

Yêu cầu chung

- Tạo repo `f8-fullstack-day26`, deploy với Github Pages
 - Mỗi bài tập tạo 1 file HTML riêng biệt kèm JS thuần
 - Giao diện tự làm CSS, đơn giản nhưng rõ ràng
 - Code clean, có comment giải thích các phần quan trọng
-

Bài 1: Location API Demo

1.1. Location Properties Inspector

File: `location-inspector.html`

Yêu cầu:

1. Tạo giao diện hiển thị **tất cả thuộc tính** của `window.location` trong một bảng hoặc danh sách với format rõ ràng. Cụ thể hiển thị các thuộc tính sau:
 - `href` - URL đầy đủ hiện tại
 - `protocol` - Giao thức đang sử dụng (http: hoặc https:)
 - `hostname` - Tên miền (ví dụ: google.com)
 - `port` - Số cổng (ví dụ: 3000, 8080, hoặc rỗng nếu là cổng mặc định)
 - `pathname` - Đường dẫn sau tên miền (ví dụ: /products/detail)
 - `search` - Phần query string bao gồm dấu ? (ví dụ: ?id=123&name=abc)
 - `hash` - Phần fragment identifier bao gồm dấu # (ví dụ: #section1)
 - `origin` - Origin của URL (protocol + hostname + port)
2. Mỗi thuộc tính hiển thị theo format:

Property Name: `location.href`

Current Value: `https://example.com:3000/products?id=123#details`

1.2. URL Manipulation Tool

File: `url-manipulator.html`

Yêu cầu:

1. Tạo form với các input cho phép user tạo URL mới:

- Protocol (dropdown với options: http:, https:)
- Hostname (text input - ví dụ: google.com)
- Port (number input - ví dụ: 3000, có thể để trống)
- Pathname (text input - ví dụ: /products/detail)
- Search/Query (text input - ví dụ: ?id=123&name=abc)
- Hash (text input - ví dụ: #section1)

2. Có button "Navigate" sử dụng `location.assign()` để chuyển tới URL được tạo từ form

3. Có button "Replace" sử dụng `location.replace()` để thay thế URL hiện tại

4. Có button "Reload" sử dụng `location.reload()` để tải lại trang

Bài 2: History API Navigation

2.1. Single Page Navigation

File: `spa-navigation.html`

Yêu cầu:

1. Tạo menu navigation với 4 "pages": Home, About, Services, Contact. **Lưu ý:** Không tạo các file HTML tương ứng `home.html`, `about.html` ... mà chỉ sử dụng 1 file HTML duy nhất và thay đổi nội dung bằng JavaScript.

2. Sử dụng `history.pushState()` để thay đổi URL khi click menu mà **không reload trang**

3. Sử dụng `window.addEventListener('popstate')` để xử lý khi user nhấn back/forward.

Vấn đề cần giải quyết: Khi user click menu, URL thay đổi nhưng nếu user nhấn nút back/forward của browser, trang sẽ không tự động cập nhật nội dung tương ứng với URL mới. Event `popstate` sẽ được fired khi user navigation qua back/forward buttons, cho phép chúng ta detect và cập nhật nội dung phù hợp.

4. Mỗi "page" hiển thị nội dung khác nhau trong cùng 1 file HTML:

- **Home:** "Chào mừng đến với trang chủ! Đây là nội dung của trang Home."
- **About:** "Thông tin về chúng tôi. Chúng tôi là công ty chuyên về công nghệ web."
- **Services:** "Các dịch vụ của chúng tôi: Thiết kế web, Lập trình ứng dụng, Tư vấn công nghệ."
- **Contact:** "Liên hệ với chúng tôi: Email: contact@example.com, Phone: 0123456789"

5. **Yêu cầu F5:** Khi user F5 (refresh) trang ở bất kỳ URL nào (ví dụ: `/about` , `/services`), trang vẫn phải hiển thị đúng nội dung tương ứng với URL đó, không về lại Home.

6. Hiển thị thông tin history hiện tại:

```
History Length: 5
```

Hiển thị giá trị của `history.length`

Bài 3: Navigator API Applications

3.1. Clipboard Manager

File: `clipboard-manager.html`

Yêu cầu:

1. Copy to Clipboard:

- Text area để nhập nội dung cần copy
- Button "Copy" sử dụng `navigator.clipboard.writeText()` để copy nội dung

2. Paste from Clipboard:

- Button "Paste" sử dụng `navigator.clipboard.readText()`
- Hiển thị nội dung vừa paste vào text area khác

3.2. Camera Capture App

File: `camera-capture.html`

Yêu cầu:

1. Video Stream:

- Button "Start Camera" sử dụng `navigator.mediaDevices.getUserMedia()` để bật camera
- Hiển thị video stream trong thẻ `<video autoplay>`
- Button "Stop Camera" để tắt camera. **Hướng dẫn stop camera:** Sử dụng `stream.getTracks().forEach(track => track.stop())` để dừng tất cả tracks của stream

2. Photo Capture:

- Button "Take Photo" để chụp ảnh từ video stream
- Sử dụng `<canvas>` để capture frame hiện tại từ video với kích thước canvas bằng kích thước của phần tử video
- Hiển thị ảnh vừa chụp trong thẻ ``
- Button "Download Photo" để tải ảnh về máy. **Hướng dẫn download:**

```
// Tạo element <a> để download
const link = document.createElement('a');
// Đặt tên file download
link.download = 'photo.png';
// Chuyển canvas thành data URL để tạo link download
link.href = canvas.toDataURL();
// Tự động click để trigger download
link.click();
```