

CRUD Case Study: Student Management System

1. Introduction

This case study is designed to help trainees understand how to perform all four CRUD operations (Create, Retrieve, Update, and Delete) in a full stack of application contexts. The scenario chosen is the Student Management system, where the user can manage a registry of students.

2. Objective

Build a web-based Student Management System using HTML, CSS and JavaScript/TypeScript (Frontend) and any backend technology of your choice (e.g. Spring Boot / Servlet)

3. Technologies Allowed

- Frontend: HTML, CSS, JavaScript (external) /Angular(TypeScript)
- JavaScript/TypeScript (external)
- Backend of your choice (Spring Boot/Servlets)
- Database: H2 or derby

4. Pages to be Designed

1. Home Page
2. Add Student Details Page
3. View Student Details Page
4. Update Student Details Page
5. Delete Student Details Functionality

A Nav Bar with the logo are compulsory on each page.

5. Database table structure:

Table Name: **Student_Management**

<u>Field Name</u>	<u>Data Type</u>	<u>Constraints</u>	<u>Example</u>	<u>Remarks</u>
Student_id	Integer	Primary Key	1001	Auto-generated
Name	VARCHAR(100)	NOT NULL	John Doe	Name of the student
DOB	DATE	NOT NULL	2000-01-01	DOB of the student
Class	VARCHAR(5)	NOT NULL	8th	Class number of the Student
Division	VARCHAR(5)	NOT NULL	A/B/C/D	Division of the Student
Subject	VARCHAR(50)	Default Science	Maths	Stream or Subject that the student is studying
Address	VARCHAR(100)	NOT NULL	Utopian Heights, Makers Galaxy	Permanent Address of the Student

6. Tasks

6.1 Create (Add Student)

Design a form to insert a new Student into the database. The student id for the application should be auto – generated. Please find attached screen shot for reference:

The screenshot shows a web-based form titled "Add Student". At the top left, there is a navigation bar with links: "Student Management", "Home", "Add Student", and "View Student". The main form area has a blue header bar with the title "Add Student". Below this, there are seven input fields: "Student Id" (with a placeholder "mm/dd/yyyy" and a calendar icon), "Student Name", "Date Of Birth" (with a placeholder "mm/dd/yyyy" and a calendar icon), "Class", "Division", "Subject", and "Student Address". At the bottom of the form is a blue "Submit" button.

6.2 Retrieve (View Student Details)

Design a web page to have 1 table, to display the student information

The table will display all the details that we collect at the time of inserting new records into the database. The table will have 2 buttons at the end of each student's record.

Following is the functionality of the two buttons:

- **Update Student Details:** With the update button, we can update the details of the student (except the student_id which is auto generated).
- **Delete Student Details:** With delete button, we will be able to delete(soft-delete) the student record. You can let the record be

present in the table but make the record inaccessible or disabled so that no further changes can be made to the student record on which we have performed the delete operation.

Please find the reference for the page.

Student Management									
Home Add Student View Student									
Id	Name	DOB	Class	Division	Subject	Address	Edit	Delete	
1	Mark	2000-01-01	8th	A	English	Mumbai	Update	Delete	
2	John	2000-01-01	8th	A	Maths	Navi Mumbai	Update	Delete	
3	Jake	2000-01-01	8th	A	Maths	Mumbai	Update	Delete	
4	Amy	2000-01-01	11th	B	Maths	Gujrat	Update	Delete	
5	Holt	2000-01-01	9th	B	Social Studies	Thane	Update	Delete	
6	Terry	2000-01-01	7th	C	Science	Delhi	Update	Delete	

6.3 Update (Update Student)

Once the user clicks on the update, the application should redirect the user to the update page with prefilled information. User should be able to update all details except student_id.

Please find the attached screenshot for reference.

Student Management [Home](#) [Add Student](#) [View Student](#)

Update Student

Student Id

Student Name

Date Of Birth
 (i)

Class

Division

Subject

Student Address

6.4 Delete Action

For the delete action, make sure that when interacting with the delete button, the record of the student must be deleted, and the View Details page should be updated accordingly.

7. Notes & Submission Instructions

- Maintain proper file and folder structure. Follow MVC Architecture
- Use only external CSS and JS.
- Make sure all the actions performed have appropriate validations and acknowledgement messages/pop-ups
- Submit your project folder as a zipped file with naming convention:
Web_EmpName_EmpID.zip.