

## Web Application Development Lab 04

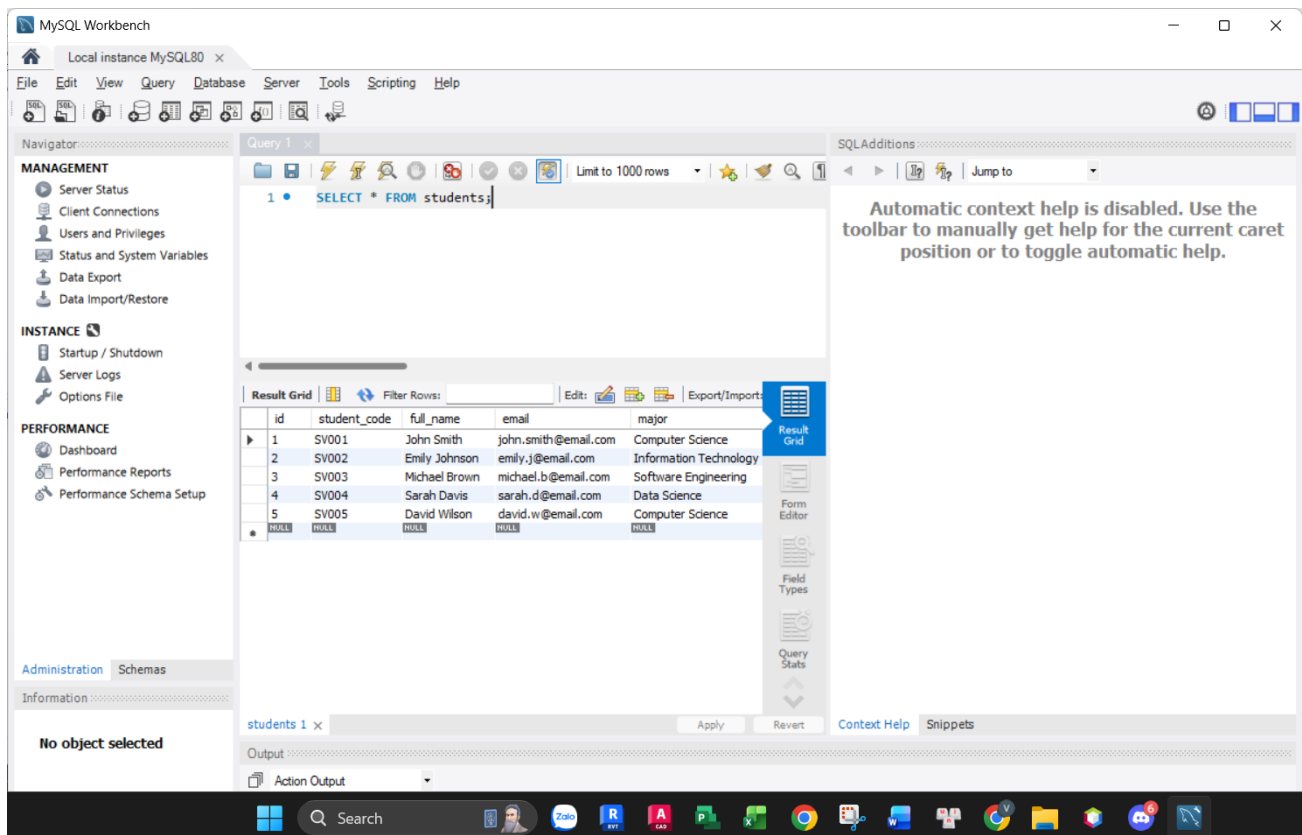
### JSP + MySQL

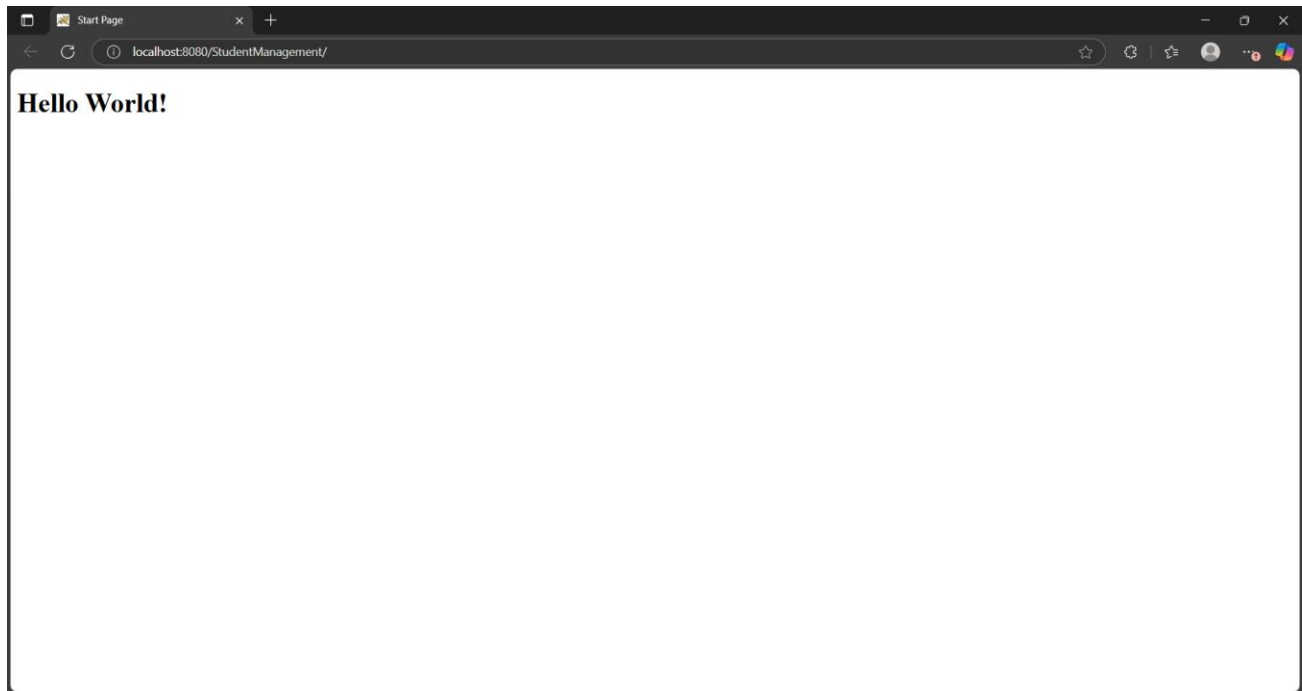
#### *Part A – In-class practices and exercises*

### **I. Exercise 1: SETUP AND DISPLAY**

#### Task 