<!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>

  </head>

  <style>

    /\* General Styles \*/

body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

    line-height: 1.6;

    color: #333;

}

h1, h2 {

    margin: 0;

    padding: 0;

    color: #333;

}

section {

    padding: 20px;

    margin: 20px auto;

    background-color: #fff;

    border-radius: 8px;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Responsive Layout \*/

#app-title, #searchAndAddButton, #stdDataTable, #pagination {

    max-width: 1200px;

    width: 90%;

    margin: auto;

}

#app-title h1 {

    text-align: center;

    font-size: 2.5em;

    color: #4CAF50;

}

/\* Search and Add Button Section \*/

#searchAndAddButton {

    display: flex;

    flex-wrap: wrap;

    justify-content: space-between;

    align-items: center;

    padding: 10px 0;

}

#sortDropdown,

#searchInput {

    padding: 10px;

    font-size: 1em;

    border: 1px solid #ddd;

    border-radius: 4px;

    width: 200px;

    margin-bottom: 10px;

}

.searchBtn {

    padding: 10px 15px;

    font-size: 1em;

    background-color: #4CAF50;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    margin-left: 10px;

}

.ADbtn {

    padding: 10px 20px;

    font-size: 1em;

    background-color: #007BFF;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    margin-top: 10px;

}

.ADbtn:hover, .searchBtn:hover {

    background-color: #0056b3;

}

/\* Modal Styles \*/

#myModal {

    display: none;

    position: fixed;

    z-index: 1;

    left: 0;

    top: 0;

    width: 100%;

    height: 100%;

    overflow: auto;

    background-color: rgba(0, 0, 0, 0.4);

    padding-top: 60px;

}

.modal-content {

    background-color: #fff;

    margin: 5% auto;

    padding: 20px;

    border: 1px solid #888;

    width: 80%;

    max-width: 600px;

    border-radius: 8px;

}

.modal-content h2 {

    font-size: 1.5em;

    margin-bottom: 20px;

    color: #333;

}

.modal-content label {

    font-weight: bold;

}

.modal-content input[type="text"], .modal-content input[type="submit"] {

    width: 100%;

    padding: 10px;

    margin: 5px 0 10px 0;

    display: inline-block;

    border: 1px solid #ccc;

    border-radius: 4px;

    box-sizing: border-box;

}

.modal-content input[type="submit"] {

    background-color: #4CAF50;

    color: white;

    border: none;

    cursor: pointer;

}

.modal-content input[type="submit"]:hover {

    background-color: #45a049;

}

/\* Student Data Table \*/

.table {

    width: 100%;

    border-collapse: collapse;

    margin: 20px 0;

    overflow-x: auto;

}

.table th, .table td {

    padding: 12px;

    border: 1px solid #ddd;

    text-align: left;

}

.table th {

    background-color: #f2f2f2;

    font-weight: bold;

}

.table tr:nth-child(even) {

    background-color: #f9f9f9;

}

.table tr:hover {

    background-color: #f1f1f1;

}

/\* Pagination \*/

#pagination {

    display: flex;

    justify-content: center;

    margin-top: 20px;

}

.pg {

    padding: 10px 15px;

    margin: 0 5px;

    background-color: #4CAF50;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

}

.pg:hover {

    background-color: #45a049;

}

.editBtn, .deleteBtn {

    padding: 8px 12px;

    font-size: 1em;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    transition: background-color 0.3s ease, transform 0.2s ease;

}

/\* Edit Button \*/

.editBtn {

    background-color: #ffc107;

    margin-right: 5px;

}

.editBtn:hover {

    background-color: #e0a800;

    transform: scale(1.05);

}

/\* Delete Button \*/

.deleteBtn {

    background-color: #dc3545;

}

.deleteBtn:hover {

    background-color: #c82333;

    transform: scale(1.05);

}

/\* Responsive Adjustments \*/

@media (max-width: 768px) {

    #app-title h1 {

        font-size: 2em;

    }

    #searchAndAddButton {

        flex-direction: column;

        align-items: flex-start;

    }

    #sortDropdown, #searchInput {

        width: 100%;

        margin-bottom: 10px;

    }

    .ADbtn {

        width: 100%;

    }

}

@media (max-width: 480px) {

    #app-title h1 {

        font-size: 1.8em;

    }

    .modal-content {

        width: 90%;

    }

    .table th, .table td {

        padding: 8px;

        font-size: 0.9em;

    }

}

  </style>

  <body>

    <!-- App Title here -->

    <section id="app-title">

      <h1>

        <strong><em>Student Database Management System</em></strong>

      </h1>

    </section>

    <!-- App Search, Sort and Add Button here -->

    <section id="searchAndAddButton">

      <select id="sortDropdown" onchange="sortStudents(this.value)">

        <option value="">Sort by</option>

        <option value="name">Sort by Name</option>

        <option value="studentID">Sort by ID</option>

        <option value="class">Sort by Class</option>

      </select>

      <div>

          <input

            type="text"

            id="searchInput"

            oninput="searchStudents()"

            placeholder="Search by name, ID, or class..."

