3. Sự khác biệt chính

Thẻ Link:

Điều hướng tĩnh, thường xuất hiện trong JSX như một phần của giao diện người dùng.

Không thực hiện bất kỳ logic nào trước khi điều hướng.

Tương tự như thẻ <a>, nhưng không tải lại toàn bộ trang, giúp ứng dụng

React của bạn giữ trạng thái hiện tại.

Hook useNavigate:

Điều hướng có lập trình (programmatic navigation), dựa trên các sự kiện

hoặc logic trong ứng dụng.Thường được sử dụng bên trong các hàm xử lý

sự kiện hoặc sau khi xử lý logic.Không cần phải có liên kết trực

tiếp trong giao diện người dùng.

NavLink:

NavLink là một biến thể của Link, cung cấp thêm khả năng tự động thêm các class CSS dựa trên trạng thái điều hướng.

Được sử dụng chủ yếu trong các menu điều hướng để hiển thị trạng thái hiện tại của đường dẫn (ví dụ: đường dẫn đang được chọn sẽ có một class đặc biệt).

4. Khi nào sử dụng cái nào:

Sử dụng Link:

Khi bạn chỉ cần một liên kết điều hướng đơn giản, như trong các menu,

danh sách điều hướng, hoặc liên kết trong nội dung.

Khi bạn không cần điều hướng có điều kiện hoặc không cần thực

hiện bất kỳ logic nào trước khi điều hướng.

Sử dụng useNavigate:

Khi bạn cần điều hướng có điều kiện, sau khi thực hiện một hành động nào

đó, như đăng nhập, đăng ký, hoặc submit form.Khi điều hướng không

được xác định trước, hoặc bạn cần kiểm soát chính xác thời điểm điều hướng xảy ra.

Sử dụng NavLink:

<NavLink to="/about" activeClassName="active">About</NavLink>

activeClassName: Thuộc tính này cho phép bạn đặt một class CSS đặc biệt khi đường dẫn tương ứng với NavLink đang được kích hoạt (người dùng đang ở trang đó).

exact: Có thể được sử dụng để chỉ áp dụng class khi đường dẫn chính xác khớp với URL hiện tại.

Sử dụng NavLink khi bạn cần tạo một menu điều hướng và muốn hiển thị trạng thái đường dẫn hiện tại (ví dụ: làm nổi bật mục menu đang được chọn).

**tag <select>**

**Mô tả:**

* Thẻ <select> được sử dụng để tạo ra một danh sách tùy chọn (dropdown), cho phép người dùng chọn một hoặc nhiều giá trị từ danh sách đã định nghĩa.

**Cấu trúc:**

html

Sao chép mã

<select name="options">

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

**Thuộc tính chính:**

* name: Tên của trường này khi gửi dữ liệu lên máy chủ.
* multiple: Nếu thuộc tính này được đặt, người dùng có thể chọn nhiều tùy chọn cùng lúc.
* size: Số lượng tùy chọn hiển thị mà không cần cuộn danh sách.

Mục đích của htmlFor là để gọi thẻ input có value id chính là value của htmlFor. Khi nhấn vào label thì thẻ input sẽ xuất hiện

 <label htmlFor="image"></label>

<input onClick={(e) => setImage(e.target.files[0])} type="file" id="image" hidden required />

**event.target:** Đại diện cho phần tử (element) mà sự kiện xảy ra, trong trường hợp này là một trường input.

**event.target.name:** Trả về giá trị của thuộc tính name của phần tử input, thường được sử dụng để xác định loại dữ liệu (ví dụ: name="username").

**event.target.value:** Trả về giá trị mà người dùng đã nhập vào trường input.

Deploy on render part redirect rewrite

**Khi bạn thay đổi cấu trúc URL**: Redirect để đảm bảo rằng người dùng và công cụ tìm kiếm vẫn có thể tìm thấy nội dung mà không gặp lỗi 404.

**Khi bạn muốn tạo các URL thân thiện**: Rewrite để làm cho các URL dễ đọc hơn và thân thiện với SEO.

**Khi bạn cần chuyển hướng từ HTTP sang HTTPS**: Redirect tất cả các yêu cầu từ HTTP sang HTTPS để tăng cường bảo mật.

**Khi bạn đang triển khai ứng dụng SPA (Single Page Application)**: Dùng Rewrite để điều hướng tất cả các yêu cầu không tìm thấy đến index.html để cho phép điều hướng client-side.

**Rewrite tất cả các yêu cầu không khớp với file cụ thể đến index.html (dùng cho SPA)**:

plaintextSource: /.\*

Destination: /index.html

Status: 200

**Khi nào dùng FormData và khi nào không cần:**

* **Dùng FormData** khi form của bạn có tệp tin hoặc bạn cần gửi dữ liệu theo định dạng multipart/form-data.
* **Không cần FormData** khi form của bạn không có tệp tin và bạn chỉ cần gửi dữ liệu dưới dạng JSON hoặc các định dạng văn bản khác.

Tóm lại, hãy sử dụng FormData khi bạn cần xử lý tệp tin hoặc dữ liệu phức tạp mà useState không hỗ trợ tốt, còn useState sẽ phù hợp cho các form đơn giản, không có tệp tin.