

# BÀI TẬP THỰC HÀNH SỐ 3

Học phần: Nền tảng phát triển Web – Dành cho: K65

---

## PHẦN 1: LUYỆN TẬP JAVASCRIPT (180 phút)

### 1. Mục tiêu

- Sinh viên ôn lại cú pháp JS, các kiểu dữ liệu, hàm, vòng lặp, câu lệnh điều kiện.
- Thao tác với **DOM**: thêm, xóa, sửa nội dung HTML, thay đổi CSS qua JavaScript.
- Xử lý sự kiện (click, change, submit, ...).
- Áp dụng kiến thức **OOP** trong JavaScript (nếu cần) để tổ chức mã.

### 2. Nội dung & Hoạt động

Dưới đây là gợi ý một **bài tập thực hành** tổng hợp, chia thành các bước nhỏ. Giảng viên có thể điều chỉnh nội dung hoặc chia thành nhiều bài tập riêng lẻ tùy theo mức độ của sinh viên.

**Bài tập: Xây dựng “Quản Lý Công Việc (Todo List)” bằng JavaScript thuần**

#### 1. Chuẩn bị giao diện (HTML & CSS cơ bản) (10-15 phút)

- Giảng viên cung cấp hoặc hướng dẫn tạo một trang HTML đơn giản có các thành phần:
  - **Input** để nhập nội dung công việc mới.
  - **Nút** “Thêm công việc”.
  - **Danh sách** các công việc (ban đầu có thể trống).
- Sinh viên có thể sử dụng **Bootstrap** hoặc **CSS thuần** (hoặc SASS/SCSS nếu muốn) để tạo giao diện cơ bản, nhưng **phần chính** sẽ là thao tác JS.

#### 2. Thêm công việc (30 phút)

- **Hướng dẫn của giảng viên:**
  - Cách **lấy giá trị** từ `<input>` khi người dùng bấm nút “Thêm”.
  - Tạo **DOM element** (ví dụ `<li>` hoặc `<div>`) để hiển thị công việc mới.
  - Thêm phần tử này vào danh sách công việc (thông qua `appendChild()` hoặc các phương thức DOM khác).
  - Quản lý **mảng** (array) hoặc **danh sách** công việc trong JS để dễ dàng thêm/xóa/sửa.
- **Nhiệm vụ của sinh viên:**
  - Viết hàm `addTask()` thực hiện các bước trên.
  - Kiểm tra trống/null: nếu ô nhập rỗng thì cảnh báo, không thêm.
  - Xóa ô nhập sau khi thêm thành công.

#### 3. Đánh dấu hoàn thành / Xóa công việc (45 phút)

- **Hướng dẫn của giảng viên:**
  - Cách **gắn sự kiện** (event listener) cho mỗi công việc mới tạo.
  - Sự kiện **click** để “đánh dấu hoàn thành” (có thể gạch ngang hoặc đổi màu).
  - Sự kiện **click** để **xóa** công việc (xóa DOM element và cập nhật mảng).
  - Cách sử dụng **event delegation** (nếu muốn nâng cao) hoặc gắn sự kiện trực tiếp từng nút.
- **Nhiệm vụ của sinh viên:**
  - Thêm một nút “Xóa” bên cạnh mỗi công việc. Khi bấm nút, công việc biến mất.
  - Thêm tính năng “đánh dấu hoàn thành” (ví dụ, khi bấm vào tiêu đề công việc).
  - Cập nhật trạng thái trong mảng (nếu dùng mảng đối tượng, có thể có thuộc tính `isDone: true/false`).

#### 4. Lưu dữ liệu cục bộ (Local Storage) (30 phút)

- **Hướng dẫn của giảng viên:**
  - Cách sử dụng `localStorage` để lưu danh sách công việc dưới dạng JSON.
  - Mỗi khi thêm/xóa/sửa, cần cập nhật lại `localStorage`.
  - Khi tải lại trang, đọc dữ liệu từ `localStorage` để hiển thị danh sách.
- **Nhiệm vụ của sinh viên:**
  - Viết hàm `saveTasksToLocalStorage()` và `loadTasksFromLocalStorage()`.
  - Đảm bảo mọi thay đổi đều được đồng bộ với `localStorage`.

#### 5. Mở rộng tính năng (tùy chọn) (30 phút)

- **Hướng dẫn của giảng viên:**
  - Gợi ý thêm tính năng lọc công việc “Tất cả”, “Chưa hoàn thành”, “Đã hoàn thành”.
  - Hoặc thêm tính năng **chỉnh sửa** (edit) nội dung công việc.
- **Nhiệm vụ của sinh viên (tùy chọn):**
  - Tự do sáng tạo, thêm tính năng tùy thích (sắp xếp công việc theo ngày, deadline, ...).
  - Tối ưu giao diện bằng CSS nâng cao (SASS/SCSS, Bootstrap).

#### 6. Kiểm tra & Chấm điểm (30 phút)

- **Hướng dẫn của giảng viên:**
  - Yêu cầu sinh viên chạy thử ứng dụng, thêm công việc, đánh dấu hoàn thành, xóa, tải lại trang xem dữ liệu còn hay không.
  - Quan sát code, đánh giá mức độ sử dụng các khái niệm JS (mảng, vòng lặp, DOM manipulation, event handling, `localStorage`, v.v.).
- **Nhiệm vụ của sinh viên:**
  - Trình bày ngắn gọn logic code.
  - Giải đáp các câu hỏi của giảng viên (nếu có).

