|  |  |  |
| --- | --- | --- |
| 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 VUE 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**

[BÁO CÁO TÌM HIỂU VỀ VUE JS 1](#_Toc23853306)

[**I.** **Giới thiệu** 1](#_Toc23853307)

[**II.** **Cài đặt để sử dụng** 1](#_Toc23853308)

[**III.** **Rendering** 6](#_Toc23853309)

[**IV.** **Điều kiện vòng lặp** 7](#_Toc23853310)

[**V.** **Xử lý thông tin của người dùng** 8](#_Toc23853311)

[**VI.** **Component** 10](#_Toc23853312)

[**VII.** **Demo tích hợp vào module** 12](#_Toc23853313)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Trang download vue.js 1](#_Toc23853280)

[Hình 2. Cài đặt bằng cdn. 1](#_Toc23853281)

[Hình 3. Cài đặt bằng npm. 2](#_Toc23853282)

[Hình 4. Trang chủ download Nodejs. 2](#_Toc23853283)

[Hình 5. File nodejs 3](#_Toc23853284)

[Hình 6. Giao diện cài đặt nodejs. 3](#_Toc23853285)

[Hình 7. Cài đặt nodejs 4](#_Toc23853286)

[Hình 8. Giao diện hoàn thành cài đặt nodejs. 4](#_Toc23853287)

[Hình 9. Kiểm tra phiên bản nodejs và nmp. 5](#_Toc23853288)

[Hình 10. Cài đặt Vue-CLI 5](#_Toc23853289)

[Hình 11. Hiển thị dữ liệu đơn giản với Vue js. 6](#_Toc23853290)

[Hình 12. Demo v-bind.. 6](#_Toc23853291)

[Hình 13. Điều kiện trong Vue js 7](#_Toc23853292)

[Hình 14. Vòng lặp trong vue js. 8](#_Toc23853293)

[Hình 15. Hiển thị ví dụ vòng lặp. 8](#_Toc23853294)

[Hình 16. Sự kiện tương tác với người dùng trong Vue js. 9](#_Toc23853295)

[Hình 17. Hiển thị v-on. 9](#_Toc23853296)

[Hình 18. Sau khi click vào button. 9](#_Toc23853297)

[Hình 19. Demo v-model. 10](#_Toc23853298)

[Hình 20. Hiển thị v-model. 10](#_Toc23853299)

[Hình 21. Định nghĩa component trong vue js. 10](#_Toc23853300)

[Hình 22. Tạo đỗi tượng của component để hiển thị. 11](#_Toc23853301)

[Hình 23. Ví dụ về props. 11](#_Toc23853302)

[Hình 24. Ví dụ về component (HTML). 11](#_Toc23853303)

[Hình 25. Ví dụ về component (Vue js). 12](#_Toc23853304)

[Hình 26. Hiển thị ví dụ component. 12](#_Toc23853305)

# BÁO CÁO TÌM HIỂU VỀ VUE JS

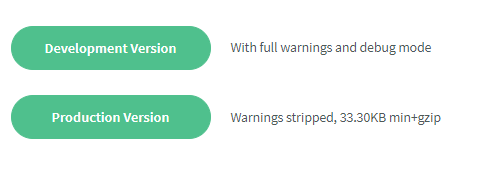
1. **Giới thiệu**

Vue.js là một framework của Javascript, dùng để xây dựng giao diện người dùng. Phần lõi của Vue.js tập trung chủ yếu vào phần view.

1. **Cài đặt để sử dụng**

Để sử dụng được vue js thì trước hết phải cài đặt. Có 4 cách cài đặt là chủ yếu:

**Cách 1:** Sử dụng thẻ **<script>** trực tiếp bằng cách dowload trực tiếp trên trang chủ Vue js về.



Hình . Trang download vue.js

**Cách 2:** Cài đặt bằng **CDN**



Hình . Cài đặt bằng cdn.

