|  |  |  |
| --- | --- | --- |
| Macintosh HD:Users:hoangnguyen:Documents:TDC:fitlog_blue.png | TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC  **Khoa Công Nghệ Thông Tin** | Macintosh HD:Users:hoangnguyen:Documents:TDC:logoTDC_blue.png |
|  |  |  |

**Chuyên đề phát triển web 2** | HKI – [2019 – 2020]

PRJ – REPORT

**BÁO CÁO TÌM HIỂU CÔNG NGHỆ REACT JS**

**Nhóm J:**

* **Thái Quang Hưng –** 17211TT3105 (Nhóm trưởng)
* **Hà Xuân Phi –** 17211TT0607 (Nhóm phó)
* **Trần Quốc Bữu –** 17211TT3375
* **Võ Đình Chinh –** 17211TT3587
* **Trần Bình Văn –** 17211TT0053

**MỤC LỤC**

[REACT JS 1](#_Toc23924418)

[**I.** **CÀI ĐẶT MÔI TRƯỜNG** 1](#_Toc23924419)

[**1.** **React Developer Tools** 1](#_Toc23924420)

[**2.** **Node.JS** 1](#_Toc23924421)

[**3.** **YARN** 1](#_Toc23924422)

[**4.** **IDE** 2](#_Toc23924423)

[**II.** **KHỞI TẠO CHƯƠNG TRÌNH DEMO** 2](#_Toc23924424)

[**1.** **Cài đặt React App** 2](#_Toc23924425)

[**2.** **Tạo Project** 2](#_Toc23924426)

[**3.** **Chạy chương trình DEMO.** 2](#_Toc23924427)

[**4.** **Cài đặt PORT cho từng Project** 3](#_Toc23924428)

[**5.** **Chỉnh sửa nội dụng Project** 4](#_Toc23924429)

[**6.** **Cấu trúc thư mục** 4](#_Toc23924430)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Kiểm tra phiên bản. 1](#_Toc23924397)

[Hình 2: Cài đặt yarn thay thế cho npm. 2](#_Toc23924398)

[Hình 3: Cài đặt react app 2](#_Toc23924399)

[Hình 4: Tạo Project với tên là demo 2](#_Toc23924400)

[Hình 5: Chạy chương trình DEMO 3](#_Toc23924401)

[Hình 6:Kết quả chương trình DEMO chạy 3](#_Toc23924402)

[Hình 7: Cài đặt port 1511 3](#_Toc23924403)

[Hình 8: Nội dung của Project 4](#_Toc23924404)

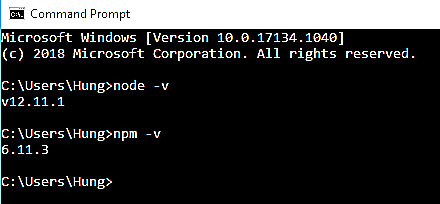
# REACT JS

1. **CÀI ĐẶT MÔI TRƯỜNG**
2. **React Developer Tools**

* React-detector: là Chorme-extension để xác định trang web nào sử dụng React.
* Show-me-the-react: firefox và chrome.
* React Developer Tools: là một Plusgin có thể mở rộng chức năng của các công cụ phát triển của trình duyệt. Nó tạo ra một tab mới trong các công cụ dành cho nhà phát triển.

1. **Node.JS**

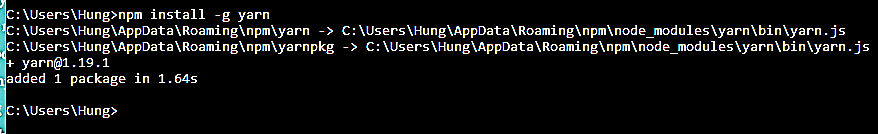
* Là một ngôn ngử lập trình server.
* Là nền tảng để xây dựng ứng dụng.
* Là mã nguồn mở.
* Không cần phải biết nếu chỉ học ReactJS. Chỉ cần biết sử dụng Node Package Manager (npm).
* Cú pháp cài đặt các package bằng npm: npm install [package-name] –save.



