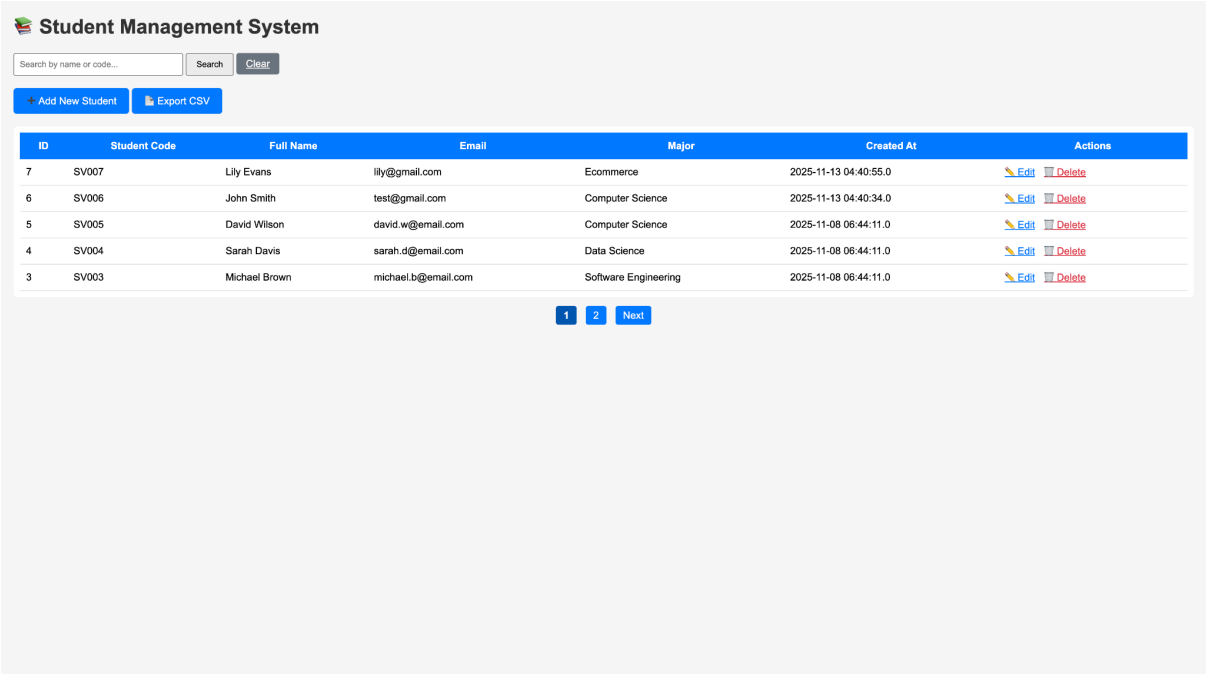
Name: Phạm Gia Phúc

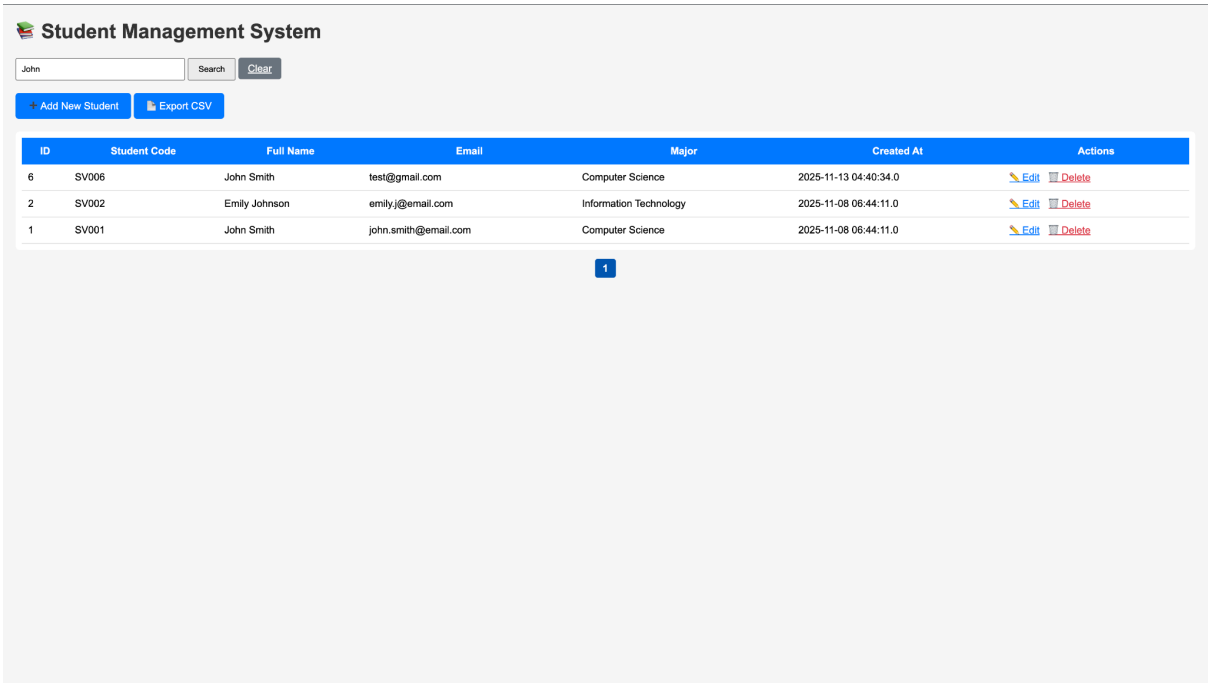
ID: ITCSIU22178

Web App Development Laboratory - Lab 04 Assignment

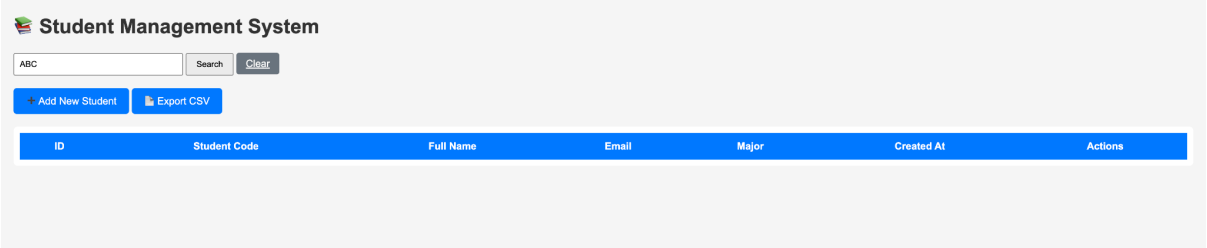