Có thể xem được thêm các mã nguồn của package trên npm tại: <https://cdn.jsdelivr.net/npm/vue/>

**Cách 3:** Cài đặt vue js bằng **NPM**

Nên sử dụng npm để dụng khi muốn xây dụng một ứng dụng lớn với Vue.



Hình . Cài đặt bằng npm.

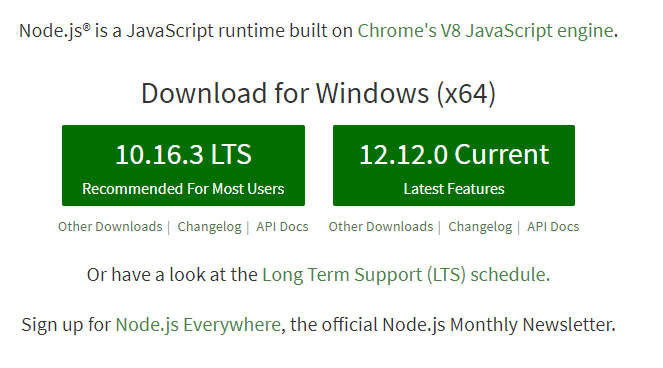
**Cách 4:** Cài đặt **Vue - CLI**

Các công cụ dòng lệnh (CLI – Command Line Interface) hiện là những tính năng không thể thiếu cho một framework, nó giúp xây dựng các template một cách nhanh chóng.

Để cài đặt được Vue-CLI thì trước hết phải cài đặt Nodejs.

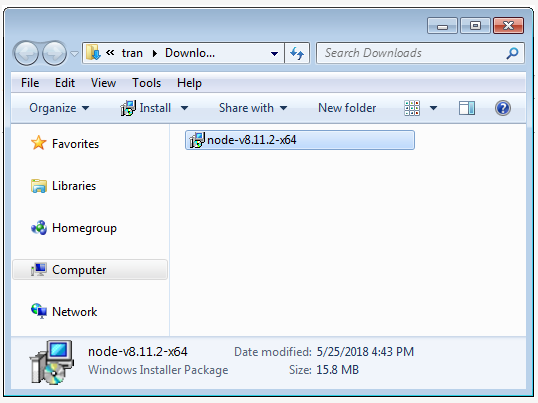
Các bước cài đặt Nodejs.

**Bước 1:** Download nodejs về máy bằng cách vào trang chủ của nodojs có địa chỉ như sau: <https://nodejs.org/en/> và tìm đúng phiên bản thích hợp để download.



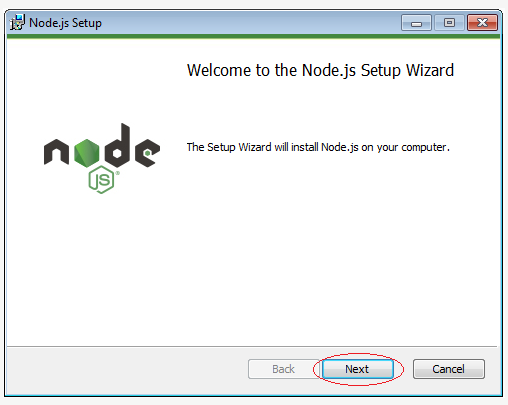
Hình . Trang chủ download Nodejs.

.



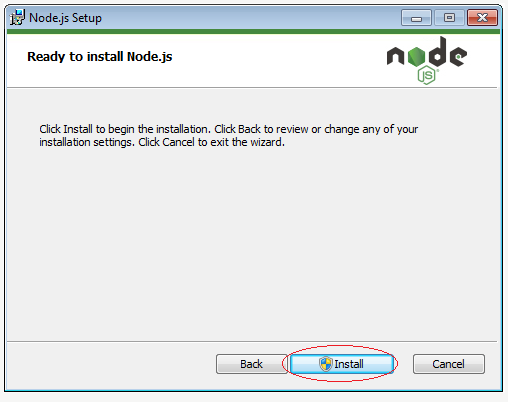
Hình . File nodejs

**Bước 2:** Sau khi download thành công thì tiến hành cài đặt nodejs bằng cách double click vào file vừa download về sau đó cứ làm đúng yêu cầu rồi **Next** để cài đặt.



