

BÀI TẬP THỰC HÀNH SỐ 04

Học phần: Nền tảng Web – Dành cho: K64

BÀI TẬP 01: Bài tập thực hành: Ứng dụng quản lý sinh viên (Javascript)

Mục tiêu:

- Áp dụng kiến thức về JavaScript, DOM manipulation, localStorage và JSON.
- Xây dựng một ứng dụng CRUD (Create, Read, Update, Delete) đơn giản.
- Làm quen với việc lưu trữ dữ liệu trên trình duyệt.

Yêu cầu:

1. Giao diện người dùng:

- Tạo một form nhập liệu để thêm sinh viên mới (họ tên, mã số sinh viên, ngày sinh, lớp).
- Hiển thị danh sách sinh viên trong một bảng (table) với các cột tương ứng.
- Cung cấp các nút/chức năng để sửa và xóa sinh viên.

2. Lưu trữ dữ liệu:

- Sử dụng localStorage để lưu trữ danh sách sinh viên dưới dạng JSON.
- Khi ứng dụng khởi động, đọc dữ liệu từ localStorage và hiển thị lên bảng.
- Khi thêm, sửa hoặc xóa sinh viên, cập nhật dữ liệu trong localStorage.

3. Xử lý sự kiện:

- Viết các hàm xử lý sự kiện cho việc thêm, sửa, xóa sinh viên.
- Validate dữ liệu đầu vào trước khi thêm hoặc sửa sinh viên.
- Xác nhận trước khi xóa sinh viên.

Gợi ý:

- **Cấu trúc dữ liệu:** Sử dụng một mảng để lưu trữ các đối tượng sinh viên (ví dụ: [{ hoTen: "...", maSV: "...", ngaySinh: "...", lop: "..."}]).
- **localStorage:**
 - Lưu dữ liệu: `localStorage.setItem("danhSachSinhVien", JSON.stringify(danhSach))`
 - Đọc dữ liệu: `JSON.parse(localStorage.getItem("danhSachSinhVien")) || []`
- **DOM manipulation:** Sử dụng các phương thức như `document.getElementById`, `document.createElement`, `element.innerHTML`,...
- **Event handling:** Sử dụng `addEventListener` để lắng nghe các sự kiện như "click", "submit"

Lưu ý:

- Đây là một bài tập cơ bản, bạn có thể tự do mở rộng và thêm các tính năng nâng cao (tìm kiếm, sắp xếp, phân trang,...).
- Hãy cố gắng tự mình hoàn thành bài tập để củng cố kiến thức. Nếu gặp khó khăn, bạn có thể tìm kiếm sự trợ giúp từ cộng đồng hoặc tham khảo các tài liệu hướng dẫn.

CODE gợi ý (sử dụng CSS thuần)

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Ứng dụng quản lý sinh viên</title>
  <style>
    /* CSS cơ bản cho giao diện */
    table { width: 100%; border-collapse: collapse; }
    th, td { border: 1px solid #ddd; padding: 8px; }
    input[type=text], input[type=date] { width: 100%; padding: 8px;
}
    button { padding: 8px 16px; background-color: #4CAF50; color:
white; border: none; }
  </style>
</head>
<body>

<h2>Thêm sinh viên</h2>
<form id="formSinhVien">
  Họ tên: <input type="text" id="hoTen"><br><br>
  Mã SV: <input type="text" id="maSV"><br><br>
  Ngày sinh: <input type="date" id="ngaySinh"><br><br>
  Lớp: <input type="text" id="lop"><br><br>
  <button type="submit">Thêm</button>
</form>

<h2>Danh sách sinh viên</h2>
<table id="bangSinhVien">
  <tr>
    <th>Họ tên</th>
    <th>Mã SV</th>
    <th>Ngày sinh</th>
    <th>Lớp</th>
    <th>Hành động</th>
  </tr>
</table>

<script>
  // JavaScript (bạn sẽ viết các hàm xử lý sự kiện tại đây)
  document.getElementById("formSinhVien").addEventListener("submit",
function(event) {
    event.preventDefault(); // Ngăn chặn form submit mặc định

    // Lấy dữ liệu từ form
    // ...

