**Lab 5**

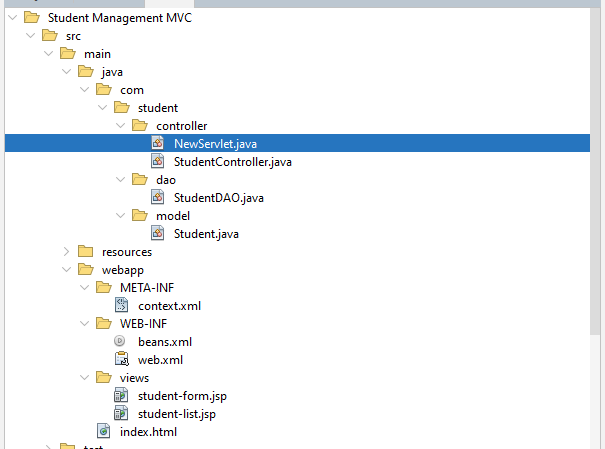
**Name:** Trần Quốc Nam

**ID:** ITITIU21250

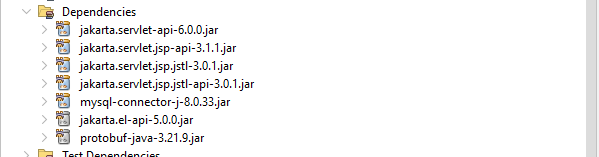
**Exercise 1: Project Setup & Model Layer**

Task 1.1: Create Project Structure

Project structure:

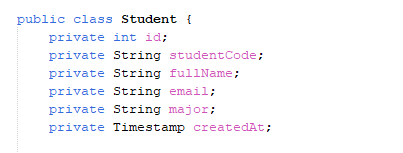


Libraries:

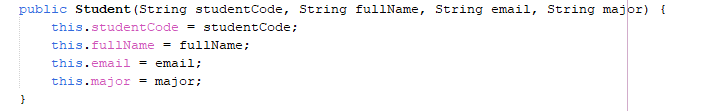


Task 1.2: Create Student JavaBean

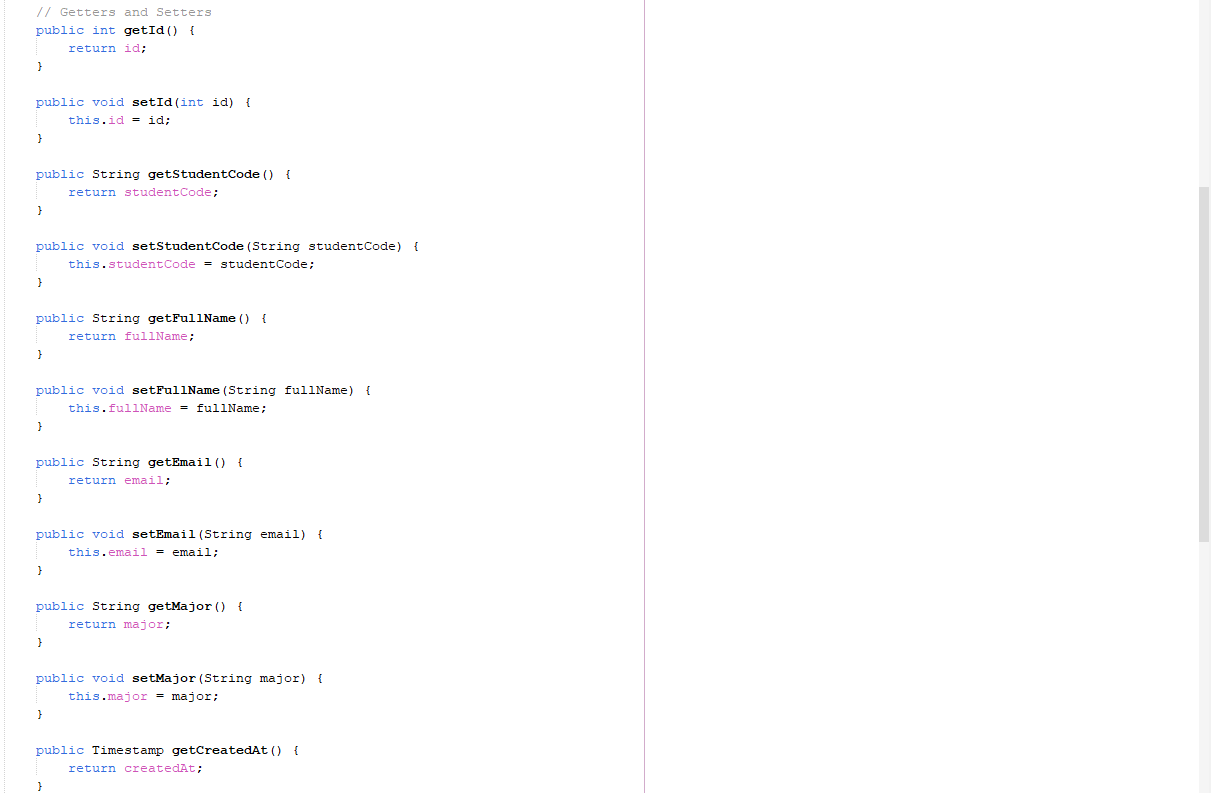
Attributes declared:



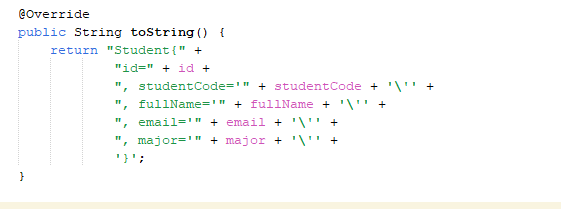
Constructor implemented:



Getters/ Setters



toString() implemented

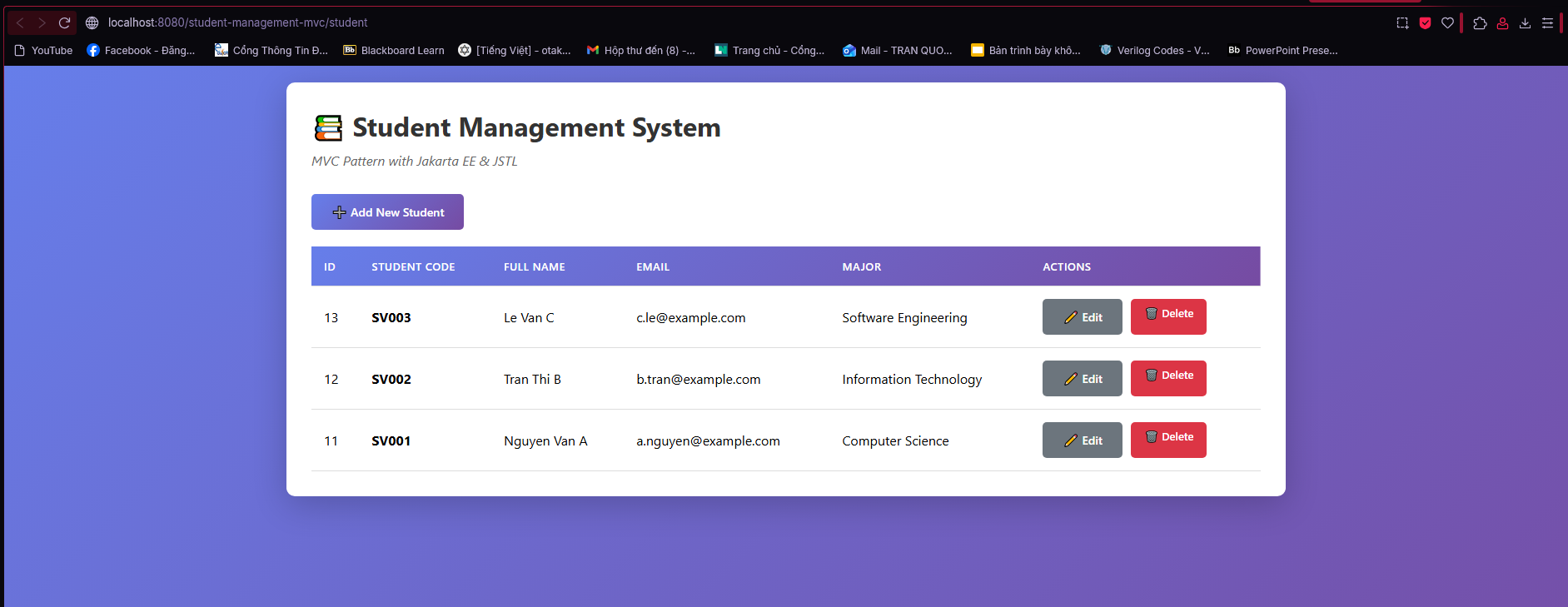


Task 1.3: Create StudentDAO

getConnection():

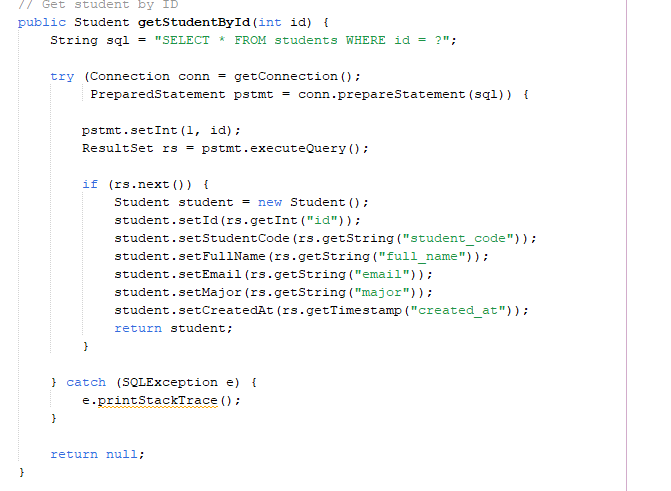


getAllStudents() returns data:



