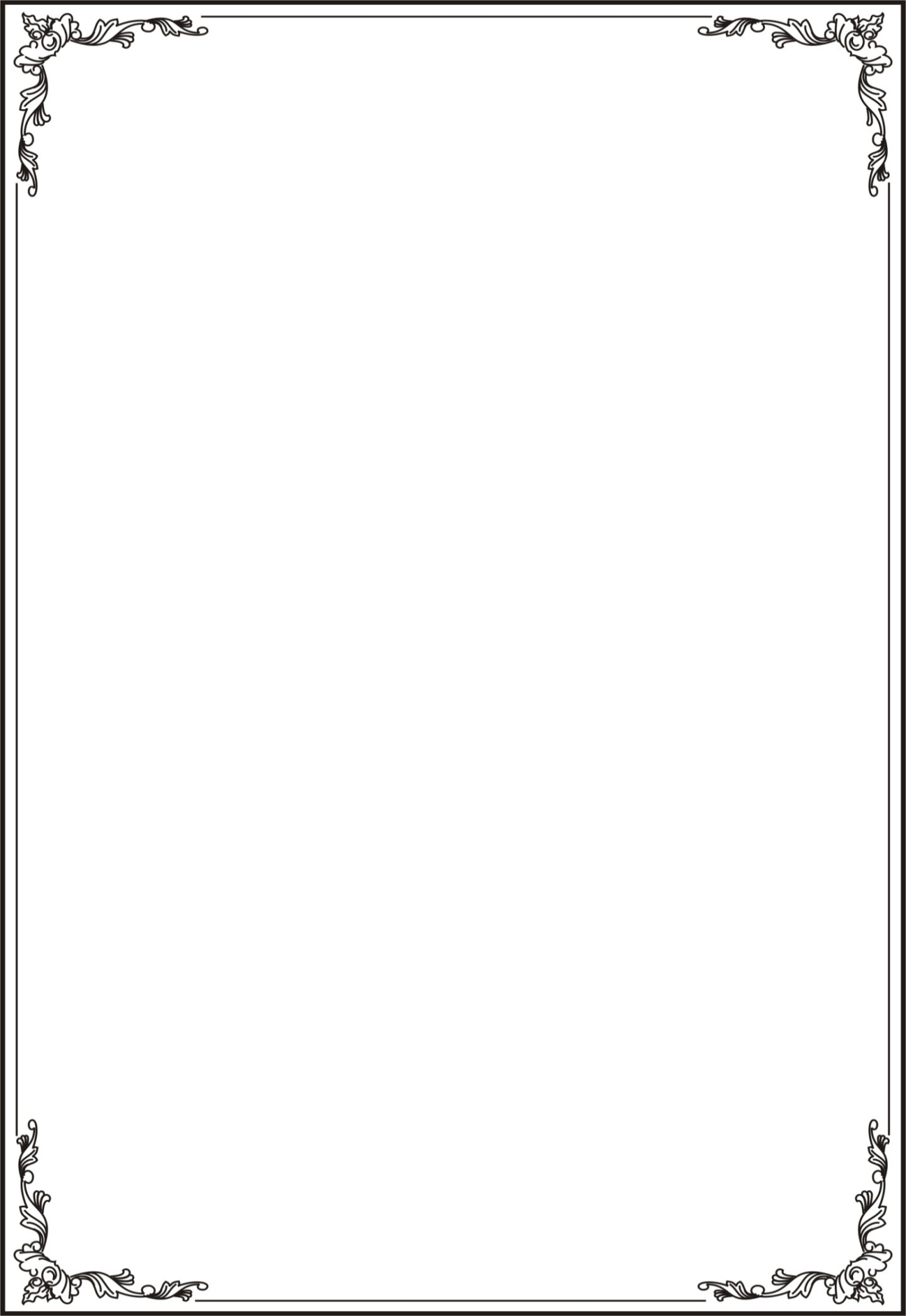
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC THÀNH ĐÔNG**

----------o0o----------



**ĐỒ ÁN TỐT NGHIỆP**

**TÌM HIỂU, NGHIÊN CỨU CÔNG NGHỆ REACTJS – THƯ VIỆN JAVASCRIPT ĐỂ XÂY DỰNG GIAO DIỆN NGƯỜI DÙNG, ỨNG DỤNG TẠO MỘT APP TODOLIST.**

Giảng viên hướng dẫn : **Th.SHoàng Anh Tuấn**

Sinh viên thực hiện : Vũ Duy Tạ

Lớp : **CNTTK10**

*Hải Dương, ngày 21 tháng 10 năm 2017*

# 

# LỜI CẢM ƠN

|  |  |
| --- | --- |
|  | *Hải Dương, tháng 10 năm 2020*  Sinh viên thực hiện  **Vũ Duy Tạ** |

# MỤC LỤC

# DANH MỤC CÁC HÌNH

# DANH MỤC CÁC BẢNG

# 

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| REACTJS | Công cụ phát triển web ứng dụng |
| JSX | cú pháp mở rộng dành cho ngôn ngữ JavaScript viết theo kiểu XML |
| NPM | công cụ tạo và quản lý các thư viện lập trình Javascript |
| HTML | Ngôn ngữ siêu văn bản |
| STT | Số thứ tự |
| JS | Thư viện web Javascript |
| Single-way data flow | Luồng dữ liệu một chiều |

# LỜI MỞ ĐẦU

## 1. Tên đề tài

**Tìm hiểu, nghiên cứu công nghệ reactjs – Thư viện JavaScript để xây dựng giao diện người dùng, Ứng dụng tạo một App TodoList.**

## 2. Lý do chọn đề tài

Với các công nghệ web đã khá phổ biến trước đó như html, jquery,…, Developer phải ứng dụng và [học lập trình](https://nordiccoder.com/) rất nhiều đoạn code khác nhau. Khi đó, họ phải sử dụng nhiều framework khác nhau, và luôn gặp thách thức code lại từ đầu nếu xuất hiện bất kỳ sự thay đổi nào từ component. Những gì họ muốn khi đó là một framework có thể phá vỡ những cấu trúc phức tạp, cũng như có thể tái sử dụng lại các đoạn code trước kia nhằm giúp project hoàn thành nhanh hơn.

Và ReactJS xuất hiện đúng lúc, cung cấp ngay giải pháp mà họ đang tìm kiếm. Công nghệ này sử dụng JSX (viết tắt của cụm từ JavaScript XML, giống như một dạng cú pháp mở rộng giúp code nhanh và dễ sử dụng hơn). Công nghệ này cực kỳ hữu dụng trong việc cấu trúc các đoạn code sử dụng trong lĩnh vực trí tuệ nhân tạo, hay đồng bộ các component vào một file chỉ cần xác thực một lần.

Ngày nay, ReactJS đang ngày một phổ biến hơn vì tính đơn giản và linh hoạt. Nhiều người còn cho rằng, đây sẽ là tương lai của ngành web development. Người ta ước tính có khoảng hơn 1.300 developer và hơn 94.000 website đang sử dụng ReactJS.

