

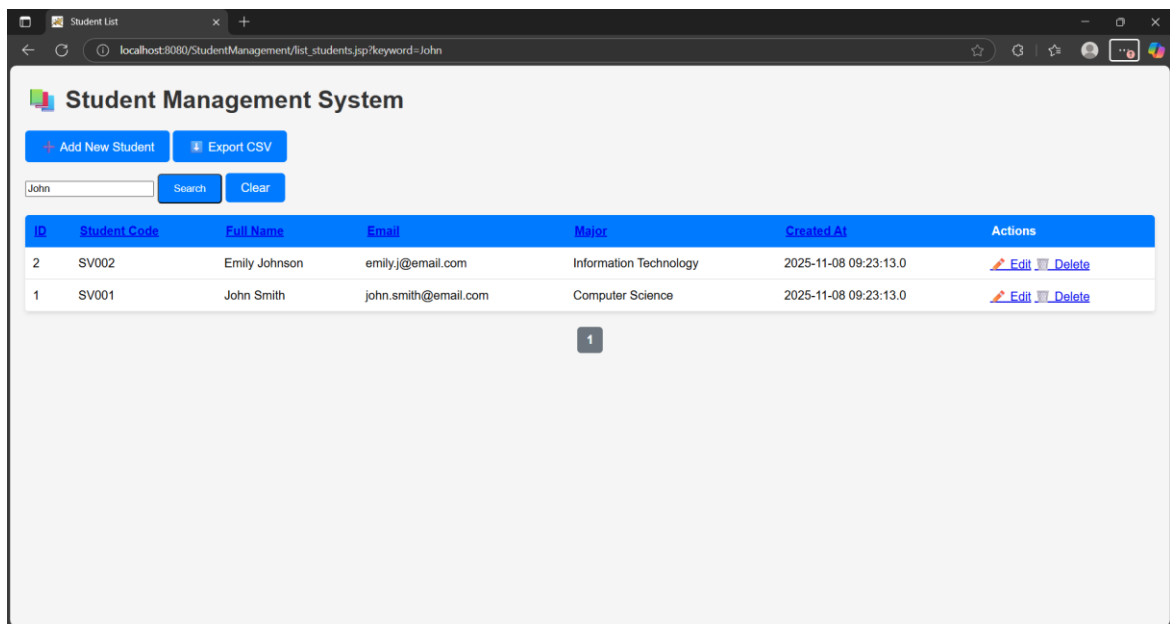
Web Application Development Lab 04

JSP + MySQL

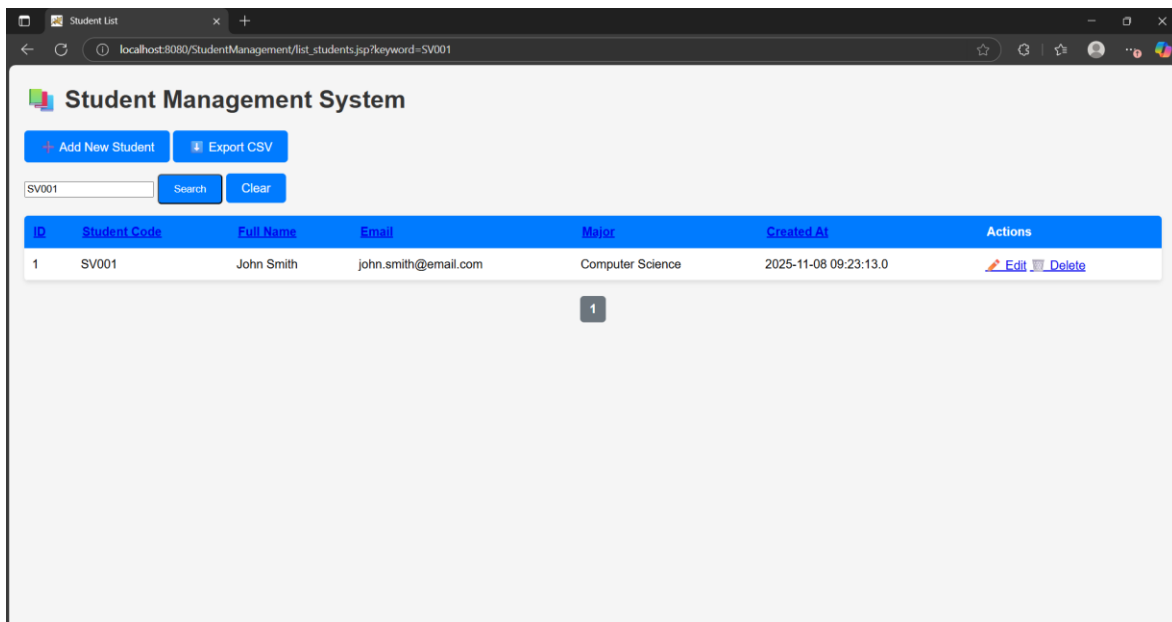
Part B – HOMEWORK

I. EXERCISE 5: SEARCH FUNCTIONALITY

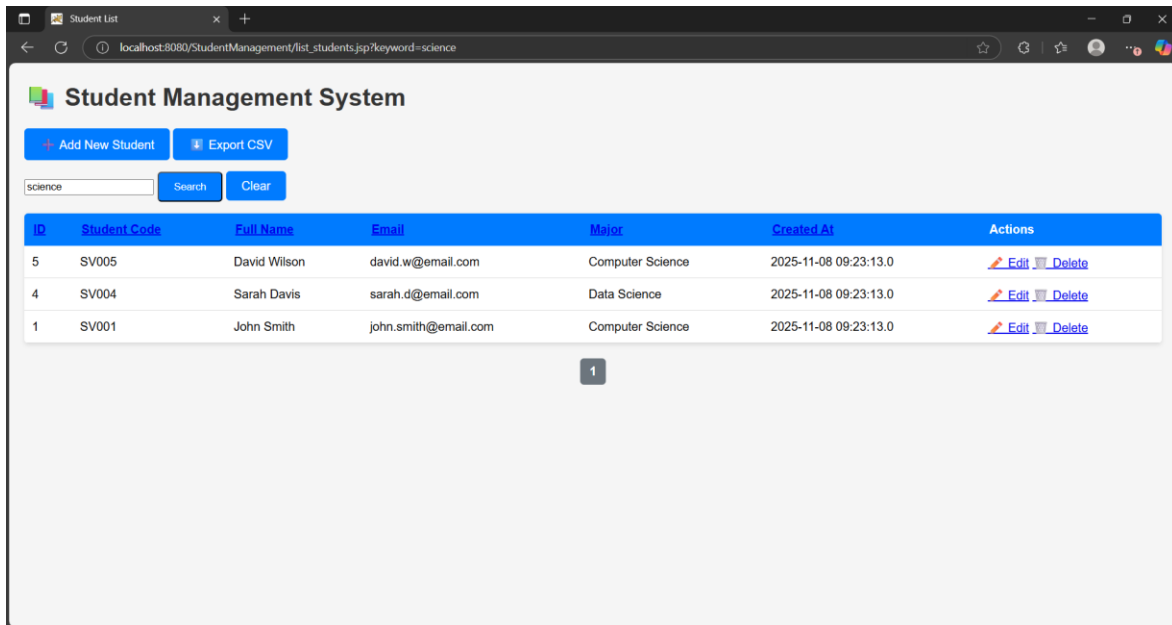
*The screenshots are taken after most of the functionalities are implemented



Searching for “John”



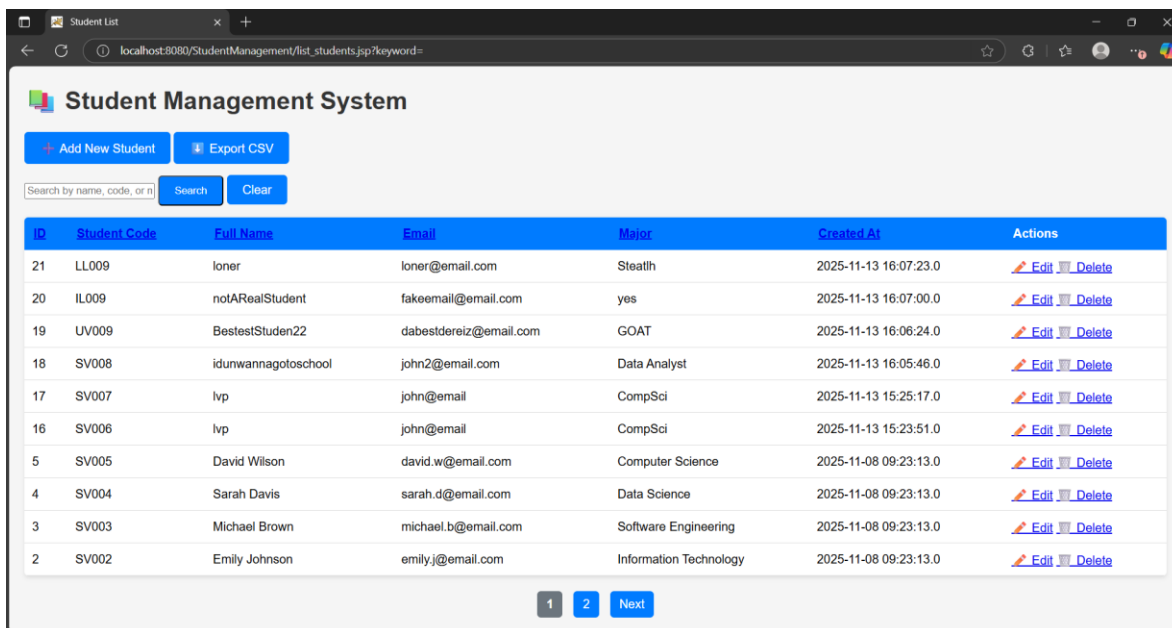
Searching by student code



The screenshot shows the 'Student Management System' interface. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below these is a search bar with the text 'science' and buttons for 'Search' and 'Clear'. The table below displays the results of the search, showing three students. Each row has columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The 'Actions' column contains links for 'Edit' and 'Delete'.

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 the 'Student Management System' interface. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below these is a search bar with the text 'Search by name, code, or n' and buttons for 'Search' and 'Clear'. The table below displays the results of the search, showing 21 students. Each row has columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The 'Actions' column contains links for 'Edit' and 'Delete'.

