Filter(item=>()) là để lọc ra những cái nào khác điều tiện trong này nó hoạt động như forEach.  
1.JSX là gì:JSX là code html được viết trong 1 function component ở file js chứ không phải ở file html

2.SPA là gì: SPA là viết tắt của Sigle Page Application, nó có nghĩa là 1 trang web chỉ chạy trên 1 trạng html và nó giúp chúng ta chuyển trang mà không cần phải load lại trang giúp cải thiện trải nghiệm của người dùng.

3.Trong 1 function component bắt buộc phải return là nó return về 1 jsx, và khối jsx đó chỉ được có 1 parent duy nhất.

{/\* thuộc tính taget blank là để khi bấm vào link đó thì sang tab mới \*/}

        {/\* inline style trong react  sẽ phải dùng 2 dấu {{}} dấu {} đầu tiên sẽ là tôi biết bạn muốn sử dụng css react là đấu thứ 2 sẽ là object

        và ở trong đó bắt đầu chuyền key và value\*/}

4.lưu ý trong react khi mà chúng ta them 1 sự kiện vào và chuyền 1 hàm vào thì nếu chúng ta chuyền trực tiếp tên hàm vào như này

<button onClick={solveEvent()}>click me</button>

Thì hàm này sẽ được chạy trước 1 lần khi mà ta chưa bấm vào nó.Để giải quyết thì ta sẽ cho 1 arrow func ở đẳng trước nó.

<button onClick={()=>{solveEvent()}}>click me</button>

**useState():**bản chất là trả ra 1 object

-useState là một hook trong React cho phép chúng ta quản lý state trong các component function. State là một biến cục bộ của component, có thể được thay đổi theo thời gian và sẽ phản ánh trên DOM.

* -Phần tử thứ nhất: Đây là biến state.
* -Phần tử thứ hai: Đây là hàm dùng để cập nhật giá trị của state.

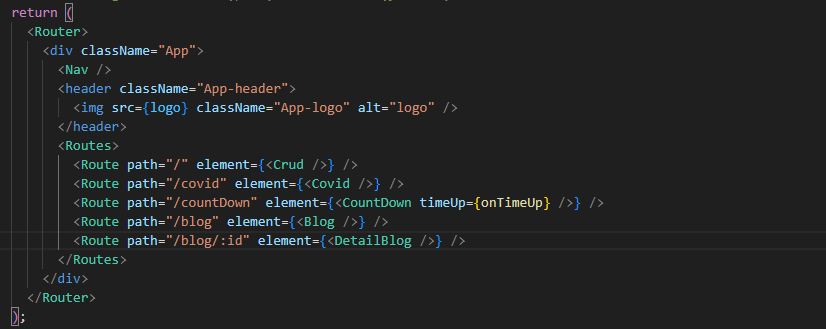
**useEffect():**Sẽ được chạy khi component bị rerender.nếu ta chuyền 1 ham mà hàm đó làm rerender lại component thì sẽ sảy ra hiện tượng là vòng lặp vô hạn.Vì thế nên sẽ them 1 parameter vào đó chính là phụ phuộc useEffect( ()=>{},[] ) và trong cặp ngoặc vuông đó sẽ chuyền vào 1 dependences, nếu mà cái dependences đấy nó bị thay đổi thì cái useEffect nó mới được chạy. Ta có thể gọi để sử dụng nhiều UseEffect cùng 1 lúc được, hoặc là chuyền thêm 1 dependence vào ngoặc vuông đó.

-Hàm cleanup là return()=>{} trong useEffect sẽ được gọi khi mà component bị unmount hoặc là dependences bị thay đổi.