1.1. - Setting up the DB and Project

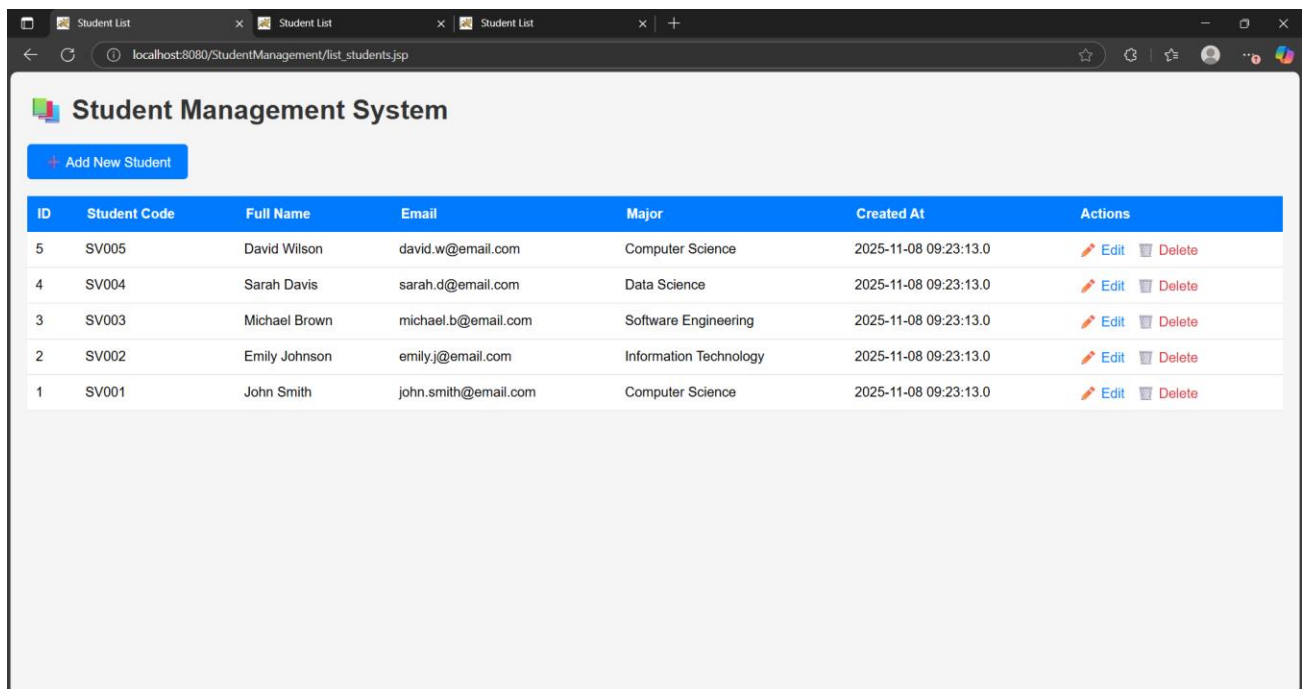
- Database is set up with MySQL and executing “SELECT \* FROM students;” returns exactly 5 students.
- NetBeans project configured with Tomcat server runs smoothly.





