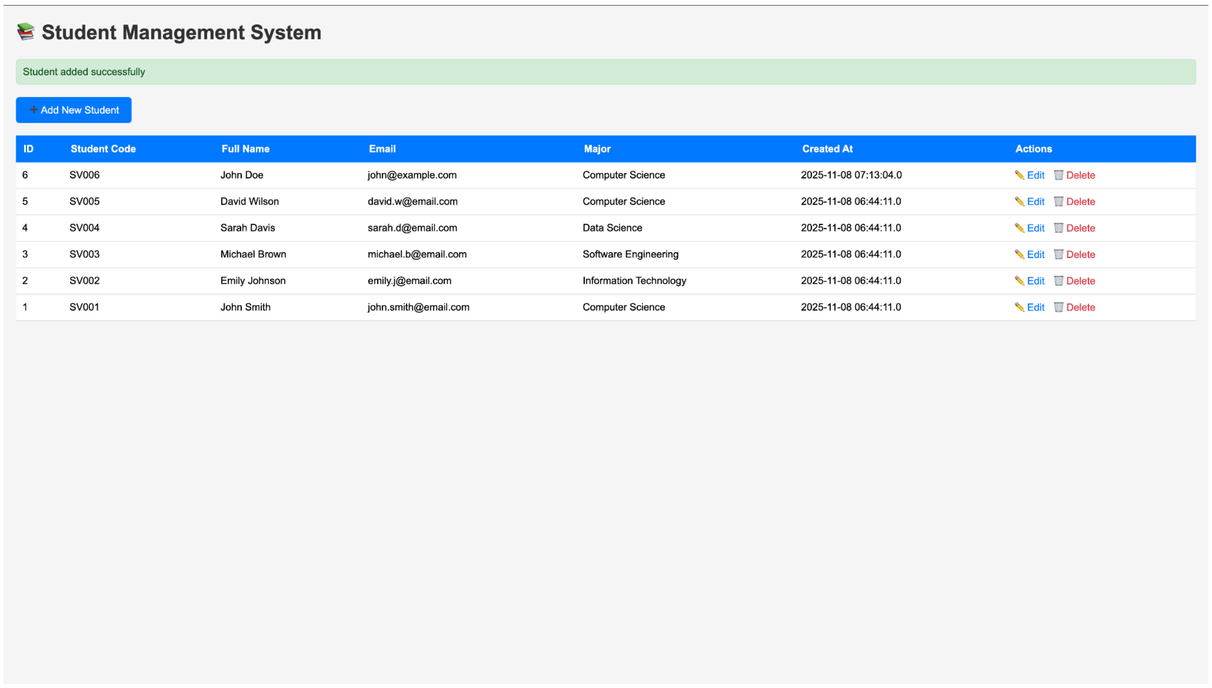
Email
student@email.com

Major
e.g., Computer Science

Save Student    Cancel

### Task 2.2: Process add student

- Test 1: Valid data



📚 **Student Management System**

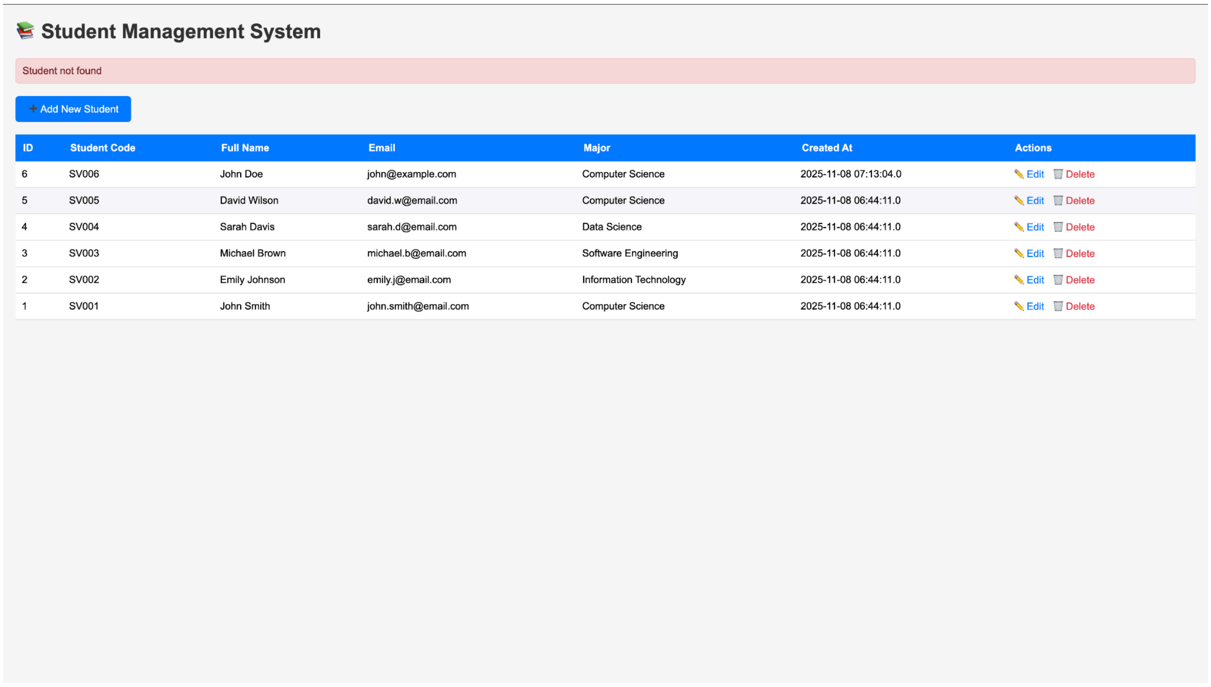
Student added successfully

＋ Add New Student

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 6 | SV006 | John Doe | john@example.com | Computer Science | 2025-11-08 07:13:04.0 | ✏️ Edit 🗑 Delete |
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |

→ Explanation: When we fill all the data into the inputs in the form and click the submit button, the request with parameters is sent