Sự phổ biến của ReactJS một phần nhờ vào các ông trùm về công nghệ như Facebook, PayPal, Uber, Instagram và Airbnb, sử dụng để giải quyết các vấn đề liên quan đến UI. Từ đó kéo theo làn sóng rất nhiều người và công ty tiếp tục tin tưởng sử dụng ReactJS.

Từ việc nghiên cứu cách thức hoạt động này, sẽ giúp em phát triển được các ứng dụng nhỏ, phục vụ trải nghiệm tốt cho người dùng, nhận thức được tầm quan trọng đó em đã tìm hiểu, nghiên cứu và quyết định chọn đề tài “***Tìm hiểu, nghiên cứu công nghệ reactjs – Thư viện JavaScript để xây dựng giao diện người dùng, Ứng dụng tạo một App TodoList***” làm đồ án tốt nghiệp.

## 3. Đối tượng nghiên cứu

* Nghiên cứu cách sử dụng Thư viện JavaScript để xây dựng giao diện người dùng.
* Tìm hiểu, nghiên cứu công nghệ Reactjs
* Ứng dụng tạo một App TodoList.

## 4. Mục đích nghiên cứu

* Hiểu được cơ chế hoạt động của component trong reactjs
* Nắm bắt và sử dụng được thư viện web javascript
* Sản xuất được một ứng dụng thực tế.
* Tạo thêm nguồn tài liệu cho lập trình ứng dụng web

## 5. Phương pháp nghiên cứu

* Tìm hiểu cách sử dụng công nghệ reactjs trên trang chủ https://reactjs.org/.
* Nghiên cứu thư viện javascript để làm nền tảng cho công nghệ reactjs
* Áp dụng reactjs để làm một app todolist(Ứng dụng quản lý công việc)

## 6. Ý nghĩa lý luận và thực tiễn của đề tài

* Qua việc tìm hiểu và thực hiện làm luận văn, đã giúp em nâng cao khả năng am hiểu về lĩnh vực làm app web.
* Tạo được một ứng dụng thực tế dựa trên những kiến thức nghiên cứu.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Trong chương này em sẽ trình bày cơ sở lý thuyết của 2 nền tảng Javascript, reactjs.

* 1. **Javascript**
     1. **Javascript là gì?**

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

* + 1. Các thư viện Javascript đình đám hiện nay

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được. Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

* **AngularJS**: Một thư viện dùng để xây dựng ứng dụng Single Page
* **NodeJS**: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime
* **Sencha Touch**: Một Framework dùng để xây dựng ứng dụng Mobile
* **ExtJS**: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications)
* **jQuery**: Một thư viện rất mạnh về hiểu ứng
* ReactJS: Một thư viện viết ứng dụng mobie
* Và còn nhiều thư viện khác
  1. Reacjs

1.2.1 Reacjs là gì?

**ReactJS** là một thư viện JavaScript có tính hiệu quả và linh hoạt để xây dựng các thành phần giao diện người dùng (UI) có thể sử dụng lại. **ReactJS** giúp phân chia các UI phức tạp thành các thành phần nhỏ (được gọi là component). Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. **ReactJS** ban đầu được phát triển và duy trì bởi Facebook và sau đó được sử dụng trong các sản phẩm của mình như **WhatsApp & Instagram**.

ReactJS được dùng để xây dựng các ứng dụng [single page application] (SPA). Một trong những điểm hấp dẫn của ReacJS là nó không chỉ được xây dựng bên phía clients mà còn sử dụng được bên phía server.

1.2.2. Các khái niệm cơ bản về Reactjs

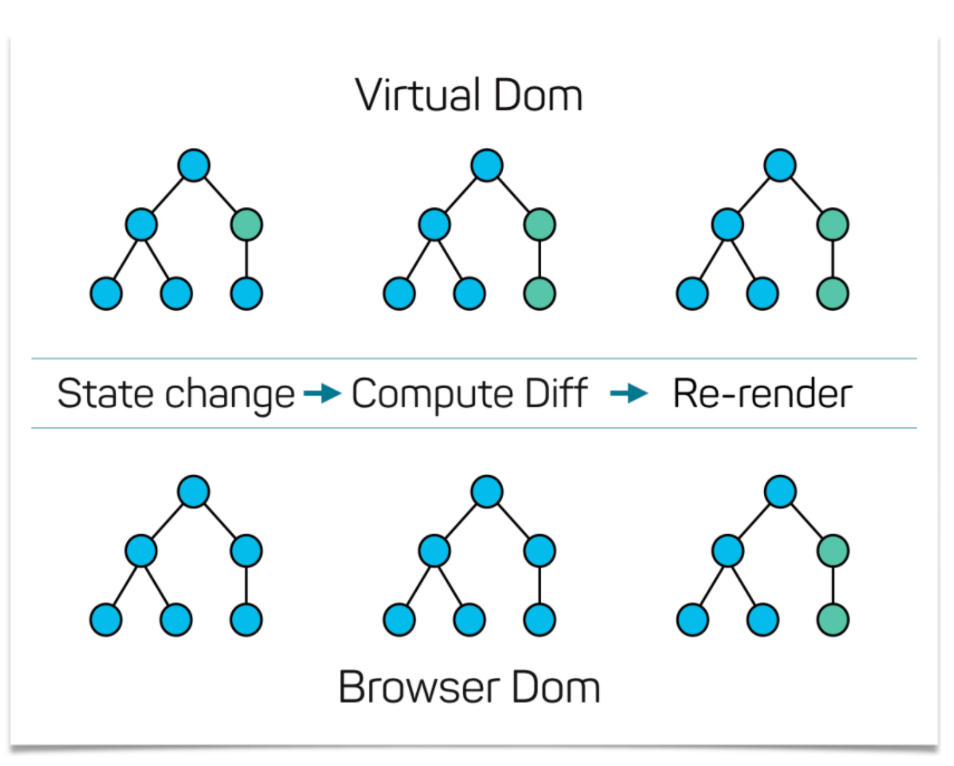
Khi bắt đầu làm quen với **ReactJS**, chúng ta nên làm quen với các khái niệm cơ bản của nó trước, bởi các khái niệm này sẽ đi cùng với chúng ta trong suốt quá trình học tập và làm việc với **ReactJS** sau này.

### Virtual DOM

Để hiểu rõ khái niệm về Virtual DOM, chúng ta cùng nhau đi tìm hiểu về DOM trước. DOM là một Document Object Model và là một cấu trúc trừu tượng của text. Các đoạn mã HTML được gọi là HTML DOM. Mỗi elements trong HTML là các nodes của DOM đó.

Tại sao có DOM rồi lại cần Virtual DOM (DOM ảo)? Khi chúng ta làm việc với một DOM, khi một nodes thay đổi thì tất cả các nodes cũng phải thay đổi thay. Giả sử, chúng ta có một list danh sách gồm 10 items, nếu chúng ta thay đổi 1 items thì DOM cũng thay đổi 9 items còn lại về trạng thái ban đầu của nó.

