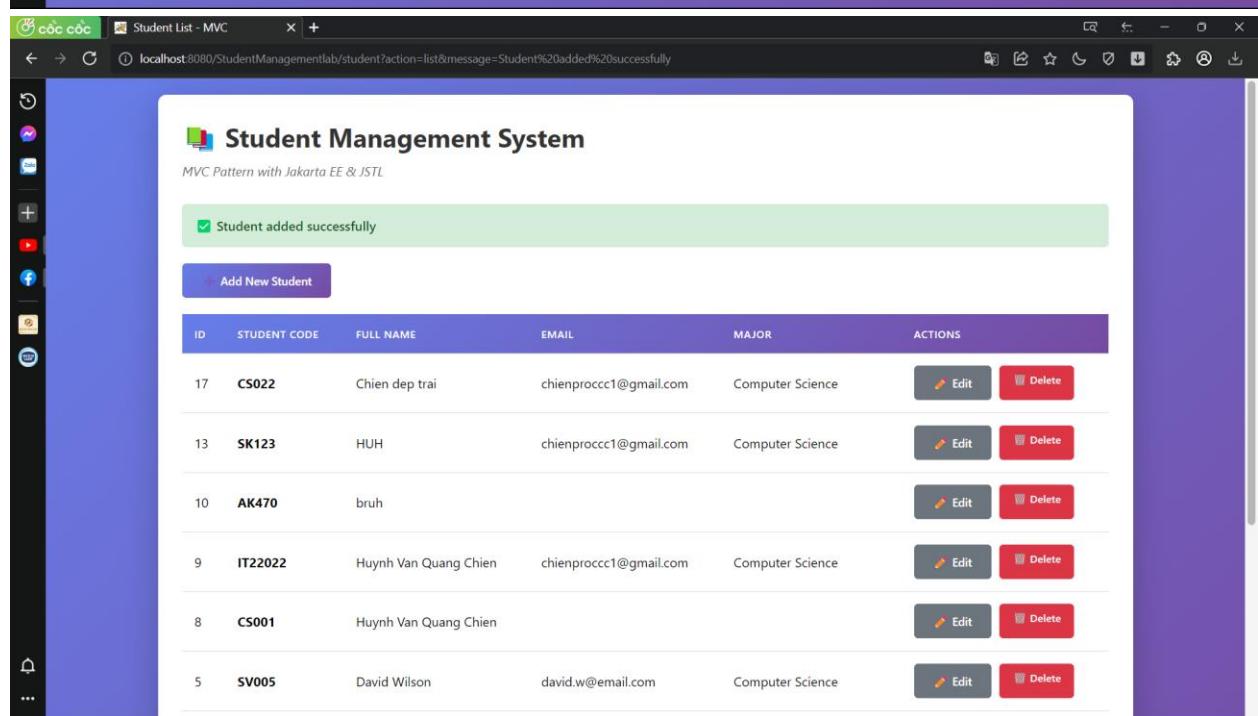
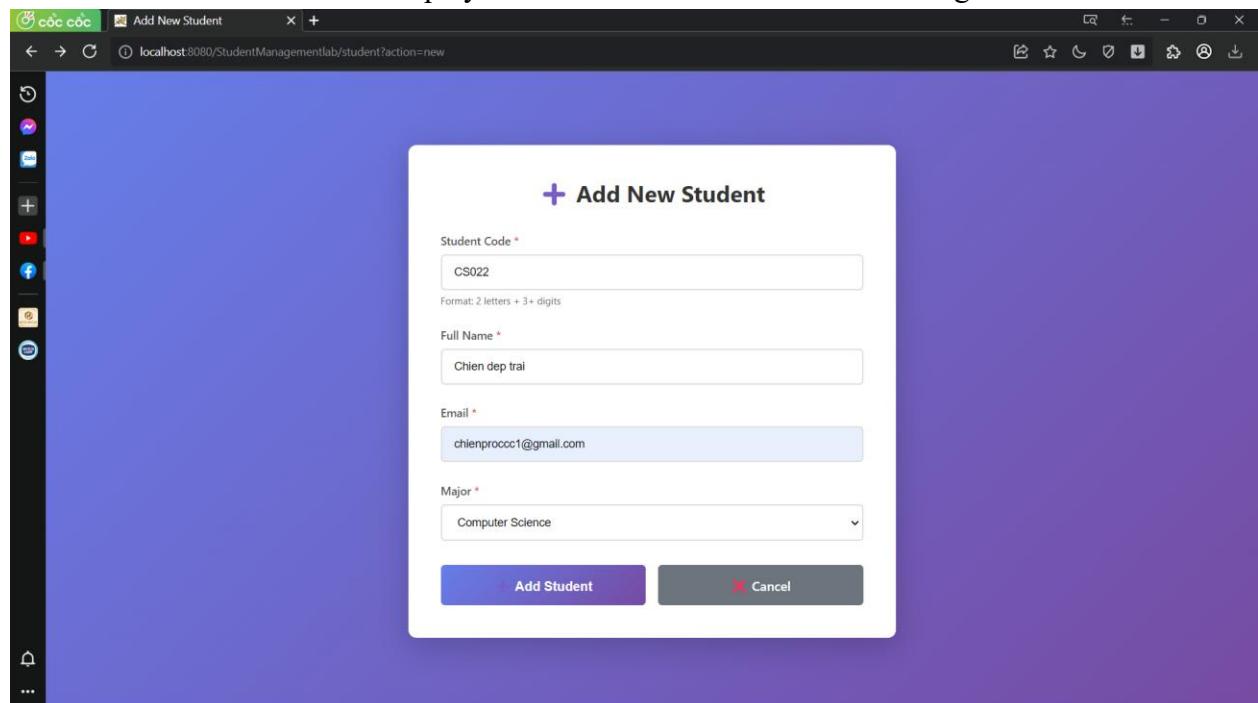


## I. Add Student:

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.



| ID | STUDENT CODE | FULL NAME             | EMAIL                 | MAJOR            | ACTIONS                                       |
|----|--------------|-----------------------|-----------------------|------------------|---|
| 17 | CS022        | Chien dep trai        | chienprocc1@gmail.com | Computer Science | <button>Edit</button> <button>Delete</button> |
| 13 | SK123        | HUH                   | chienprocc1@gmail.com | Computer Science | <button>Edit</button> <button>Delete</button> |
| 10 | AK470        | bruh                  |                       |                  | <button>Edit</button> <button>Delete</button> |
| 9  | IT22022      | Huynh Van Quang Chien | chienprocc1@gmail.com | Computer Science | <button>Edit</button> <button>Delete</button> |
| 8  | CS001        | Huynh Van Quang Chien |                       |                  | <button>Edit</button> <button>Delete</button> |
| 5  | SV005        | David Wilson          | david.w@email.com     | Computer Science | <button>Edit</button> <button>Delete</button> |

## 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.

The screenshot shows two browser windows side-by-side. The top window is titled 'Edit Student' and contains a form with fields for Student Code (CS022), Full Name (Chien sieu dep trai), Email (chiendeptraivc@gmail.com), and Major (Computer Science). The bottom window is titled 'Student List - MVC' and displays a table of student records with columns: ID, STUDENT CODE, FULL NAME, EMAIL, and MAJOR. A success message 'Student updated successfully' is visible above the table. The table data is as follows:

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

### 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.

