Khởi tạo project

* Giới thiệu : create-react-app
* Cài đặt bằng lệnh : npm install –g create-react-app
* Tạo mới ứng dụng : create – react- app tên project
* Chạy ứng dụng : npm start
* Đổi port tại package.json:
* “start”:”set port=4200 && react-scripts start”

\* những kiến thức khi làm việc với component:

+ cách nhúng các component vào các page là chỉ cần tạo ra 1 page con và ném vào page cha. Như vậy khi hiển thị thì nó sẽ show lên cả thuộc tính của lớn con

Ví dụ : lớp cha là dashboard trong src tạo ra 1 folder project trong đó tạo 1 component con . muốn nhúng component con vào thì dashboard.js thêm thẻ class con của ProjectItem là kết quả khi show gia trị cha thì nó còn show luôn cả giá trị của thằng con . ở đây component được nhúng thỏa thích để bạn show dữ liệu chỉ cần gọi tên class của nó lên và việc dung

* Reactjs

tạo nguyên mẫu, nhanh hơn

Giao tiếp tốt hơn với các trình duyệt giao diện người dùng

hiển thị trường hợp API của bạn cho các quảng cáo trả lời kinh doanh

- Redux

- Lấy cảm hứng từ kiến ​​trúc Flux của facebook

lưu trữ trạng thái ứng dụng - hỗ trợ apis không trạng thái

- Hoàn hảo khi bạn có lượng dữ liệu hợp lý thay đổi theo thời gian

**1.Giới thiệu**

* React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application.

- React là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

2. React là gì

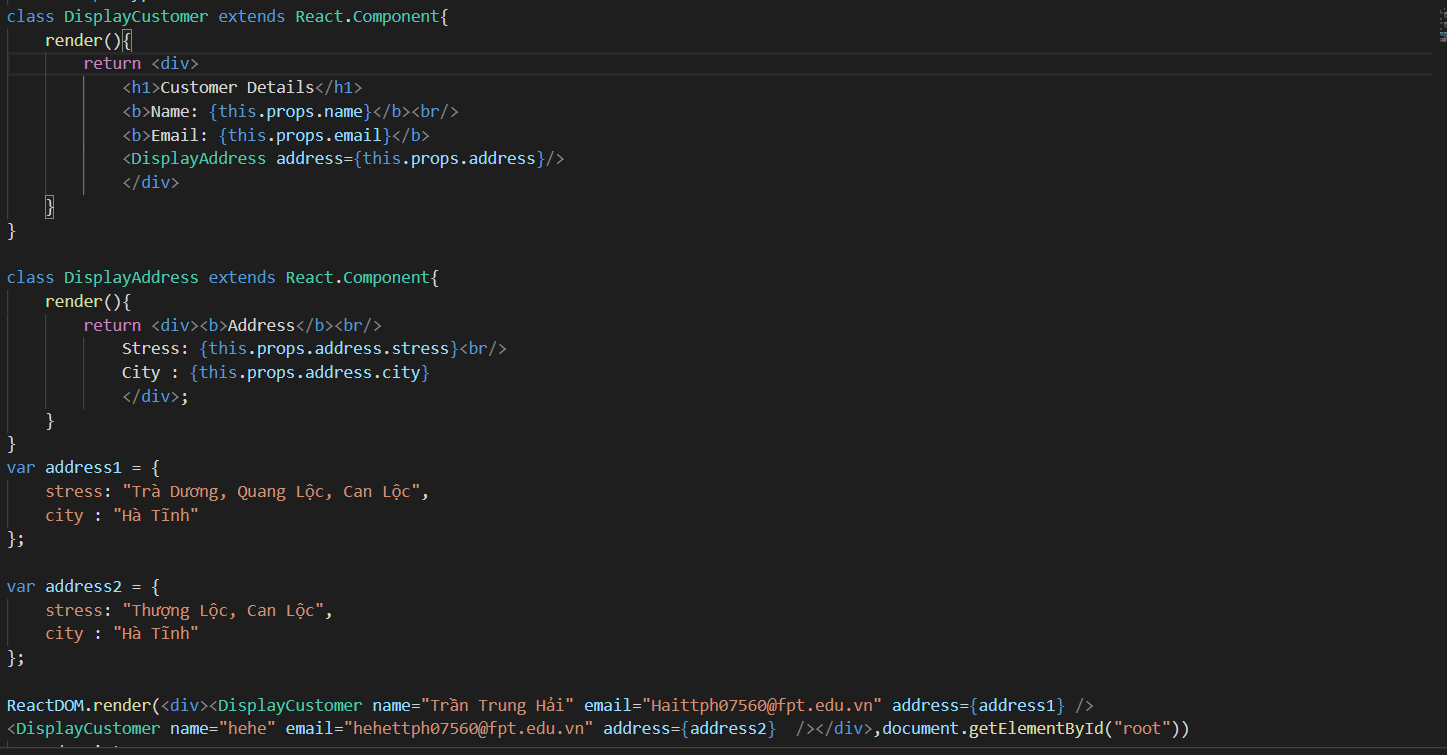
React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production