Hình . Kiểm tra phiên bản.

1. **YARN**

* Tương tự npm
* Nhanh hơn npm.
* Cú pháp npm install –g yarn.



Hình : Cài đặt yarn thay thế cho npm.

1. **IDE**

* Có thể sử dụng bất kỳ công cụ nào có thể code. VD: Sublime Text 3, Visual Studio Code, …
* Cài đặt JSX/Babel Syntax Highlighting.

1. **KHỞI TẠO CHƯƠNG TRÌNH DEMO**
2. **Cài đặt React App**

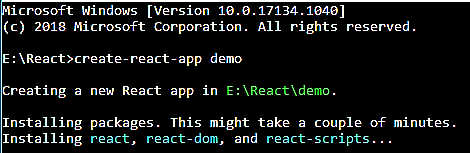
Sử dụng câu lệnh install –g create-react-app



Hình : Cài đặt react app

1. **Tạo Project**

Sử dụng câu lệnh create-react-app “Ten project”

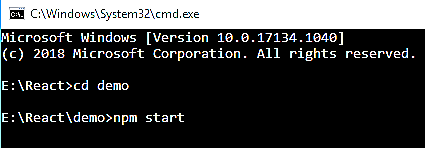


Hình : Tạo Project với tên là demo

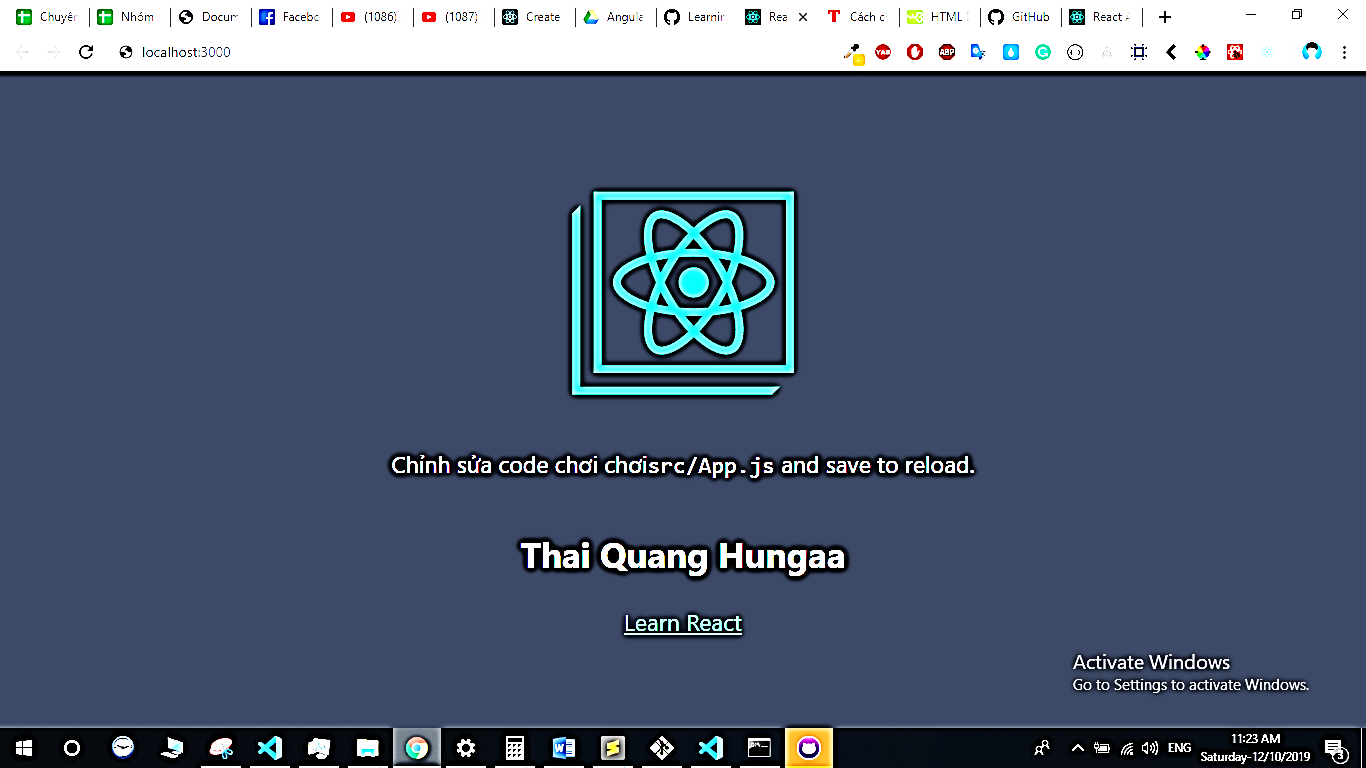
1. **Chạy chương trình DEMO.**

Sử dụng 2 câu lệnh:

* cd “Tên Project”
* npm start



Hình : Chạy chương trình DEMO



Hình :Kết quả chương trình DEMO chạy

1. **Cài đặt PORT cho từng Project**