to the process_add url. In the **process_add.jsp** file, we get the values from request parameters, create a SQL prepared statement and execute the statement to insert the student into the database.

- Test 2: After testing the url **http://localhost:8080/StudentManagement/edit_student.jsp?id=999**, the view:



📚 **Student Management System**

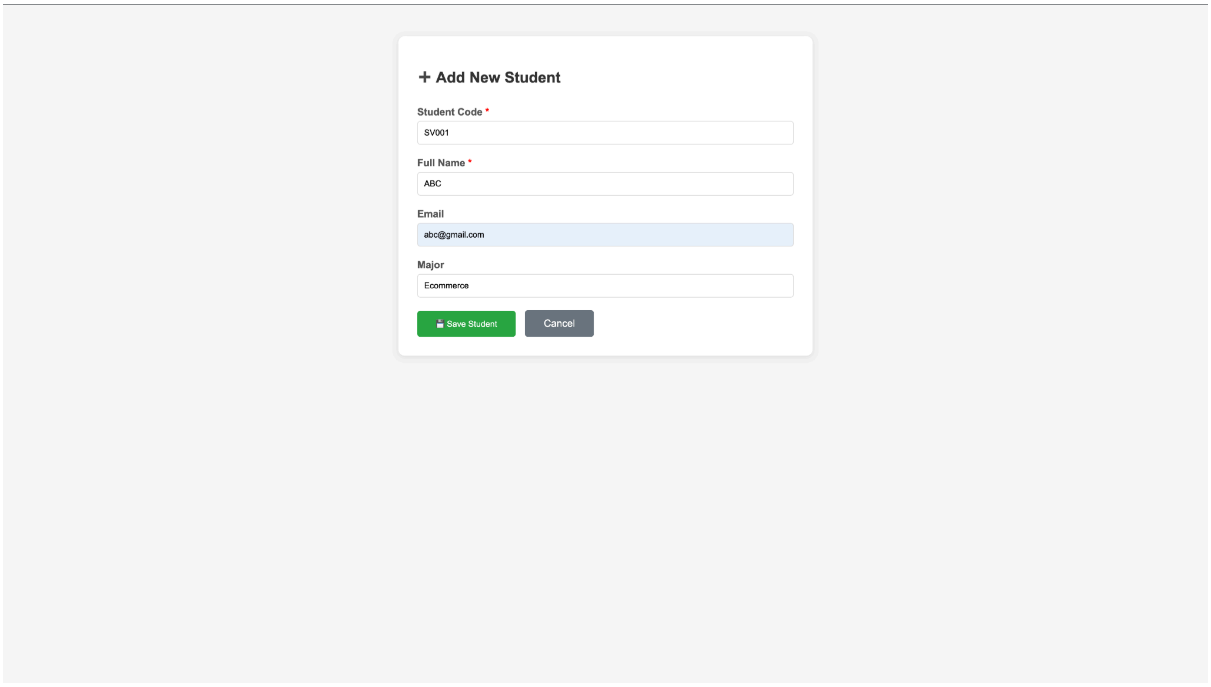
Student not found

＋ Add New Student

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 6 | SV006 | John Doe | john@example.com | Computer Science | 2025-11-08 07:13:04.0 | ✏️ Edit 🗑 Delete |
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑 Delete |