Điều này là không cần thiết, mặc dù tốc độ xử lý của DOM khá nhanh nhưng đối với các ứng dụng SPA việc thay đổi các DOM này là liên tục nên nó sẽ xảy ra khá chậm và không khả thi đi xây dựng ứng dụng lớn. Lúc này Virtual DOM sẽ được dùng để thay thế. Nó được xây dựng dựa trên DOM thật, có một vài thuộc tính của DOM thật nhưng khi thay đổi Virtual DOM sẽ không thực hiện thay đổi trên màn hình giống như DOM thật.



Khi chúng ta thực hiện render một JSX element, mỗi Virtual DOM object sẽ được cập nhật, khi virtual DOM được cập nhật, **ReactJS** sẽ so sánh virtual DOM với virtual DOM trước đó để kiểm tra trước khi thực hiện cập nhật và sau đó sẽ cập nhật trên một phần của DOM thật. Thay đổi của DOM thật sẽ được hiển thị ra màn hình.

Quay lại ví dụ bên trên, thì lúc này khi chúng ta sử dụng Virtual DOM thì nó chỉ cập nhật duy nhất 1 items, lúc này tài nguyên sẽ được tiết kiệm cũng như tốc độ xử lý cũng nhanh hơn rất nhiều.

### JSX

**JSX** là viết tắt là Javascript XML, nó cho phép bạn viết các đoạn mã HTML trong React một cách dẽ dàng và có cấu trúc hơn. Về cú pháp cũng gần tương tự như HTML, giả sử mình có 1 đoạn mã HTML như sau:

|  |  |
| --- | --- |
|  | <p class="text">Đây là đoạn văn</p> |

Thì trong JSX thì sẽ được viết như này :

|  |  |
| --- | --- |
|  | <p className="text">Đây là đoạn văn</p> |

chỉ cần thay class thành className là xong.

### Components

Khi bạn làm việc với một dự án lớn, UI có độ phức tạp cao chia thành các phần khác nhau. Việc chia nhỏ các thành phần trong UI là một điều cần thiết, các phần nhỏ này được gọi là các components, cho phép render các đoạn mã HTML,... Trong **ReactJS** cách viết components được chia thành 2 loại:

* class components
* function components.

|  |  |
| --- | --- |
|  | //Function component  function Clock(props) {    return (      <div>        <h1>Hello, world!</h1>      </div>    );  } |
|  | //Class component  class Clock extends React.Component {    render() {      return (        <div>          <h1>Hello, world!</h1>        </div>      );    }  } |

### Props và State

## Props là gì?

Props (Properties) là một object được truyền vào một component, mỗi component sẽ nhận vào một props và trả về một [React Element](https://reactjs.org/docs/rendering-elements.html).

Các component có thể giao tiếp với nhau nhờ vào props, có thể giao tiếp giữa component cha – con và ngược lại.

Bạn cũng có thể sử dụng một số lib để có thể quản lý và sử dụng props cho toàn bộ app như Redux, Flux, …

Khi một component con nhận props từ component cha thì nó chỉ có thể đọc (read only) mà không thể chỉnh sửa giá trị của props đó.

Để có thể truyền một props từ cha sang con thì bạn tạo mới một attributes giống như HTML

Bạn có thể xem ví dụ bên dưới

const App = () => <Welcome name="VoManhKien.com" />

Ở ví dụ trên mình đã tạo mới một attributes có tên name và truyền giá trị là VoManhKien.com bây giờ component Welcome đã có thể show props có chứa biến name ra rồi.

## Sử dụng props trong component

Chúng ta cùng đi xét ví dụ bên dưới

class Welcome extends React.Component {

const { name, age, gender } = this.props;

render() {

return <h1>Hello {name}, năm nay {age}, và bạn là {gender}</h1>;

}

}

const element = <Welcome name="Kien" age={18} gender="male" />;

Như phần trước mình đã trình bày thì mình vừa tạo một element

const element = <Welcome name="Kien" age={18} gender="male" />;

truyền vào compoent Welcome biến name, age, gender có giá trị lần lượt bằng Kien, 18, male

Và ở component Welcome mình đã lấy giá trị name, age, male ra từ trong props và hiển thị ra màn hình.

Bạn có thể truyền bất kỳ kiểu dữ liệu nào qua props như string, number, object, array, function, ...

## Lưu ý khi sử dụng Props

Khi sử dụng props thì bạn cần lưu ý [không nên thay đổi trực tiếp props](https://reactjs.org/blog/2015/10/07/react-v0.14.html#new-deprecations-introduced-with-a-warning) bằng cách gán cho nó một giá trị mới props.name = "new value"

Bạn không thể sử dụng những cách thay đổi props như setProps replaceProps …

Nếu bạn muốn thay đổi props thì bắt buộc chỉ có thể thay đổi ở component cha nơi truyền props sang component con. Phần này chúng ta sẽ tìm hiểu trong những phần tiếp theo của khoá học này.

## Ví dụ thực tế

Giả sử mình có component Author, ở component App (file src/App.js) mình sẽ truyền một props name, avatar sang cho component Author. Component Author sẽ hiển thị thông tin ra.

// src/App.js

import React from 'react';

import './App.css';

import Author from './Author';

function App() {

return (

<div className="App">

<Author name="Vo Manh Kien" avatar="https://static.vomanhkien.com/files/20200924102140/kien-vo-e1600923028433.jpg" />

</div>

);

}

export default App;

// src/Author.js

import React from "react";

function Author(props) {

const { name, avatar } = props;

return (

<div>

<img src={avatar} alt="Avatar" width="209" style={{ borderRadius: 100 }} />

<h2>{name}</h2>

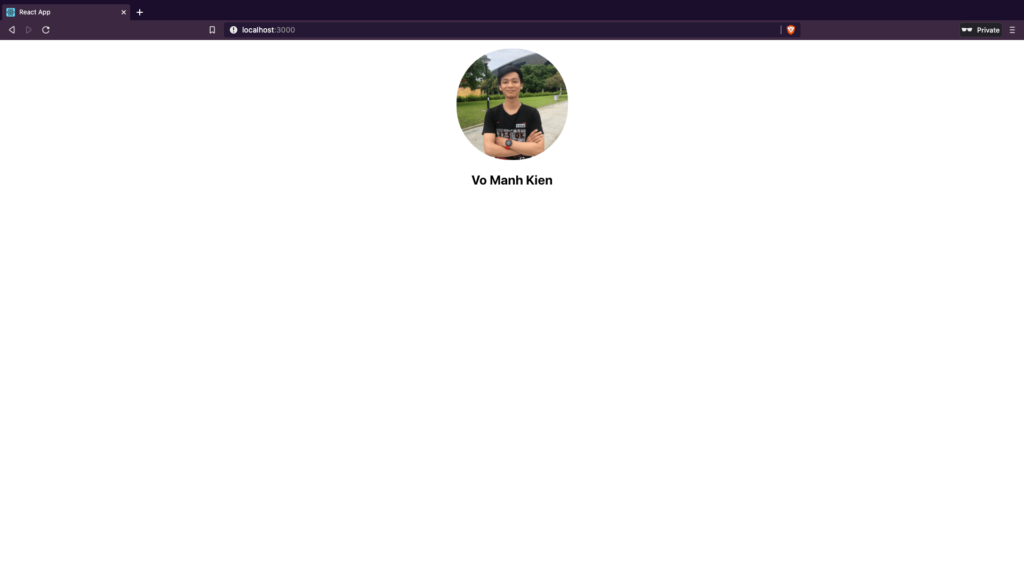
</div>

);

}

export default Author;

Sau khi bạn chạy lệnh yarn start thì sẽ ra kết quả như sau:

Props trong react demo

Bây giờ bạn có thể sử dụng component Author ở bất kỳ đâu, chỉ cần truyền name, avatar vào là được.