    // Validate dữ liệu
    // ...

    // Thêm sinh viên vào bảng và localStorage
    // ...

```

```

});

// Hàm hiển thị danh sách sinh viên từ localStorage khi trang tải
function hienThiDanhSachSinhVien() {
    // ...
}

// Gọi hàm hiển thị danh sách khi trang tải
hienThiDanhSachSinhVien();
</script>

</body>
</html>

```

CODE gợi ý (sử dụng TailwindCSS)

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Ứng dụng quản lý sinh viên</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="container mx-auto p-4">

<h2 class="text-2xl font-bold mb-4">Thêm sinh viên</h2>
<form id="formSinhVien" class="mb-8">
    <div class="mb-4">
        <label for="hoTen" class="block text-gray-700 text-sm font-bold
mb-2">Họ tên:</label>
        <input type="text" id="hoTen" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight
focus:outline-none focus:shadow-outline">
    </div>
    <div class="mb-4">
        <label for="maSV" class="block text-gray-700 text-sm font-bold
mb-2">Mã SV:</label>
        <input type="text" id="maSV" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight
focus:outline-none focus:shadow-outline">
    </div>
    <div class="mb-4">
        <label for="ngaySinh" class="block text-gray-700 text-sm font-
bold mb-2">Ngày sinh:</label>
        <input type="date" id="ngaySinh" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight
focus:outline-none focus:shadow-outline">
    </div>
    <div class="mb-4">

```

```

        <label for="lop" class="block text-gray-700 text-sm font-bold
mb-2">Lớp:</label>
        <input type="text" id="lop" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight
focus:outline-none focus:shadow-outline">
    </div>
    <div class="flex items-center justify-between">
        <button type="submit" class="bg-blue-500 hover:bg-blue-700
text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-
outline">
            Thêm
        </button>
    </div>
</form>

<h2 class="text-2xl font-bold mb-4">Danh sách sinh viên</h2>
<table id="bangSinhVien" class="table-auto w-full">
    <thead>
    <tr>
        <th class="px-4 py-2">Họ tên</th>
        <th class="px-4 py-2">Mã SV</th>
        <th class="px-4 py-2">Ngày sinh</th>
        <th class="px-4 py-2">Lớp</th>
        <th class="px-4 py-2">Hành động</th>
    </tr>
    </thead>
    <tbody>
    </tbody>
</table>

<script>
    // JavaScript (bạn sẽ viết các hàm xử lý sự kiện tại đây)
    document.getElementById("formSinhVien").addEventListener("submit",
function(event) {
        event.preventDefault(); // Ngăn chặn form submit mặc định

        // Lấy dữ liệu từ form
        // ...

        // Validate dữ liệu
        // ...

        // Thêm sinh viên vào bảng và localStorage
        // ...
    });

    // Hàm hiển thị danh sách sinh viên từ localStorage khi trang tải
    function hienThiDanhSachSinhVien() {
        // ...
    }

    // Gọi hàm hiển thị danh sách khi trang tải
    hienThiDanhSachSinhVien();
</script>

```

```
</body>
</html>
```

BÀI TẬP 02: Bài tập thực hành: Ứng dụng quản lý sinh viên (jQuery)

Mục tiêu:

- Áp dụng kiến thức về jQuery, DOM manipulation, AJAX và JSON.
- Xây dựng một ứng dụng CRUD (Create, Read, Update, Delete) cho sinh viên.
- Làm quen với việc đọc và ghi dữ liệu từ JSON file.
- Sử dụng Tailwind CSS để thiết kế giao diện.