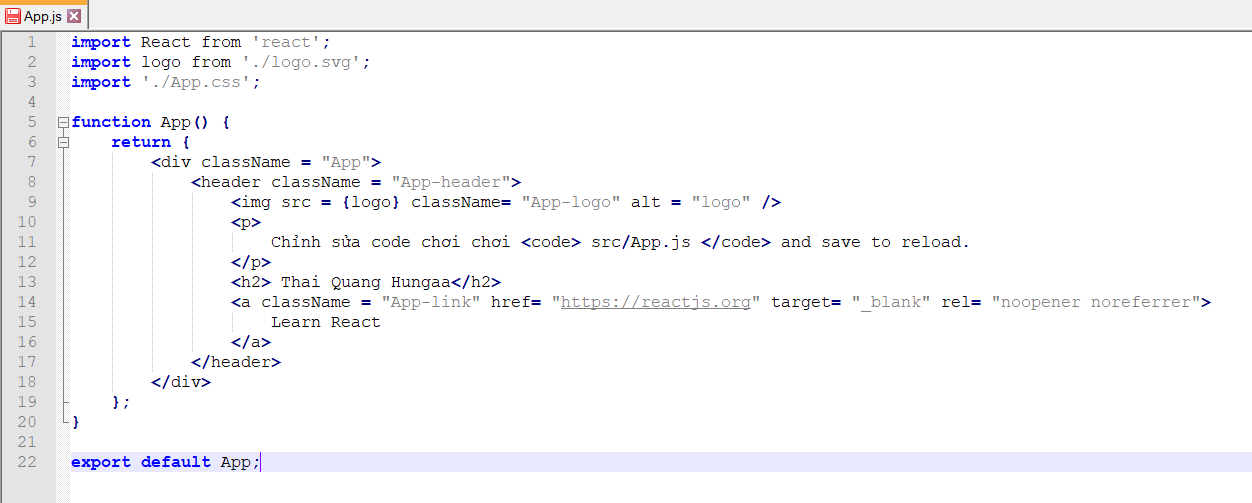
Mở file package.json tìm dòng: "start": "react-scripts start" và thay thế thành "start": " set port = GiaTriPort react-scripts start"



Hình : Cài đặt port 1511

1. **Chỉnh sửa nội dụng Project**

Tất cả nội dung của Project được code trong file App.js nằm trong folder src.



Hình : Nội dung của Project

1. **Cấu trúc thư mục**
2. Package.json

Là nơi khai báo: Tên của Project, Version, các thư viện, package,…

1. Node\_module

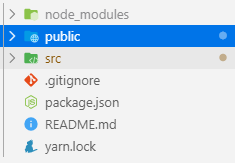
Là nơi chưa thư viện, package, …

1. Public/index.html

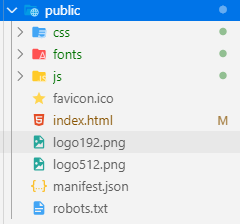
Là file chạy đâu tiên

1. Src

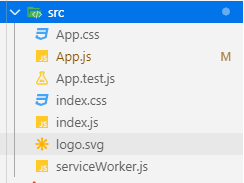
Là nới chúng ta làm việc nhiều nhất. Tất cả các thanh đổi về js đều được code trong thư mục src. Nơi quản lý các Component (Các module theo như môn học).