## Task 1.2 – Display Student List

O th



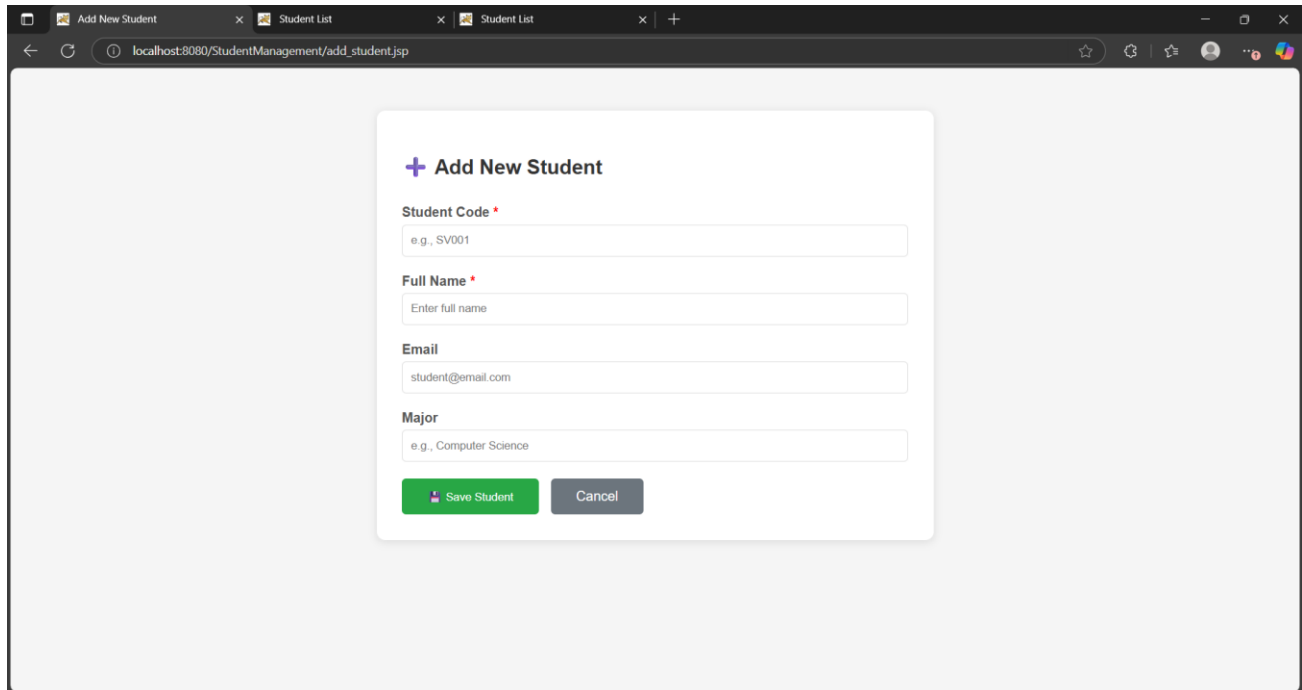
### - Code walkthrough:

- First off, establish connection the database with JDBC connection
- Execute the query “SELECT \* FROM students ORDER BY id DESC” in order to get all the student data in descending order of their ID
- Then, for each student in our data base, take their information and parse

them into a table.

