

Web Application Development Lab 04

JSP + MySQL

Part B – HOMEWORK

I. EXERCISE 5: SEARCH FUNCTIONALITY

*The screenshots are taken after most of the functionalities are implemented

A screenshot of a web browser displaying the 'Student Management System'. The URL in the address bar is `localhost:8080/StudentManagement/list_students.jsp?keyword=John`. The page title is 'Student List'. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below them is a search input field containing 'John', a 'Search' button, and a 'Clear' button. The main area shows a table of student data with the following columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. Two rows are visible:

ID	Student Code	Full Name	Email	Major	Created At	Actions
2	SV002	Emily Johnson	emilyj@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

Searching for “John”

A screenshot of a web browser displaying the 'Student Management System'. The URL in the address bar is `localhost:8080/StudentManagement/list_students.jsp?keyword=SV001`. The page title is 'Student List'. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below them is a search input field containing 'SV001', a 'Search' button, and a 'Clear' button. The main area shows a table of student data with the following columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. One row is visible:

ID	Student Code	Full Name	Email	Major	Created At	Actions
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

Searching by student code

The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?keyword=science". The page title is "Student Management System". At the top, there are buttons for "Add New Student" and "Export CSV", and a search bar containing "science" with "Search" and "Clear" buttons. Below the search bar is a table with columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. The table contains three rows of student data. A navigation bar at the bottom shows page 1 of 1.

ID	Student Code	Full Name	Email	Major	Created At	Actions
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

Searching by major

The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?keyword=". The page title is "Student Management System". At the top, there are buttons for "Add New Student" and "Export CSV", and a search bar containing "Search" and "Clear" buttons. Below the search bar is a table with columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. The table contains ten rows of student data. A navigation bar at the bottom shows page 1 of 2.

ID	Student Code	Full Name	Email	Major	Created At	Actions
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete

Empty search (keyword= ' ')

ID	Student Code	Full Name	Email	Major	Created At	Actions
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete

Clear link redirect to list_students.jsp

Search implementation:

```

if (keyword != null && !keyword.trim().isEmpty()) {
    sql = "SELECT * FROM students WHERE full_name LIKE ? OR
student_code LIKE ? OR major LIKE ?";
    pstmt = conn.prepareStatement(sql);
    String search = "%" + keyword.trim() + "%";
    pstmt.setString(1, search);
    pstmt.setString(2, search);
    pstmt.setString(3, search);
} else {
    sql = "SELECT * FROM students ORDER BY id DESC";
    pstmt = conn.prepareStatement(sql);
}
rs = pstmt.executeQuery();

```

II. EXERCISE 6: VALIDATION ENHANCEMENT

6.1: Email Validation

The provided regex did not produce the expected results so a new regex is implemented, taken from: <https://www.regular-expressions.info/email.html>

(from a reddit thread)

Implementation(process_add):

```
if (email != null && !email.isEmpty()) {  
    if (!email.matches("^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-  
z]{2,}+$")) {  
        response.sendRedirect("add_student.jsp?error=Invalid email format");  
        return;  
    }  
}  
  
if (studentCode != null && !studentCode.isEmpty()) {  
    if (!studentCode.matches("[A-Z]{2}[0-9]{3,}")) {  
        response.sendRedirect("add_student.jsp?error=Invalid ID format, student  
code MUST follows pattern: 2 uppercase letters plus 3 digits");  
        return;  
    }  
}
```

Implementation(process_edit):

```

if (email != null && !email.isEmpty()) {
    if (!email.matches("[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-
z]{2,}"))
        response.sendRedirect("edit_student.jsp?error=Invalid email format");
    return;
}
}

```

ID	Student Code	Full Name	Email	Major	Created At	Actions
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	Ivp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	Ivp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete

1 2 [Next](#)

Testing john@email.com

The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?message=Student%20added%20successfully". The page displays a success message "✓ Student added successfully" and a "Student Management System" header. Below the header are buttons for "Add New Student" and "Export CSV", and search fields. A table lists student data with columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions (Edit, Delete). The table contains 10 rows of student information.