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	dabestderez@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):

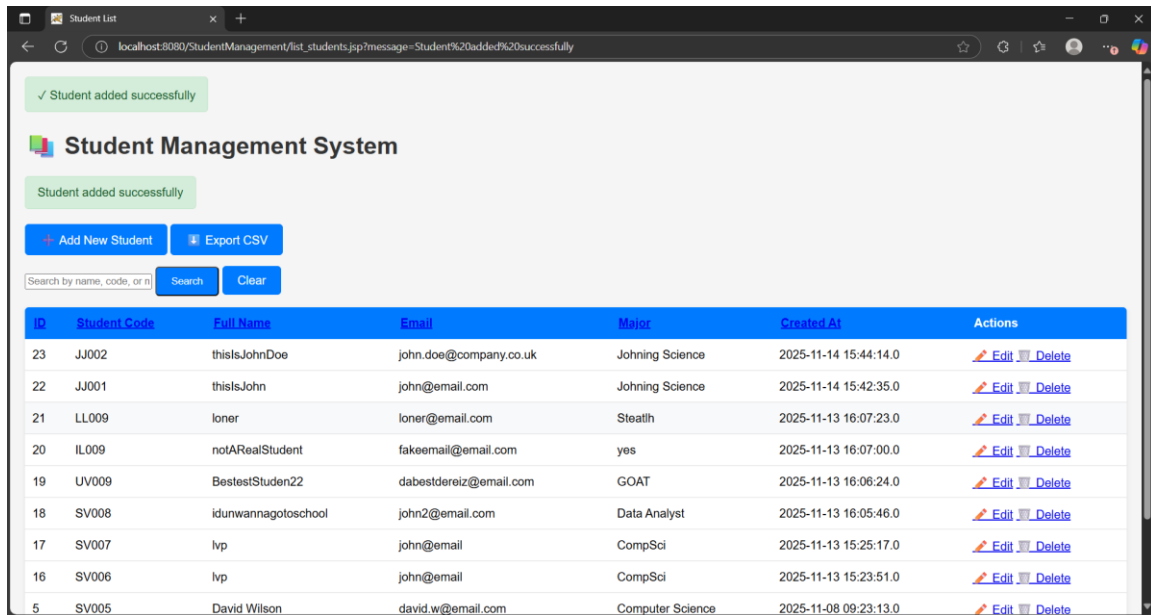
```

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	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
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

Testing john@email.com



✓ Student added successfully

Student Management System

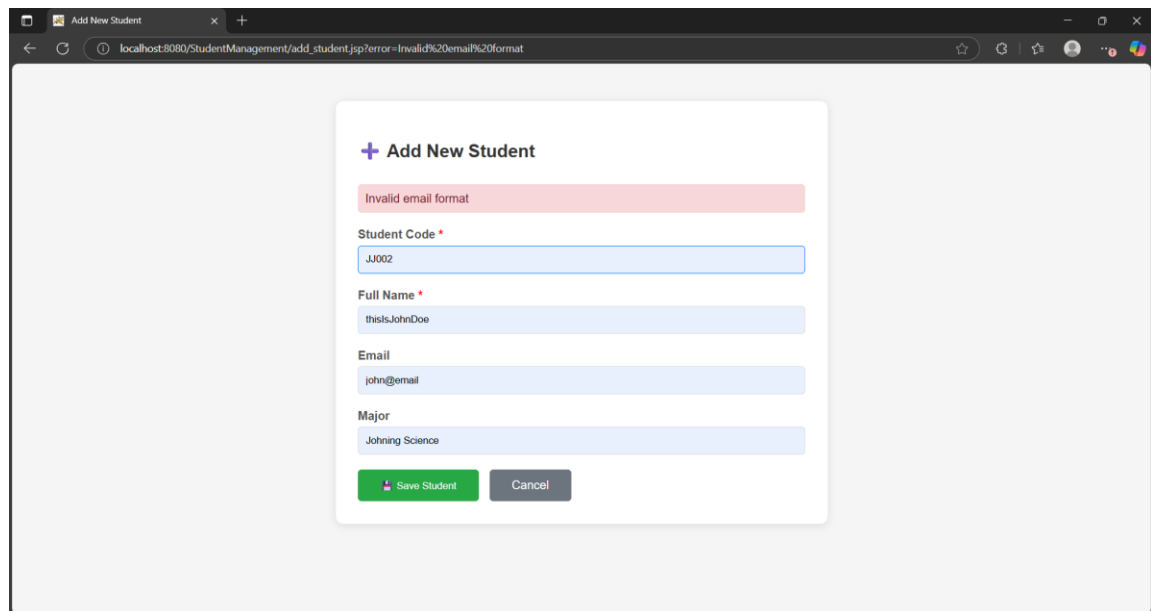
Student added successfully

[+ Add New Student](#) [Export CSV](#)

Search by name, code, or n

ID	Student Code	Full Name	Email	Major	Created At	Actions
23	JJ002	thisIsJohnDoe	john.doe@company.co.uk	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

Testing john.doe@company.co.uk



+ Add New Student

Invalid email format

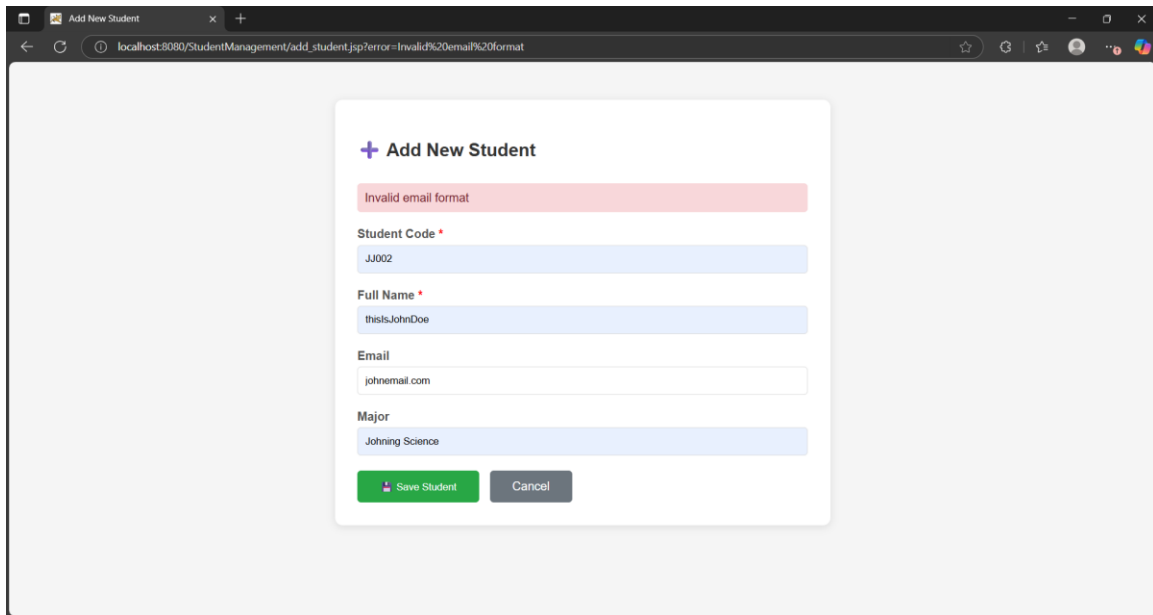
Student Code *

Full Name *

Email

Major

Testing john@email

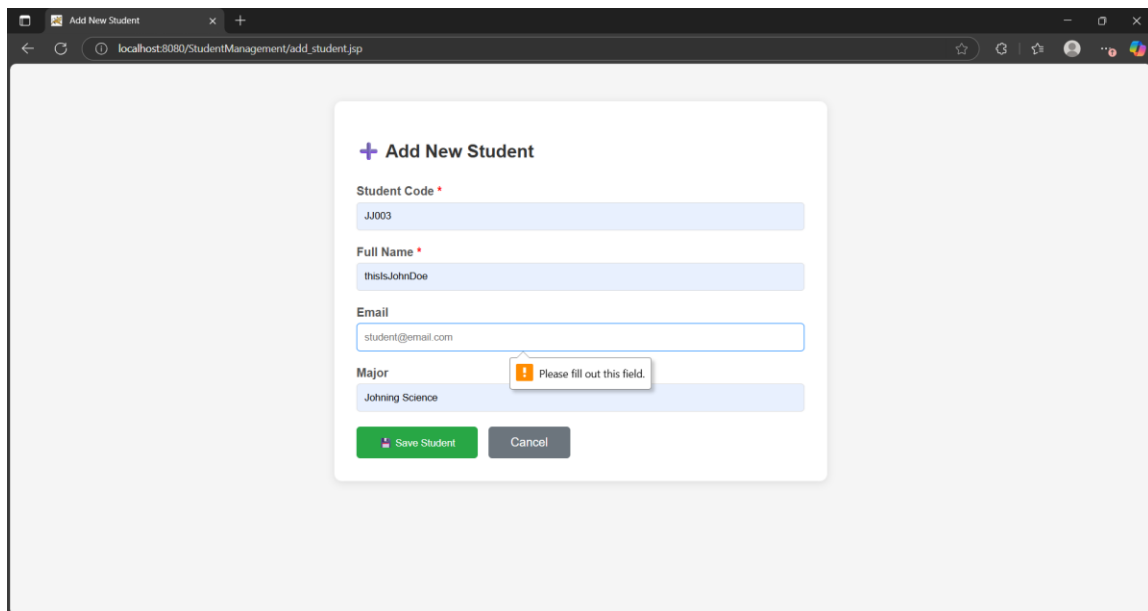