## II. EXERCISE 2: CREATE OPERATION

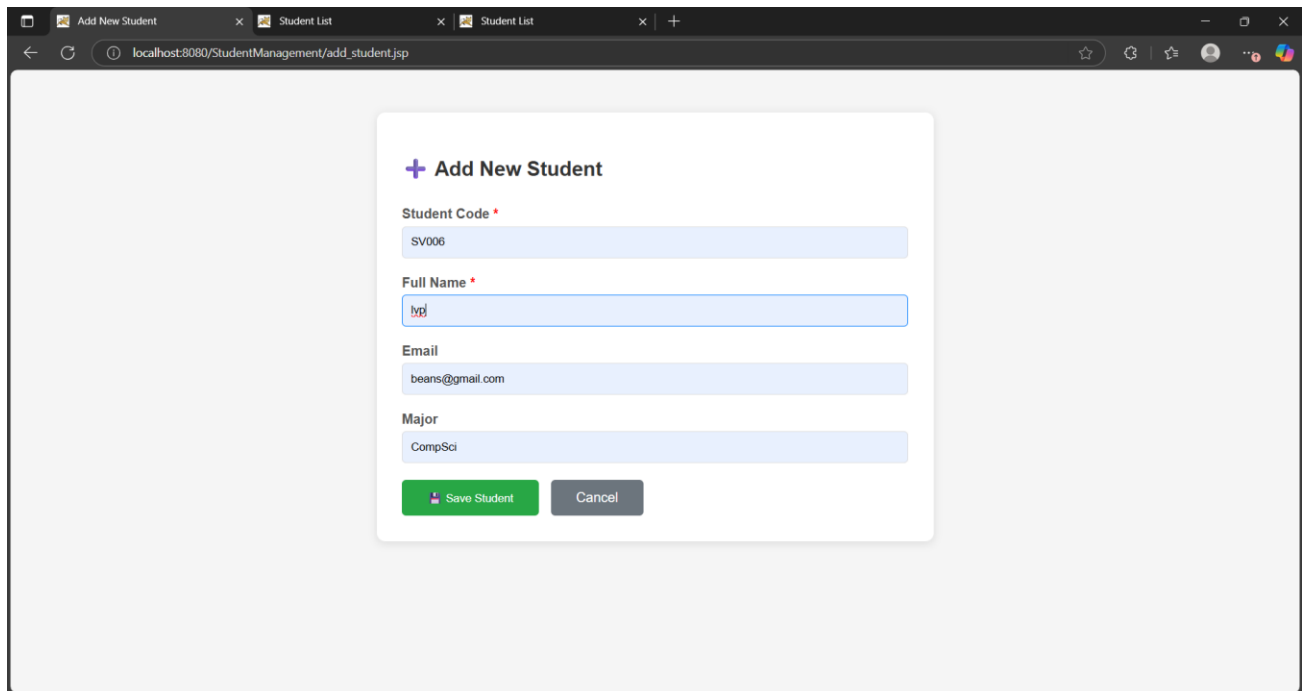
### Task 2.1: Create Add Student Form



The screenshot shows a web browser window with the address bar displaying `localhost:8080/StudentManagement/add_student.jsp`. The browser has three tabs: 'Add New Student', 'Student List', and 'Student List'. The main content area displays a form titled '+ Add New Student'. The form contains four input fields: 'Student Code' (with a red asterisk and placeholder 'e.g., SV001'), 'Full Name' (with a red asterisk and placeholder 'Enter full name'), 'Email' (with placeholder 'student@email.com'), and 'Major' (with placeholder 'e.g., Computer Science'). At the bottom of the form are two buttons: a green 'Save Student' button and a grey 'Cancel' button.

- Code walkthrough:
  - Simple form in html with appropriate input
  - Submit button process user's form with `process_add.jsp` in task 2.2