Exercise 5: Search Functionality



- **Test case 1: Search for rows that have rows contained “John”**



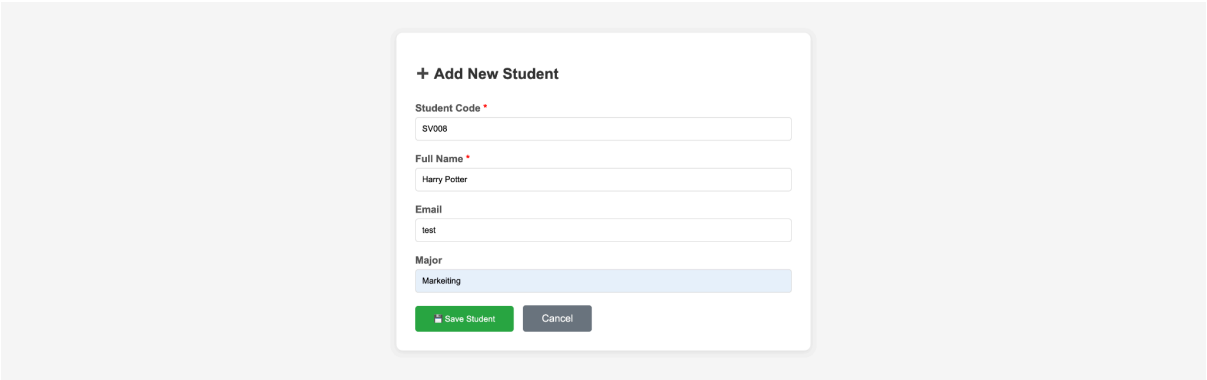
- **Test case 2: Empty rows when search**

