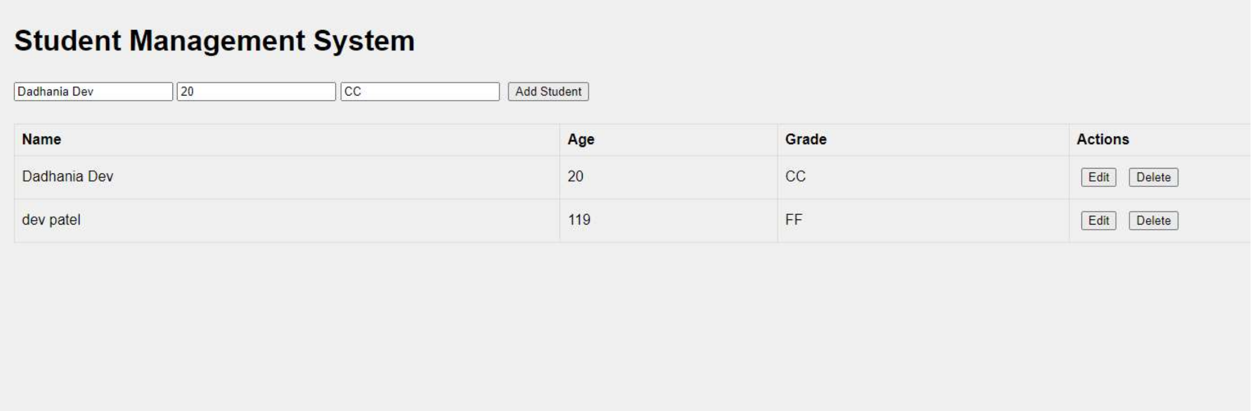
# Practical 13 - WT

JavaScript Arrays and Objects
Task: Student Management System

- Objective: Create a simple student management system using JavaScript arrays and objects.
- Activities:
- Store student data (name, age, grade) in an array of objects.
- Display the student data in an HTML table.
- Add functionality to add, edit, and delete student records.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Management System</title>
    <link rel="stylesheet" href="pr13.css">
</head>
<body>
    <h1>Student Management System</h1>
    <form id="student-form">
        <input type="text" id="name" placeholder="Name" required>
        <input type="number" id="age" placeholder="Age" required>
        <input type="text" id="grade" placeholder="Grade" required>
        <button type="submit">Add Student</button>
    </form>
    <table id="student-table">
        <thead>
            <tr>
                <th>Name</th>
                <th>Age</th>
                <th>Grade</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody id="student-body"></tbody>
    </table>
    <script src="pr13.js"></script>
</body>
</html>

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    padding: 20px;
}
```

```

form {
    margin-bottom: 20px;
}

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
}

th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
}

th {
    background-color: #f2f2f2;
}

button {
    margin: 5px;
}

let students = [];
let editingIndex = null;

document.getElementById('student-form').addEventListener('submit', function(event) {
    event.preventDefault();

    const name = document.getElementById('name').value;
    const age = document.getElementById('age').value;
    const grade = document.getElementById('grade').value;

    if (editingIndex !== null) {

        students[editingIndex] = { name, age, grade };
        editingIndex = null;
    } else {

        students.push({ name, age, grade });
    }


    document.getElementById('student-form').reset();


    displayStudents();
});

function displayStudents() {
    const studentBody = document.getElementById('student-body');
    studentBody.innerHTML = '';
    students.forEach((student, index) => {
        const row = document.createElement('tr');

        row.innerHTML = `
            <td>${student.name}</td>
            <td>${student.age}</td>
            <td>${student.grade}</td>
            <td>
                <button onclick="editStudent(${index})">Edit</button>
                <button onclick="deleteStudent(${index})">Delete</button>
            </td>
        `;

        studentBody.appendChild(row);
    });
}

function editStudent(index) {
    const student = students[index];
    document.getElementById('name').value = student.name;
    document.getElementById('age').value = student.age;
    document.getElementById('grade').value = student.grade;

    editingIndex = index;
}

function deleteStudent(index) {
    students.splice(index, 1);
    displayStudents();
}

```



## Student Management System

Dadhania Dev | 20 | CC | Add Student

| Name | Age | Grade | Actions |
|---|---|---|---|
| Dadhania Dev | 20 | CC | Edit Delete |
| dev patel | 119 | FF | Edit Delete |

Learning Outcomes:-

- Learners will understand how to organize and manage data using JavaScript arrays and objects. They will grasp how to represent real-world entities (like students) as objects, allowing for easy manipulation and retrieval of data.

- Participants will gain practical experience in manipulating the Document Object Model (DOM) using JavaScript. They will learn how to dynamically create, update, and remove HTML elements, enhancing their skills in building interactive web applications.

- Learners will become familiar with handling user events, such as form submissions and button clicks. They will understand how to prevent default form behavior and manage form data, including adding validation and interaction logic.

- Participants will develop an understanding of basic Create, Read, Update, and Delete (CRUD) operations in a web application context. They will learn how to implement these functionalities using JavaScript, which are foundational concepts in web development and data management.