## Task 2.2: Process Add Student



**+ Add New Student**

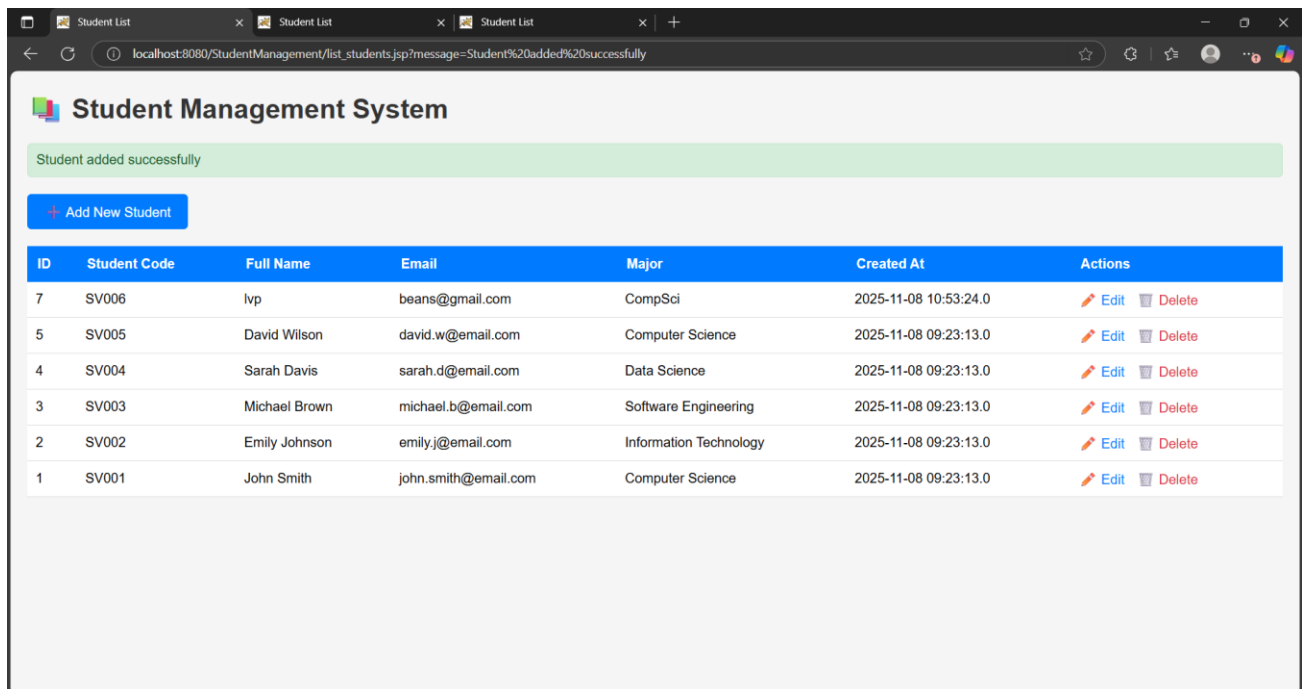
Student Code \*  
SV006

Full Name \*  
lvp

Email  
beans@gmail.com

Major  
CompSci

[Save Student](#) [Cancel](#)



**Student Management System**

Student added successfully

[+ Add New Student](#)

ID	Student Code	Full Name	Email	Major	Created At	Actions
7	SV006	lvp	beans@gmail.com	CompSci	2025-11-08 10:53:24.0	<a href="#">Edit</a> <a href="#">Delete</a>
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>

- Code walkthrough:
  - Get parameters from form with request.getParameters
  - Establish connection with database server
  - Insert the user information in the current table
  - Redirect the user to the student display website if successful and an error message at the add page if not.

- Error handling for form validation

**+ 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

**Save Student** **Cancel**

**+ Add New Student**

Student code already exists

**Student Code \***  
e.g., SV001

**Full Name \***  
lvp

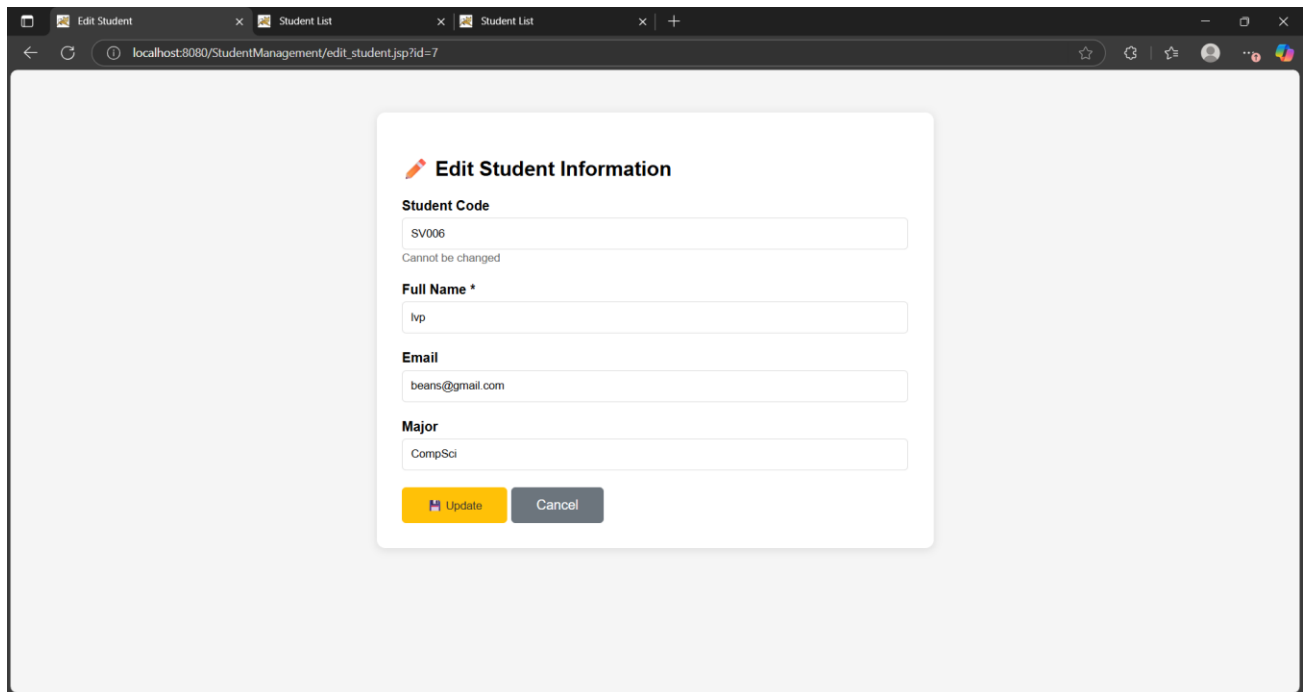
**Email**  
beans@gmail.com

**Major**  
CompSci

**Save Student** **Cancel**

Please fill out this field.