The screenshot shows a web browser window with the title 'Add New Student'. The address bar displays 'localhost:8080/StudentManagement/add_student.jsp?error=Invalid%20email%20format'. The form itself is titled '+ Add New Student' and contains the following fields:

- Student Code ***: JJ002
- Full Name ***: thisIsJohnDoe
- Email**: johnemail.com
- Major**: Johning Science

At the top of the form, there is a red error message: 'Invalid email format'. At the bottom, there are two buttons: 'Save Student' (green) and 'Cancel' (grey).

Testing johnemail.com

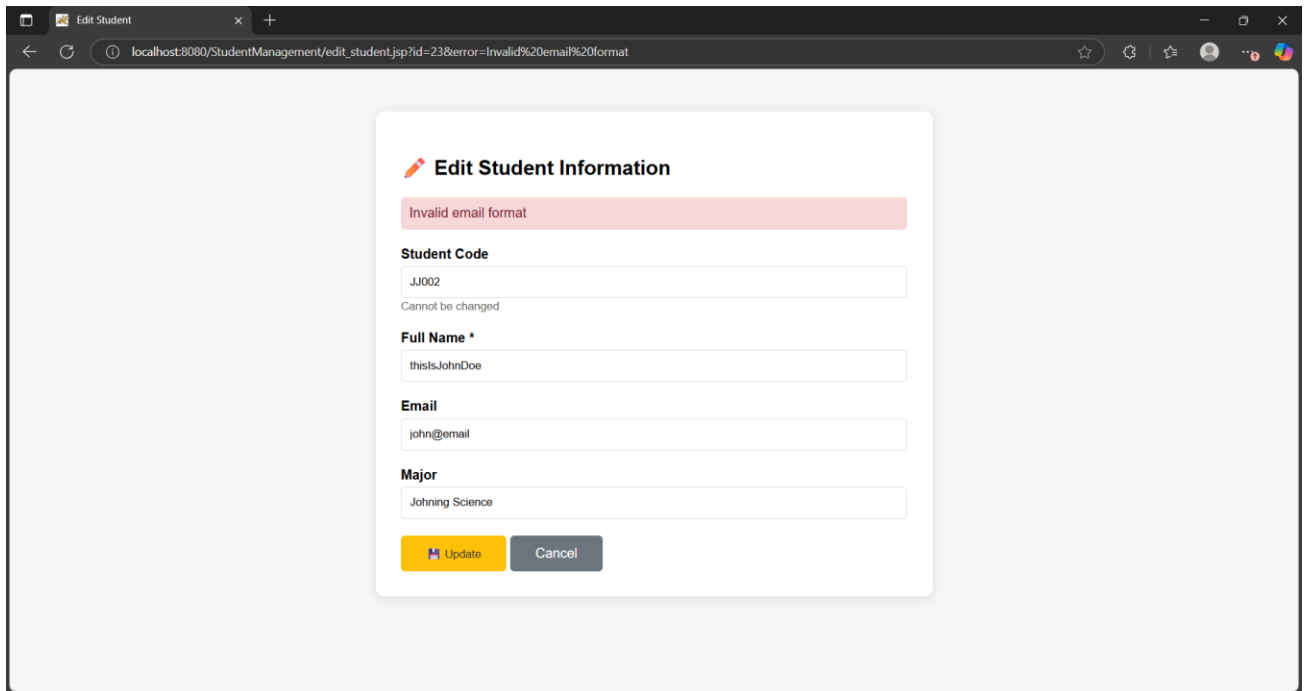


The screenshot shows the same 'Add New Student' form, but with different data and a validation message. The fields are:

- Student Code ***: JJ003
- Full Name ***: thisIsJohnDoe
- Email**: student@email.com
- Major**: Johning Science

A tooltip with an orange warning icon and the text 'Please fill out this field.' is positioned over the 'Major' field. The 'Save Student' and 'Cancel' buttons are still present at the bottom.

Empty field

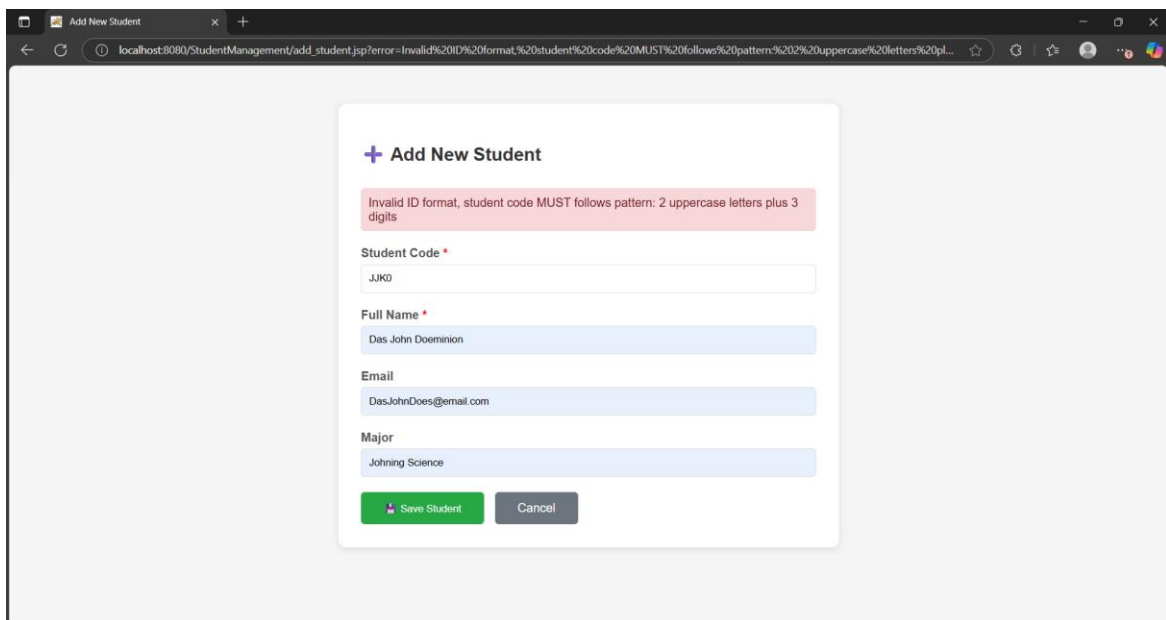