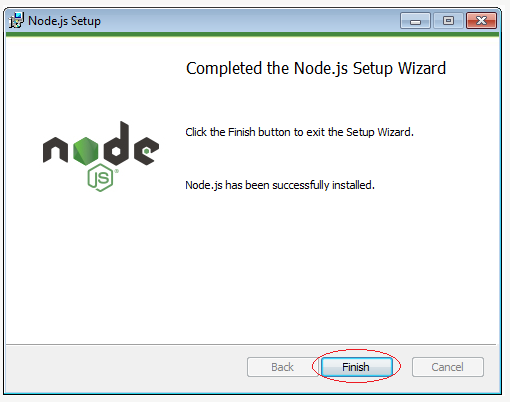
Hình . Giao diện cài đặt nodejs.

**Bước 3:** Nhấn Install để cài đặt nodejs.



Hình . Cài đặt nodejs

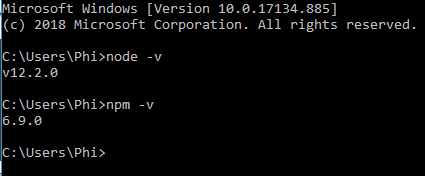
**Bước 4:** Chờ đến khi quá trình cài đặt hoàn tất và nhấn **Finish** để hoàn thành.



Hình . Giao diện hoàn thành cài đặt nodejs.

**Bước 5:** Kiểm tra và cấu hình Nodejs.

Vào cửa sổ **CMD** để thực hiện kiểm tra phiên bản của **Node Js** và **NPM**



Hình . Kiểm tra phiên bản nodejs và nmp.

Để có thể run anywhere với vue-cli hãy install nó global



Hình . Cài đặt Vue-CLI

1. **Rendering**

Lõi của Vue.js là một hệ thống cho phép render dữ liệu đến DOM bằng cú pháp template rõ ràng, đơn giản:



Hình . Hiển thị dữ liệu đơn giản với Vue js.

Dữ liệu của các DOM được liên kết với nhau. Khi sửa app.message=”hello world 12345”trong console thì lập tức chuỗi sẽ chuyển sang thành “hello world 12345”

Ngoài việc quản lí và chỉnh sửa văn bản, cũng có thể bind (ràng buộc) các thuộc tính của phần tử web, như sau:

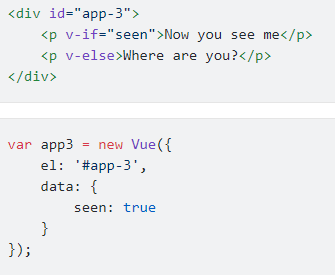


Hình . Demo v-bind..

Khi rê chuột lên chữ “Loading” một vài giây thì sẽ hiển thị thuộc tính title được bind. Thuộc tính v-bind được gọi là một directive trong Vue, các directive sẽ được bắt đầu bằng v- . Trong ví dụ trên, directive về cơ bản đang giữ thuộc tính của title và luôn đồng nhất với đối tượng vue được khởi tạo.

1. **Điều kiện vòng lặp**

Để ẩn hay hiển thị một phần tử trong Vue rất dễ dàng:



Hình . Điều kiện trong Vue js

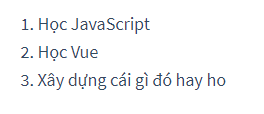
Dòng chữ “Now you see me” nếu như giá trị của seen trong data là true còn ngược lại nếu là false thì sẽ hiển thị dòng chữ “Where are you?”.