#### Thời lượng gợi ý:

- Chuẩn bị & Giới thiệu bài: ~15 phút
- Thêm công việc: 30 phút
- Hoàn thành / Xóa công việc: 45 phút
- Local Storage: 30 phút
- Mở rộng tính năng: 30 phút
- Kiểm tra & Chấm điểm: 30 phút

**Tổng: 180 phút**

---

## PHẦN 2: LUYỆN TẬP JQUERY (120 phút)

Sau khi đã làm quen với **Vanilla JS**, sinh viên chuyển sang áp dụng **jQuery** để rút gọn cú pháp và làm quen với các thư viện JS phổ biến.

### 1. Mục tiêu

- Hiểu cách **khởi tạo** và **sử dụng** jQuery trong dự án.
- Luyện tập **cú pháp chọn phần tử** (selectors) và **chaining** trong jQuery.
- Xây dựng tính năng **AJAX** đơn giản để lấy dữ liệu (hoặc gửi form).
- Thêm hiệu ứng / animation cơ bản (hide, show, slide, fade, ...).

### 2. Nội dung & Hoạt động

#### Bài tập: “Form Đăng Ký với jQuery và AJAX”

##### 1. Khởi tạo dự án jQuery (10 phút)

- **Hướng dẫn của giảng viên:**
  - Thêm link CDN jQuery (hoặc tải file jquery.min.js).
  - Kiểm tra jQuery đã hoạt động hay chưa (dùng `$(document).ready()` hoặc `$(function() {})`).
- **Nhiệm vụ của sinh viên:**
  - Tạo file HTML cơ bản có form đăng ký (họ tên, email, mật khẩu, ...).
  - Nhúng jQuery, in ra console “jQuery is ready!” để kiểm tra.

##### 2. Xử lý sự kiện với jQuery (30 phút)

- **Hướng dẫn của giảng viên:**
  - Sử dụng **selectors** để lấy các phần tử form: `$('#form')`, `$('#email')`, ...
  - Lắng nghe sự kiện **submit** (`$('#form').on('submit', function(e) { ... })`).
  - Ngăn chặn hành vi mặc định (`e.preventDefault()`).
- **Nhiệm vụ của sinh viên:**
  - Khi người dùng bấm nút “Đăng ký”, kiểm tra **ràng buộc** (họ tên không trống, email hợp lệ, mật khẩu đủ dài, ...).
  - Nếu lỗi, hiển thị thông báo bên dưới trường input (bằng cách chèn `<span>` hoặc thay đổi CSS).
  - Nếu không lỗi, cho phép gửi form hoặc thực hiện bước tiếp theo.

##### 3. AJAX cơ bản (30 phút)

- **Hướng dẫn của giảng viên:**
  - Giải thích khái niệm AJAX: gửi dữ liệu lên server mà không cần reload trang.
  - Sử dụng phương thức `$.ajax()` hoặc `$.get()`, `$.post()` để gửi dữ liệu form lên server giả lập (có thể dùng một API mock hoặc JSON Placeholder).
- **Nhiệm vụ của sinh viên:**
  - Thay vì submit form theo cách truyền thống, gọi AJAX để gửi dữ liệu (tên, email, ...) lên endpoint.
  - Xử lý **thành công**: Hiển thị thông báo “Đăng ký thành công” (hoặc hiển thị JSON response).
  - Xử lý **thất bại**: Thông báo lỗi (VD: “Server bận, vui lòng thử lại sau”).

##### 4. Hiệu ứng / Animation (20 phút)

- **Hướng dẫn của giảng viên:**
  - Giới thiệu nhanh các hàm hiệu ứng như `.hide()`, `.show()`, `.fadeIn()`, `.slideToggle()`, ...
  - Cách chaining nhiều phương thức jQuery.

- **Nhiệm vụ của sinh viên:**
  - Khi form gửi thành công, ẩn form đăng ký bằng hiệu ứng `.slideUp()` hoặc `.fadeOut()`, rồi hiển thị một khối `<div>` chúc mừng.
  - Hoặc tạo nút “Xem chi tiết” để hiển thị thêm thông tin bằng hiệu ứng.
- 5. **Tổng kết & Đánh giá (30 phút)**
  - **Hướng dẫn của giảng viên:**
    - Yêu cầu sinh viên trình bày, demo tính năng.
    - Kiểm tra thao tác jQuery selectors, event handling, AJAX, animation.
  - **Nhiệm vụ của sinh viên:**
    - Giải đáp thắc mắc, nhận phản hồi để hoàn thiện bài.
    - Tham khảo cách viết code ngắn gọn, rõ ràng, tách riêng file JS.

**Thời lượng gợi ý:**

- Khởi tạo dự án jQuery: 10 phút
- Xử lý sự kiện với jQuery: 30 phút
- AJAX cơ bản: 30 phút
- Hiệu ứng / Animation: 20 phút
- Tổng kết & Đánh giá: 30 phút

**Tổng: 120 phút**

---