### Giới thiệu về JSX

* JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch.
* Vừa giúp ta custome 1 html theo ý mình
* JSX giúp cho việc xây dựng các ứng dụng React một cách nhanh hơn, dễ tối ưu trong việc complie code sang javascript.
* Nó khá giống với HTML nên dễ dàng cho việc viết chuyển đổi.



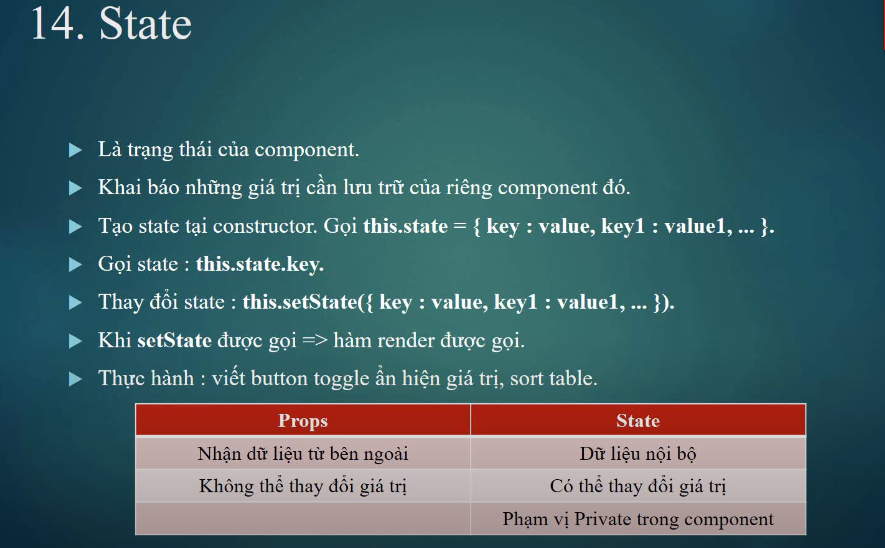
* Có thể nhúng các element tự custome rồi nhúng vào các thẻ khác ngay trên html



### Props và State:

Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI.

* Lúc nào dùng {this.props.children} : khi muốn lấy text ở trong class đó
* Vidu: <Text>Hello</Text>
* Khi dùng {this.props.title} : khi muốn lấy properties trong class Text
* Ví dụ <Text title=”Xin chào” />
* Var thì được khai báo trong render hoặc ngoài class
* Props thì được truyền từ component cha.
* State thì được truyền từ một components
* Global State nên dùng cho nhiều components.
* State chỉ có thể được sử dụng ở trong một component sinh ra nó.
* Nếu dự đoán được một component cần quản lý state, thì nó nên được tạo ở trong một class component chứ không phải là một function component.