Hình . Cấu trúc tổng của thư mục.



Hình . Cấu trúc folder public.



Hình . Cấu trúc folder src.

1. **CÁC KỸ THUẬT REACTJS**
2. **Component**

* Component được hiểu như là một khối, một đoạn hoặc một màn hình trong một trang web.
* Việc sử dụng conponent giúp quản lý cũng như sửa đổi rất nhanh chóng và hiệu quả.
* Cách tạo ra một Conponent:
  + Tạo một folder với tên là components nằm trong folder src.



Hình . Tạo thư mục components trong thư mục src.

* + Tạo một file js với tên Conpoment nhưng không bắt buộc cùng tên.



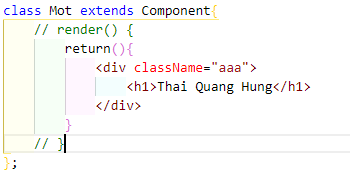
Hình . Tạo file với tên là 1.js

* + Thêm dòng code import React, { Component } from 'react'; vào ngay đầu file.



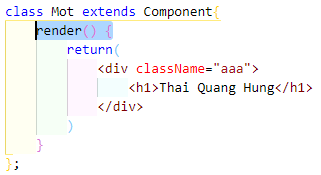
Hình . Thêm dòng code import vào đầu file 1.js.

* + Đối với cách cũ, tạo theo dạng function.



Hình . Tạo theo function trong file 1.js.

* + Đối với ES6, tạo theo dạng class dù là cách tạo nào thì ký tự đầu tiên cũng phải viết hoa.



Hình . Tạo theo Class trong file 1.js.

* + Export Component để có thể gọi và sử dụng.



Hình . Export Component mot để có thể gọi và sử dụng.

* + Export Component vào file App.js để có thể sử dụng



Hình . Export Component Mot vào file App.js.

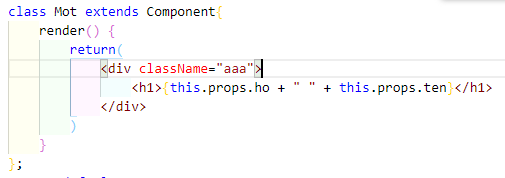
* + Gọi Component để sử dụng bằng cách <Tên Component/> hoặc <Tên Component></Tên Component>



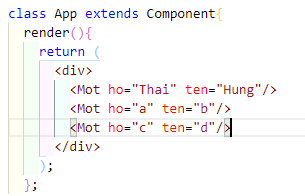
Hình . Gọi – sử dụng Component Mot theo 2 cách.

1. **Props**

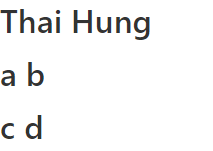
* Là các thuộc tính của 1 component.
* Truyền dữ liệu theo dạng key=>value từ file cha sang file con.
* Cú pháp: this.props.key.
* Giúp dễ quản lý cũng như sửa đổi.



Hình . Tạo Component Mot với các từ khóa this – file con.

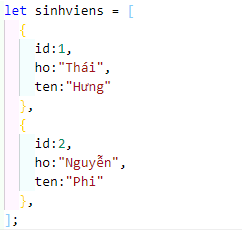


Hình . Gọi Component Mot với các cặp key=>value truyền vào.

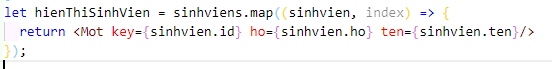


Hình Kết của ví dụ trên.

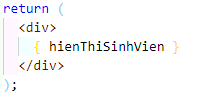
* Sử dụng vòng lặp để có thể props sản phẩm ra một cách nhanh chóng.



Hình . Tạo một mảng sinhviens có 2 đối tượng.



Hình . Tạo một biến và sửa dụng từ khóa map để return tất cả sinhvien.