Check point 1:

For StudentDAO it has create the object Student and set all the attribute of the Student class using setters:

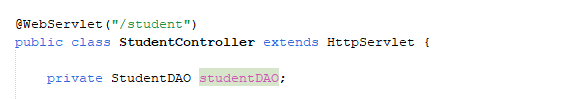




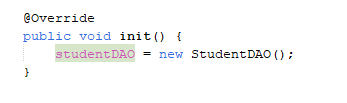
**Exercise 2: Controller Layer**

Task 2.1: Create Basic Servlet

Servlet configure:



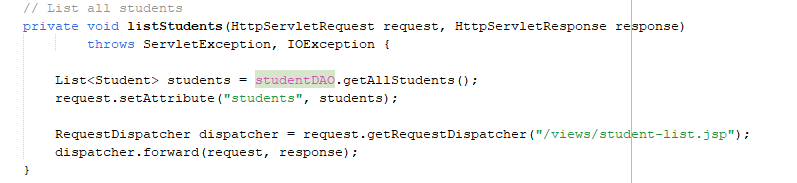
init() initializes DAO:



doGet() routes:



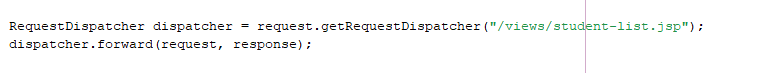
listStudents() gets data from DAO:



Sets request attribute:

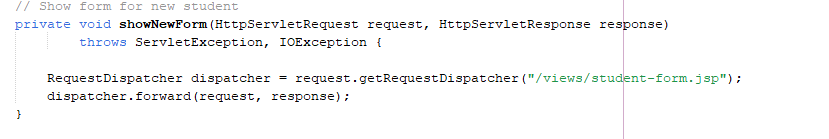


Forward to JSP:

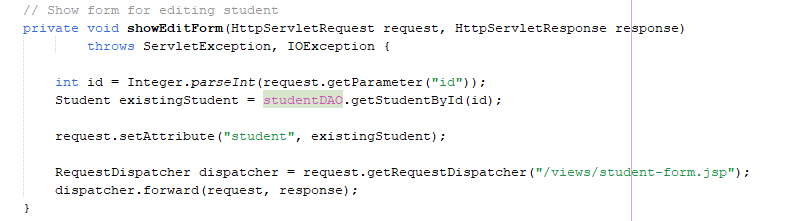


Task 2.2: Add More CRUD Methods

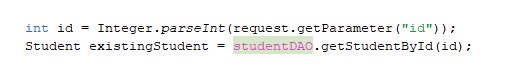
showNewForm() forwards to form:



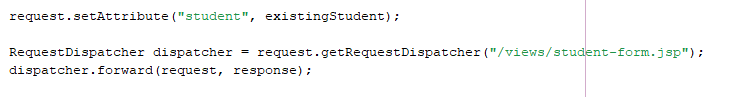
showEditForm() loads student and forwards:



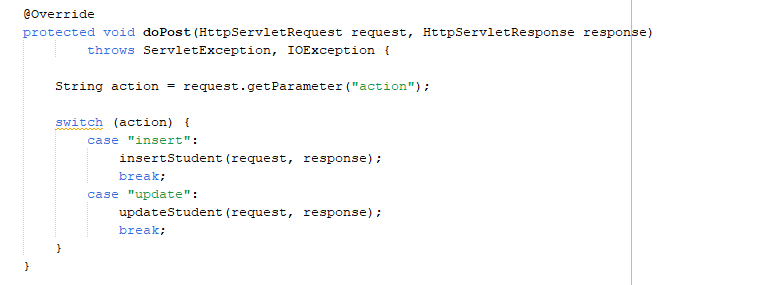
Loads student:



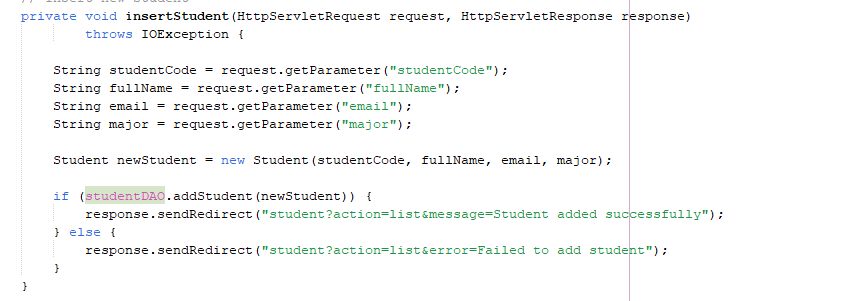
Forwards to form:



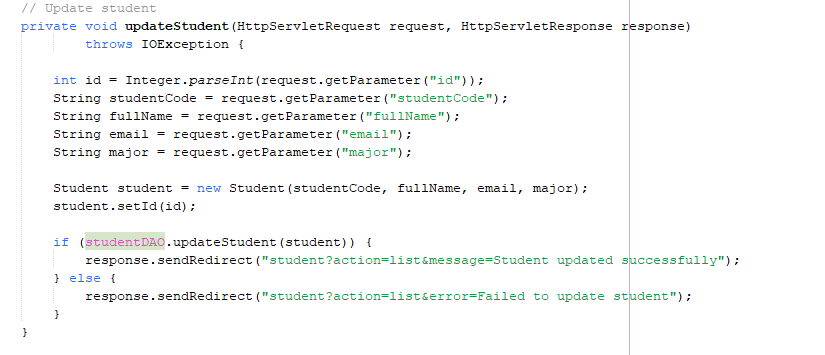
doPost() routes:



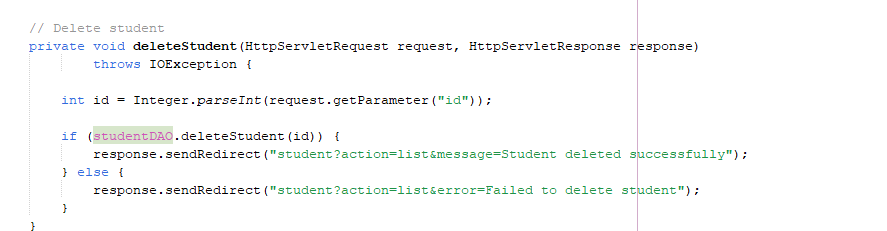
insertStudent() creates student:



updateStudent() modifies student:



deleteStudent() removes student:



Checkpoint #2:

**Scenario 1:** User clicks “Add New Student”

1. Browser 🡪 GET /student?action=new

2. Controller (**StudentController**) receives request and sets data as request attribue

* Extracts *action=new* parameter
* Calls *request.getRequestDispatcher()* and forward to *student-form.jsp*

3. View (JSP) display using form



4. Browser receives HTML response

**Scenario 2:** User clicks “Edit”

1. Browser 🡪 GET /student?action=edit

2. Controller (StudentController) receives request

* Extract *action=edit* parameter
* Calls *studentDAO.getStudentById()*

3. Model (StudentDAO) executes SQL

* Return *Student*

4. Controller sets data as request attribute

* Fowards to *student-form.jsp*

5. View (JSP) display using form



6. Browser receives HTML response

**Scenario 3:** User clicks “Add New Student” and fills the information and clicks “Add Student”

1. Browser 🡪 POST /student?action=insert

2. Controller (StudentController) receives request

* Extracts *action=insert* parameter
* Calls *studentDAO.addStudent()*

3. Model (StudentDAO) executes SQL

* Return True or False

4. View (JSP) displays using JSTL

*<c:forEach items="${students}">*

5. Browser receives HTML response and a message

**Scenario 4:** User clicks “Edit” and fills the information and clicks “Update Student”

1. Browser 🡪 POST /student?action=update

2. Controller (StudentController) receives request

* Extracts *action=update* parameter
* Calls *studentDAO.updateStudent()*

3. Model (StudentDAO) executes SQL

* Return True or False

4. View (JSP) displays using JSTL

*<c:forEach items="${students}">*

5. Browser receives HTML response and a message

**Scenario 5:** User clicks “Delete”

1. Browser 🡪 GET /student?action=delete

2. Controller (StudentController) receives request

* Extracts *action=delete* parameter
* Calls *studentDAO.deleteStudent()*

3. Model (StudentDAO) executes SQL

* Return True or False

4. View (JSP) displays using JSTL

*<c:forEach items="${students}">*

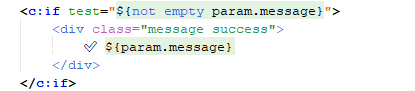
5. Browser receives HTML response and a message

**Exercise 3: View Layer with JSTL**

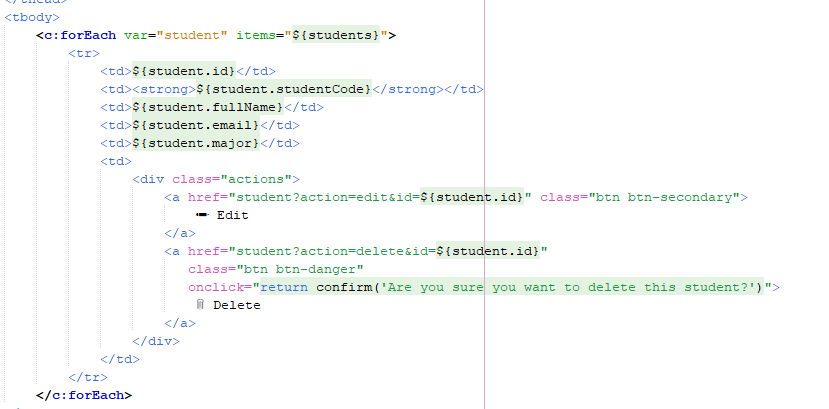
Task 3.1: Create Student List View

JSTL taglib included:

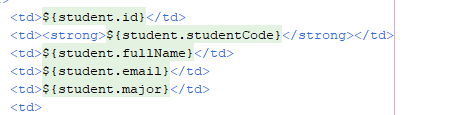
<c: if> for conditional display – will show message if exist



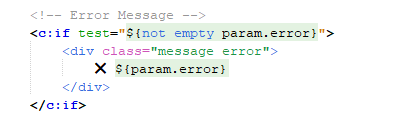
<c:forEach> for looping through students



EL for data access:

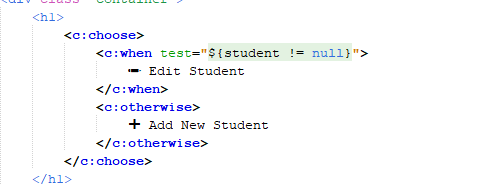


Empty list handled:

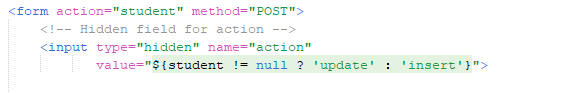


Task 3.2: Create Student Form View

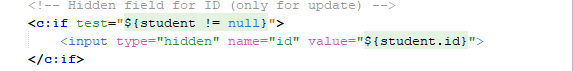
<c: choose> for title:



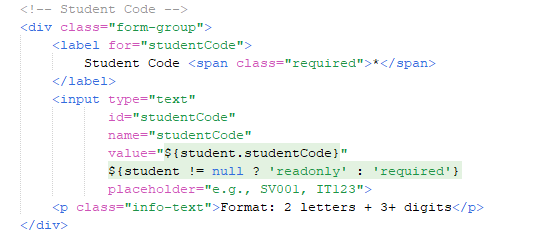
Dynamic action field:



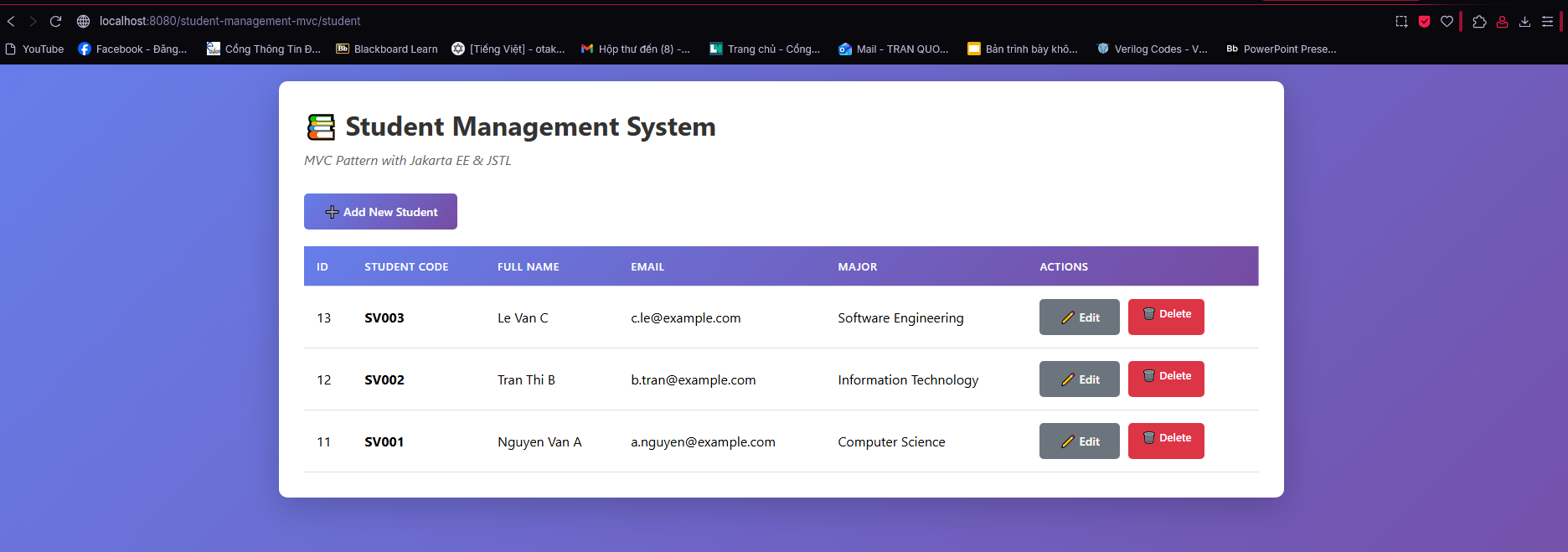
Conditional id field:



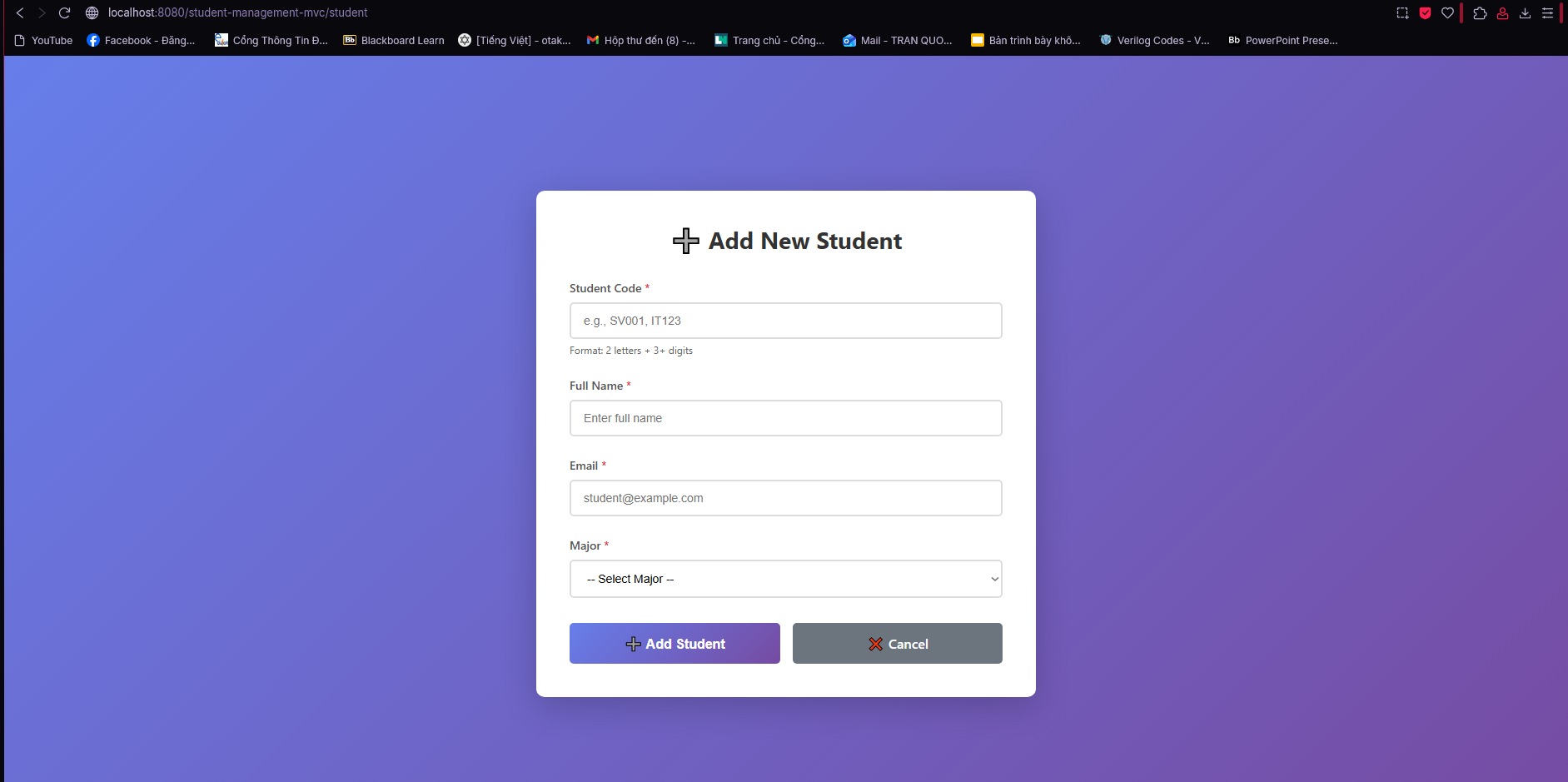
Pre-filled values:



