

I. Add Student:

The process starts when the user clicks on “Add New Student” button requests `action=new`, prompting the Controller to forward to an empty form from `student-form.jsp`. After submitting the form, the Controller captures the data, creates a student object, and assigns it to the DAO for the SQL INSERT. To prevent duplicate entries, the Controller then employs the POST-Redirect-GET (PRG) pattern, redirecting the user to `student?action=list` to display the new student and a success message.

The image shows two screenshots of a web application. The top screenshot displays the 'Add New Student' form, which is a white modal box centered on a purple background. The form contains the following fields: 'Student Code' (with value 'CS022'), 'Full Name' (with value 'Chien dep trai'), 'Email' (with value 'chienproccc1@gmail.com'), and 'Major' (a dropdown menu set to 'Computer Science'). At the bottom of the form are two buttons: 'Add Student' (purple) and 'Cancel' (grey). The browser's address bar shows the URL `localhost:8080/StudentManagementlab/student?action=new`.

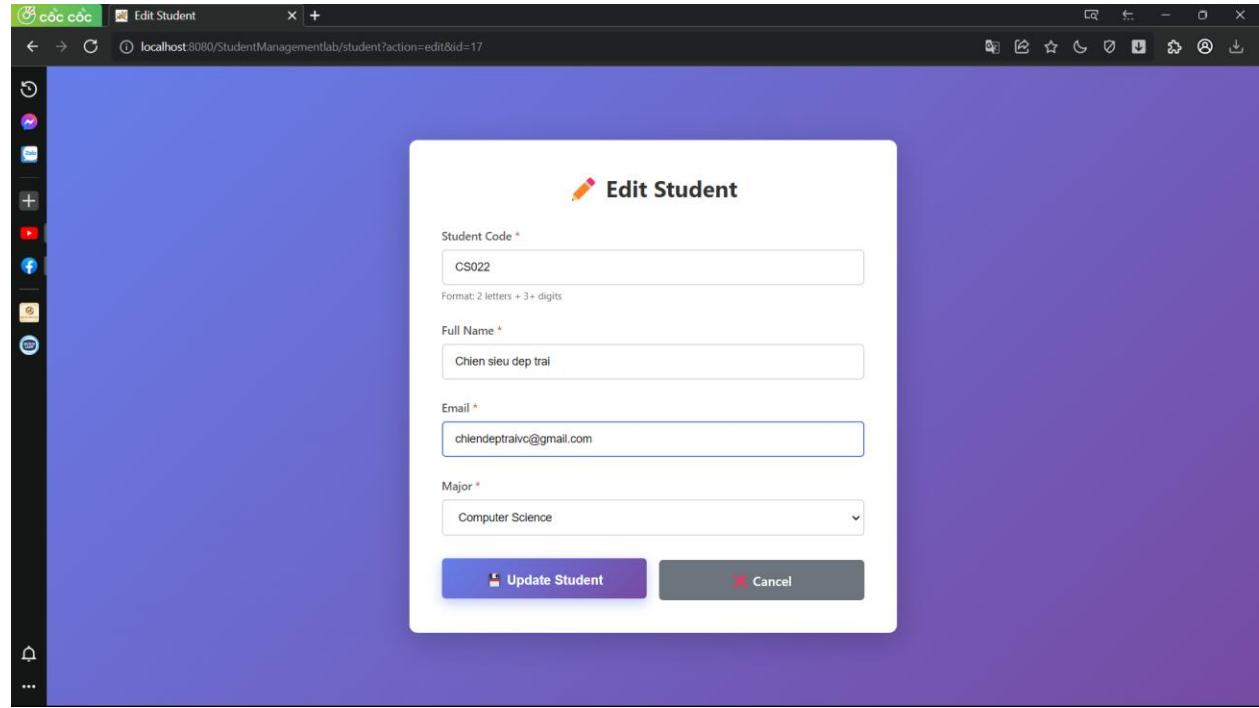
The bottom screenshot displays the 'Student List' page, which has a purple header and a white main content area. At the top of the content area is a green success message: 'Student added successfully'. Below this is a purple button labeled 'Add New Student'. Underneath is a table with the following data:

ID	STUDENT CODE	FULL NAME	EMAIL	MAJOR	ACTIONS
17	CS022	Chien dep trai	chienproccc1@gmail.com	Computer Science	Edit Delete
13	SK123	HUH	chienproccc1@gmail.com	Computer Science	Edit Delete
10	AK470	bruh			Edit Delete
9	IT22022	Huynh Van Quang Chien	chienproccc1@gmail.com	Computer Science	Edit Delete
8	CS001	Huynh Van Quang Chien			Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	Edit Delete

The browser's address bar shows the URL `localhost:8080/StudentManagementlab/student?action=list&message=Student%20added%20successfully`.

II. Edit:

Editing begins with a request including the student ID (action=edit&id=(id)). The Controller uses the DAO (getStudentById) to get the current data, sets it as an attribute, and pre-fill the student-form.jsp. After the submits the changes (POST, action=update), the Controller validates the data and calls the DAO to execute the SQL UPDATE query. Finally, redirecting to student?action=list to show the updated list.