The screenshot shows a web browser window with the address bar displaying `localhost:8080/StudentManagement/edit_student.jsp?id=23&error=Invalid%20email%20format`. The page title is "Edit Student". The form, titled "Edit Student Information", contains a red error message: "Invalid email format". The form fields are: "Student Code" (JJ002, with a note "Cannot be changed"), "Full Name *" (thisIsJohnDoe), "Email" (john@email), and "Major" (Johning Science). At the bottom are "Update" and "Cancel" buttons.

Testing john@email edit

The process_edit results closely mimics that of the process_add

6.2: Student Code Pattern Validation



The screenshot shows a web browser window with the address bar displaying `localhost:8080/StudentManagement/add_student.jsp?error=Invalid%20ID%20format,%20student%20code%20MUST%20follows%20pattern%20%20uppercase%20letters%20pl...`. The page title is "Add New Student". The form, titled "+ Add New Student", contains a red error message: "Invalid ID format, student code MUST follows pattern: 2 uppercase letters plus 3 digits". The form fields are: "Student Code *" (JJK0), "Full Name *" (Das John Doeminion), "Email" (DasJohnDoes@email.com), and "Major" (Johning Science). At the bottom are "Save Student" and "Cancel" buttons.

Testing JJK0

Student List

localhost:8080/StudentManagement/list_students.jsp?message=Student%20added%20successfully

✓ Student added successfully

Student Management System

Student added successfully

Add New Student

Export CSV

Search by name, code, or n

Search

Clear

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

Testing for JJK001

III. EXERCISE 7: USER EXPERIENCE IMPROVEMENTS

7.1: Pagination

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
21	LL009	loner	loner@email.com	Steathh	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

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>

    <% } %>

    <% } %>

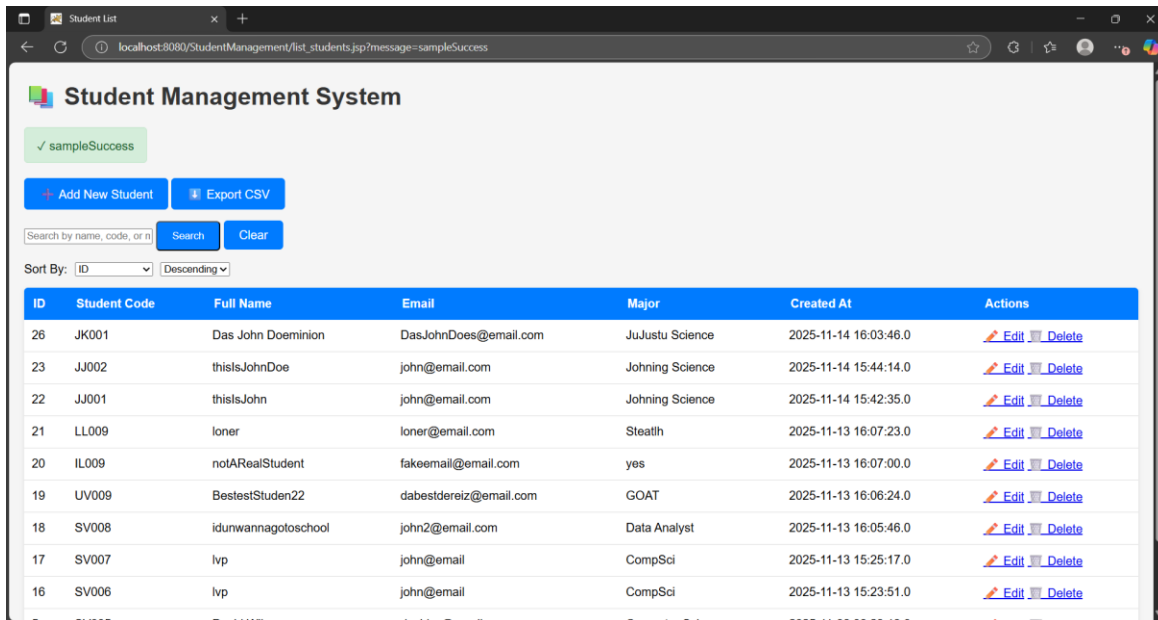
    <% if (currentPage < totalPages) { %>

        <a href="list_students.jsp?page=<%= currentPage + 1 %>">Next</a>

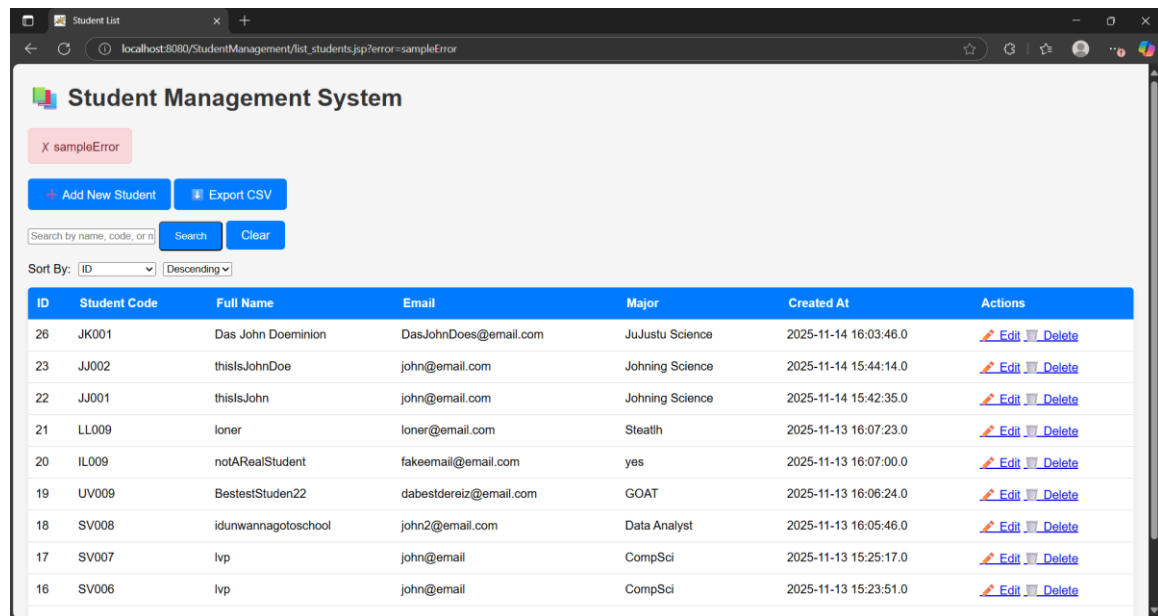
    <% } %>

</div>
```

