

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Môn: THỰC TẬP CƠ SỞ

Báo cáo tuần

Website quản lý lớp học dành cho giáo viên và học sinh

Giảng viên: Kim Ngọc Bách
Học và tên: Nguyễn Tuấn Anh
Mã sinh viên: B22DCVT026
Lớp: E22CQCN05-B

Hà Nội - 2025

Kính gửi Thầy,

Em xin gửi báo cáo cá nhân tuần 7 cho dự án trang web quản lý lớp học. Trong tuần này, em đã tập trung tích hợp giao diện frontend (được xây dựng ở Tuần 3-4) với backend (Tuần 5-6), hoàn thiện một phần chức năng lịch và việc cần làm, đồng thời bắt đầu tối ưu trải nghiệm người dùng. Báo cáo này trình bày chi tiết các công việc em đã thực hiện một mình để đảm bảo tiến độ dự án.

Mục tiêu

- Kết nối giao diện frontend với API backend để hiển thị dữ liệu thực tế (danh sách lớp học, bài tập).
- Bắt đầu xây dựng logic hiển thị lịch thời gian thực, tích hợp dữ liệu bài tập.
- Bắt đầu phát triển chức năng việc cần làm, hỗ trợ hiển thị và lọc danh sách công việc.
- Kiểm tra tính đồng bộ giữa các chức năng và bắt đầu tối ưu trải nghiệm người dùng.

Công việc Thực hiện

1. Tích hợp Frontend với Backend

Em đã sử dụng JavaScript (với Fetch API) để kết nối các trang frontend (HTML/CSS từ Tuần 3-4) với các endpoint API (PHP từ Tuần 5-6):

- **Trang Chủ:**
 - **Tích hợp:** Em gọi endpoint `/api/classes/list` để hiển thị danh sách lớp học của người dùng (giáo viên hoặc học sinh) ngay sau khi đăng nhập.
 - **Thực hiện:** Sử dụng Fetch API để lấy dữ liệu JSON, sau đó hiển thị các lớp học dưới dạng thẻ (cards) bằng cách thêm các phần tử `<div>` động vào DOM.
 - **Kết quả:** Trang chủ hiển thị danh sách lớp học với tên, mã lớp và mô tả, lấy từ cơ sở dữ liệu.
- **Trang Đăng ký và Đăng nhập:**
 - **Tích hợp:** Em gọi các endpoint `/api/register` và `/api/login` để xử lý biểu mẫu. Sau khi đăng ký hoặc đăng nhập thành công, em lưu token phiên vào localStorage để sử dụng cho các yêu cầu API sau.
 - **Thực hiện:** Thêm xử lý lỗi (ví dụ: hiển thị thông báo “Email đã tồn tại” khi đăng ký thất bại) bằng cách phân tích phản hồi JSON từ API.

- **Kết quả:** Người dùng có thể đăng ký và đăng nhập, với thông báo thành công hoặc lỗi hiển thị trên giao diện.
- **Bảng Điều khiển:**
 - **Tích hợp:** Gọi `/api/classes/list` và `/api/submissions/list` để hiển thị danh sách lớp học và bài nộp gần đây.
 - **Thực hiện:** Sử dụng JavaScript để tạo các thẻ động cho lớp học và bài nộp, với các liên kết dẫn đến trang chi tiết lớp học.
 - **Kết quả:** Bảng điều khiển hiển thị thông tin thực tế, như danh sách lớp học và trạng thái bài nộp.

2. Chức năng Lịch (Bắt đầu)

- **Logic Hiển thị:** Em bắt đầu xây dựng logic cho trang lịch bằng JavaScript, lấy dữ liệu bài tập từ endpoint `/api/assignments/list` (tạm thời giả lập endpoint này để lấy danh sách bài tập theo `class_id`).
- **Tích hợp Dữ liệu:** Em hiển thị các thời hạn bài tập trên lịch tháng (từ HTML/CSS ở Tuần 4) bằng cách đánh dấu các ngày có bài tập với màu nền đặc biệt.
- **Thực hiện:** Sử dụng JavaScript để phân tích dữ liệu JSON từ API, sau đó thêm các sự kiện vào các ô ngày tương ứng trong bảng `<table>`.
- **Kết quả:** Lịch hiển thị các thời hạn bài tập với tiêu đề và lớp học khi di chuột qua, nhưng chưa hoàn thiện phần hiển thị chi tiết sự kiện.