Bây giờ giả sử trường hợp mình không truyền tham số name hoặc avatar vào thì chẳng phải sẽ bị lỗi hay sao.

Vậy nên React đã cung cấp cho bạn cách để khai báo props mặc định

Bạn có thể khai báo như sau:

import React from "react";

function Author(props) {

const { name, avatar } = props;

return (

<div>

<img src={avatar} alt="Avatar" width="209" style={{ borderRadius: 100 }} />

<h2>{name}</h2>

</div>

);

}

// Thêm đoạn này vào

Author.defaultProps = {

name: "No Name",

avatar: "default avatar url"

}

// cho đến đây

export default Author;

Cú pháp để khai báo props default đó là tên component.defaultProps trong trường hợp trên thì là Author.defaultProps sẽ bằng với một object props tương ứng.

Như vậy nếu bạn không truyền vào thì nó cũng không bị lỗi mà sẽ có giá trị mặc định để hiển thị

**State** là một obejct mà lưu trữ giá trị của các thuộc tính bên trong components và chỉ tồn tại trong phạm vi của component đó. Mỗi khi bạn thay đổi giá trị của một **state** thì component đó sẽ được render lại.

* Sử dụng state

Để sử dụng được state thì bạn có 2 cách để khai báo

// Cách 1:

this.state = { name : 'VoManhKien.com' }

Với cách 1 thì bạn có thể dùng từ khoá this.state đặt trong một class để khai báo.

// Cách 2:

constructor(props) {

super(props);

this.state = { website: "VoManhKien.com" };

}

Với cách 2 thì bạn có thể để trong thẻ khởi tạo constructor

eact đề xuất bạn nên sử dụng cách 2 để [khởi tạo state](https://reactjs.org/docs/state-and-lifecycle.html#adding-local-state-to-a-class).

Đối với một [function component](https://vomanhkien.com/components-trong-reactjs/#Function_Component) thì bạn phải sử dụng useState() để có thể khai báo một state. Đây là một [React Hooks](https://reactjs.org/docs/hooks-intro.html) chúng ta sẽ học trong những bài sau.

Nếu bạn muốn sử dụng state mới khai báo thì bạn có thể gọi đến object state như this.state.name name ở đây là phần tử trong object bạn đã khai báo ở trên.

Bạn hãy thử log ra xem

console.log(this.state.name) // VoManhKien.com

Bạn có thể xem một ví dụ nhỏ bên dưới, mình khai báo state bên trong constructor và sau đó gọi ra ở bên trong hàm render để sử dụng.

import React from "react";

class App extends React.Component {

constructor(props) {

super(props);

this.state = { name: "VoManhKien.com" };

}

render() {

return (

<div>

<h1>Học ReactJS tại {this.state.name} </h1>

</div>

);

}

}

export default App;

Nếu bạn muốn cập nhật một state thì bạn chỉ cần gọi phương thức this.setState({key: value})

Lưu ý: bạn không thể gán giá trị mới cho state bằng this.state.name = "Kien.pro"

Ví dụ mình cập nhật giá trị name ở trên thì mình sẽ làm như sau:

this.setState({ name: "Kien.pro" })

Ngoài ra bạn cũng có thể lấy giá trị trước của state trước khi cập nhật

this.setState((state) => {

return {

count: state.count + 1,

};

});

}}

Bạn có thể xem ví dụ bên dưới, mình đã làm theo 2 cách setState

import React from "react";

import "./App.css";

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

count: 1,

};

}

render() {

return (

<div className="App">

<p>Count: {this.state.count}</p>

<button

onClick={() => {

this.setState({

count: this.state.count + 1,

});

}}

>

Thêm 1

</button>

<button

onClick={() => {

this.setState((state) => {

return {

count: state.count - 2,

};

});

}}

>

Bớt 2

</button>

</div>

);

}

}

export default App;

## Sự khác nhau giữa Props và State

State và Props có sự khác nhau cơ bản như sau:

* State: State là dữ liệu nội bộ trong một component, cho nên chỉ có thể sử dụng trong component đã khai báo nó mà thôi.
* Props: Dữ liệu đực chuyền qua các component, component cha truyền cho component con và khi component con nhận thì chỉ có thể đọc chứ không thế thay đổi giá trị

## Một số lưu ý khi sử dụng State

Để việc sử dụng state được dễ dàng và tránh gây ra lỗi ngoài ý muốn thì mình sẽ liệt kê một số lưu ý như sau:

* Bạn nên để cấu trúc dữ liệu của state đơn giản nhất có thể, không nên tạo cấu trúc quá lằng nhằng sẽ khó thao tác và ảnh hưởng hiệu năng
* Không thay đổi state một cách trực tiếp
* State được update không đồng bộ nên bạn cần lưu ý khi sử dụng hàm setState cần giá trị từ state trước

## Khi nào nên sử dụng state và khi nào nên dùng props

Vậy khi nào thì nên sử dụng state và khi nào thì nên dùng Props?

