|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC LẬP TRÌNH WEB**

**NODEJS**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **AN TOÀN THÔNG TIN**

Giảng viên hướng dẫn : Thầy Dương Thành Phết

Sinh viên thực hiện :

MSSV: 1811770177 Tên: Nguyễn Minh Triết

MSSV: 1811770189 Tên: Võ Khánh Hưng

MSSV: 1811063472 Tên: Trần Văn Nhân

MSSV: 1811770180 Têm: Nguyễn Quang Trường

Lớp: 18DATA1

TP. Hồ Chí Minh, 2021

# **LỜI MỞ ĐẦU**

 Javascript ngày càng trở nên phổ biến hơn với nhiều tính năng và các thư viện được hỗ trỡ cho developer, điều đó khiến cho các giao diện web càng trở nên sinh động hơn. Mọi thứ mà chúng ta có thể làm được trên web ngày nay là Javascript có thể chạy được trên server, cũng như chạy được trên browser, điều này là khó tưởng tượng trong những năm trở lại đây, hoặc nó chỉ đóng gói trong môi trường sandboxed như Flash hoặc JavaApplets. Thật vậy, rõ ràng rằng từ xưa đến nay chúng ta vẫn quan niệm rằng lập trình bên phía server chỉ dùng được những ngôn ngữ như php, ruby, ... nhưng từ khi node js ra đời nó mang đến tư tưởng mới cho việc lập trình cả bên phía server cũng như bên phía client. Và đối với node js thì chúng ta biến những điều mà trước kia chỉ có thể thao tác được bên phía client thì nay cũng thao tác và xử lý được trên server, và ngược lại.

Node js được viết bằng ngôn ngữ javascript, nó là một trình biên đóng gói của Google’s V8 JavaScript engine, libuv platform abstraction layer, và một thư viện lõi được viết bằng Javascript. Mục tiêu của Node js là làm cho web có khả năng push như trong một số ứng dụng gmail. Node js cung cấp công cụ giúp lập trình viên có thể làm việc trong non-blocking, mô hình I/O . Sau hơn 20 năm nghiên cứu, xây dựng và phát triển, nhóm kĩ sư đã cho ra đời sản phẩm ứng dụng web node js chạy thời gian thực và kết nối 2 chiều client và server, cho phép trao đổi dữ liệu một cách tự do.

# **LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành đến Thầy Dương Thành Phết - giảng viên bộ môn “Lập trình Web ” thuộc khoa Công nghệ thông tin, chuyên ngành An toàn thông tin đã trang bị cho chúng em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đề tài nghiên cứu này.

Tuy nhiên trong quá trình nghiên cứu đề tài, do kiến thức chuyên ngành còn hạn chế nên nhóm em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày đề tài. Rất mong được sự quan tâm, góp ý của thầy để đề tài của nhóm em được đầy đủ và hoàn thiện hơn.

Xin chân thành cảm ơn.

### 1. NodeJS là gì?

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Giống như khi bạn lướt Facebook thì mỗi khi bạn comment hay like một topic nào đó thì ngay lập tức chủ topic và những người đã comment trên đó sẽ nhận được thông báo là bạn đã comment.

