

Báo cáo cài đặt nhóm chức năng 1: Quản lý Bảng cấp

Môn học: *Đánh giá và kiểm định chất lượng phần mềm*-1-3-24(N01.TH7)*

Đơn vị thực hiện:

- **Tên nhóm:** Nhóm 02
 - **Thành viên thực hiện:**
 - Trịnh Hoài Nam
 - Phạm Hoàng Anh (L)
 - Nguyễn Đức Anh
 - Hoàng Thị Thảo Nhi
-

Nội dung báo cáo:

- **Báo cáo cài đặt:** do bạn Nguyễn Đức Anh và Trịnh Hoài Nam thực hiện.
-

1. Môi trường cài đặt

- **Phía Client (Người dùng):**
 - Trình duyệt web hiện đại hỗ trợ HTML5, CSS3 và JavaScript ES6 (ví dụ: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).
- **Phía Server (Giả định cho một ứng dụng hoàn chỉnh):**
 - Do tệp cung cấp là HTML/CSS/JS thuần túy, dữ liệu được quản lý tạm thời trong JavaScript (sampleData).
 - Trong một ứng dụng thực tế, cần có:
 - **Web Server:** Ví dụ: Nginx, Apache.
 - **Runtime Môi trường Backend:** Ví dụ: Node.js (với Express.js), Java (với Spring Boot), Python (với Django/Flask), PHP (với Laravel).
 - **Hệ quản trị Cơ sở dữ liệu:** Ví dụ: PostgreSQL, MySQL, MongoDB.
- **Công cụ phát triển (dựa trên file):**
 - Trình soạn thảo mã: Bất kỳ (VSCode, Sublime Text, ...).
 - File HTML có sử dụng các class của Tailwind CSS (ví dụ: flex, bg-cyan-800, rounded-lg), cho thấy Tailwind CSS có thể là một phần của môi trường phát triển giao diện.

2. Kiến trúc chung

- **Kiến trúc hiện tại (dựa trên file HTML):**
 - **Single Page Application (SPA) Mock-up:** Giao diện người dùng được quản lý động bằng JavaScript trên một trang HTML duy nhất.
 - **Client-Side Rendering (CSR):** Nội dung (danh sách bảng cấp, form) được tạo và cập nhật bởi JavaScript phía trình duyệt.
 - **Data In-Memory:** Dữ liệu (sampleData trong JavaScript) được lưu trữ và thao tác trực tiếp trong bộ nhớ của trình duyệt khi trang được tải. Dữ liệu sẽ mất khi tải lại trang.
 - **Component-based UI (mô phỏng):** Các hàm JavaScript (contentTemplates, modalTemplates) tạo ra các khối HTML tái sử dụng được cho các phần khác nhau

của giao diện (danh sách, modal).

- **Kiến trúc đề xuất cho ứng dụng hoàn chỉnh:**
 - **Kiến trúc 3 lớp (Three-Tier Architecture) hoặc Microservices:**
 1. **Tầng Trình diễn (Presentation Tier):** HTML, CSS (Tailwind CSS), JavaScript (như trong file) xử lý giao diện người dùng và tương tác cơ bản.
 2. **Tầng Ứng dụng (Application Tier/Logic Tier):** Backend API (ví dụ: RESTful API) xử lý logic nghiệp vụ, xác thực, và giao tiếp với tầng dữ liệu.
 3. **Tầng Dữ liệu (Data Tier):** Cơ sở dữ liệu lưu trữ thông tin bằng cấp, giáo viên, khoa, v.v.