### III. EXERCISE 3: UPDATE OPERATION



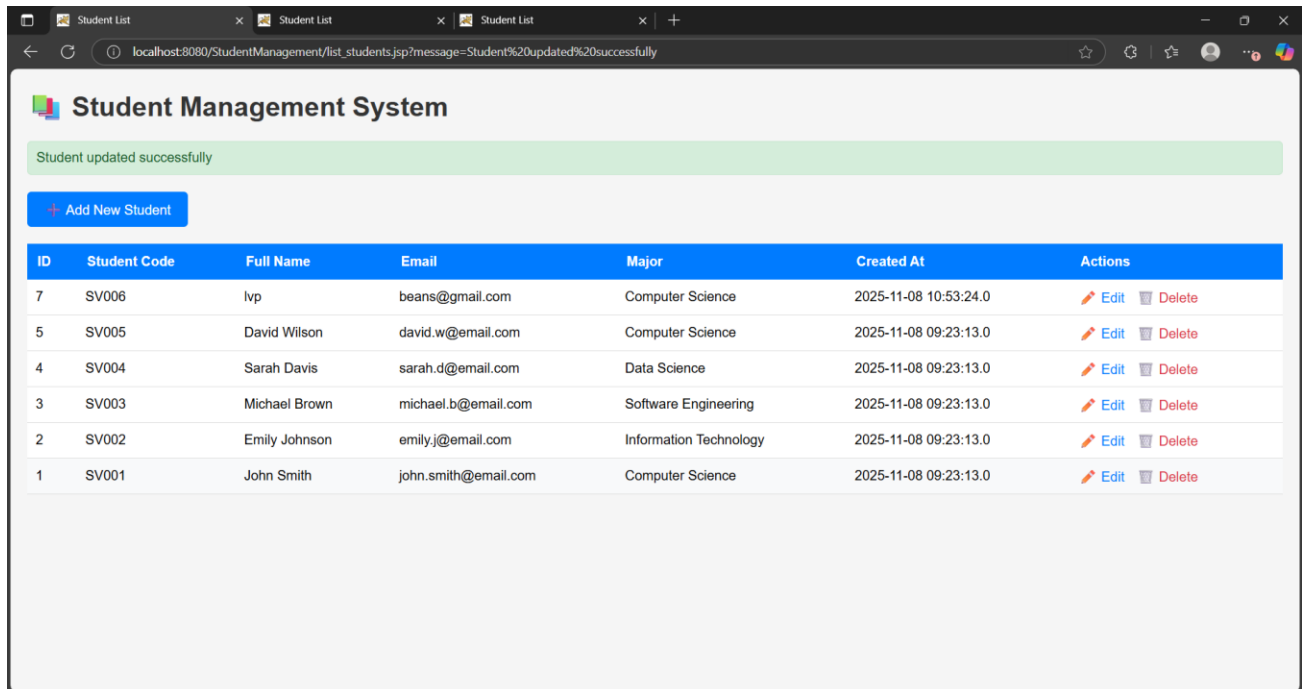
The screenshot shows a web browser window with the URL `localhost:8080/StudentManagement/edit_student.jsp?id=7`. The browser has three tabs: 'Edit Student', 'Student List', and 'Student List'. The main content area displays a form titled 'Edit Student Information' with a pencil icon. The form contains the following fields:

- Student Code**: Input field with value 'SV006'. Below it, the text 'Cannot be changed' is displayed.
- Full Name \***: Input field with value 'lvp'.
- Email**: Input field with value 'beans@gmail.com'.
- Major**: Input field with value 'CompSci'.

At the bottom of the form are two buttons: a yellow 'Update' button with a pencil icon and a grey 'Cancel' button.