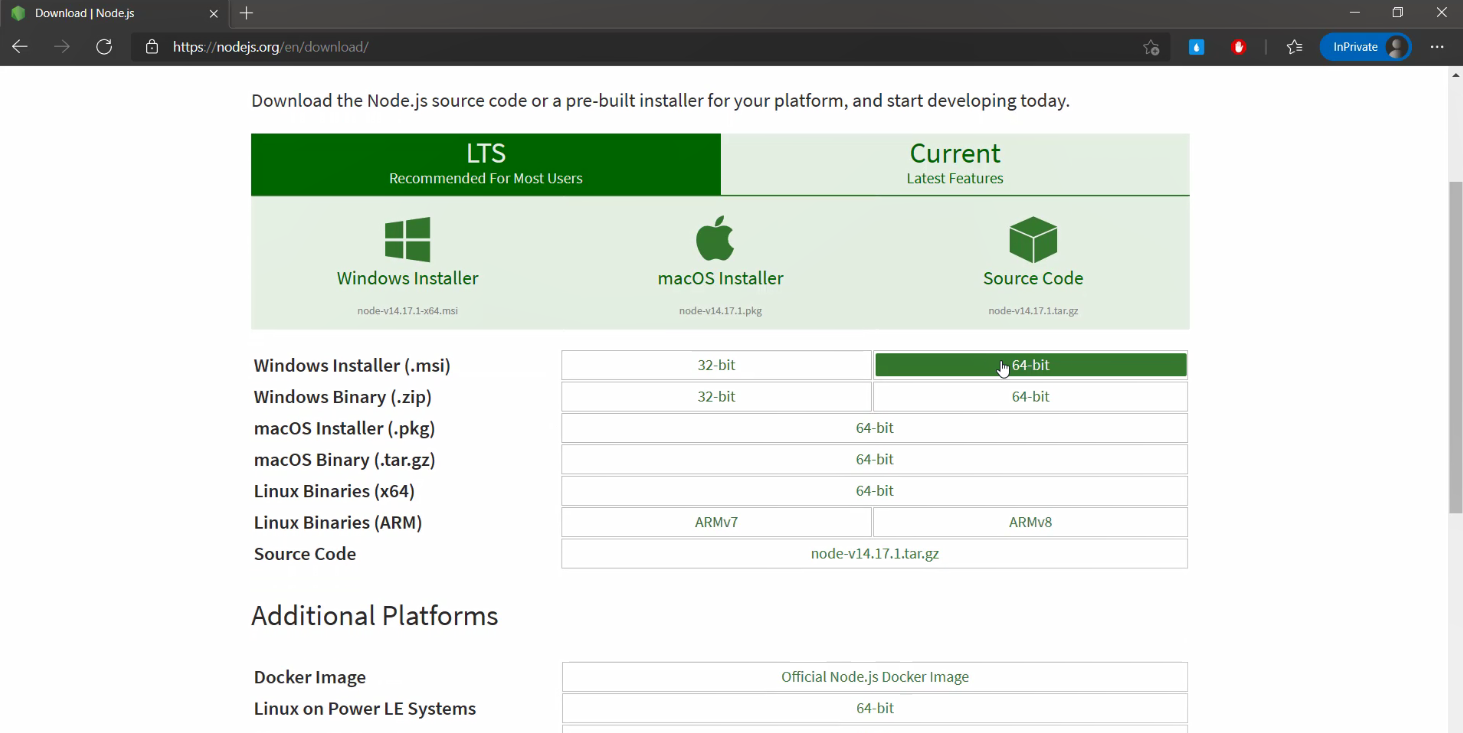
### 2. Các đặc tính của NodeJS

Qua phần tìm hiểu NodeJS là gì mình có giới thiệu một đặc tính rất quan trọng đó là Realtime, tuy nhiên vẫn còn khá nhiều đặc tính mà bạn cần phải biết trước khi học NodeJS.

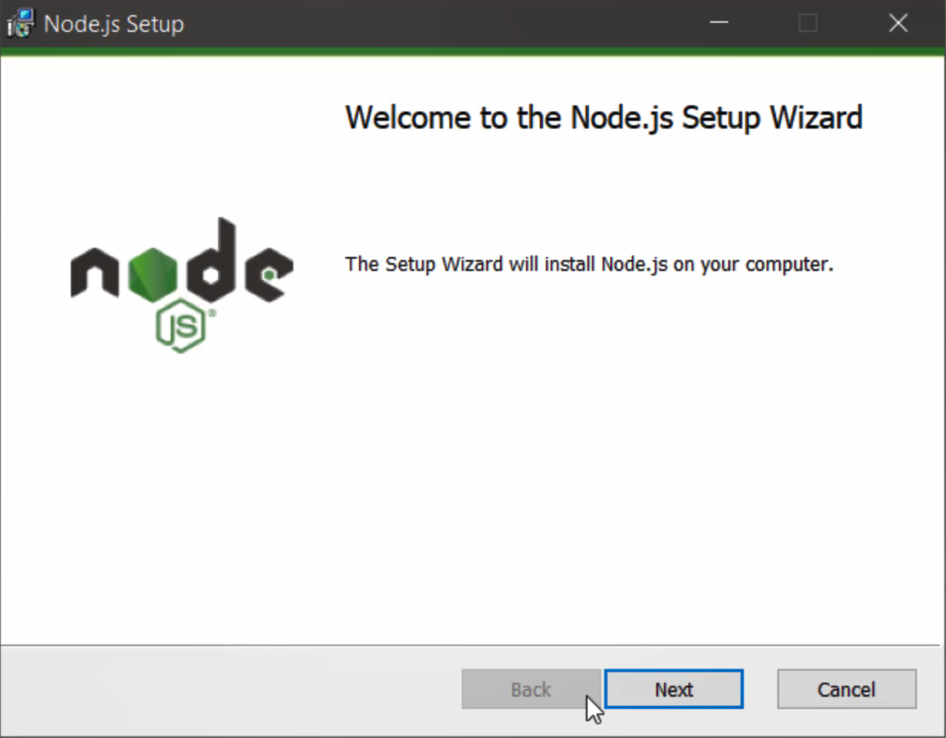
* Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.