ID	Student Code	Full Name	Email	Major	Created At	Actions
23	JJ002	thisIsJohnDoe	john.doe@company.co.uk	Johnning Science	2025-11-14 15:44:14.0	
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	
20	IL009	notARRealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	

Testing john.doe@company.co.uk

The screenshot shows a web browser window titled "Add New Student" with the URL "localhost:8080/StudentManagement/add_student.jsp?error=Invalid%20email%20format". The form has a title ">Add New Student" and fields for Student Code, Full Name, Email, and Major. The "Email" field contains "john@email" and is highlighted in red, indicating an error. A message "Invalid email format" is displayed above the input field. The "Save Student" button is visible at the bottom.

Testing john@email

Add New Student

Invalid email format

Student Code *

JJ002

Full Name *

thisIsJohnDoe

Email

johnemail.com

Major

Joining Science

Save Student Cancel

Testing johnemail.com

Add New Student

Student Code *

JJ003

Full Name *

thisIsJohnDoe

Email

student@email.com

Major

Joining Science

Please fill out this field.

Save Student Cancel

Empty field

Invalid email format

Student Code
JJ002
Cannot be changed

Full Name *
thisIsJohnDoe

Email
john@email

Major
Johnning Science

Update **Cancel**

Testing john@email edit

The process_edit results closely mimics that of the process_add

6.2: Student Code Pattern Validation

Invalid ID format, student code MUST follows pattern: 2 uppercase letters plus 3 digits

Student Code *
JK0

Full Name *
Das John DoeMinion

Email
DasJohnDoes@email.com

Major
Johnning Science

Save Student **Cancel**

Testing JJK0

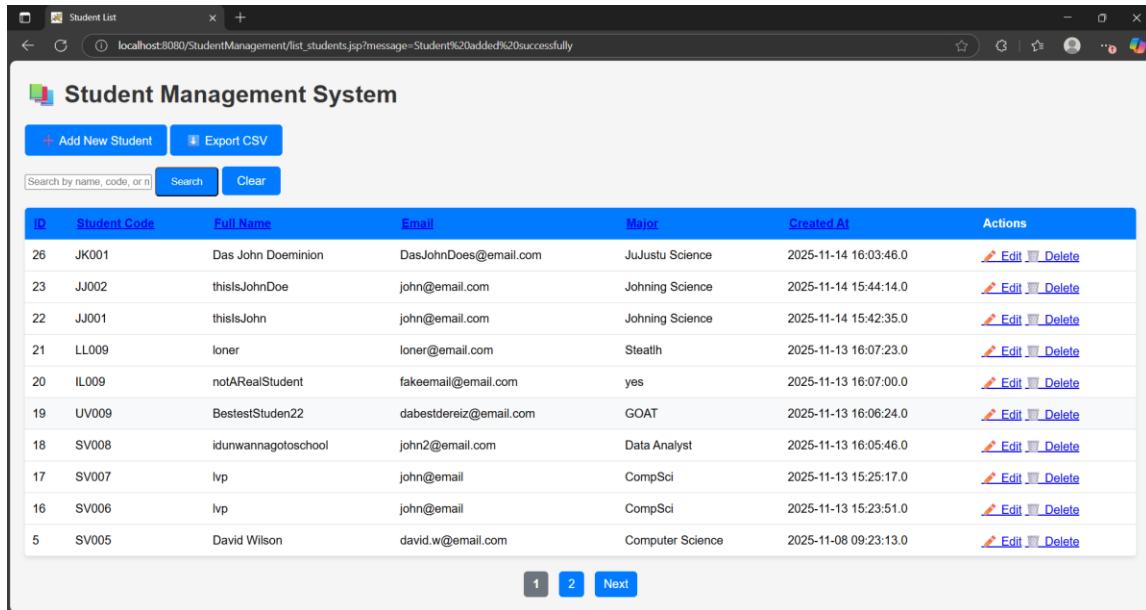
The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?message=Student%20added%20successfully". The page displays a success message "✓ Student added successfully" and a general message "Student added successfully". It features two buttons: "Add New Student" and "Export CSV". Below these are search fields "Search by name, code, or n:" with "Search" and "Clear" buttons. A table lists student data with columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. The table contains 16 rows of student information, including newly added ones.