**1) React-Router là gì?**

React-Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khẳng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng.

Phiên bản mới nhất hiện tại là v4.

Cài đặt:

npm install react-router-dom –save

***2.2) Route***

Route: Định nghĩa một ánh xạ (mapping) giữa một URL và một Component. Điều đó có nghĩa là khi người dùng truy cập theo một URL trên trình duyệt, một Component tương ứng sẽ được render trên giao diện.

<Router>

<div className="App">

<Route path="/" exact component={Home} />

<Route path="/about" component={About} />

<Route path="/contact" component={Contact} />

<Route component={NotFound}/>

</div>

</Router>

Trong đó:

* **path:** Là đường dẫn trên URL.
* **exact:** Giúp cho route này này chỉ hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.
* **component:** Là component sẽ đươc load ra tương ứng với Route đó.

**Fragment :** là một common pattern được giới thiệu kể từ khi React 16 ra đời. Nó cho phép bạn return nhiều element từ một component mà không làm sinh ra những DOM element ko cần thiết.

render() {

return (

<React.Fragment>

<ChildA />

<ChildB />

<ChildC />

</React.Fragment>

);

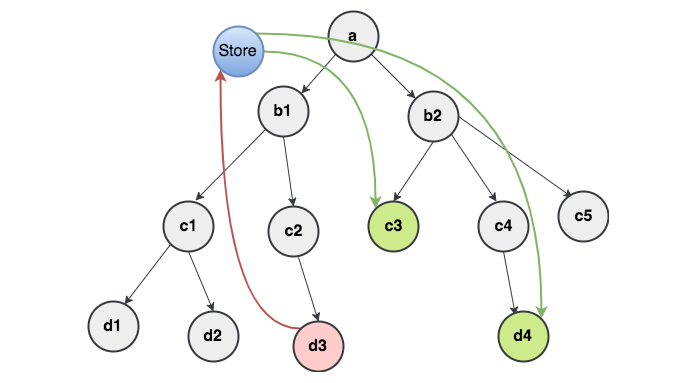
}

**Constructor(props) :**

* Hàm này Thực hiện việc thiết lập state cho component.
* Việc sử dụng super(props) là để có thể sử dụng this.props trong phạm vi hàm constructor này
* this.onChange = this.onChange.bind(this); nếu thiếu bind(this) thì giá trị trong function onChange sẽ **undefined**
* **preventDefault**: Huỷ bỏ event nếu nó có thể huỷ mà không dừng sự lan rộng(propagation) của event tới phần khác.
* **stopPropagation** Ngăn chặn sự lan rộng của sự kiện hiện tại tới thằng khác.
* **stopImmediatePropagation** ngăn chặn những listeners cũng đang đang lắng nghe cùng event được gọi.

## **Redux là gì?**

**Redux** là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. **[Redux](https://topdev.vn/blog/?s=redux)** ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ **Elm** và kiến trúc **Flux**của Facebook. Do vậy Redux thường dùng kết hợp với React.



*Cái cách mà Redux hoạt động là khá đơn giản. Nó có 1 store lưu trữ toàn bộ state của app. Mỗi component có thể access trực tiếp đến state được lưu trữ thay vì phải send drop down props từ component này đến component khác.*

3 thành phần của Redux: Actions, Store, Reducers.

1. Actions

Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

**Actions** được gửi bằng cách sử dụng store.dispatch() method, chúng phải có một type property để biểu lộ loại action để thực hiện. Chúng cũng phải có một payload chứa thông tin. Actions được tạo thông qua một action creator

2. Reducers

**Reducers** là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

**3. Store**

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listensaers thông qua helper methods.

[Giải thích về connect React-redux (viblo.asia)](https://viblo.asia/p/giai-thich-ve-connect-react-redux-eW65GPwLKDO)