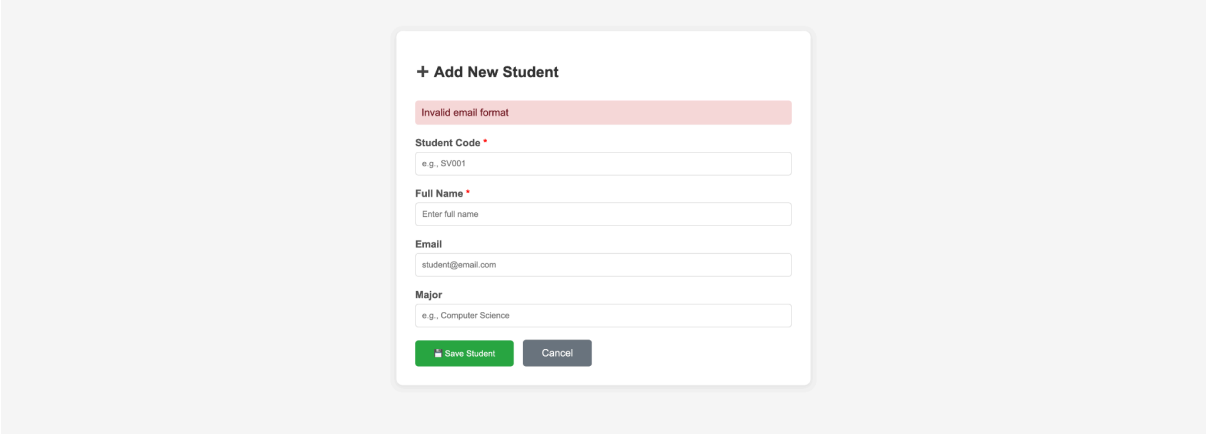


→ **Explanation:** The search feature allows users to filter the student list by entering a name or student code in the search box. When the user types a keyword and submits the form, the page reloads with the keyword included in the URL as a GET parameter. The server then checks whether this keyword exists and is not empty. If a keyword is provided, the SQL queries are modified to include a WHERE clause using LIKE '%keyword%' so that only records whose full_name or student_code match the keyword are returned. The first search query calculates the total number of matching records, which is used to generate accurate pagination. Next, the main query retrieves only the records for the current page using LIMIT and OFFSET. If no keyword is entered, the system simply fetches all students without filtering. The search box also keeps the old keyword as its default value so users can see their previous search term. Together, this logic allows the table to display filtered results while still supporting pagination and navigation between pages.

Exercise 6: Validation Enhancement

- Before and after enter the fields:

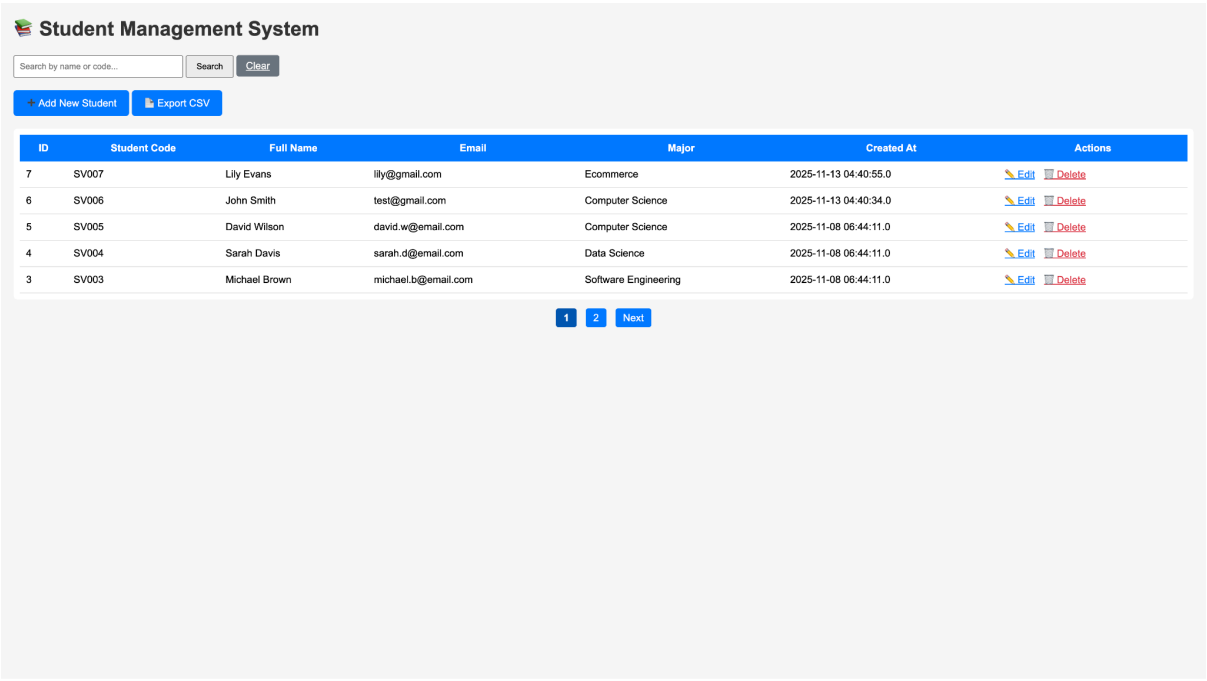




→ **Explanation:** The code first checks whether the required fields student code and full name are empty; if either is missing, the user is returned to the form with an error. If an email is provided, it must match a valid email pattern; otherwise, an error is shown. The student code also must follow the format of two uppercase letters followed by at least three digits (e.g., SV001). If the input fails any of these checks, the form is rejected. Only when all validations pass does the program attempt to insert the record into the database. If the database detects a duplicate student code or another SQL error, the user is redirected with the appropriate error message.