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete

Testing for JK001

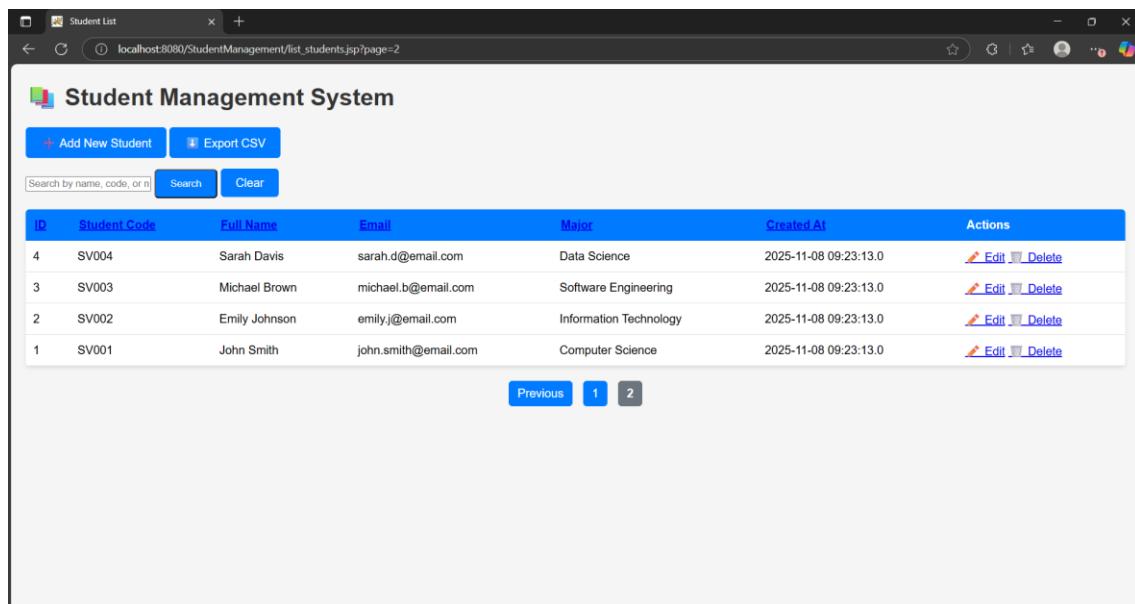
III. EXERCISE 7: USER EXPERIENCE IMPROVEMENTS

7.1: Pagination



The screenshot shows a web-based student management system. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below that is a search bar with placeholder text 'Search by name, code, or n...' and 'Search' and 'Clear' buttons. The main area displays a table of student data with the following columns: ID, Student Code, Full Name, Email, Major, Created At, and Actions. There are 10 rows of data. At the bottom, there are navigation buttons for '1', '2', and 'Next'.

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	Das.JohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johning Science	2025-11-14 15:42:35.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete



The screenshot shows the same student management system, but the data is paginated. It displays the first 10 students from a total of 20. The current page is 2. Navigation buttons at the bottom include 'Previous', '1', and '2'.