3. Cách cài đặt NodeJS

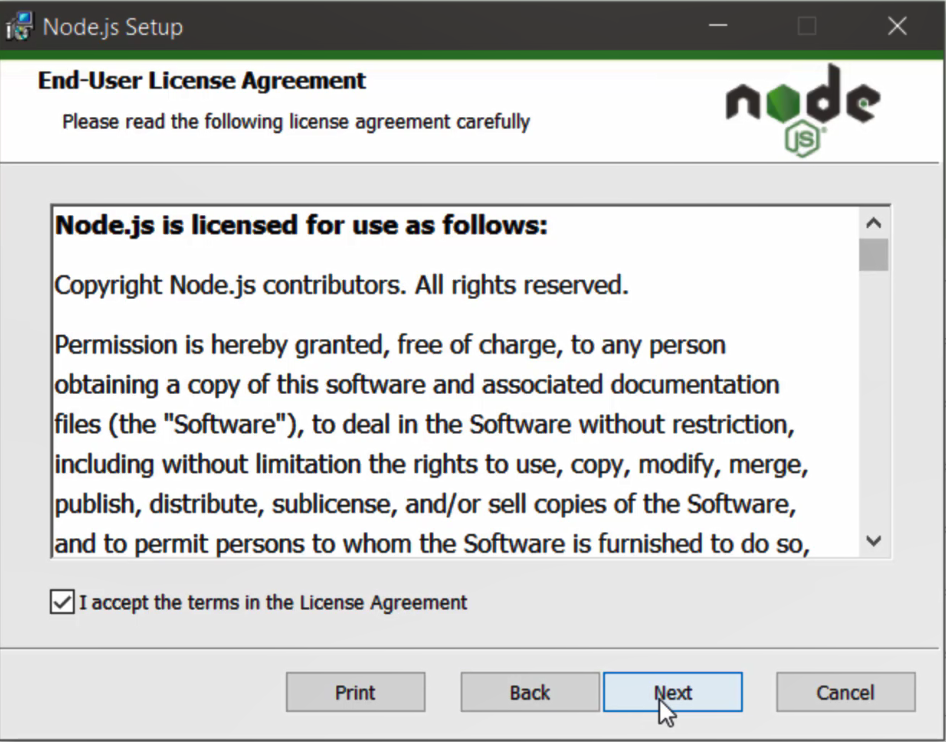
-Bước 1: Truy cập trang nodejs.org. Nhấn vào mục Download và chọn phiên bản cài đặt phù hợp với hệ điều hành. (Khuyến khích sử dụng phiên bản Long Term Support.)