7.2: Improved UI/UX



Success message



Sample error message

Student Management System

Add New Student Export CSV

science Processing... Clear

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

Student Management System

Add New Student Export CSV

science Processing... Clear

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 (pstmt != null) pstmt.close();
    if (conn != null) conn.close();
}
%>

```

students (2).csv · Excel

File Home Insert Draw Page Layout Formulas Data Review View Developer Help Foxit PDF XDTH

Clipboard Font Alignment Number Styles Cells Editing Add-ins

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

ID	Student Cc Full Name	Email	Major
26 JK001	Das John Doeminion	DasJohnDoes@email.com	JuJustu Science
23 JI002	thisisJohnDoe	john@email.com	Johning Science
22 JI001	thisisJohn	john@email.com	Johning Science
21 LI009	loner	loner@email.com	Steath
20 IL009	notARealStudent	fakeemail@email.com	yes
19 UV009	BestestStuden22	dabestderez@email.com	GOAT
18 SV008	idunwannagotoschool	john2@email.com	Data Analyst
17 SV007	lvp	john@email	CompSci
16 SV006	lvp	john@email	CompSci
5 SV005	David Wilson	david.w@email.com	Computer Science
4 SV004	Sarah Davis	sarah.d@email.com	Data Science
3 SV003	Michael Brown	michael.b@email.com	Software Engineering
2 SV002	Emily Johnson	emily.j@email.com	Information Technology
1 SV001	John Smith	john.smith@email.com	Computer Science

students (2)

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.

The screenshot shows a web browser window titled "Student List" with the URL `localhost:8080/StudentManagement/list_students.jsp?sort=id&order=asc`. The page displays the "Student Management System" header with buttons for "Add New Student" and "Export CSV". Below is a search bar and a table of students. The table is sorted by ID in ascending order. The table has 7 columns: 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
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

At the bottom of the table, there are pagination controls showing "1", "2", and "Next".

Sort by ID

The screenshot shows the same "Student Management System" interface, but the URL is `localhost:8080/StudentManagement/list_students.jsp?sort=full_name&order=desc`. The table is now sorted by Full Name in descending order. The table structure and headers are identical to the previous screenshot.

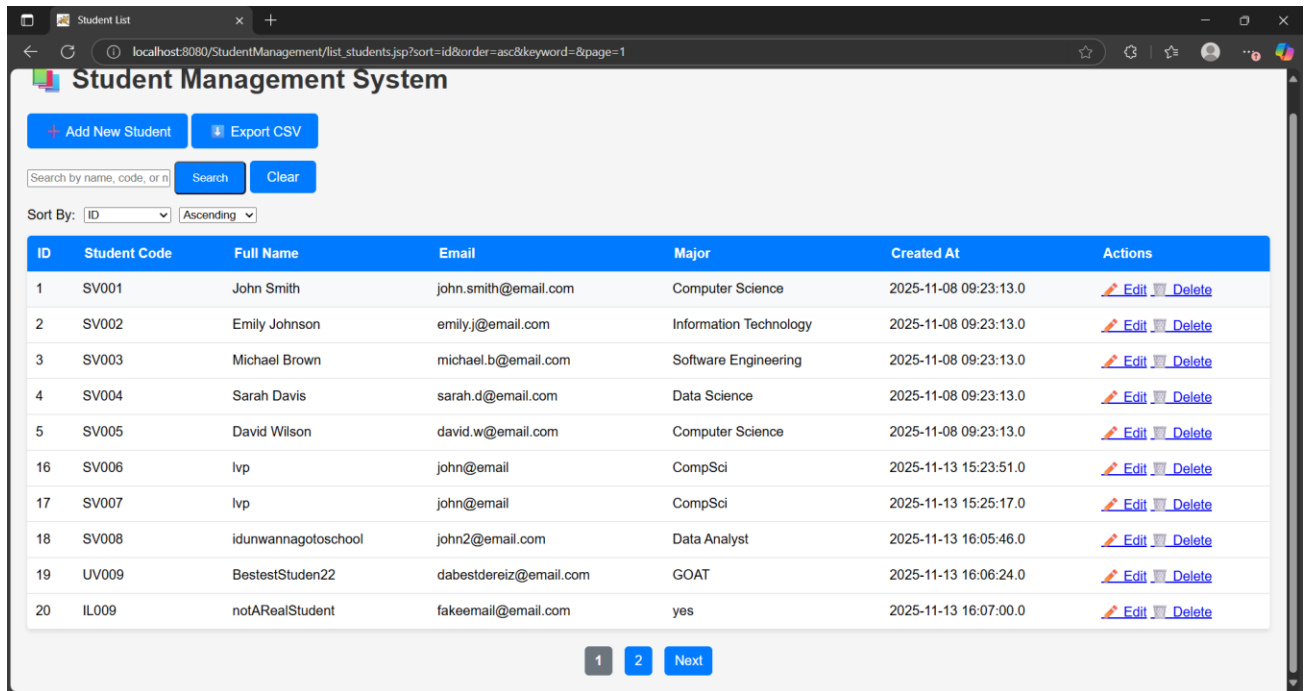