ID	Student Code	Full Name	Email	Major	Created At	Actions
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

Implementation

```
int currentPage = 1;

if (pageParam != null) {
    try { currentPage = Integer.parseInt(pageParam); } catch
    (NumberFormatException e) { currentPage = 1; }

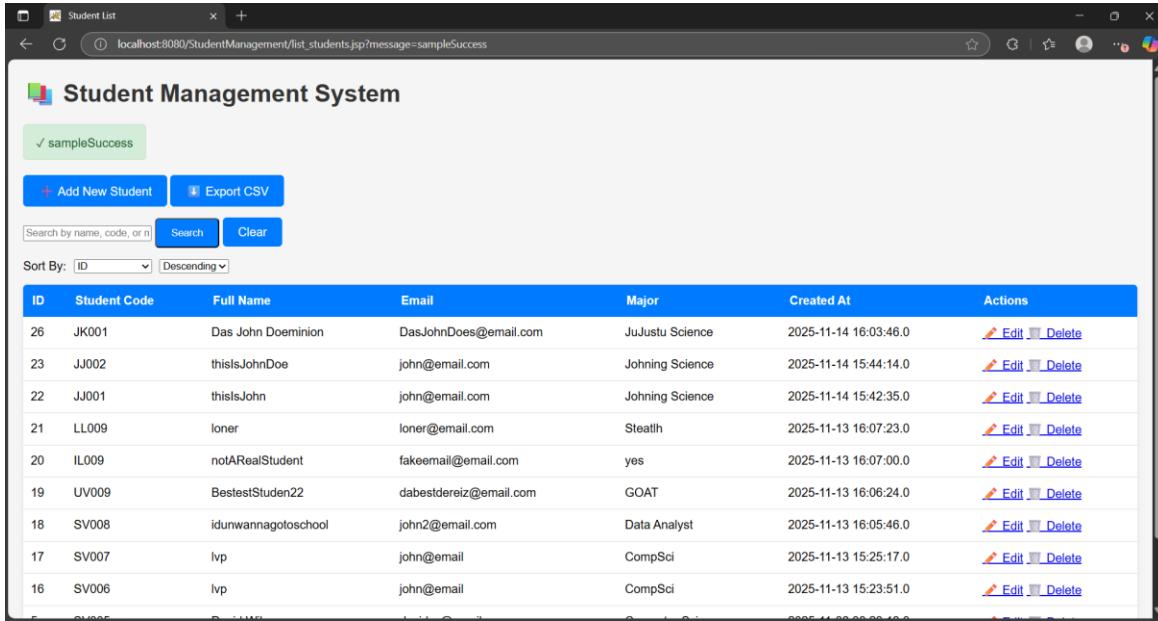
}
```

```
int recordsPerPage = 10;  
int offset = (currentPage - 1) * recordsPerPage;  
  
String countSql;  
  
PreparedStatement countStmt;  
  
if (keyword != null && !keyword.trim().isEmpty()) {  
  
    countSql = "SELECT COUNT(*) FROM students WHERE full_name  
LIKE ? OR student_code LIKE ? OR major LIKE ?"; //Replace the search  
query  
  
    countStmt = conn.prepareStatement(countSql);  
  
    String search = "%" + keyword.trim() + "%";  
  
    countStmt.setString(1, search);  
  
    countStmt.setString(2, search);  
  
    countStmt.setString(3, search);  
  
} else {  
  
    countSql = "SELECT COUNT(*) FROM students";  
  
    countStmt = conn.prepareStatement(countSql);  
  
}  
  
  
ResultSet countRs = countStmt.executeQuery();  
countRs.next();  
  
int totalRecords = countRs.getInt(1);  
  
int totalPages = (int) Math.ceil((double) totalRecords / recordsPerPage);  
countRs.close();  
countStmt.close();  
  
<div class="pagination">
```

```
<% if (currentPage > 1) { %>
    <a href="list_students.jsp?page=<%= currentPage - 1
%>">Previous</a>
    <% } %>

<% for (int i = 1; i <= totalPages; i++) { %>
    <% if (i == currentPage) { %>
        <strong><%= i %></strong>
    <% } else { %>
        <a href="list_students.jsp?page=<%= i %>"><%= i %></a>
    <% } %>
    <% } %>
<% if (currentPage < totalPages) { %>
    <a href="list_students.jsp?page=<%= currentPage + 1 %>">Next</a>
<% } %>
</div>
```