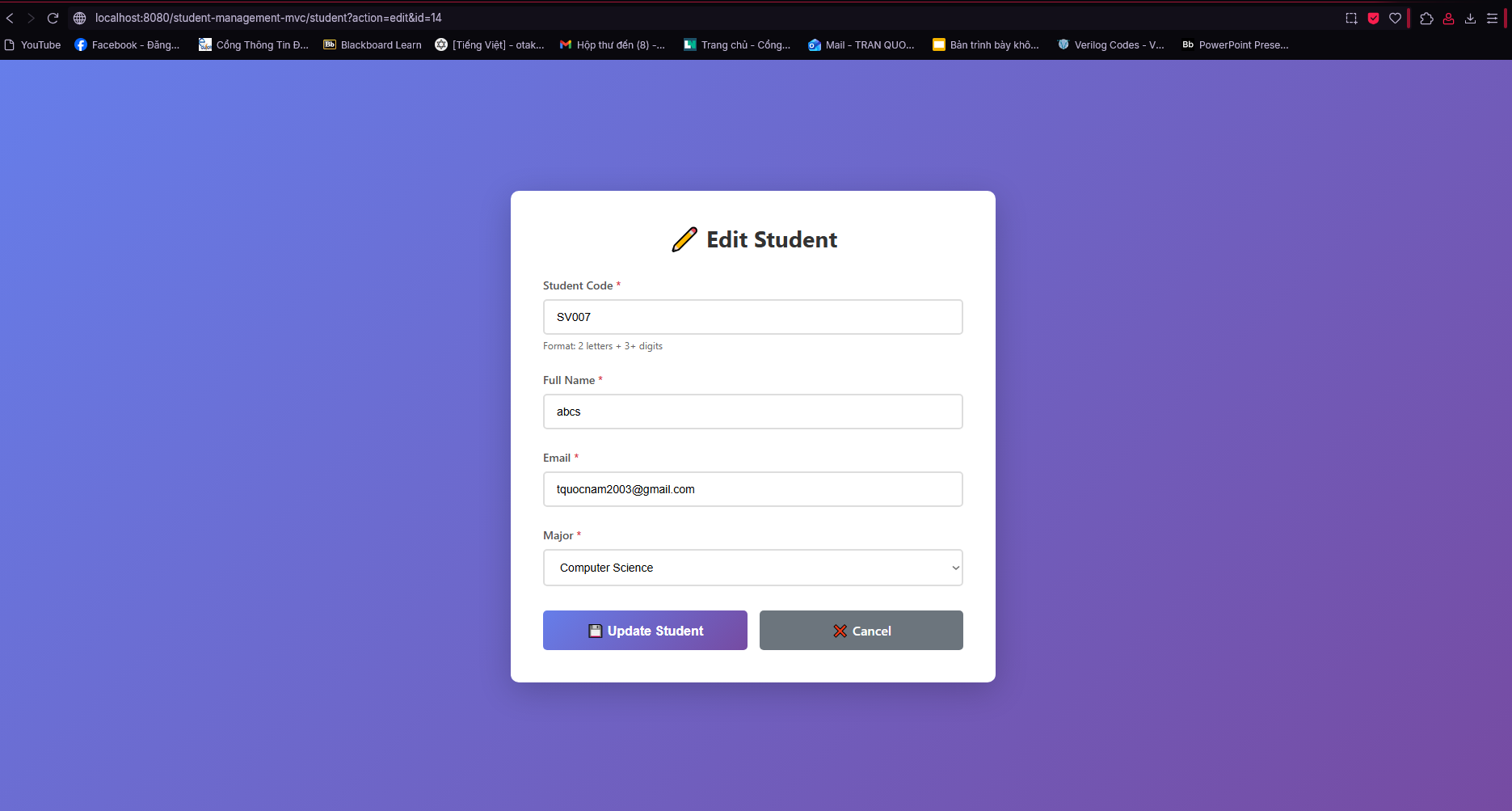
CheckPoint #3:

By default the web will show a list of student with edit/delete option for each student instance 