Khi bạn chỉ muốn sử dụng dữ liệu nội bộ trong component đó thì có thể sử dụng state, vì nó có thể thay đổi giá trị tuỳ ý

Còn nếu bạn muốn truyền giá trị qua một component khác thì bạn cần dùng đến props

## Ví dụ thực tế

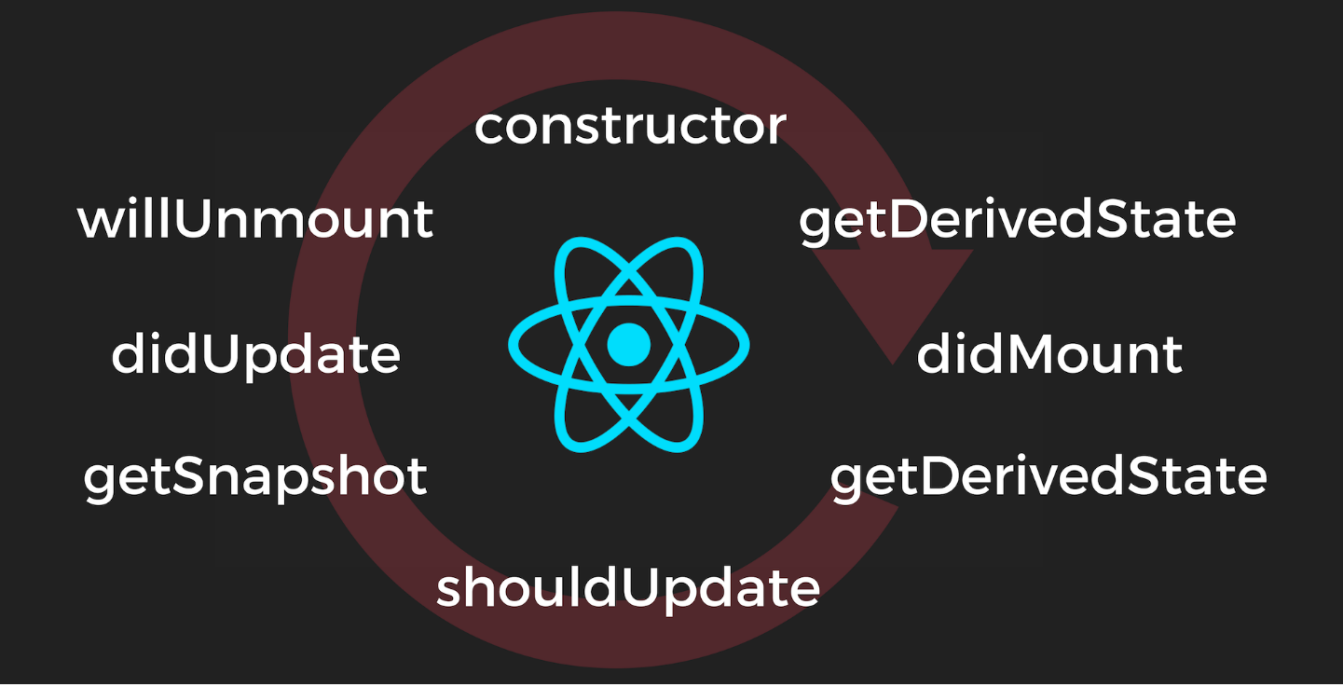
Bây giờ chúng ta sẽ xét một ví dụ thực tế, kết hợp giữa state và props

Vấn đề: Có 2 component App.js và Author.js. Trong component App sẽ chứa state name hiển thị phần Your name: name còn component Author sẽ chứa phần input

Làm sao để khi input ở component Author sẽ có thể update state ở component App

### React Lifecycle

**React Lifcecycle** là một vòng đời của component, khi chúng ta tiến hành render một component thì **ReactJS** thực hiện nhiều tiến trình khác nhau, các tiến trình này được lặp đi lặp lại đối với các component.



Giả sử khi một component được gọi trước tiên nó sẽ cài đặt props và state, sau đó tiến hành mouting, update, unmouting,...việc tham gia vào quá trình này bạn cần sử dụng đến các hàm hỗ trợ của lifcecycle.

# CHƯƠNG 2: CÀI ĐẶT VÀ CẤU HÌNH

## 2.1. Cài đặt môi trường chạy Reactjs

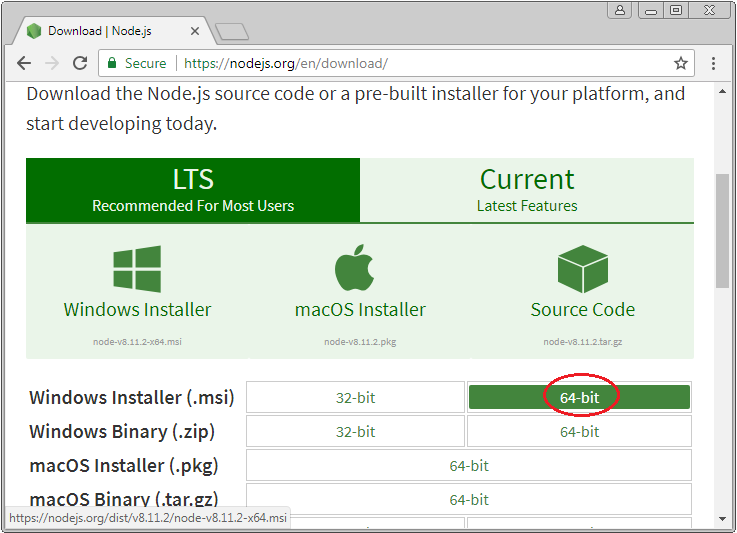
* + 1. Cài đặt NodeJS và NPM

Để cài đặt môi trường chạy ReactJS trước tiên bạn phải cài đặt NodeJS và NPM, nó là một nền tảng bắt buộc. Thông thường NPM sẽ được cài đặt kèm theo khi bạn cài đặt NodeJS.

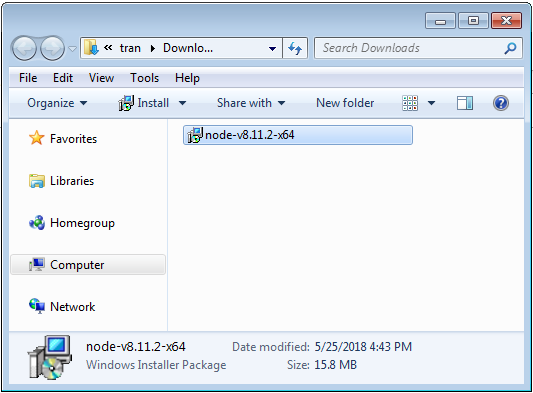
+ Hướng dẫn cài đặt NodeJS trên Window

### Bước 1: Download NodeJS từ trang chủ Nodejs.org

Bạn truy cập vào địa chỉ [nodejs.org](http://nodejs.org/) sẽ xuât hiện một trang web với giao diện như sau:

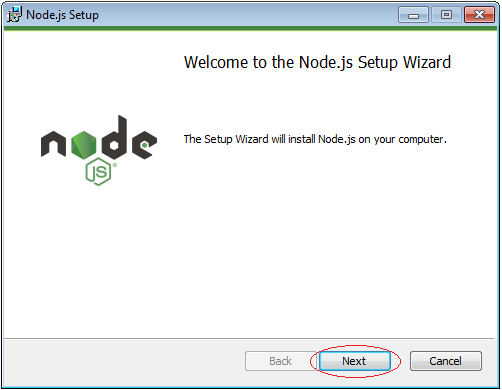


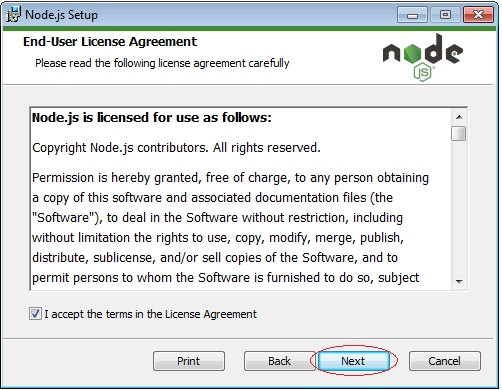
Sau khi download thành công bạn có một file.

