

Exp No. 9	Student Management System using MySQL, Node.js, DOM and AJAX

Problem Statement

Develop a simple Student Management Website that allows users to add, view, search, and delete student records. The application must use HTML/CSS for the user interface, JavaScript DOM for dynamic webpage manipulation, AJAX for communication between the browser and server without reloading the page, Node.js for the backend server, and MySQL for storing student information.

Do not use any frontend framework such as React, Angular, or Vue.

1. MySQL Database

Create a database named:

student_db

Create a table named:

students

Field	Data Type	Constraint
id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(100)	
roll_no	VARCHAR(20)	UNIQUE
department	VARCHAR(50)	
semester	INT	
email	VARCHAR(100)	

Insert at least 5 sample student records.

2. Website Design

Create a webpage containing a Student Registration Form with the following fields:

- Student Name
- Roll Number
- Department
- Semester

- Email

Provide an “Add Student” button.

Below the form, display a Student List in an HTML table with columns such as:

ID	Name	Roll No	Department	Semester	Email	Action
1	Rahul	CS101	CSE	6	rahul@gmail.com	Delete

Use CSS to make the page reasonably attractive and readable.

3. Node.js Backend

Create a Node.js server that connects to MySQL and provides the following operations:

GET — Display students

GET /students — Retrieve all student records from MySQL.

POST — Add student

POST /students — Receive student information from the browser and insert it into MySQL.

DELETE — Delete student

DELETE /students/:id — Delete the student corresponding to the given ID.

The Node.js server should return appropriate JSON responses. For example:

```
{ "message": "Student added successfully" }
```

4. AJAX and DOM Manipulation

Use AJAX to communicate with the Node.js server. Students may use either XMLHttpRequest or fetch().

When the page loads, retrieve the student records using AJAX and display them dynamically using JavaScript DOM methods.

Use methods such as document.createElement() or element.innerHTML() to generate/update the student table.

Important:

After adding or deleting a student, do not reload the webpage. The student table must be updated using AJAX and DOM manipulation.

5. Search Feature

Add a search box:

Search by Name or Roll Number: [_____] [Search]

When the user clicks Search, use AJAX to retrieve matching student records and display the results dynamically in the student table.

Expected Functionality

The final website should support the following workflow:

- Add Student → AJAX → Node.js → MySQL INSERT
- View Students → AJAX → Node.js → MySQL SELECT
- Search Student → AJAX → Node.js → MySQL SELECT with a condition
- Delete Student → AJAX → Node.js → MySQL DELETE
- AJAX Response → DOM Update without reloading the webpage

Suggested Time Allocation

Task	Time
MySQL database/table	10 minutes
HTML/CSS interface	15 minutes
Node.js + MySQL connection	20 minutes
GET/POST/DELETE APIs	20 minutes
AJAX + DOM manipulation	20 minutes
Search + testing	15 minutes
Total	100 minutes

Submission Requirements

Submit the following project structure:

student-management/

server.js

package.json

index.html

style.css

script.js

Also provide the SQL commands used to create the database and table.

Restrictions

Do not use Express, React, Angular, Bootstrap, or any other framework. Use Node.js, MySQL, HTML, CSS, JavaScript DOM, and AJAX only.