3. Cấu trúc mã nguồn (cho chức năng Quản lý Bảng cấp)

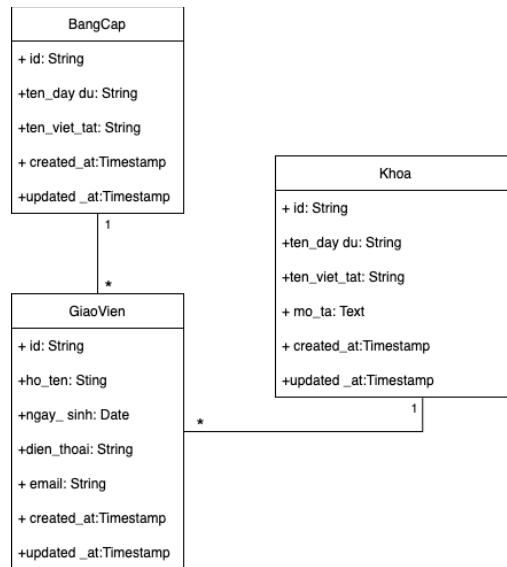
- **Code:**
 - **Phần HTML (cấu trúc giao diện):**
 - Thanh điều hướng bên trái (<aside>) có mục "Bảng cấp" (Bảng cấp).
 - Khu vực nội dung chính (<main>) sẽ được cập nhật động bởi JavaScript.
 - Template cho danh sách bảng cấp (trong contentTemplates['Bảng cấp']):
 - Tiêu đề "Danh sách Bảng cấp".
 - Nút "Thêm mới" (<button onclick="showModal('Bảng cấp', 'add')">).
 - Ô tìm kiếm (<input type="text">) và nút tìm kiếm.
 - Bảng (<table>) để hiển thị các cột: Mã, Tên đầy đủ, Tên viết tắt, Hành động.
 - Template cho modal thêm/sửa/xem bảng cấp (trong modalTemplates['Bảng cấp']):
 - Tiêu đề modal động ("Thêm mới Bảng cấp", "Chỉnh sửa Bảng cấp", "Chi tiết Bảng cấp").
 - Form (<form>) với các trường input:
 - Mã (chỉ hiển thị khi xem chi tiết, disabled).
 - Tên đầy đủ (<input type="text" name="fullName">).
 - Tên viết tắt (<input type="text" name="shortName">).
 - Các nút "Hủy", "Lưu" (ẩn khi xem chi tiết).
 - **Phần CSS (định dạng):**
 - Sử dụng các lớp tiện ích của Tailwind CSS.
 - CSS tùy chỉnh trong thẻ <style> hoặc file CSS ngoài (trong file HTML có <link rel="stylesheet" href="https://stackedit.io/style.css" /> và CSS nội bộ).
 - Định dạng cho bảng, modal, nút, hiệu ứng hover, toast message, scrollbar.
 - **Phần JavaScript (logic xử lý):**
 - **sampleData.degrees:** Mảng lưu trữ đối tượng bảng cấp. Mỗi đối tượng có id, fullName, shortName.
 - **generateId(prefix, existingIds):** Hàm tạo ID tự động (ví dụ: "DEG004").
 - **contentTemplates['Bảng cấp']():** Hàm JavaScript trả về chuỗi HTML

cho việc hiển thị danh sách bằng cấp. Dữ liệu được lấy từ `sampleData.degrees` và lặp qua để tạo các hàng của bảng. Mỗi hàng có các nút "Xem chi tiết" (view-btn), "Sửa" (edit-btn), "Xóa" (delete-btn) với data-id tương ứng.

- **modalTemplates['Bằng cấp'](title, action, item):** Hàm JavaScript trả về chuỗi HTML cho form trong modal. Dựa vào action ('add', 'edit', 'view') và item (dữ liệu bằng cấp) để điền thông tin và cấu hình trạng thái (enabled/disabled) của các trường.
- **showModal(section, action, item):** Hiển thị modal, truyền section = 'Bằng cấp', action và item (nếu sửa/xem).
- **hideModal():** Ẩn modal và reset form.
- **setupModalEvents():** Gắn sự kiện submit cho form trong modal. Khi submit:
 - Lấy dữ liệu từ form.
 - Nếu `currentAction === 'add'`: Tạo ID mới, thêm vào `sampleData.degrees`.
 - Nếu `currentAction === 'edit'`: Cập nhật `sampleData.degrees` dựa trên `currentItem.id`.
 - Hiển thị toast message, đóng modal, làm mới danh sách.
- **setContentEvents(section):** Gắn sự kiện click cho các nút "Thêm mới", "Xem chi tiết", "Sửa", "Xóa" trong danh sách bằng cấp.
 - Nút "Thêm mới": Gọi `showModal('Bằng cấp', 'add')`.
 - Nút "Xem chi tiết": Lấy id từ data-id, tìm item trong `sampleData.degrees`, gọi `showModal('Bằng cấp', 'view', item)`.
 - Nút "Sửa": Tương tự nút xem, nhưng gọi `showModal('Bằng cấp', 'edit', item)`.
 - Nút "Xóa": Lấy id, hiển thị `confirm()`, nếu OK thì gọi `deleteItem('Bằng cấp', id)`.
- **deleteItem(section, id):** Tìm và xóa bằng cấp khỏi `sampleData.degrees` bằng `splice()`, hiển thị toast, làm mới danh sách.
- **performSearch(input):** Lấy giá trị từ ô tìm kiếm, lặp qua các hàng (<tr>) của bảng, ẩn/hiện hàng dựa trên `textContent` có chứa từ khóa tìm kiếm hay không.
- **switchContent(section):** Cập nhật `mainContent.innerHTML` bằng kết quả từ `contentTemplates[section]()` và gọi `setContentEvents(section)`.
- **showToast(message, type):** Hiển thị thông báo nhanh (toast).