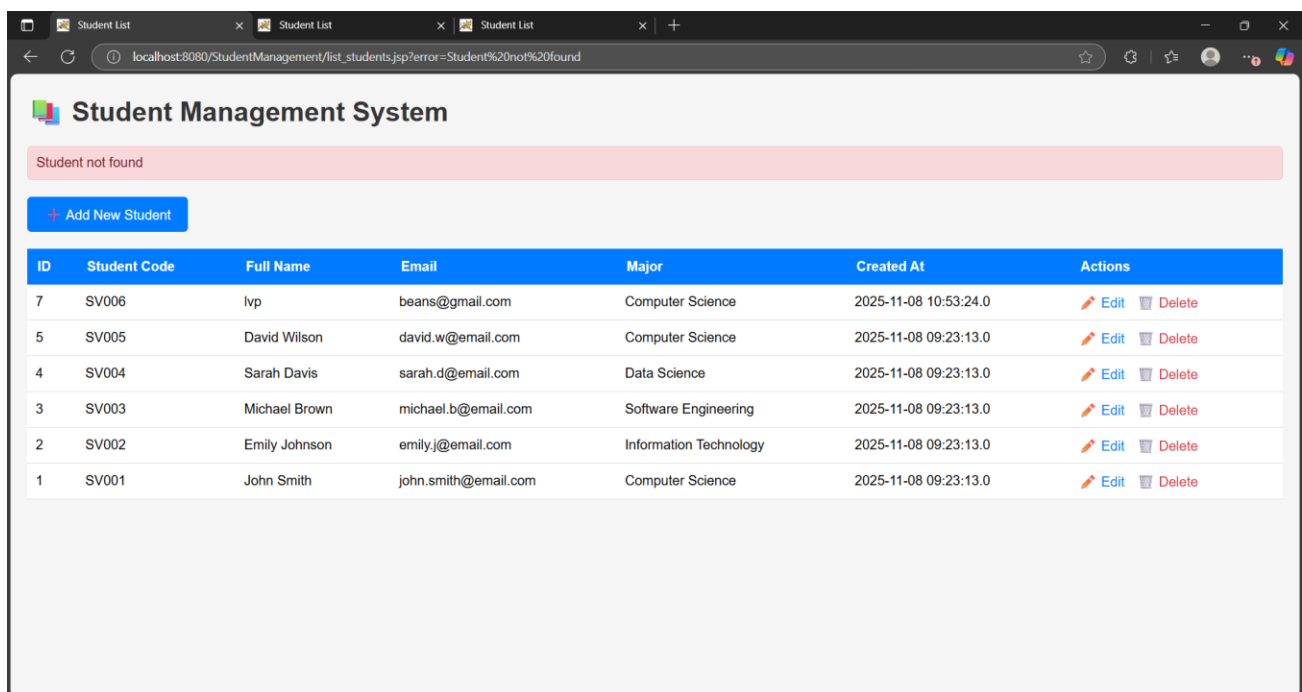
- Code walkthrough:
  - Establish a connection to the database
  - Retrieve user information based on their ID
  - Parse them onto a form
  - On submit form, process the form information make the necessary updates to the database.

#### Task 3.2: Process Update



The screenshot shows a web browser window with the URL `localhost:8080/StudentManagement/list_students.jsp?message=Student%20updated%20successfully`. The page title is "Student Management System". A green banner at the top displays the message "Student updated successfully". Below this is a blue button labeled "Add New Student". A table lists seven students with columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The Actions column contains "Edit" and "Delete" links for each student.

ID	Student Code	Full Name	Email	Major	Created At	Actions
7	SV006	lvp	beans@gmail.com	Computer Science	2025-11-08 10:53:24.0	<a href="#">Edit</a> <a href="#">Delete</a>
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>