→ Explanation: When we access the testing url with invalid id, we will execute to get the current student’s information by querying the data from the database. If the id param is invalid or the student is not found, it will redirect to the list_students page with error “**Invalid student ID**”, “**Invalid ID format**” and “**Student not found**”

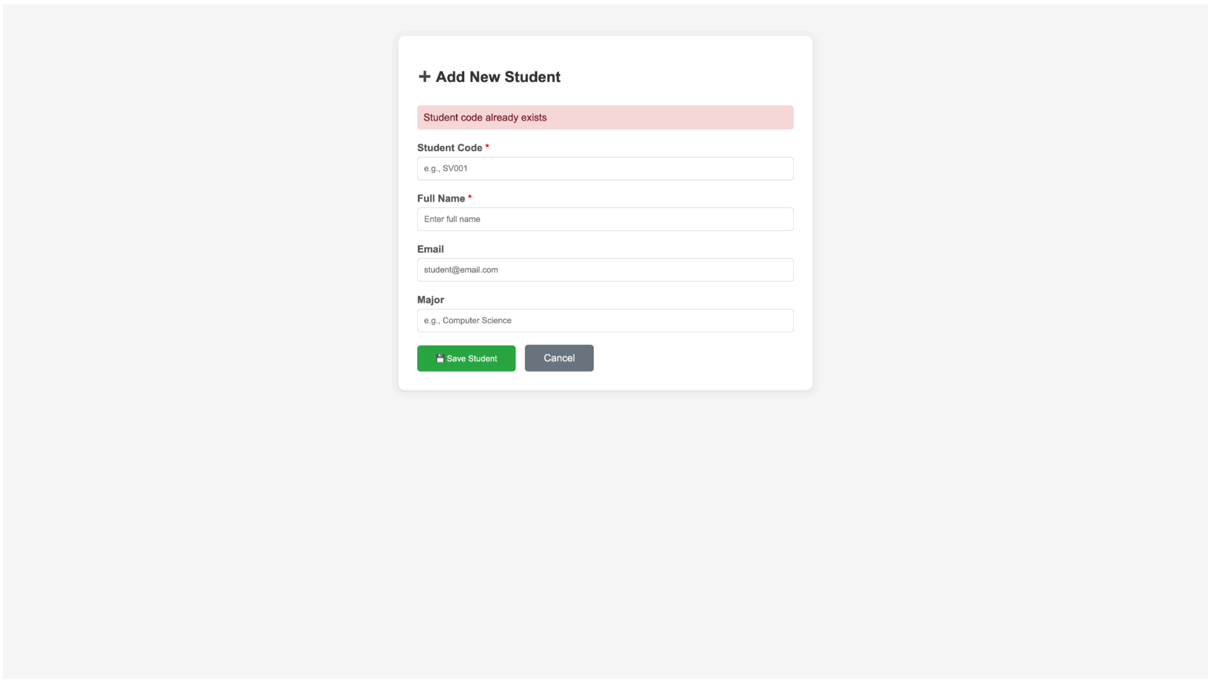
- Test 3:
  + Test data:



**+ Add New Student**

| | |
|---|---|
| Student Code * | SV001 |
| Full Name * | ABC |
| Email | abc@gmail.com |
| Major | Ecommerce |

  + After submitting, the view:



**+ Add New Student**

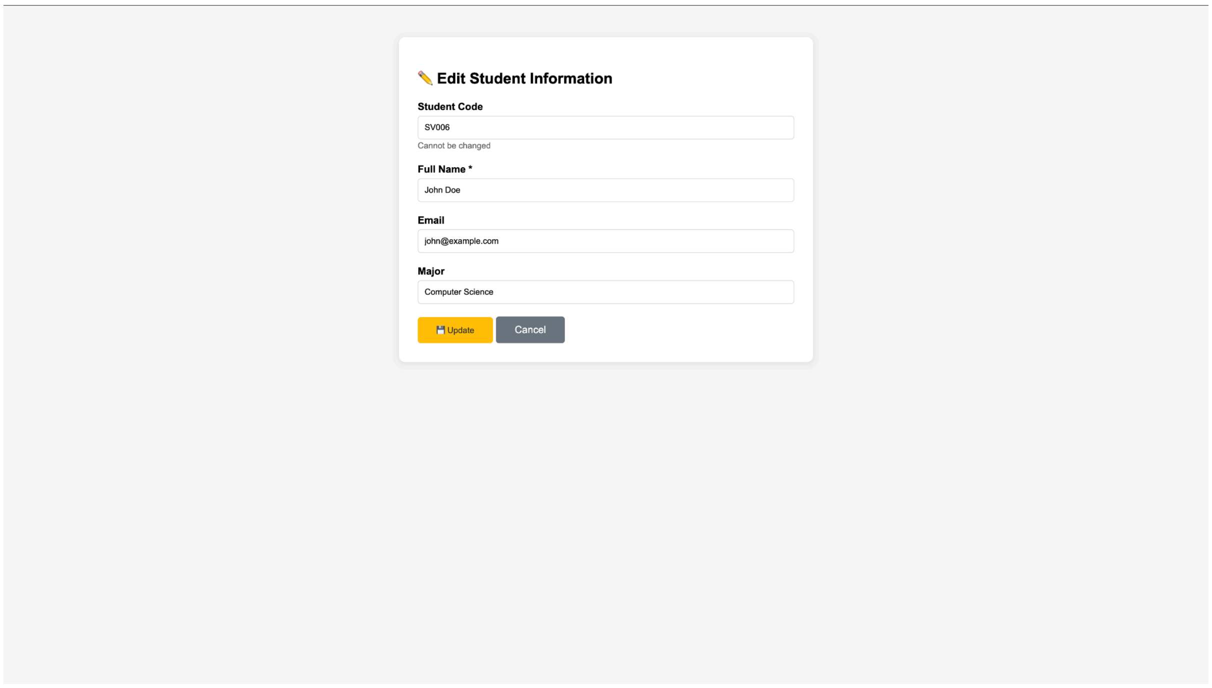
Student code already exists

| | |
|---|---|
| Student Code * | e.g., SV001 |
| Full Name * | Enter full name |
| Email | student@email.com |
| Major | e.g., Computer Science |

→ Explanation: When we insert the duplicated value in the field student code, the request will be sent to the **process_add** file. A prepared statement is executed during the request. If the database returns the errors and there is an error called “**Duplicate entry**” in the errors, it will redirect to the add_student page with error “**Student code already exists**”

**Exercise 3: Update operation**
**Task 3.1: Create Edit Form**

- The edit student form:

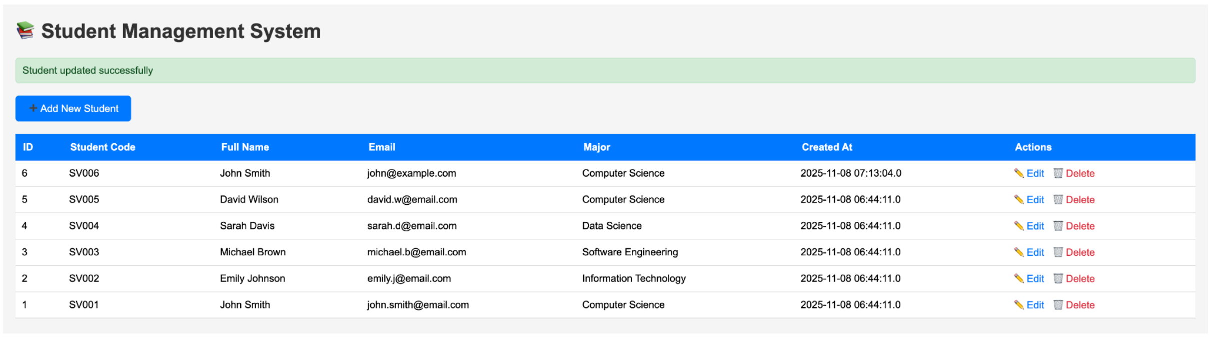


✏️ Edit Student Information

|  |  |
|---|---|
| Student Code | SV006 (Cannot be changed) |
| Full Name * | John Doe |
| Email | john@example.com |
| Major | Computer Science |