Ngoài v-bind hay v-if thì còn có thể v-for dùng để trình bày các giá trị dữ liệu từ một mảng:



Hình . Vòng lặp trong vue js.

Kết quả sẽ hiển thị là:



Hình . Hiển thị ví dụ vòng lặp.

Ta có thể thấy :

+ **todo**: là phần tử thứ x trong mảng.

+ **todos**: là tên mảng.

+ **{{todo.text}}**: là để lấy giá trị trong mảng lên.

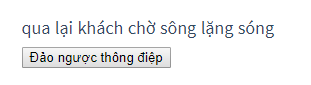
1. **Xử lý thông tin của người dùng**

Để người dùng tương tác với ứng dụng, có thể dùng directive **v-on** để đính kèm các event listener (phần tử lắng nghe sự kiện) và gọi các method (phương thức) trên đối tượng Vue.



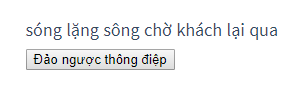
Hình . Sự kiện tương tác với người dùng trong Vue js.

Hiển thị:



Hình . Hiển thị v-on.

Khi click vào button “Đảo ngược ….” Thì chuỗi hiển thị sẽ đảo ngược lại.



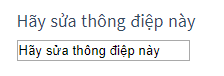
Hình . Sau khi click vào button.

Vue cũng cung cấp directive **v-model** giúp cho việc ràng buộc hai chiều (two-way binding) giữa form input và trạng thái ứng dụng trở nên vô cùng dễ dàng:



Hình . Demo v-model.

Hiển thị:

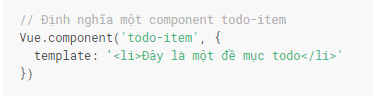


Hình . Hiển thị v-model.

Có thể chỉnh sửa thông điệp trong form input.

1. **Component**

Định nghĩa về component trong Vue js



Hình . Định nghĩa component trong vue js.

Giờ thì bạn có thể sử dụng component todo-item trong template của các c omponent khác:



Hình . Tạo đỗi tượng của component để hiển thị.

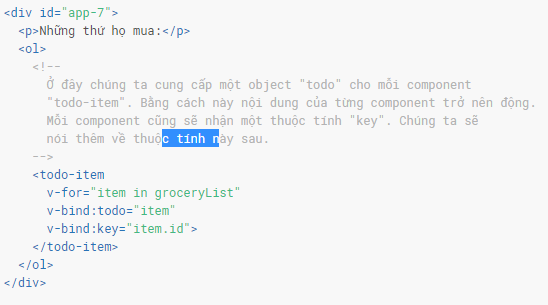
Có điều ví dụ này chưa được thú vị cho lắm, vì các đề mục todo hiện đang giống nhau hoàn toàn. Có thể truyền dữ liệu từ scope (phạm vi) cha vào các component con bằng cách sử dụng một **prop**:



Hình . Ví dụ về props.

Bây giờ có thể truyền **todo** vào mỗi component bằng cách sử dụng **v-bind**:

HTML:



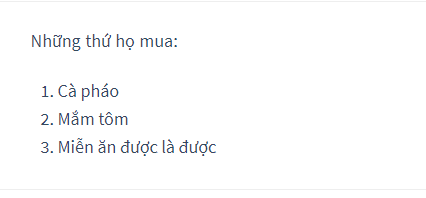
Hình . Ví dụ về component (HTML).

Vue js:



Hình . Ví dụ về component (Vue js).

Hiển thị:



Hình . Hiển thị ví dụ component.

1. **Demo tích hợp vào module**

Các bước thực hiện:

**Bước 1**: Tiến hành cài đặt thư viện vue.js vào module.

Có 2 cách thông dụng để cài đặt vue js là

Cách 1: Sử dụng **CDN** để cài đặt:



Cách 2: Download thư viện **vue.js** từ trang chủ của Vue js về và đặt đường dẫn tới thư mục tải xuống bằng thẻ <script>.