5.Outputing Lists – Render loop

6.Props là dùng để chuyền dữ liệu từ các component từ thằng cha xuống cho thằng con

**7.Router**

**-**

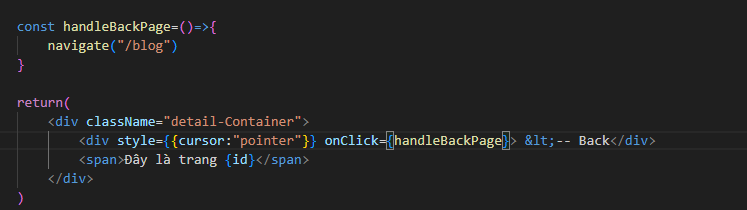


-path của Route phải trùng với to của Link hoặc NavLink

-ngoài ra bên phần path của Route có hỗ trợ /:id là nó sẽ tự hiểu là ta sẽ chuyền 1 cái id vào trong và có 1 hàm useParams() trả về 1 object là 1 id mà mình chuyền về.

**8.useNavigate**

-Dùng để di chuyển đến 1 trang nào đó trong app



**9.Để Dọn dẹp lại 1 component trước khi nó bị unmounted thì return trong useEffect sẽ là willunmount.**

**10.UseContext()**

-Đầu tiên để sử dụng được useContext ta phải trải qua 3 bước:

+)B1:Create Context

+)B2:Provider  
+)B3:Consumer

-Để tạo 1 createContext ta phải khai báo createContext from ‘react’🡪 create context đó là 1 object, trong object đó nó chưa 2 thằng đó là provider và consumer.

-Và sau Khi tạo createContext xong thì sẽ thực hiện gọi createContext đó truy cập tới Provider để thực hiện bọc component cha để thực hiện truyền dữ liệu, và trong provider sẽ thực hiện truyền vào 1 value và những thằng component con maf được bọc bởi provider sẽ nhận được cái value đó.

VD: export const Context=createConText();

<Context.Provider value”myValue”>

</Context.Provider>

-Và để thực hiện consumer cái value đó thì thằng components con sẽ thực hiện import cái thằng Context ở component cha và tạo 1 thằng useContext from ‘react’ để thực hiện tiêu thụ.

VD: const myValue=useContext(Context)

**11.useMemo()**

-Là 1 component để thực hiện ghi nhớ kết quả của 1 hàm,useMemo() nhận vào 2 parameter là callback và dependences, về callback nó chả về giá trị được tính toán cần ghi nhớ, và dependences thì cũng giống như useEffect().

**12.UseLayoutEffect()**

-Hook này sử dụng như useEffect nhưng nó khác về thứ tự thực hiện công việc:

VD: useEffect:

1.Cập nhật lại state

2.Cập nhật DOM

3.render UI

4.cleanup

5.callback

useLayoutEffect:

1.cập nhật state

2.cập nhật dom

3.clean up

4.call back

5.renderUI

**13.UseRef()**

-Dùng để lưu trữ giá trị tham chiều và có thể truy xuất đến cách thành phần DOM.

-useRef() là 1 object và muốn lấy được giá trị của nó đang được lưu trữ thì bắt buộc phải truy cập đến key current trong đó.

VD: let value=useRef() -> lấy giá trị value.current

-Truy xuất đến dom ta có thể dùng như là : <div ref={useRef()} >hi</div> thì lúc này ta có thể nhận được cả element <div>hi</div> kia

**14.React.memo()->Higher Orders Components**

-Tác dụng của React.memo là tránh làm các component con bị re-reder không cần thiết.

-Nó nhận vào 1 component là chính nó và sẽ thực hiện việc không bị làm bản thân bị tự động bị re-render.Trong trường hợp mà nó nhận vào props thì nó sẽ so sánh giá trị

**15.UseCallback()**

-Dùng để tối ưu hóa trương trình, khi tạo 1 useCallback nó sẽ tạo ra 1 tham chiếu mới và tham chiếu của nó chỉ thay đổi khi dependences của nó thay đổi.

**16.useMemo()**

-nó nhận vào 2 tham số đó là callback và denpendences.Và nó sẽ trả về kết quả tính toán trong hàm callback