## Task 3.2: Process Update

- Test case 1: Valid update

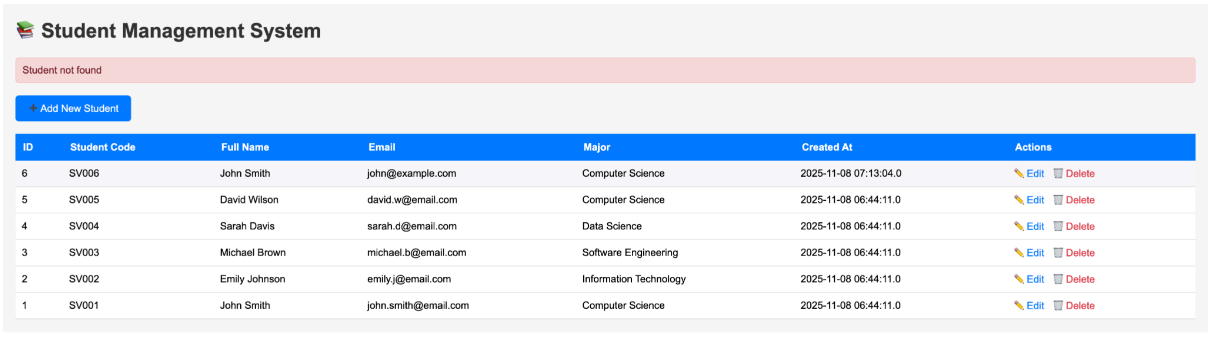


📚 Student Management System

Student updated successfully

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 6 | SV006 | John Smith | john@example.com | Computer Science | 2025-11-08 07:13:04.0 | ✏️ Edit 🗑️ Delete |
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |

- Test case 2: Invalid ID

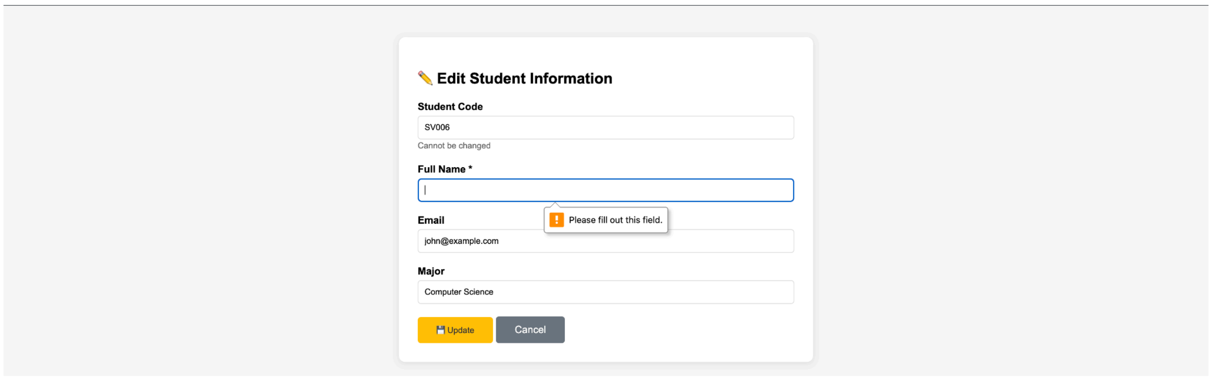


📚 Student Management System

Student not found

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 6 | SV006 | John Smith | john@example.com | Computer Science | 2025-11-08 07:13:04.0 | ✏️ Edit 🗑️ Delete |
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |

- Test case 3: Empty name



✏️ Edit Student Information

Student Code: SV006
Cannot be changed

Full Name *: 
Please fill out this field.