**Bước 2**: Tiến hành tích hợp vue js vào module (Phần Menu)

Phần Menu

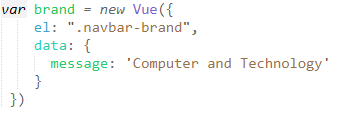


Đoạn code để thay thế cho phần thương hiệu

HTML:



VUE JS



Để hiển thị dữ liệu trong **message: “…”** lên website ta chỉ cần khai báo {{message}} ở vị trí cần hiển thị dữ liệu

Tiếp đến là phần menu sản phẩm

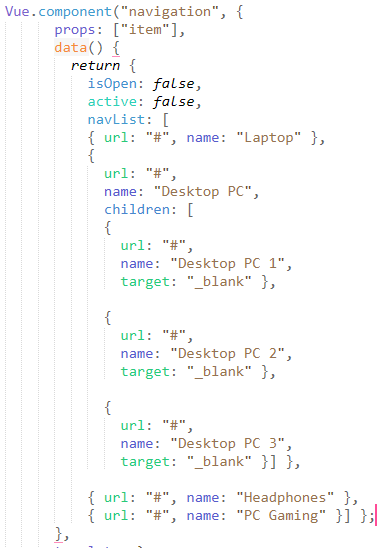
HTML:



Trong đó **navigation** là component được hiển thị bằng vue js.

VUE JS:

Khởi tạo **Component**



Trong đó:

+ **Vue.component** (“navigation”, {/\*Thuộc tính của component\*/}); là dùng để khởi tạo component

+ **isOpen**: Dùng để đại diện cho class **isOpen** để hiển thị menu con khi click chuột vào menu có phần tử con.

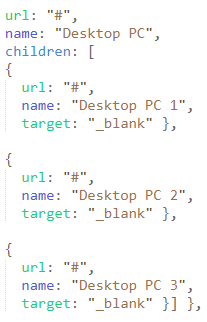
+ **isActive**: Dùng để đại diện cho class **isActive** khi chọn một menu.

+ **navList**: Mảng dữ liệu của menu cha.

+ **url**: là liên kết của menu.

+ **name**: là tên của menu.

+ **children**: là menu con.



Tiếp đến là **Template**



Template đùng để hiện thị đoạn code HTML ra website bằng vue js

Trong đó :

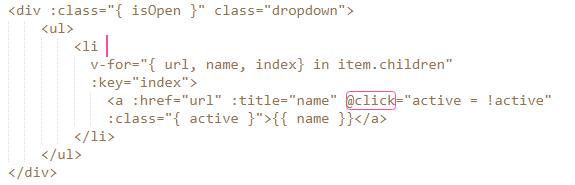
+ **v**-**for** **= “item in navList”:** dùng để chạy vòng lặp dữ liệu trong của phần tử **item** trong mảng **navList.**

**+ v-if** **= “item.children”:** là điều kiện đùng để kiểm tra menu nào có chứa menu con và hiển thị menu con ra website.

**+ :href** **= “item.url”:** trong đó **:href** dùng để thuộc tính **href** trong thẻ <a> còn **item.url** dùng để lấy dữ liệu **url** trong mảng **navList** ra.

+ **:title="item.name":** trong đó **:title** dùng để hiển thị thuộc tính **title** trong thẻ <a> còn **item. name** dùng để lấy dữ liệu **name** trong mảng **navList** ra.

+ **@click="isOpen = !isOpen, active = !active" :class="{ active }">{{ item.name }}:** trong đó @click là sự kiện khi click vào menu dùng để hiển thị menu con hoặc là active menu, còn **{{item.name}}** dùng để hiển thị tên danh mục trong menu



+ Phần danh mục con củng tương tự như danh mục cha.

+ **new Vue ({ el: “#navbar” }); :** dùng để khai báo vị trí cần thêm dữ liệu