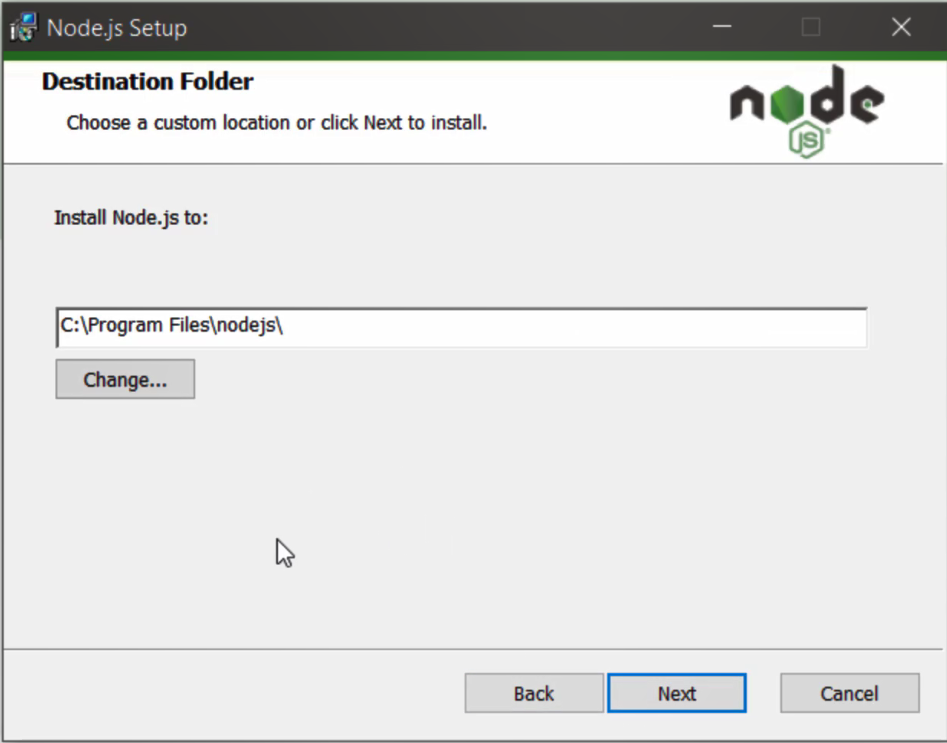
-Bước 2: Mở file cài đặt. Chọn Next để bắt đầu tiến hành cài đặt.



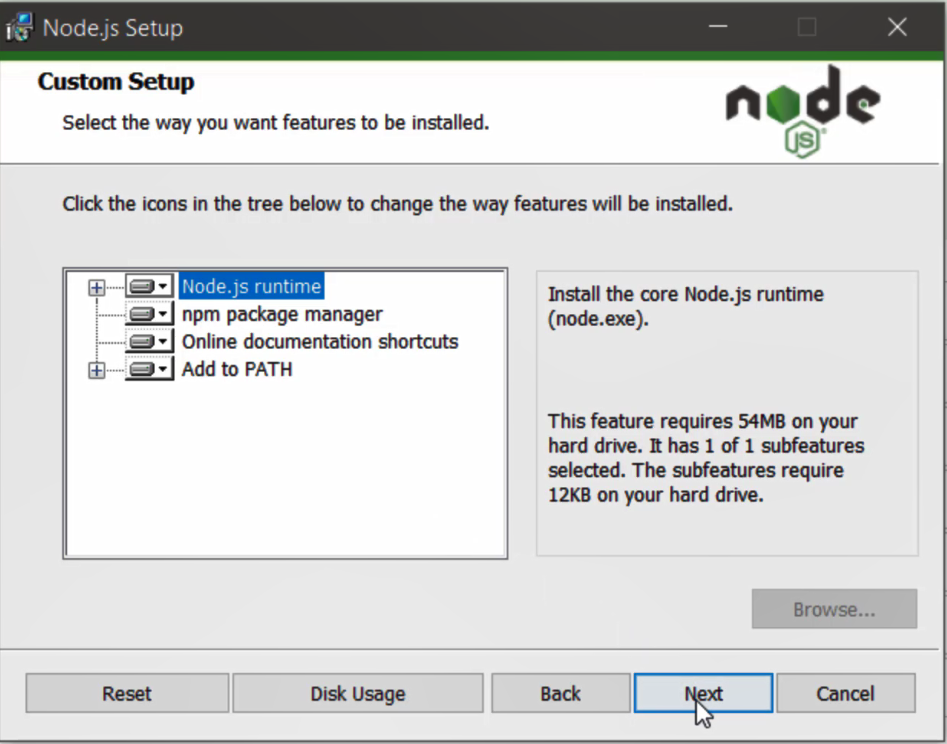
-Bước 3: Check vào ô Đồng ý với điều khoản sử dụng và chọn Next.