3. Chức năng Việc cần làm (Bắt đầu)

- **Hiển thị Danh sách:** Em gọi endpoint `/api/submissions/list` (với tham số `student_id`) để lấy danh sách bài tập chưa nộp của học sinh.
- **Lọc Công việc:** Em thêm một bộ lọc đơn giản bằng JavaScript, cho phép hiển thị bài tập theo lớp học (sử dụng `<select>` để chọn `class_id`).
- **Thực hiện:** Sử dụng Fetch API để lấy dữ liệu, sau đó hiển thị danh sách công việc dưới dạng `` với các mục `` chứa tiêu đề, thời hạn và lớp học.
- **Kết quả:** Trang việc cần làm hiển thị danh sách bài tập chưa hoàn thành, với bộ lọc cơ bản theo lớp học.

4. Tối ưu Trải nghiệm Người dùng

- **Tốc độ Phản hồi:** Em tối ưu các yêu cầu Fetch API bằng cách lưu trữ dữ liệu tạm thời trong `localStorage` để giảm số lần gọi API khi chuyển trang.
- **Thông báo:** Em thêm thông báo thời gian thực (sử dụng `alert` tạm thời) khi đăng ký, đăng nhập hoặc nộp bài thành công.

- **Điều chỉnh Giao diện:** Dựa trên tự đánh giá, em điều chỉnh khoảng cách (padding) và kích thước chữ trên bảng điều khiển để cải thiện tính dễ đọc trên di động.

5. Kiểm tra Đồng bộ

- Em kiểm tra tính đồng bộ bằng cách mô phỏng hành vi người dùng:
 - Đăng ký một tài khoản mới, đăng nhập, tạo lớp học, và kiểm tra xem lớp học có hiển thị đúng trên bảng điều khiển.
 - Kiểm tra dữ liệu bài tập được lấy từ API và hiển thị chính xác trên trang chủ.
- Em phát hiện lỗi nhỏ khi danh sách lớp học không cập nhật ngay sau khi tạo lớp mới, đã khắc phục bằng cách làm mới dữ liệu sau mỗi thao tác POST.

Kết quả

- **Tích hợp Frontend-Backend:** Các trang chủ, đăng ký, đăng nhập và bảng điều khiển được kết nối với API, hiển thị dữ liệu thực tế.
- **Chức năng Lịch:** Hoàn thành 60% logic hiển thị thời hạn bài tập trên lịch tháng.
- **Chức năng Việc cần làm:** Hoàn thành hiển thị danh sách và bộ lọc cơ bản.
- **Tối ưu UX:** Cải thiện tốc độ phản hồi và giao diện trên di động.

Thách thức và Giải pháp

- **Thách thức:** Kết nối nhiều endpoint API với frontend trong thời gian ngắn.
 - **Giải pháp:** Em ưu tiên tích hợp các trang chính trước (trang chủ, bảng điều khiển) và sử dụng hàm Fetch API tái sử dụng để tiết kiệm thời gian.
- **Thách thức:** Dữ liệu lịch không hiển thị đúng ngày do định dạng sai.
 - **Giải pháp:** Em chuyển đổi định dạng deadline từ API sang Date trong JavaScript để khớp với lịch.

Kế hoạch Tuần 8

- Hoàn thiện chức năng lịch và việc cần làm (thêm chi tiết sự kiện và đánh dấu trạng thái).
- Xây dựng chức năng cài đặt (đổi mật khẩu, cập nhật ảnh đại diện).

- Tăng cường bảo mật (giới hạn đăng nhập thất bại) và kiểm tra đồng bộ toàn diện.