Yêu cầu:

- Cấu trúc dữ liệu:**
 - Sử dụng một JSON file (ví dụ: data.json) để lưu trữ danh sách các lớp và sinh viên.
 - Mỗi lớp có các thuộc tính: tên lớp, danh sách sinh viên.
 - Mỗi sinh viên có các thuộc tính: họ tên, mã số sinh viên, ngày sinh.
- Giao diện người dùng (Tailwind CSS):**
 - Tạo một dropdown list để chọn lớp.
 - Hiển thị danh sách sinh viên của lớp đã chọn trong một bảng (table).
 - Cung cấp các nút/chức năng để thêm, sửa và xóa sinh viên.
 - Tạo form nhập liệu để thêm/sửa thông tin sinh viên.
- Xử lý sự kiện (jQuery):**
 - Viết các hàm xử lý sự kiện cho việc chọn lớp, thêm, sửa, xóa sinh viên.
 - Sử dụng AJAX để đọc dữ liệu từ JSON file khi trang web được tải và khi có sự thay đổi (thêm, sửa, xóa).
 - Validate dữ liệu đầu vào trước khi thêm hoặc sửa sinh viên.
 - Xác nhận trước khi xóa sinh viên.
- Lưu trữ dữ liệu:**
 - Khi thêm, sửa hoặc xóa sinh viên, cập nhật dữ liệu trong JSON file thông qua AJAX.

Gợi ý:

- **Đọc dữ liệu từ JSON file:** Sử dụng `$.getJSON("data.json", function(data) { ... })`.
- **Ghi dữ liệu vào JSON file:** Sử dụng `$.ajax({ url: "save.php", method: "POST", data: { data: JSON.stringify(newData) }, ... })` (bạn cần tạo một file PHP save.php để xử lý việc ghi dữ liệu).
- **DOM manipulation (jQuery):** Sử dụng các phương thức như `$("#elementId").append(), .html(), .on("click", function() { ... })`,...
- **Tailwind CSS:** Sử dụng các lớp tiện ích của Tailwind CSS để tạo giao diện.

Ví dụ (cấu trúc JSON file data.json):

```
[
  {
    "tenLop": "K64KTPM1",
    "sinhVien": [
      { "hoTen": "Nguyễn Văn A", "maSV": "123456", "ngaySinh": "2002-
```

```

05-15" },
    { "hoTen": "Trần Thị B", "maSV": "654321", "ngaySinh": "2003-08-
22" }
  ],
},
{
  "tenLop": "K64KTPM2",
  "sinhVien": [
    { "hoTen": "Lê Văn C", "maSV": "987654", "ngaySinh": "2002-12-10"
},
    { "hoTen": "Phạm Thị D", "maSV": "456789", "ngaySinh": "2003-03-
05" }
  ]
}
]

```

CODE giao diện gợi ý:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Quản lý sinh viên</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://code.jquery.com/jquery-
3.6.0.min.js"></script>
</head>
<body class="container mx-auto p-4">

<h2 class="text-2xl font-bold mb-4">Quản lý sinh viên</h2>

<div class="mb-4">
  <label for="lop" class="block text-gray-700 text-sm font-bold
mb-2">Chọn lớp:</label>
  <select id="lop" class="shadow appearance-none border rounded
w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline">
  </select>
</div>

<h3 class="text-xl font-bold mb-2">Danh sách sinh viên</h3>
<table id="bangSinhVien" class="table-auto w-full">
  <thead>
    <tr>
      <th class="px-4 py-2">Họ tên</th>
      <th class="px-4 py-2">Mã SV</th>
      <th class="px-4 py-2">Ngày sinh</th>
      <th class="px-4 py-2">Hành động</th>
    </tr>
  </thead>
  <tbody>
  </tbody>
</table>

```

```

</table>

<div id="formSinhVienModal" class="hidden fixed z-10 inset-0
overflow-y-auto" aria-labelledby="modal-title" role="dialog"
aria-modal="true">
</div>

<script>
    $(document).ready(function() {
        // Đọc dữ liệu từ data.json và hiển thị danh sách lớp vào
        dropdown
        $.getJSON("data.json", function(data) {
            // ...
        });

        // Xử lý sự kiện khi chọn lớp từ dropdown
        $("#lop").change(function() {
            // ...
        });

        // Các hàm xử lý sự kiện thêm, sửa, xóa sinh viên
        // ...
    });
</script>

</body>
</html>

```

Lưu ý:

- Đây là một bài tập nâng cao hơn, đòi hỏi kiến thức về jQuery, AJAX và xử lý JSON file.
- Bạn có thể tự do sáng tạo và thêm các tính năng nâng cao khác (tìm kiếm, sắp xếp, phân trang,...).