Exercise 7: Pagination

- View of list students page with pagination:



→ **Explanation:**

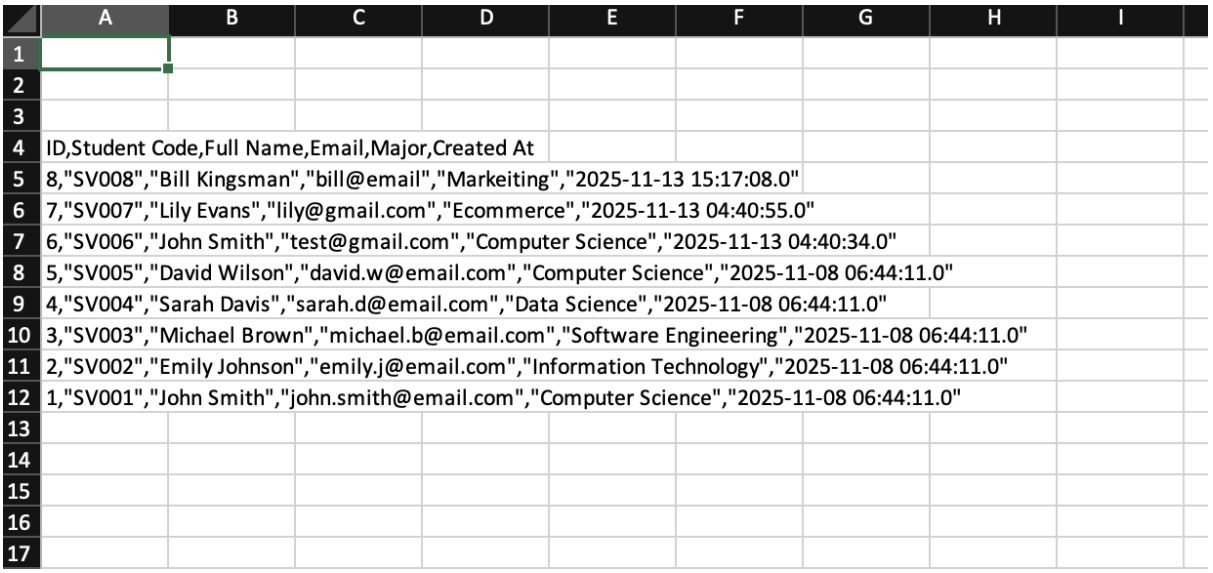
- The pagination system divides the student list into smaller pages, each showing 5 records. First, the code reads the current page number from the URL (page

parameter). If it's missing, page 1 is used. It then calculates the starting position (offset) for the SQL query using (currentPage − 1) * recordsPerPage.

- Next, a separate SQL query counts the total number of matching records (depending on whether a keyword search is used). From this total, the number of pages (totalPages) is computed using ceil(totalRecords / recordsPerPage).
- The main SQL query then retrieves only the records for the current page using LIMIT ? OFFSET ?. After displaying the results, the page generates navigation links. It shows Previous (only if not on page 1), page numbers with the current one highlighted, and Next (only if not on the last page). The keyword parameter is preserved in all links so search results remain consistent across pages.

Bonus: Export CSV

- When I click the export csv button, the result file is:



	A	B	C	D	E	F	G	H	I
1									
2									
3									
4	ID,Student Code,Full Name,Email,Major,Created At								
5	8,"SV008","Bill Kingsman","bill@email","Markeiting","2025-11-13 15:17:08.0"								
6	7,"SV007","Lily Evans","lily@gmail.com","Ecommerce","2025-11-13 04:40:55.0"								
7	6,"SV006","John Smith","test@gmail.com","Computer Science","2025-11-13 04:40:34.0"								
8	5,"SV005","David Wilson","david.w@email.com","Computer Science","2025-11-08 06:44:11.0"								
9	4,"SV004","Sarah Davis","sarah.d@email.com","Data Science","2025-11-08 06:44:11.0"								
10	3,"SV003","Michael Brown","michael.b@email.com","Software Engineering","2025-11-08 06:44:11.0"								
11	2,"SV002","Emily Johnson","emily.j@email.com","Information Technology","2025-11-08 06:44:11.0"								
12	1,"SV001","John Smith","john.smith@email.com","Computer Science","2025-11-08 06:44:11.0"								
13									
14									
15									
16									
17									

→ **Explanation:** This JSP page generates and downloads a CSV file of student data. It first sets the response type to CSV and writes the header row. Then it connects to the MySQL database and checks whether a search keyword was provided; if so, it filters students by name or student code, otherwise it loads all students. After executing the query, it loops through each row and prints the student fields into CSV format, wrapping text values in quotes to avoid formatting issues. Finally, it closes all database resources.