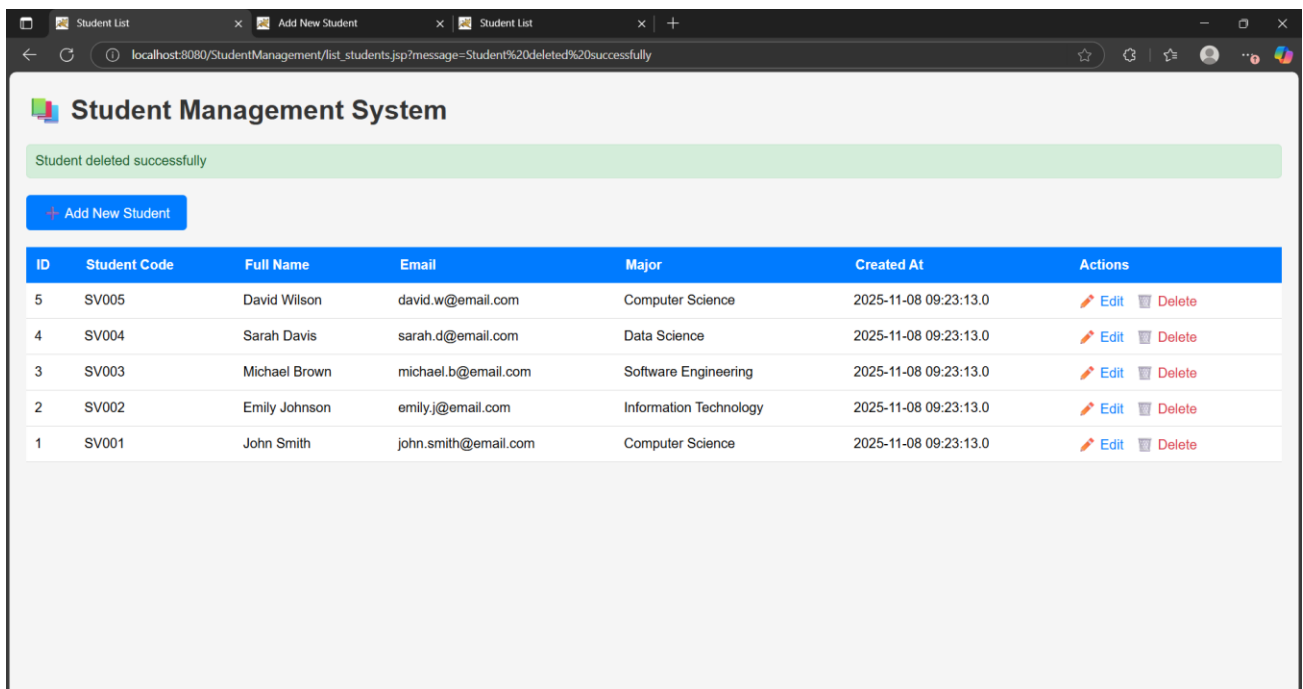
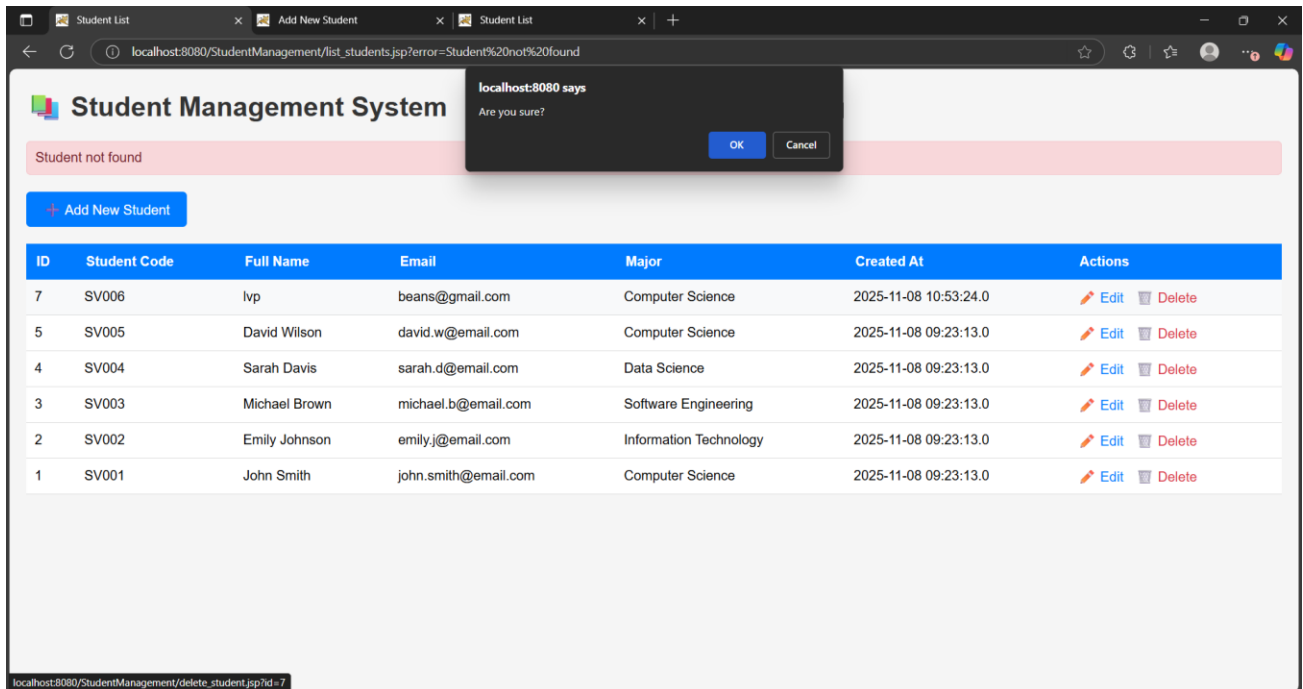


The screenshot shows a web browser window with the URL `localhost:8080/StudentManagement/list_students.jsp?error=Student%20not%20found`. The page title is "Student Management System". A pink banner at the top displays the message "Student not found". Below this is a blue button labeled "Add New Student". A table lists seven students with columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The Actions column contains "Edit" and "Delete" links for each student.

ID	Student Code	Full Name	Email	Major	Created At	Actions
7	SV006	lvp	beans@gmail.com	Computer Science	2025-11-08 10:53:24.0	<a href="#">Edit</a> <a href="#">Delete</a>
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	<a href="#">Edit</a> <a href="#">Delete</a>

- Code walkthrough:
  - o Establish connection with the database
  - o Validate some critical information, namely student name and id
  - o Perform the necessary updates
  - o Redirect the user the appropriate next page.

#### IV. EXERCISE 4: DELETE OPERATION



- Code walkthrough:
  - Establish connection with the database
  - Perform the delete update base on ID
  - Redirect user to the appropriate web page with message