ID	Student Code	Full Name	Email	Major	Created At	Actions
23	JJ002	thisIsJohnDoe	john@email.com	Johning Science	2025-11-14 15:44:14.0	Edit Delete
22	JJ001	thisIsJohn	john@email.com	Johning Science	2025-11-14 15:42:35.0	Edit Delete
4	SV004	Sarah Davis	sarah.d@email.com	Data Science	2025-11-08 09:23:13.0	Edit Delete
20	IL009	notARealStudent	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	Ioner	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

At the bottom of the table, there are pagination controls showing "1", "2", and "Next".

Sort by full name

*Issues:

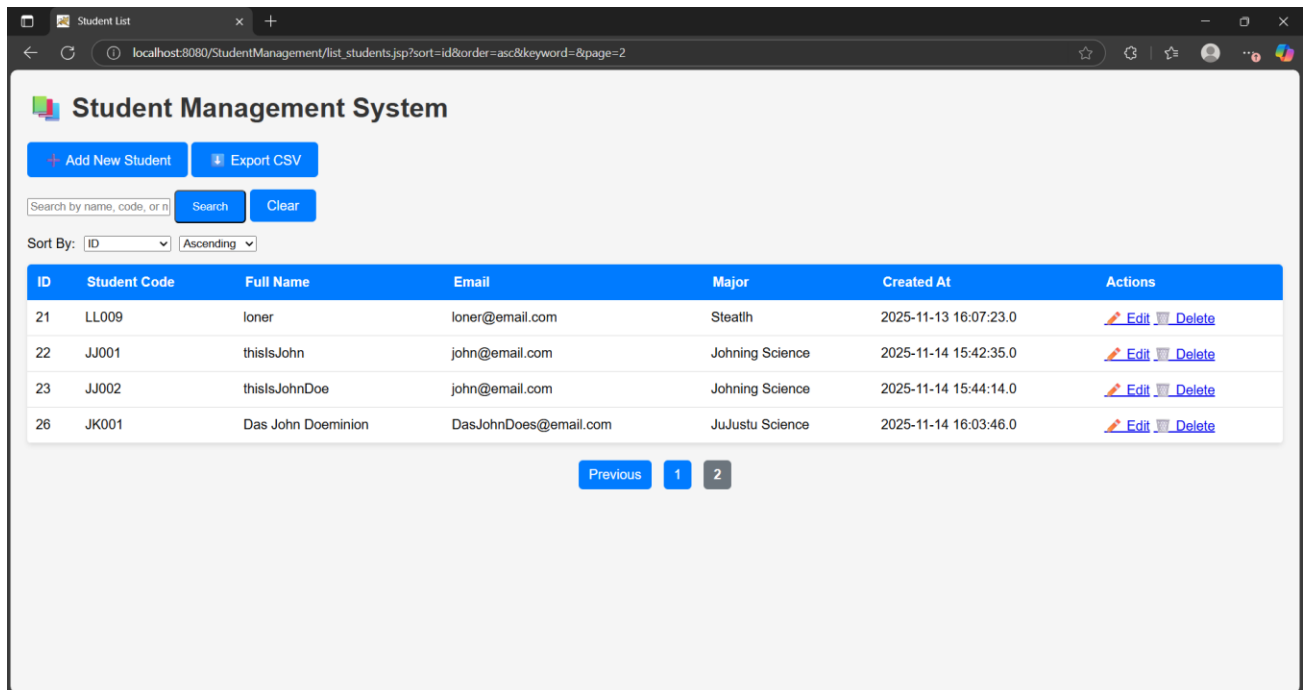
- Does not work with pagination [FIXED]
- Does not work with the search function [FIXED]



The screenshot shows the 'Student Management System' interface. At the top, there are buttons for 'Add New Student' and 'Export CSV'. Below these is a search bar with the placeholder text 'Search by name, code, or n' and buttons for 'Search' and 'Clear'. The 'Sort By' dropdown is set to 'ID' and 'Ascending'. The table lists 20 students, with columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The first 15 students are visible, and the last student is partially visible. The pagination controls at the bottom show '1' as the current page, '2' as the next page, and a 'Next' button.

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

New sort



The screenshot shows the 'Student Management System' interface on page 2. The search bar and 'Sort By' dropdown are still set to 'ID' and 'Ascending'. The table lists 4 students, with columns for ID, Student Code, Full Name, Email, Major, Created At, and Actions. The pagination controls at the bottom show 'Previous', '1', and '2' as the current page.

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	Johning Science	2025-11-14 15:42:35.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johning 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

Keep sorts through pagination

Student Management System

[Add New Student](#) [Export CSV](#)

science [Search](#) [Clear](#)

Sort By: Student Code Ascending

ID	Student Code	Full Name	Email	Major	Created At	Actions
22	JJ001	thisIsJohn	john@email.com	Johning Science	2025-11-14 15:42:35.0	Edit Delete
23	JJ002	thisIsJohnDoe	john@email.com	Johning 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

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>