4. Sơ đồ lớp của 1 chức năng

Sơ đồ lớp này mô tả ba thực thể chính: `BangCap` (Bằng Cấp), `GiaoVien` (Giáo Viên), và `Khoa` (Khoa/Đơn vị).



1. Lớp BangCap (Bằng Cấp)

Lớp này đại diện cho thông tin về các loại bằng cấp hoặc chứng chỉ.

- **Thuộc tính:**
 - id: String - Mã định danh duy nhất cho mỗi bằng cấp (Khóa chính).
 - ten_day_du: String - Tên đầy đủ của bằng cấp.
 - ten_viet_tat: String - Tên viết tắt của bằng cấp.
 - created_at: Timestamp - Thời gian tạo bản ghi.
 - updated_at: Timestamp - Thời gian cập nhật bản ghi gần nhất.
- **Mối quan hệ:**
 - **Với GiaoVien:** Một BangCap có thể được sở hữu bởi nhiều GiaoVien (Quan hệ một - nhiều). Điều này ngụ ý rằng một loại bằng cấp (ví dụ: Thạc sĩ, Tiến sĩ) có thể được nhiều giáo viên khác nhau đạt được.

2. Lớp GiaoVien (Giáo Viên)

Lớp này đại diện cho thông tin chi tiết về giáo viên.

- **Thuộc tính:**
 - id: String - Mã định danh duy nhất cho mỗi giáo viên (Khóa chính).
 - ho_ten: String - Họ và tên đầy đủ của giáo viên.
 - ngay_sinh: Date - Ngày sinh của giáo viên.
 - dien_thoai: String - Số điện thoại của giáo viên.
 - email: String - Địa chỉ email của giáo viên.
 - created_at: Timestamp - Thời gian tạo bản ghi.
 - updated_at: Timestamp - Thời gian cập nhật bản ghi gần nhất.
- **Mối quan hệ:**
 - **Với BangCap:** Nhiều GiaoVien có thể thuộc về một BangCap (Quan hệ nhiều - một, nhìn từ phía GiaoVien). Điều này có nghĩa là mỗi giáo viên sẽ có một bằng cấp chính được ghi nhận (dựa theo sơ đồ, nếu một giáo viên có nhiều bằng cấp thì cần một bảng trung gian hoặc thay đổi thiết kế).
 - **Với Khoa:** Nhiều GiaoVien thuộc về một Khoa (Quan hệ nhiều - một). Điều này có

nghĩa là mỗi giáo viên trực thuộc một khoa cụ thể.

3. Lớp Khoa (Khoa/Đơn vị)

Lớp này đại diện cho thông tin về các khoa hoặc đơn vị trong một tổ chức giáo dục.

- **Thuộc tính:**
 - id: String - Mã định danh duy nhất cho mỗi khoa (Khóa chính).
 - ten_day_du: String - Tên đầy đủ của khoa.
 - ten_viet_tat: String - Tên viết tắt của khoa.
 - mo_ta: Text - Mô tả chi tiết về khoa.
 - created_at: Timestamp - Thời gian tạo bản ghi.
 - updated_at: Timestamp - Thời gian cập nhật bản ghi gần nhất.
- **Mối quan hệ:**
 - **Với GiaoVien:** Một Khoa có thể có nhiều GiaoVien (Quan hệ một - nhiều).

Chức năng có thể liên quan:

Dựa trên sơ đồ lớp này, một số chức năng có thể được xây dựng bao gồm:

- **Quản lý thông tin giáo viên:**
 - Thêm, sửa, xóa thông tin giáo viên.
 - Gán giáo viên vào một khoa cụ thể.
 - Gán bằng cấp cho giáo viên.
 - Tìm kiếm và lọc giáo viên theo khoa, bằng cấp, hoặc các tiêu chí khác.
- **Quản lý thông tin bằng cấp:**
 - Thêm, sửa, xóa các loại bằng cấp.
 - Xem danh sách giáo viên theo từng loại bằng cấp.
- **Quản lý thông tin khoa:**
 - Thêm, sửa, xóa thông tin khoa.
 - Xem danh sách giáo viên thuộc một khoa.

Tóm tắt mối quan hệ chính:

- Một **Khoa** có nhiều **Giáo Viên**.
- Mỗi **Giáo Viên** thuộc về một **Khoa**.
- Mỗi **Giáo Viên** có một **Bằng Cấp** (theo sơ đồ hiện tại, nếu cần nhiều bằng cấp cho một giáo viên thì cần điều chỉnh).
- Một **Bằng Cấp** có thể được cấp cho nhiều **Giáo Viên**.
- Sơ đồ này cung cấp một cấu trúc cơ bản để quản lý dữ liệu liên quan đến giáo viên, bằng cấp và khoa trong một hệ thống thông tin.