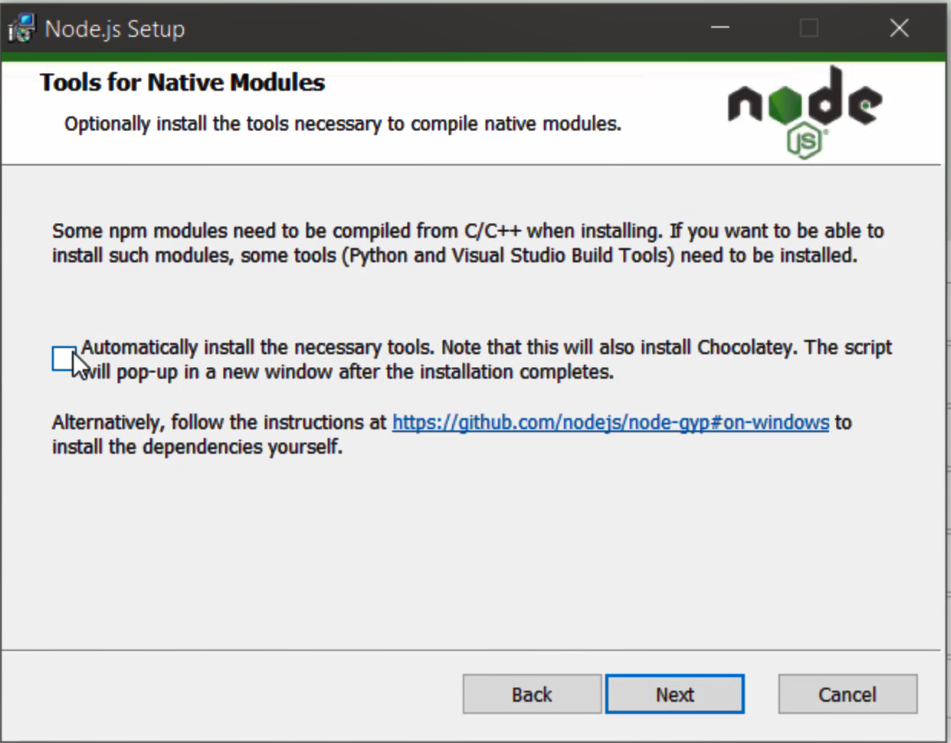
-Bước 4: Chọn đường dẫn đến thư mục chứa.



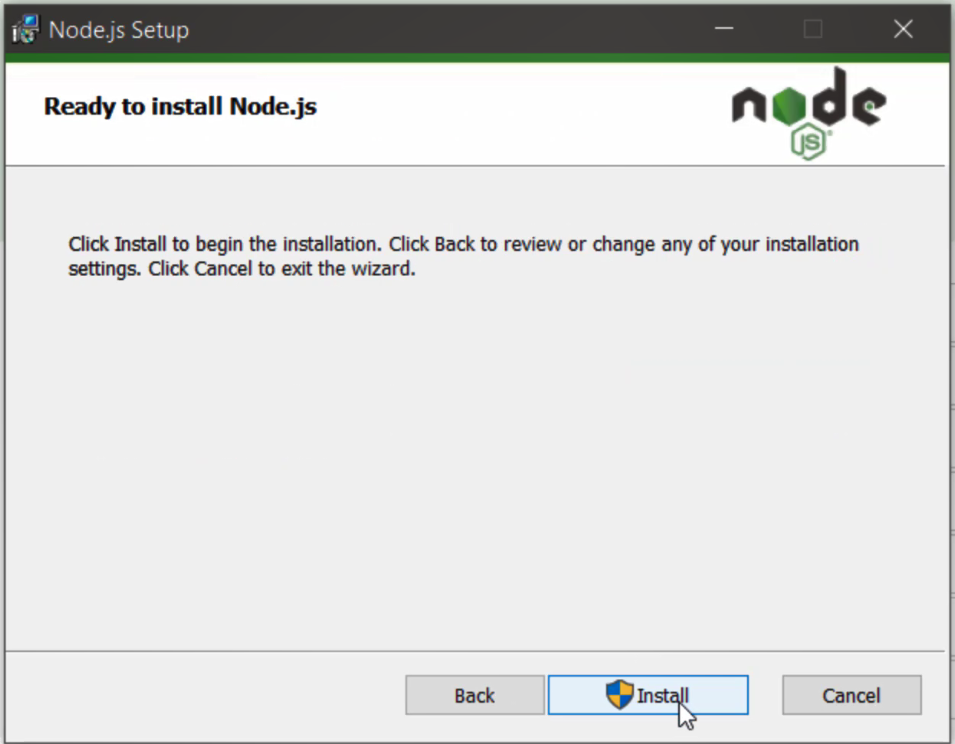
-Bước 5: Tùy chọn các tính năng cần cài đặt. (Khuyến khích chọn tất cả.)