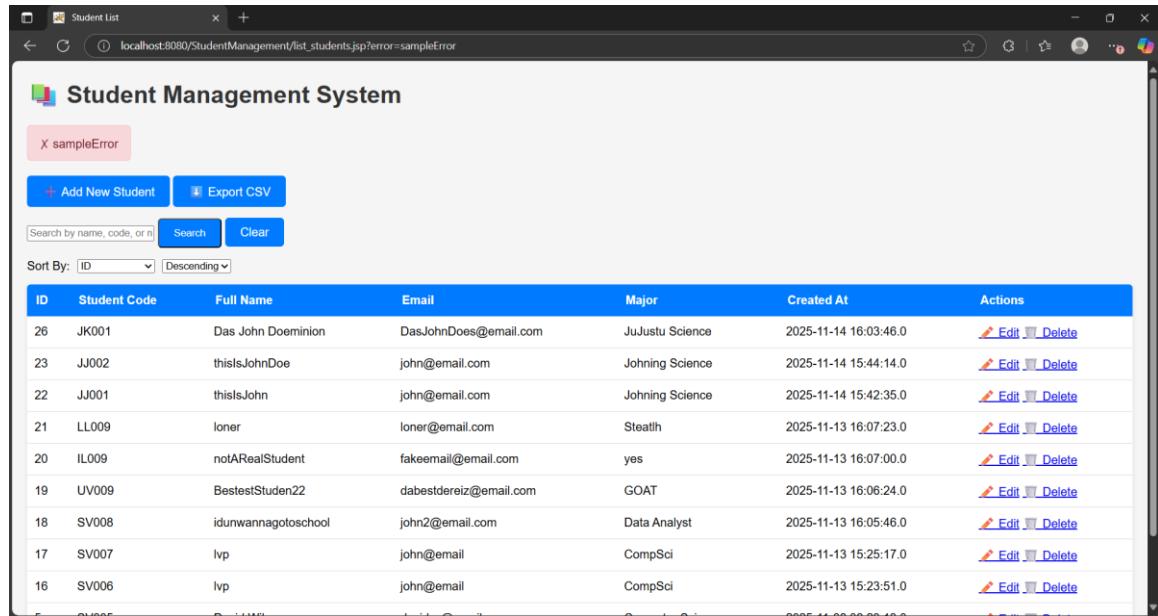
7.2: Improved UI/UX



The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?message=sampleSuccess". The page title is "Student Management System". A green success message box displays "✓ sampleSuccess". Below it are two buttons: "Add New Student" and "Export CSV". There is a search bar with placeholder text "Search by name, code, or ID" and two buttons: "Search" and "Clear". A dropdown menu for sorting is set to "ID Descending". The main content is a table with the following data:

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete

Success message



The screenshot shows a web browser window titled "Student List" with the URL "localhost:8080/StudentManagement/list_students.jsp?error=sampleError". The page title is "Student Management System". A red error message box displays "X sampleError". Below it are two buttons: "Add New Student" and "Export CSV". There is a search bar with placeholder text "Search by name, code, or ID" and two buttons: "Search" and "Clear". A dropdown menu for sorting is set to "ID Descending". The main content is a table with the same data as the previous screenshot:

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
17	SV007	lvp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
16	SV006	lvp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete

Sample error message

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johning Science	2025-11-14 15:42:35.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

1

Loading state

ID	Student Code	Full Name	Email	Major	Created At	Actions
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johning Science	2025-11-14 15:42:35.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete

Responsive table

IV. BONUS EXERCISES

1. Export to CSV

Implementation:

```
<%
response.setContentType("text/csv");
response.setHeader("Content-Disposition", "attachment;
filename=\"students.csv\"");
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/student_management",
        "",
        ""
    );
    String keyword = request.getParameter("keyword");
    String sql;
    if (keyword != null && !keyword.trim().isEmpty()) {
        sql = "SELECT * FROM students WHERE full_name LIKE ? OR
student_code LIKE ? OR major LIKE ? ORDER BY id DESC";
    }
}
```

```
pstmt = conn.prepareStatement(sql);

String search = "%" + keyword.trim() + "%";
pstmt.setString(1, search);
pstmt.setString(2, search);
pstmt.setString(3, search);

} else {

    sql = "SELECT * FROM students ORDER BY id DESC";
    pstmt = conn.prepareStatement(sql);
}

rs = pstmt.executeQuery();

out.println("ID,Student Code,Full Name,Email,Major");

while (rs.next()) {

    String id = String.valueOf(rs.getInt("id"));
    String studentCode = rs.getString("student_code");
    String fullName = rs.getString("full_name");
    String email = rs.getString("email");
    String major = rs.getString("major");

    out.println("'" + id + "','" + studentCode + "','" + fullName + "','" + email + "','" + major + "'");

}

} catch (Exception e) {
```

```

out.println("Error exporting CSV: " + e.getMessage());

} finally {

    if (rs != null) rs.close();

    if (stmt != null) stmt.close();

    if (conn != null) conn.close();

}

%>

```

ID	Student ID	Full Name	Email	Major
7	26 JK001	Das John Doeminion	DasJohnDoes@email.com	Julustu Science
8	23 J002	thisisJohnDoe	john@email.com	Johnning Science
9	22 J001	thisisJohn	john@email.com	Johnning Science
10	21 LL009	loner	loner@email.com	Steath
11	20 LL009	notARealStudent	fakeemail@email.com	yes
12	19 UV009	BestestStuden22	dabestdereiz@email.com	GOAT
13	18 SV008	idunwannagotoschool	john2@email.com	Data Analyst
14	17 SV007	lvp	john@email	Compsci
15	16 SV006	lvp	john@email	CompSci
16	5 SV005	David Wilson	david.w@email.com	Computer Science
17	4 SV004	Sarah Davis	sarah.d@email.com	Data Science
18	3 SV003	Michael Brown	michael.b@email.com	Software Engineering
19	2 SV002	Emily Johnson	emily.j@email.com	Information Technology
20	1 SV001	John Smith	john.smith@email.com	Computer Science

students.csv in excel

2. Sort by Column

Sorting is done through links, these links act as toggle for the search. For example, if the user clicks on ID, it sorts by ID in ascending order, if they click again, it is sorted by descending order.

Student Management System

ID	Student Code	Full Name	Email	Major	Created At	Actions
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
16	SV006	Ivp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
17	SV007	Ivp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
20	IL009	notARRealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete

1 2 Next

Sort by ID

Student Management System

ID	Student Code	Full Name	Email	Major	Created At	Actions
23	JJ002	thisisJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisisJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
20	IL009	notARRealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
16	SV006	Ivp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
17	SV007	Ivp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete

1 2 Next

Sort by full name

*Issues:

- Does not work with pagination [FIXED]
- Does not work with the search function [FIXED]

Student List

localhost:8080/StudentManagement/list_students.jsp?sort=id&order=asc&keyword=&page=1

Student Management System

Add New Student Export CSV

Search by name, code, or ID Search Clear

Sort By: ID Ascending

ID	Student Code	Full Name	Email	Major	Created At	Actions
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
2	SV002	Emily Johnson	emily.j@email.com	Information Technology	2025-11-08 09:23:13.0	Edit Delete
3	SV003	Michael Brown	michael.b@email.com	Software Engineering	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
16	SV006	Ivp	john@email	CompSci	2025-11-13 15:23:51.0	Edit Delete
17	SV007	Ivp	john@email	CompSci	2025-11-13 15:25:17.0	Edit Delete
18	SV008	idunwannagotoschool	john2@email.com	Data Analyst	2025-11-13 16:05:46.0	Edit Delete
19	UV009	BestestStuden22	dabestdereiz@email.com	GOAT	2025-11-13 16:06:24.0	Edit Delete
20	IL009	notARealStudent	fakeemail@email.com	yes	2025-11-13 16:07:00.0	Edit Delete

