dùng "npm create vite": tạo dự án react với vite

Mục đích của cú pháp {[itemId]: value}:

Tạo khóa động (Dynamic Keys):

Khi bạn muốn sử dụng giá trị của một biến hoặc biểu thức làm tên khóa trong một đối tượng, bạn cần sử dụng cú pháp [].

Ví dụ: Nếu itemId là "product\_123", thì khi sử dụng cú pháp [itemId]: value, đối tượng sẽ có khóa là "product\_123".

Không giới hạn tên khóa:

Thay vì phải cố định tên khóa khi tạo đối tượng, bạn có thể linh hoạt tạo khóa theo giá trị của biến hoặc biểu thức.

Ví dụ:

javascript

const keyName = "dynamicKey";

const myObject = {

[keyName]: "This is a dynamic value"

};

console.log(myObject); // { dynamicKey: 'This is a dynamic value' }

Trong ví dụ này:

keyName là biến có giá trị "dynamicKey".

Khi sử dụng [keyName]: "This is a dynamic value", đối tượng myObject sẽ có một khóa

tên là "dynamicKey" với giá trị "This is a dynamic value".

Danh sách mã kí tự đặc biệt trong html

https://xuanthulab.net/danh-sach-ma-html-cac-ky-tu-dac-biet.html

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.

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).

<https://docs.stripe.com/testing#cards>: link để lấy thẻ test stripe

useSearchParams là một hook trong thư viện react-router-dom dùng để làm việc với các tham số truy vấn (query parameters) trong URL. Các tham số này thường được sử dụng để truyền dữ liệu hoặc trạng thái qua URL mà không cần sử dụng state nội bộ của React.

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.

**Axios Interceptors** là một tính năng mạnh mẽ của Axios (một thư viện HTTP client cho JavaScript) cho phép bạn can thiệp vào các request hoặc response trước khi chúng được xử lý bởi then hoặc catch. Bạn có thể sử dụng interceptors để thêm logic tùy chỉnh vào mỗi request hoặc response, như thêm token xác thực, xử lý lỗi chung, hoặc thay đổi dữ liệu