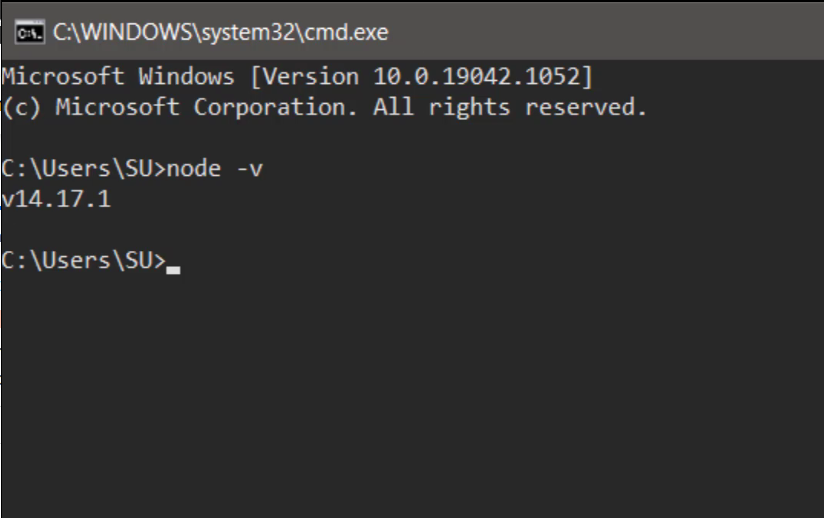
-Bước 6: Check vào ô nếu cần cài đặt thêm công cụ Chocolatey. (Một công cụ quản lí các package trên Windows.)



-Bước 7: Bấm Install để tiến hành cài đặt.



-Bước 8: Kiểm tra kết quả cài đặt bằng cách vào cmd, gõ lệnh “node -v” để xem phiên bản node hiện hành.



### 4. Tìm hiểu mối liên hệ giữa JavaScript và NodeJS

Nền tảng NodeJS là Javascript, khi bắt đầu với lập trình NodeJS chúng ta cần nắm rõ kiến thức về JavaScript để việc bắt đầu với NodeJS dễ dàng hơn. Khi lập trình với NodeJS bạn sử dụng hoàn toàn cú pháp của Javascript, code JavaScript hoạt động được trên Node, nhưng điều ngược lại thì chưa chắc, NodeJS vẫn có tích hợp một số module riêng nên chỉ sử dụng được trong NodeJS chứ trong Javascript không sử dụng được.

#### Khai báo và sử dụng biến:

var hello = 'hello world';

console.log(hello);

Kết quả nó sẽ in giá trị của biến website lên màn hình.