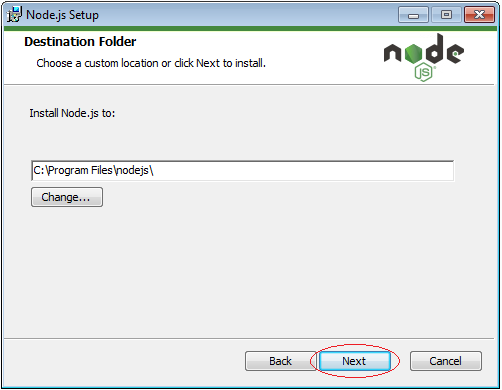


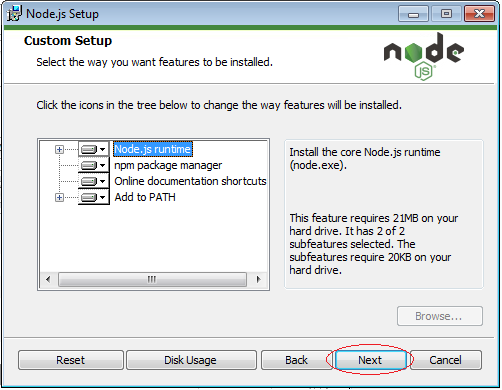
### Bước 2: Cài đặt phần mềm NodeJS

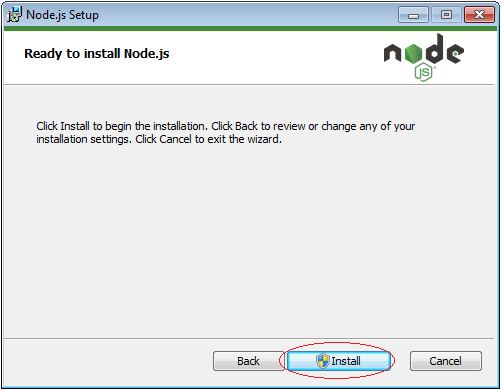
Cài đặt **NodeJS** trên **Windows** rất đơn giản, chấp nhận các tùy chọn mặc định và nhấn **"Next .. Next"** cho tới bước cuối cùng.

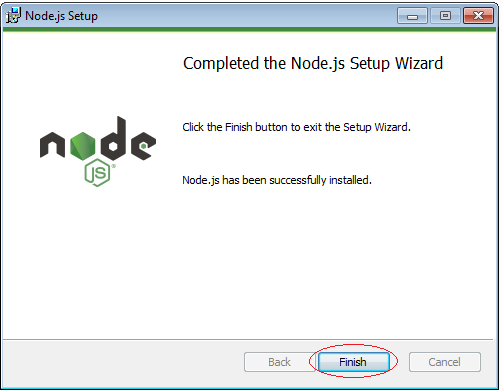










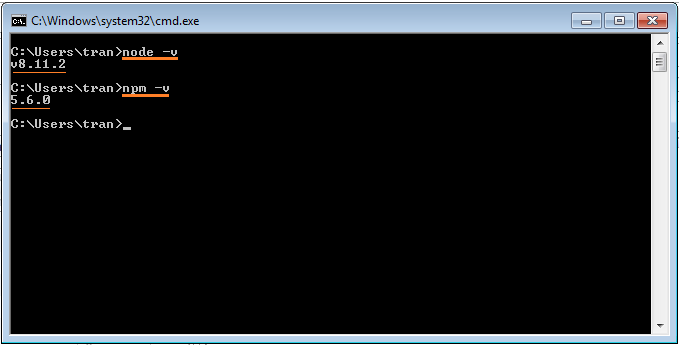


Ở bước trên chúng ta đã download và cài đặt thành công **NodeJS**. Bây giờ bạn cần kiểm tra lại kết quả cài đặt và cấu hình **NodeJS**.

Mở cửa sổ **CMD**và thực thi các lệnh sau để kiểm tra phiên bản của **NodeJS**và **NPM**:

node –v

npm –v



# 

2.1.2 Khởi tạo ReactJS App

Tiếp theo, bạn có thể cài đặt ReactJS bằng cách vào thư mục chứa dự án và mở temrinal và gõ dòng lệnh

npx create-react-app my-app

Trong đó my-app là tên thư mục chứa dự án của bạn, giả sử mình muốn khởi tạo dự án có tên **DoAn** -app, chúng ta sẽ có:

npx create-react-app DoAn-app

Sau khi chạy xong, bạn đã có ứng dụng first-app, cấu trúc thư mục của ứng dụng như sau:

first-app

├── README.md

├── node\_modules

├── package.json

├── .gitignore

├── public

│ └── favicon.ico

│ └── index.html

│ └── manifest.json

└── src

└── App.css

└── App.js

└── App.test.js

└── index.css

└── index.js

└── logo.svg

└── registerServiceWorker.js

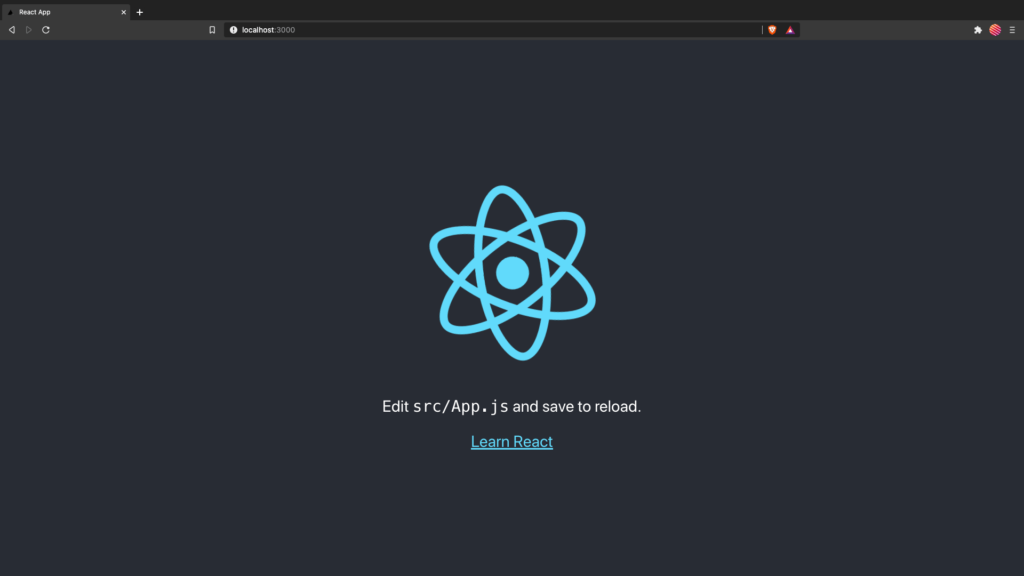
### Chạy ứng dụng

Để chạy ứng dụng, bạn đi đến thư mục DoAn-app và chạy npm start

cd first-app

npm start

Trình duyệt sẽ mở một tab mới hiển thị ứng dụng ReactJS của bạn tại địa chỉ <http://localhost:3000/>