          />

          <button class="searchBtn" onclick="">Search</button>

      </div>

      <button class="button ADbtn" onclick="openModal()">Add Student</button>

    </section>

    <!-- Add Srtudent Modal here -->

    <section id="addStdModal">

      <div id="myModal" class="modal">

        <div class="modal-content">

          <span onclick="closeModal()" style="float: right; cursor: pointer"

            >&times;</span

          >

          <h2>Add Student</h2>

          <form id="studentForm">

            <label for="studentID">Student ID:</label><br />

            <input type="text" id="studentID" name="studentID" required /><br />

            <label for="name">Name:</label><br />

            <input type="text" id="name" name="name" required /><br />

            <label for="fatherName">Father's Name:</label><br />

            <input

              type="text"

              id="fatherName"

              name="fatherName"

              required

            /><br />

            <label for="class">Class:</label><br />

            <input type="text" id="class" name="class" required /><br />

            <label for="totalMarks">Total Marks:</label><br />

            <input

              type="text"

              id="totalMarks"

              name="totalMarks"

              required

            /><br />

            <label for="obtainMarks">Obtain Marks:</label><br />

            <input

              type="text"

              id="obtainMarks"

              name="obtainMarks"

              required

            /><br /><br />

            <input type="submit" value="Submit" />

          </form>

        </div>

      </div>

    </section>

    <!-- Std Data Table -->

    <section id="stdDataTable">

      <table id="studentTable" class="table">

        <thead>

          <tr>

            <th>Student ID</th>

            <th>Name</th>

            <th>Father's Name</th>

            <th>Class</th>

            <th>Total Marks</th>

            <th>Obtain Marks</th>

            <th>Actions</th>

          </tr>

        </thead>

        <tbody id="studentData"></tbody>

      </table>

    </section>

    <section id="pagination">

        <button class="searchBtn pg" onclick="">Prev</button>

        <button class="searchBtn pg" onclick="">Next</button>

    </section>

    <script>

        let students = [];

const openModal = (studentIndex = null) => {

    console.log(studentIndex, "openModal")

    const modal = document.getElementById("myModal");

    const modalTitle = document.querySelector("#myModal h2");

    const submitButton = document.querySelector("#studentForm input[type='submit']");

    if (studentIndex !== null) {

        const student = students[studentIndex];

        document.getElementById("studentID").value = student.studentID;

        document.getElementById("name").value = student.name;

        document.getElementById("fatherName").value = student.fatherName;

        document.getElementById("class").value = student.class;

        document.getElementById("totalMarks").value = student.totalMarks;

        document.getElementById("obtainMarks").value = student.obtainMarks;

        modalTitle.textContent = "Update data";

        submitButton.value = "Update";

        submitButton.onclick = () => updateStudent(studentIndex);

    } else {

        document.getElementById("studentForm").reset();

        modalTitle.textContent = "Add Student";

        submitButton.value = "Submit";

        submitButton.onclick = addStudent;

        console.log()

    }

    modal.style.display = "block";

};

const closeModal = () => {

    document.getElementById("myModal").style.display = "none";

    closeAllMenus(); // Close all menus when modal is closed

};

// add student portion/

const addStudent = (event) => {

    event.preventDefault();

    const form = document.getElementById("studentForm");

    const formData = new FormData(form);

    const student = {};

    formData.forEach((value, key) => {

        student[key] = value;

    });

    students.push(student);

    localStorage.setItem("students", JSON.stringify(students));

    closeModal();

    displayStudents();

};

const deleteStudent = (index) => {

    students.splice(index, 1);

    localStorage.setItem("students", JSON.stringify(students));

    displayStudents();

};

const displayStudents = () => {

    const tableBody = document.getElementById("studentData");

    tableBody.innerHTML = "";

    students.forEach((student, index) => {

        const row = document.createElement("tr");

        for (const key in student) {

            const cell = document.createElement("td");

            cell.textContent = student[key];

            row.appendChild(cell);