Edit Student

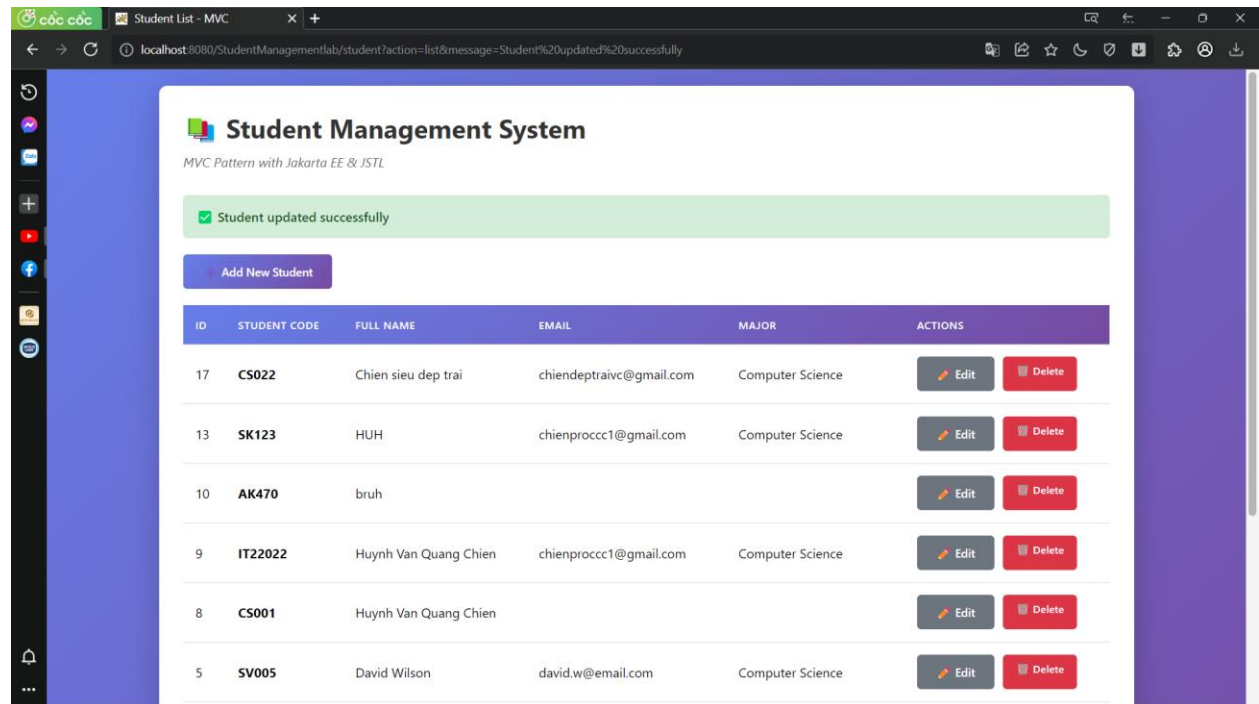
Student Code *
CS022
Format: 2 letters + 3+ digits

Full Name *
Chien sieu dep trai

Email *
chiendepraivc@gmail.com

Major *
Computer Science

Update Student **Cancel**



Student Management System
MVC Pattern with Jakarta EE & JSTL

✓ Student updated successfully

Add New Student

ID	STUDENT CODE	FULL NAME	EMAIL	MAJOR	ACTIONS
17	CS022	Chien sieu dep trai	chiendepraivc@gmail.com	Computer Science	Edit Delete
13	SK123	HUH	chienproccc1@gmail.com	Computer Science	Edit Delete
10	AK470	bruh			Edit Delete
9	IT22022	Huynh Van Quang Chien	chienproccc1@gmail.com	Computer Science	Edit Delete
8	CS001	Huynh Van Quang Chien			Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	Edit Delete

III. Delete:

The Delete process starts with a GET request containing the ID (action=delete&id=(id)). The Controller immediately sent the request to the DAO, which executes the SQL DELETE statement to permanently remove the record. To refresh the view and confirm the action, the Controller then redirects the user back to the main list view (student?action=list), which displays the list without the deleted entry.

The image displays two screenshots of a web application titled "Student Management System" (MVC Pattern with Jakarta EE & JSTL). The application is running on a browser at the URL `localhost:8080/StudentManagementlab/student?action=list&message=Student%20deleted%20successfully`.

Top Screenshot: Shows the "Student Management" interface with a confirmation dialog box. The dialog box contains the text "localhost:8080 cho biết" and "Are you sure you want to delete this student?". Below the dialog, a green message bar indicates "Student updated successfully". The table below lists students with their IDs, codes, full names, emails, majors, and actions (Edit, Delete).

ID	STUDENT CODE	FULL NAME	EMAIL	MAJOR	ACTIONS
17	CS022	Chien sieu dep trai	chiendepraivc@gmail.com	Computer Science	Edit Delete
13	SK123	HUH	chienproccc1@gmail.com	Computer Science	Edit Delete
10	AK470	bruh			Edit Delete
9	IT22022	Huynh Van Quang Chien	chienproccc1@gmail.com	Computer Science	Edit Delete
8	CS001	Huynh Van Quang Chien			Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	Edit Delete

Bottom Screenshot: Shows the same interface after the deletion. A green message bar indicates "Student deleted successfully". The table below lists the remaining students.

ID	STUDENT CODE	FULL NAME	EMAIL	MAJOR	ACTIONS
13	SK123	HUH	chienproccc1@gmail.com	Computer Science	Edit Delete
10	AK470	bruh			Edit Delete
9	IT22022	Huynh Van Quang Chien	chienproccc1@gmail.com	Computer Science	Edit Delete
8	CS001	Huynh Van Quang Chien			Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	Edit Delete