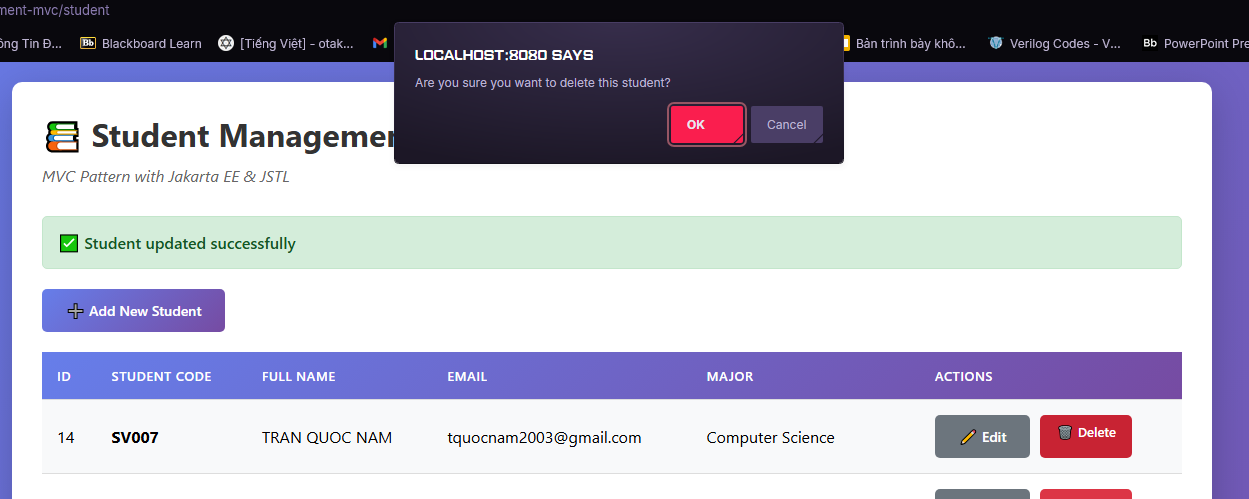
On clicking Add New Student



On clicking Edit student



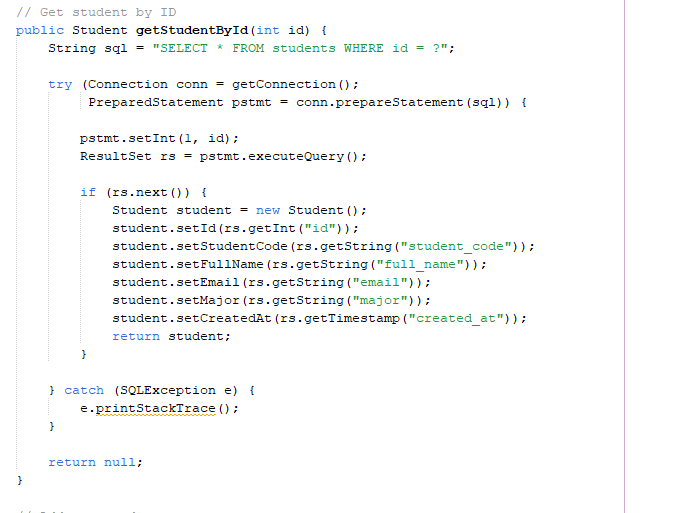
On clicking delete button



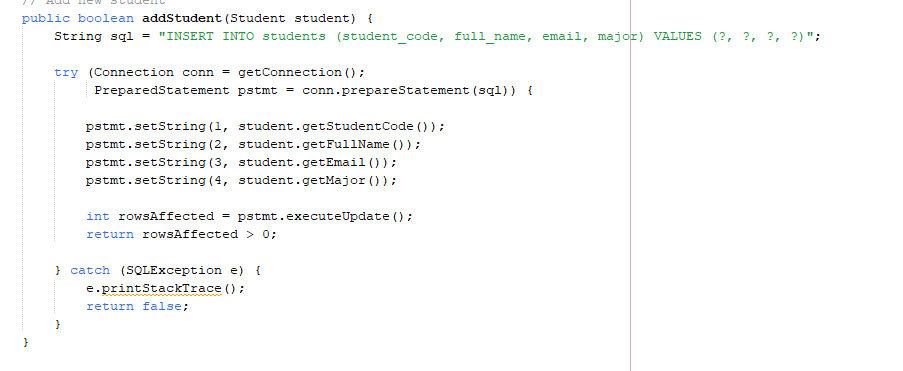
**Exercise 4: Complete CRUD Operations**

Task 4.1: Complete DAO Methods

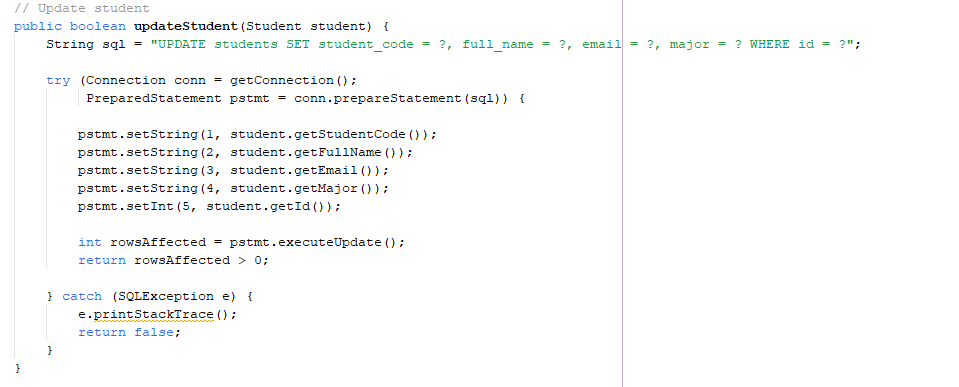
getStudentById() method:



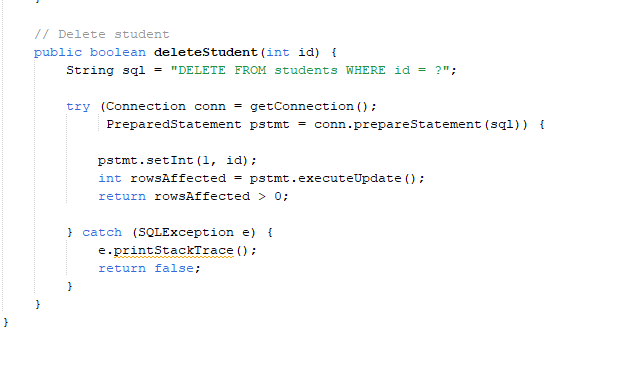
addStudent() method:



updateStudent() method:



deleteStudent() method:



Task 4.2: Integration Testing

List of students:

A screenshot of a computer

AI-generated content may be incorrect.

Add new student:

A screenshot of a computer

AI-generated content may be incorrect.

Add student successfully

A screenshot of a computer

AI-generated content may be incorrect.

Edit student:

A screenshot of a computer

AI-generated content may be incorrect.

Edit successfully

A screenshot of a computer

AI-generated content may be incorrect.

Delete student:

A screenshot of a computer

AI-generated content may be incorrect.

Delete successfully

A screenshot of a computer

AI-generated content may be incorrect.

Empty state

A screenshot of a computer

AI-generated content may be incorrect.