1 2 Next

New sort

Student List

localhost:8080/StudentManagement/list_students.jsp?sort=id&order=asc&keyword=&page=2

Student Management System

Add New Student Export CSV

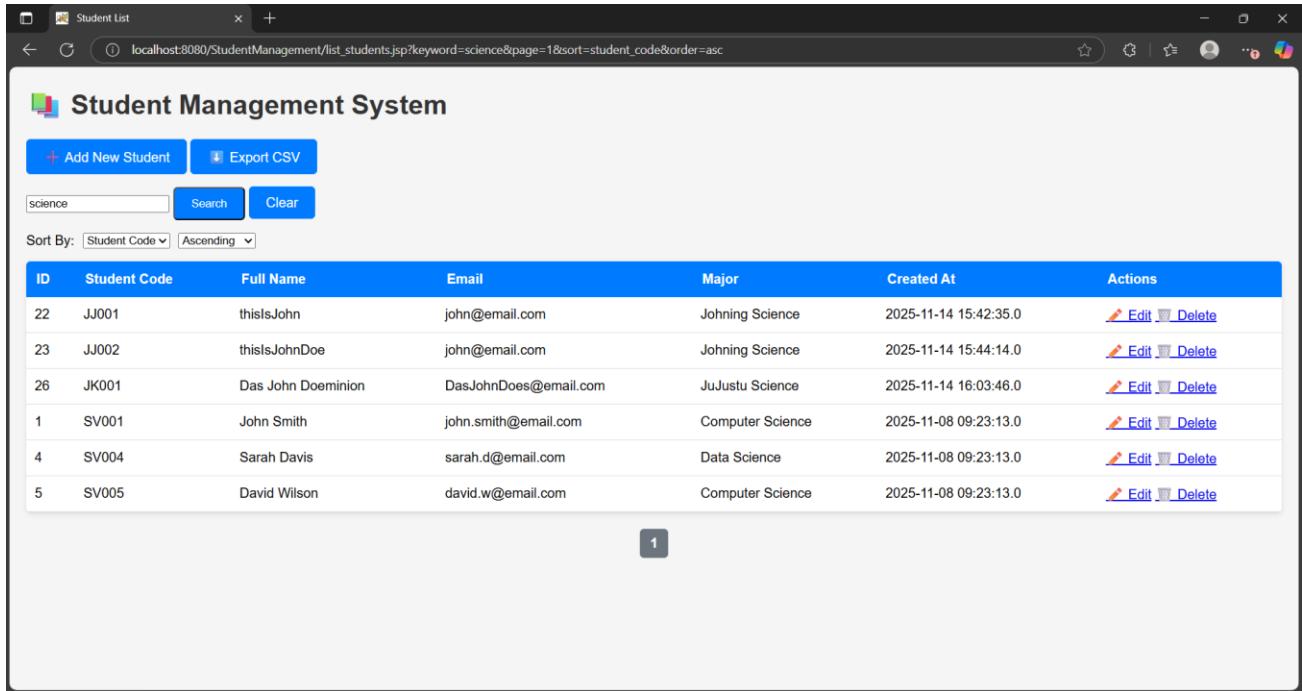
Search by name, code, or ID Search Clear

Sort By: ID Ascending

ID	Student Code	Full Name	Email	Major	Created At	Actions
21	LL009	loner	loner@email.com	Stealth	2025-11-13 16:07:23.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete

Previous 1 2

Keep sorts through pagination



The screenshot shows a web application titled "Student Management System". At the top, there are buttons for "Add New Student" and "Export CSV". Below that is a search bar with the word "Science" and buttons for "Search" and "Clear". A dropdown menu "Sort By:" is set to "Student Code" with "Ascending" selected. The main area displays a table of student data:

ID	Student Code	Full Name	Email	Major	Created At	Actions
22	JJ001	thisIsJohn	john@email.com	Johnning Science	2025-11-14 15:42:35.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johnning Science	2025-11-14 15:44:14.0	Edit Delete
26	JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science	2025-11-14 16:03:46.0	Edit Delete
1	SV001	John Smith	john.smith@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
5	SV005	David Wilson	david.w@email.com	Computer Science	2025-11-08 09:23:13.0	Edit Delete

At the bottom center of the table is a small number "1".