        }

        const actionsCell = document.createElement("td");

        const menuButton = document.createElement("button");

        menuButton.textContent = "⋮";

        menuButton.onclick = (event) => openMenu(index, event);

        actionsCell.appendChild(menuButton);

        const optionsMenu = document.createElement("div");

        optionsMenu.classList.add("options-dropdown");

        optionsMenu.id = `menu-${index}`;

        const editOption = document.createElement("button");

        editOption.textContent = "Edit";

        editOption.onclick = () => editStudent(index);

        optionsMenu.appendChild(editOption);

        const deleteOption = document.createElement("button");

        deleteOption.textContent = "Delete";

        deleteOption.onclick = () => deleteStudent(index);

        optionsMenu.appendChild(deleteOption);

        actionsCell.appendChild(optionsMenu);

        row.appendChild(actionsCell);

        tableBody.appendChild(row);

    });

};

// edit portion///

const editStudent = (index) => {

    console.log(index, "Function")

    openModal(index);

};

const updateStudent = (index) => {

    const form = document.getElementById("studentForm");

    const formData = new FormData(form);

    console.log(form, "Form in Update")

    console.log(formData, "Formdata in Update")

    const updatedStudent ={};

    console.log(updateStudent, "UpdatedStd")

    formData.forEach((value, key) => {

        updatedStudent[key] = value;

    });

    students[index] = updatedStudent;

    console.log(updateStudent, "new Line ")

    localStorage.setItem("students", JSON.stringify(students));

    closeModal();

    displayStudents();

};

/////elipsis menu///

const openMenu = (index, event) => {

    event.stopPropagation();

    closeAllMenus(); // Close all other menus before opening this one

    const menu = document.getElementById(`menu-${index}`);

    menu.classList.toggle("show");

    document.addEventListener("click", closeAllMenusOnClick); // Add event listener to close menus on click outside

};

const closeAllMenus = () => {

    const menus = document.querySelectorAll(".options-dropdown");

    menus.forEach(menu => {

        if (menu.classList.contains("show")) {

            menu.classList.remove("show");

        }

    });

    document.removeEventListener("click", closeAllMenusOnClick); // Remove event listener when menus are closed

};

const closeAllMenusOnClick = () => {

    closeAllMenus();

};

// sortby section///////

const sortStudents = (criteria) => {

    if (criteria === "name") {

        console.log(students, "Sort checking")

        students.sort((a, b) => a.name.localeCompare(b.name));

    } else if (criteria === "studentID") {

        students.sort((a, b) => a.studentID - b.studentID);

    } else if (criteria === "class") {

        students.sort((a, b) => a.class.localeCompare(b.class));

    }

    displayStudents();

};

//serach bar////

const searchStudents = () => {

    const searchTerm = document.getElementById("searchInput").value.toLowerCase();

    const filteredStudents = students.filter(student => {

        return Object.values(student).some(value => value.toString().toLowerCase().includes(searchTerm));

    });

    displayFilteredStudents(filteredStudents);

};

const displayFilteredStudents = (filteredStudents) => {

    const tableBody = document.getElementById("studentData");

    tableBody.innerHTML = "";

    filteredStudents.forEach((student, index) => {

        const row = document.createElement("tr");

        for (const key in student) {

            const cell = document.createElement("td");

            cell.textContent = student[key];

            row.appendChild(cell);

        }

        const actionsCell = document.createElement("td");

        const menuButton = document.createElement("button");

        menuButton.textContent = "⋮";

        menuButton.onclick = (event) => openMenu(index, event);

        actionsCell.appendChild(menuButton);

        const optionsMenu = document.createElement("div");

        optionsMenu.classList.add("options-dropdown");

        optionsMenu.id = `menu-${index}`;

        const editOption = document.createElement("button");

        editOption.textContent = "Edit";

        editOption.onclick = () => editStudent(index);

        optionsMenu.appendChild(editOption);

        const deleteOption = document.createElement("button");

        deleteOption.textContent = "Delete";

        deleteOption.onclick = () => deleteStudent(index);

        optionsMenu.appendChild(deleteOption);

        actionsCell.appendChild(optionsMenu);

        row.appendChild(actionsCell);

        tableBody.appendChild(row);

    });

};

document.getElementById("studentForm").addEventListener("submit", addStudent);

document.addEventListener("DOMContentLoaded", () => {

    const storedStudents = localStorage.getItem("students");

    if (storedStudents) {

        students = JSON.parse(storedStudents);

        displayStudents();

    }

});

    </script>

  </body>

</html>