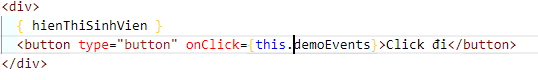
Hình . Gọi biến vừa tạo ra để hiển thị vị trí của nó.



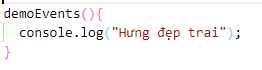
Hình . Kết quả của ví dụ trên.

1. **Events**

* Events trong ReactJS khá giống so với Jquery nhưng cách khai báo thì khác.
* Khái báo sự kiện click trong React: onTenSuKien={this.TenFunction}.
* Khởi tạo một function bằng cú pháp: TenFunction(){Nội dụng code}.
* Function phải nằm ngoài render().



Hình . Khái báo sự kiện onClick demoEvents cho nut button.

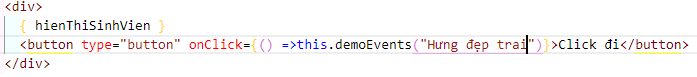


Hình . Khởi tạo function demoEvents.



Hình . Kết quả của ví dụ trên.

* Để sử dụng một sự kiện có tham số thì sử dụng cú pháp: onTenSuKien={() => this.TenFunction(tham số)}.



Hình . Khái tạo sự kiện onClick cho nút button.



Hình . Khởi tạo function demoEvents có tham số là text.

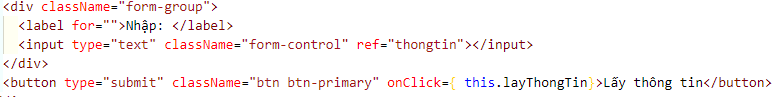


Hình . Kết quả của ví dụ trên.

* Ngoài ra còn có thể khởi tạo function bằng cú pháp: TenFunction = () => {Nội dung}.

1. **Refs**

* Refs dùng để lấy giá trị của các phần tử HTML.
* Sử dụng Refs với cú pháp: Thêm thuộc tính ref=”key”
* Cách lấy giá trị từ phần tử HTML đó là: this.refs.key.vulue

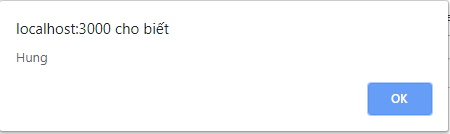


Hình . Khởi tạo sự kiện onClick cho button và thuộc tính ref cho input.





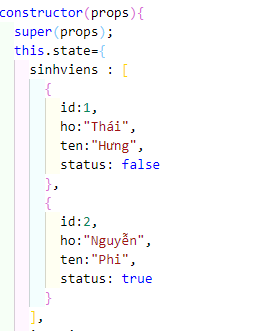
Hình . Nhập thông tin vào ô input.



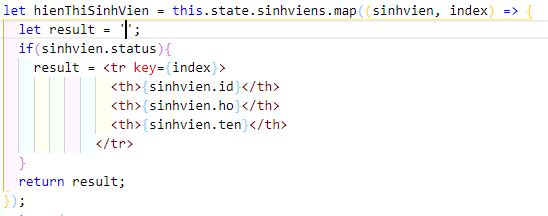
Hình . Kết quả của ví dụ trên.

1. **State**

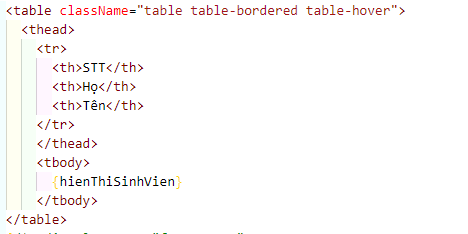
* State là một trạng thái trong ReactJS.
* State giúp khai báo những giá trị của riêng từng Components tương tự biến thuộc 1 class trong OOP.
* Cú pháp tạo State: this.state = {key: value, ....}.
* Cú pháp gọi State: this.state.key.
* Cú pháp thay đổi giá trị State: this.setState({key:value,...}).



Hình . Tạo một function constructor có đối tượng state.



Hình . Tạo một biến hienThiSinhVien trả về kết quả table nếu thuộc tính status = true.



Hình . Gọi biến hienThiSinhVien ra.