#### Lệnh kiểm tra điều kiện if else:

var hello = 'hello world';

if (hello) {

console.log(hello + ' NodeJS');

} else {

console.log(hello);

}

Kết quả như sau:

#### Vòng lặp:

var array = ['C++', 'Java', 'JavaScript', 'NodeJS'];

for(var i = 0; i < array.length; i++){

console.log(array[i]);

}

Kết quả như sau:

### 5. Tạo Project cho NodeJS

Việc thực hiện quản lý mã nguồn và các thư viện được tích hợp rất quan trọng trong lập trình. Trước đây khi sử dụng các thư viện thì ta phải vào trang chủ download về, sau đó muốn sử dụng Version cao hơn thì ta phải tìm và tải về thêm một lần nữa. Nhưng bây giờ xuất hiện Composer nên công việc đó rất nhẹ nhàng vì chỉ cần gõ vài dòng lệnh là máy tính tự động làm việc đó cho bạn. Trong bài này mình sẽ giới thiệu một chương trình tương tự như Composer đó là npm. NPM là một chương trình quản lý các thư viện tích hợp trong NodeJS, nó được tích hợp sẵn trong gói cài đặt của NodeJS nên khi cài đặt NodeJS là bạn có thể sử dụng được ngay. NPM sử dụng lệnh command line để cài đặt, gỡ bỏ, cập nhật và quản lý các Packages cho ứng dụng NodeJS. Kiểm tra việc cài đặt NPM bằng lệnh

npm --v

#### Tạo file packpage.json bằng lệnh npm init

Để tạo file package.json thì bạn mở NodeJS Command Prompt lên và nhập vào lệnh npm init, sau đó nhập dữ liệu tương với các dòng thông báo rồi nhấn Enter. Trường hợp bạn muốn để tên mặc định thì chỉ cần nhấn Enter là được. Và đây chính là cấu trúc của file package.json.

Trong đó:

* name: Tên của Project
* version: Version của Project
* description: Mô tả cho Project
* main: File chạy chính (chạy đầu tiên) của Project
* scripts: Danh sách các khai báo cấu hình bổ sung cho npm. Như trong chuỗi trên thì giá trị của test chính là câu thông báo và dừng chương trình khi bị lỗi.
* author: Tên tác giả của Project
* license: License của Project, giá trị mặc định là ISC.

#### Thêm Package cho NodeJS bằng lệnh npm install

Để thêm package cho Project NodeJS thì bạn sử dụng cú pháp sau trong cửa sổ NodeJS Prompt.

npm install package-name@version --save

Ví dụ: Cài đặt thêm gói node-persist version 0.0.6 vào NodeJS.

Bạn gõ lệnh sau:

npm install node-persist@0.0.6 --save

Kết quả:

Node-persist là một Module được xây dựng dành cho NodeJS, đây là Module có công dụng tương tự như LocalStorage trong HTML5 và Sesion trong PHP. Node-persist không sử dụng database để lưu trữ dữ liệu mà thay vào đó nó sẽ lưu vào một file trong hệ thống hoặc trong bộ nhớ với nội dung là chuỗi JSON hoặc file (có tên theo key). Vì dữ liệu lưu trữ trong bộ nhớ RAM hoặc ổ đĩa (disk) nên tốc độ xử lý dữ liệu lưu trữ của node-persist không kém phần lưu trữ trong database. Node-persist sử dụng phương thức localStorage trong HTML5 nên việc tiếp xúc nó rất dễ dàng.

#### Cách sử dụng node-persist

Sau đây là các hàm được tích hợp sẵn trong node-persist mà bạn có thể sử dụng. Trước khi sử dụng một module nào thì bạn phải sử dụng hàm require để tạo đối tượng module đó.

var storage = require('node-persist');

Hàm khởi tạo: Trước khi sử dụng thì ta phải thiết lập hàm khởi tạo để nó load tất cả các key lưu trữ trong ở cứng.

storage.initSync();

// hoặc

storage.init().then(promise);

Hàm get: Hàm get dùng để lấy giá trị của một key nào đó, nếu key không tồn tại thì nó sẽ trả về undefined.

storage.getItem('domain');

// hoặc

storage.getItemSync('domain');

Hàm set: Hàm set dùng để thiết lập giá trị cho một key nào đó.

storage.setItem('domain', 'test1');

// hoặc

storage.setItemSync('domain', 'test1');

Hàm remove: Hàm remove dùng để xóa một key nào đó.

storage.removeItem('domain');

// hoặc

storage.removeItemSync('domain');

Hàm clear: Hàm clear dùng để xóa tất cả các key trong bộ nhớ và ổ đĩa.

storage.clear();

// hoặc

storage.clearSync();