### Sửa ứng dụng

Bây giờ bạn hãy mở VSCode lên vào vào file src/App.js và thử chỉnh sửa vài chỗ trong đó xem sao

Mình sẽ sửa đoạn code thành thế này

import React from 'react';

import logo from './logo.svg';

import './App.css';

function App() {

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

<p>

Welcome to VoManhKien.com

</p>

</header>

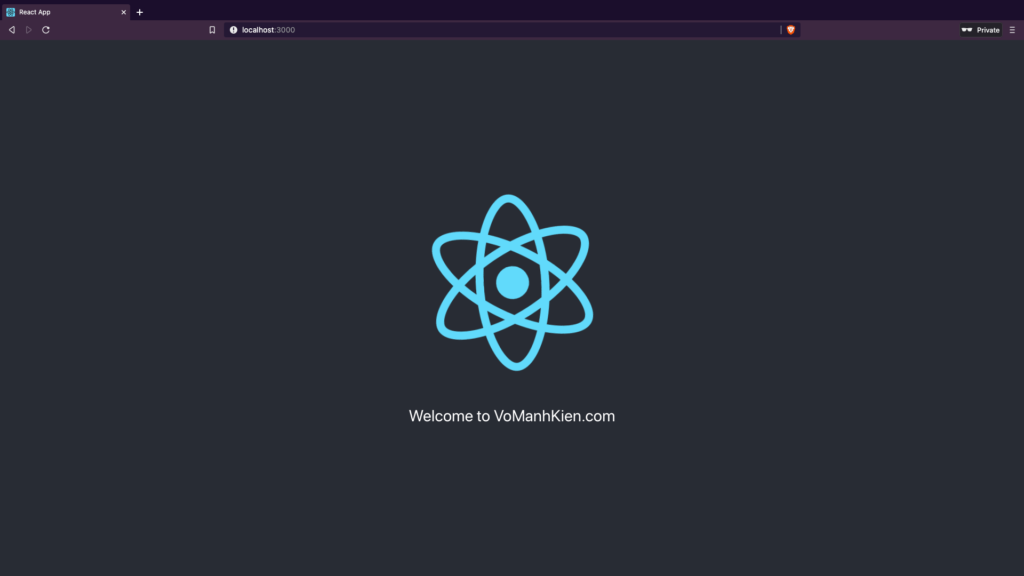
</div>

);

}

export default App;

và kết quả sẽ ra thế này đây



Bạn hãy mở file index.js lên, nó sẽ trông thế này

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import \* as serviceWorker from './serviceWorker';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

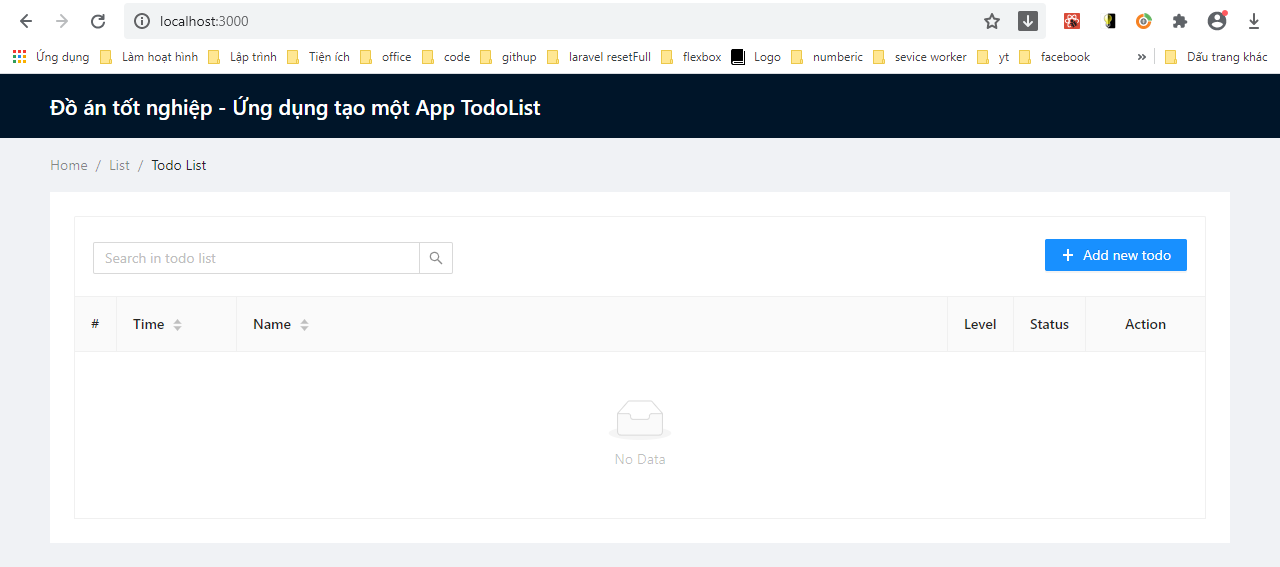
Bạn chú ý đến đoạn document.getElementById('root') và bạn lại mở thêm file public/index.html lên sẽ có đoạn <div id="root"></div>. Đây chính là vị trí mà toàn bộ code của bạn sẽ được ở trong thẻ div này. Bạn không nên sửa tên này làm gì.

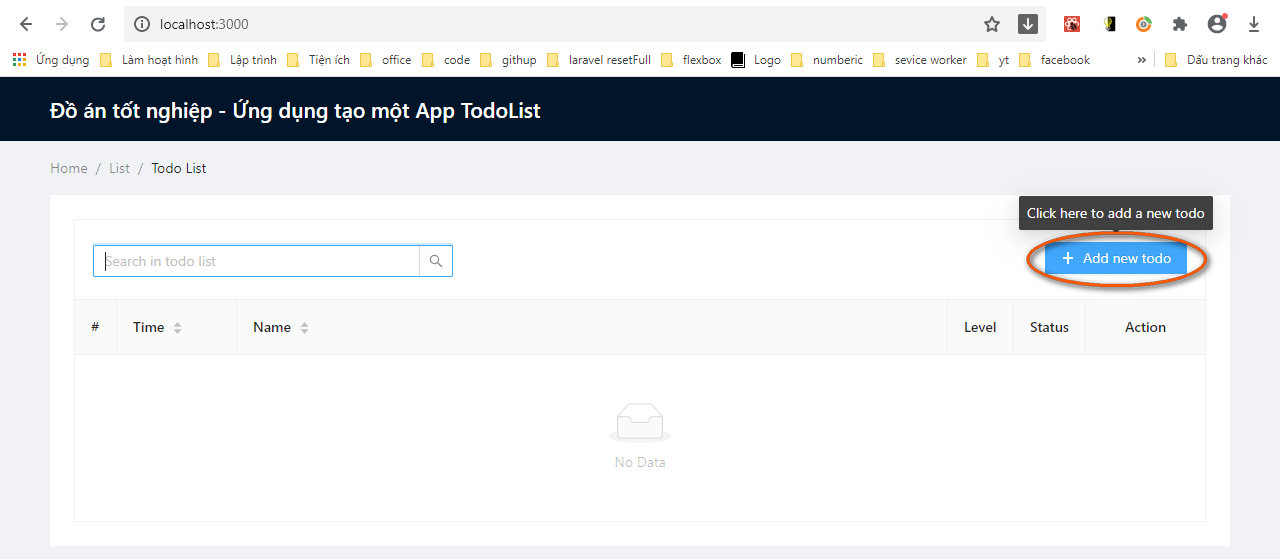
Lưu ý là React đã setup giúp bạn chức năng auto reload tức là mỗi khi bạn sửa code thì trình duyệt sẽ tự động load lại mà bạn không cần phải F5 hay tắt terminal đi bật lại nhé