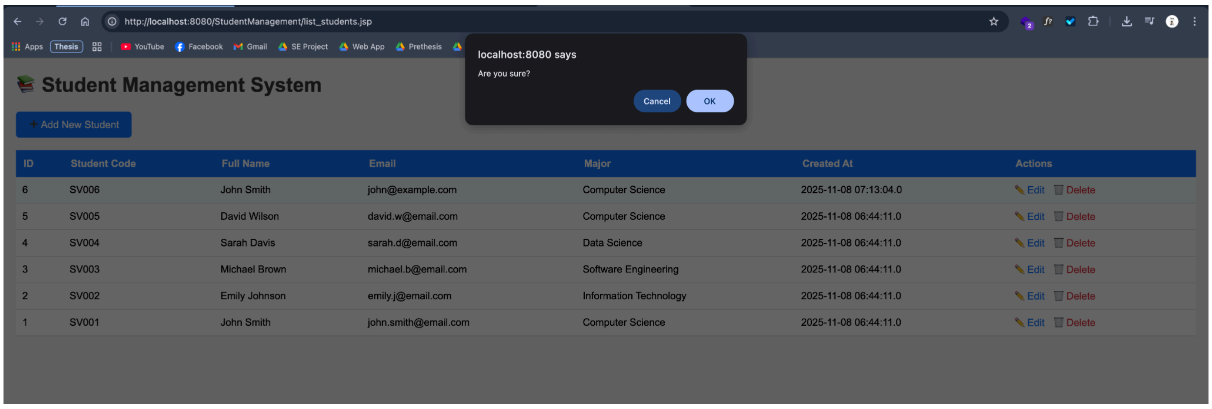
Email: john@example.com

Major: Computer Science

→ Explanation: When the edit form is submitted, the system reads all input values and executes an sql update statement to modify the student record. If the data is valid, the update succeeds, and the user is redirected to the list page with a message "**Student updated successfully**" If the student ID is invalid or not found, it redirects with "**Invalid student ID**" If a required field like name is empty, validation fails, and the form reloads showing "**Name is required**" This ensures accurate updates and maintains data integrity.

## Exercise 4: Delete operation

### Task 4.1: Implement delete



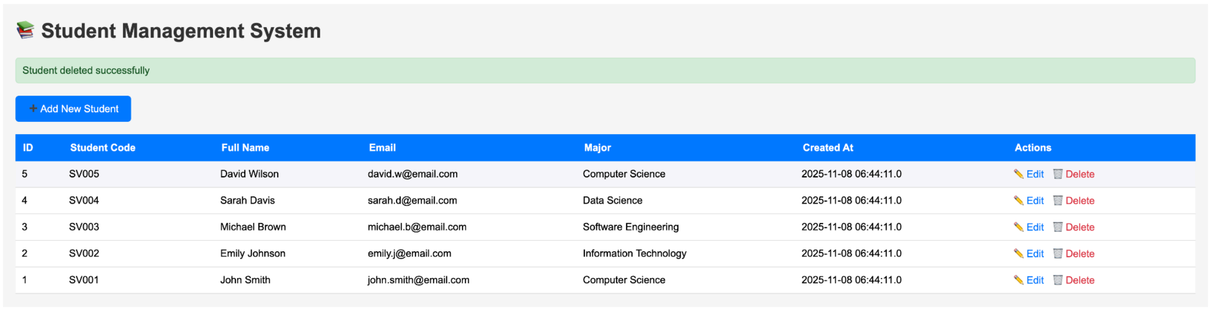
http://localhost:8080/StudentManagement/list_students.jsp

localhost:8080 says
Are you sure?

📚 Student Management System

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 6 | SV006 | John Smith | john@example.com | Computer Science | 2025-11-08 07:13:04.0 | ✏️ Edit 🗑️ Delete |
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |

### Task 4.2: Add Delete Links and Confirmation



📚 Student Management System

Student deleted successfully

| ID | Student Code | Full Name | Email | Major | Created At | Actions |
|---|---|---|---|---|---|---|
| 5 | SV005 | David Wilson | david.w@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 4 | SV004 | Sarah Davis | sarah.d@email.com | Data Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 3 | SV003 | Michael Brown | michael.b@email.com | Software Engineering | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 2 | SV002 | Emily Johnson | emily.j@email.com | Information Technology | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |
| 1 | SV001 | John Smith | john.smith@email.com | Computer Science | 2025-11-08 06:44:11.0 | ✏️ Edit 🗑️ Delete |

→ Explanation: A delete link is added next to each student record on the list page. When the user clicks the link, a JavaScript confirmation dialog appears asking "**Are you sure you want to delete this student?**" If the user confirms, the request is sent to **process_delete.jsp** with the student's ID to perform the deletion. If the user cancels, no action is taken. This confirmation step helps prevent accidental deletions and ensures that data is only removed when the user explicitly approves the action.