Sort and search

Implementation:

```
<%
String order = request.getParameter("order");
String sortBy = request.getParameter("sort");
String keyword = request.getParameter("keyword");
String pageParam = request.getParameter("page");

// Default
if (sortBy == null || sortBy.trim().isEmpty()) sortBy = "id";
if (order == null || order.trim().isEmpty()) order = "desc";

if(keyword == null){ //THIS IS A HACK TO DEAL WITH EMPTY
KEYWORD
    keyword = "";
}
```

```
    }
```



```
%>
```

```
<form method="get" style="margin-bottom: 15px; display:flex; gap:10px; align-items:center;">
```

```
<input type="hidden" name="keyword" value="<% = keyword %>">
```



```
<input type="hidden" name="page" value="<% = pageParam %>">
```



```
<label>Sort By:</label>
```



```
<select name="sort" onchange="this.form.submit()">
```

```
    <option value="id" <% = sortBy.equals("id") ? "selected" : "" %>>ID</option>
```

```
    <option value="student_code" <% = sortBy.equals("student_code") ? "selected" : "" %>>Student Code</option>
```

```
    <option value="full_name" <% = sortBy.equals("full_name") ? "selected" : "" %>>Full Name</option>
```

```
    <option value="email" <% = sortBy.equals("email") ? "selected" : "" %>>Email</option>
```

```
    <option value="major" <% = sortBy.equals("major") ? "selected" : "" %>>Major</option>
```

```
    <option value="created_at" <% = sortBy.equals("created_at") ? "selected" : "" %>>Created At</option>
```

```
</select>
```

```
<select name="order" onchange="this.form.submit()">
```

```
    <option value="asc" <% = order.equals("asc") ? "selected" : "" %>>Ascending</option>
```

```

<option value="desc" <%= order.equals("desc") ? "selected" : ""
%>>Descending</option>
</select>
```

Modified query:

```

sql = "SELECT * FROM students WHERE full_name LIKE ? OR student_code
LIKE ? OR major LIKE ? ORDER BY " + sortBy + " " + order + " LIMIT ?
OFFSET ?";
else {
    sql = "SELECT * FROM students ORDER BY " + sortBy + " " + order
+ " LIMIT ? OFFSET ?";
```

Modified pagination:

```

<div class="pagination">
<% if (currentPage > 1) { %>
    <a
    href="list_students.jsp?sort=<%=sortBy%>&order=<%=order%>&keyword=<
    %=keyword%>&page=<%= currentPage - 1 %>">Previous</a>
    <% } %>

<% for (int i = 1; i <= totalPages; i++) { %>
    <% if (i == currentPage) { %>
        <strong><%= i %></strong>
    <% } else { %>
        <a
        href="list_students.jsp?sort=<%=sortBy%>&order=<%=order%>&keyword=<
```

```
%=keyword%>&page=<%= i %>"><%= i %></a>
    <% } %>
    <% } %>
<% if (currentPage < totalPages) { %>
    <a
href="list_students.jsp?sort=<%=sortBy%>&order=<%=order%>&keyword=<%=keyword%>&page=<%= currentPage + 1 %>">Next</a>
    <% } %>
</div>
```

V. References

1. Lab manual & resources
2. <https://www.geeksforgeeks.org/java/java-program-to-search-the-contents-of-a-table-in-jdbc/>
3. <https://www.sql-practice.com/> (For SQL query)
4. <https://coderanch.com/t/287048/java/exporting-CSV>
5. <https://docs.oracle.com/javase/tutorial/jdbc/>
6. <https://docs.oracle.com/javaee/5/tutorial/doc/bnagx.html>
7. <https://coderanch.com/f/50/JSP>
8. <https://coderanch.com/c/java>