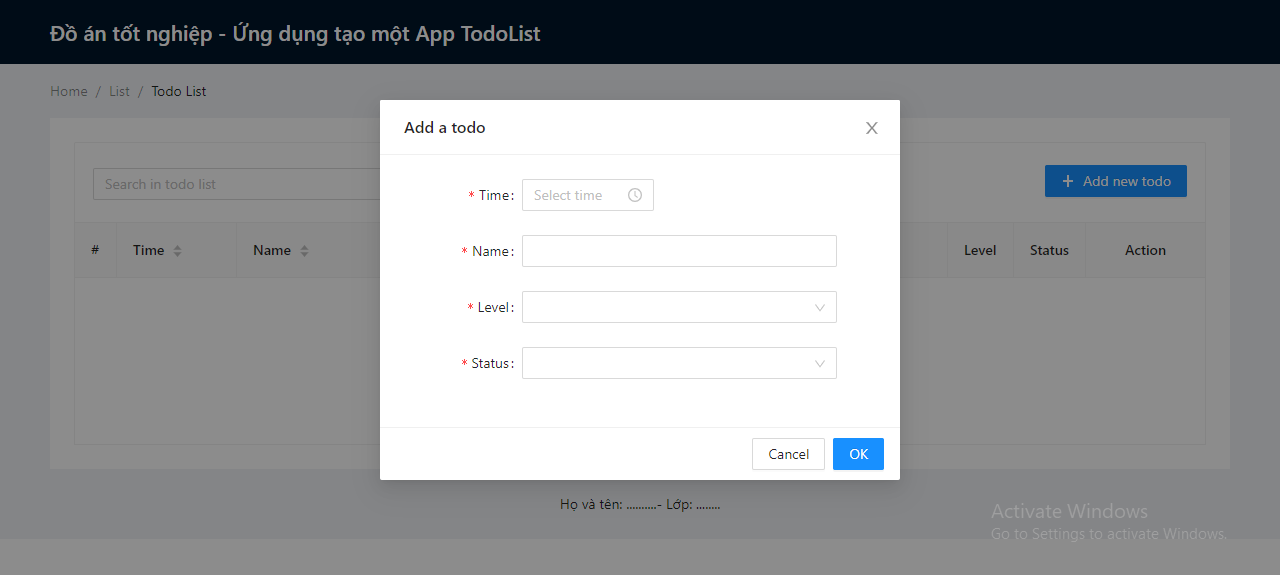
# CHƯƠNG 3: KẾT QUẢ THỰC HIỆN

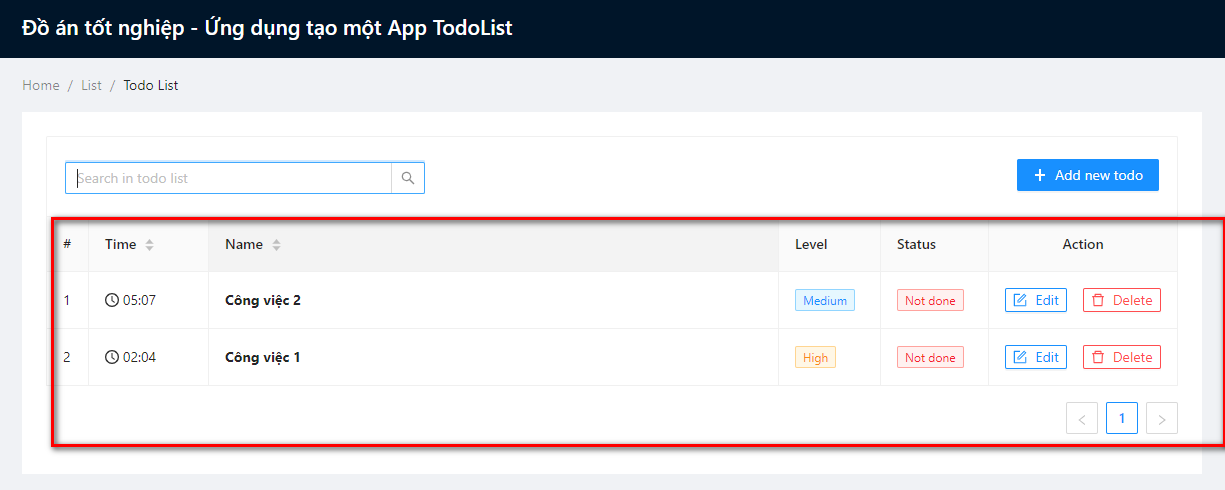
## 3.1 Tạo App list todo

3.2 Giao diện chính của app









# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Kết quả đạt được**
* Đã thực hiện thành công việc gửi thông báo từ firebase tới App học trực tuyến tại 3Dvietpro.
* Lắm bắt được cấu hình để phục vụ cho các app phát triển sau này của bản thân.
* **Hạn chế**
* Do chưa có thời gian nên chưa nghiên cứu hết các tính năng khác liên quan tới firebase.
* **Hướng phát triển**
* Mong muốn học viên muốn phát triển các ứng dụng liên quan tới truyền thông báo tới người dùng.
* Nghiên cứu thêm về khả năng gửi dữ liệu từ playfab qua firebase.

# TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Vỵ, *Phân tích và thiết kế hệ thống thông tin quản lý*, NXB Khoa học tự nhiên và công nghệ, Hà Nội – 2007.

[2] Nguyễn Thị Ngọc Mai, *Lý Thuyết Cơ sở Dữ liệu*, NXB Lao Động – Xã Hội; Hà Nội, 2007.

[3] Phạm Hữu Khang, *Quản trị SQL Server*, NXB Lao Động – Xã Hội, Hà Nôi, 2008.

Một số website học ASP.Net MVC:

https://topdev.vn/blog/nhung-tinh-nang-co-ban-cua-firebase/