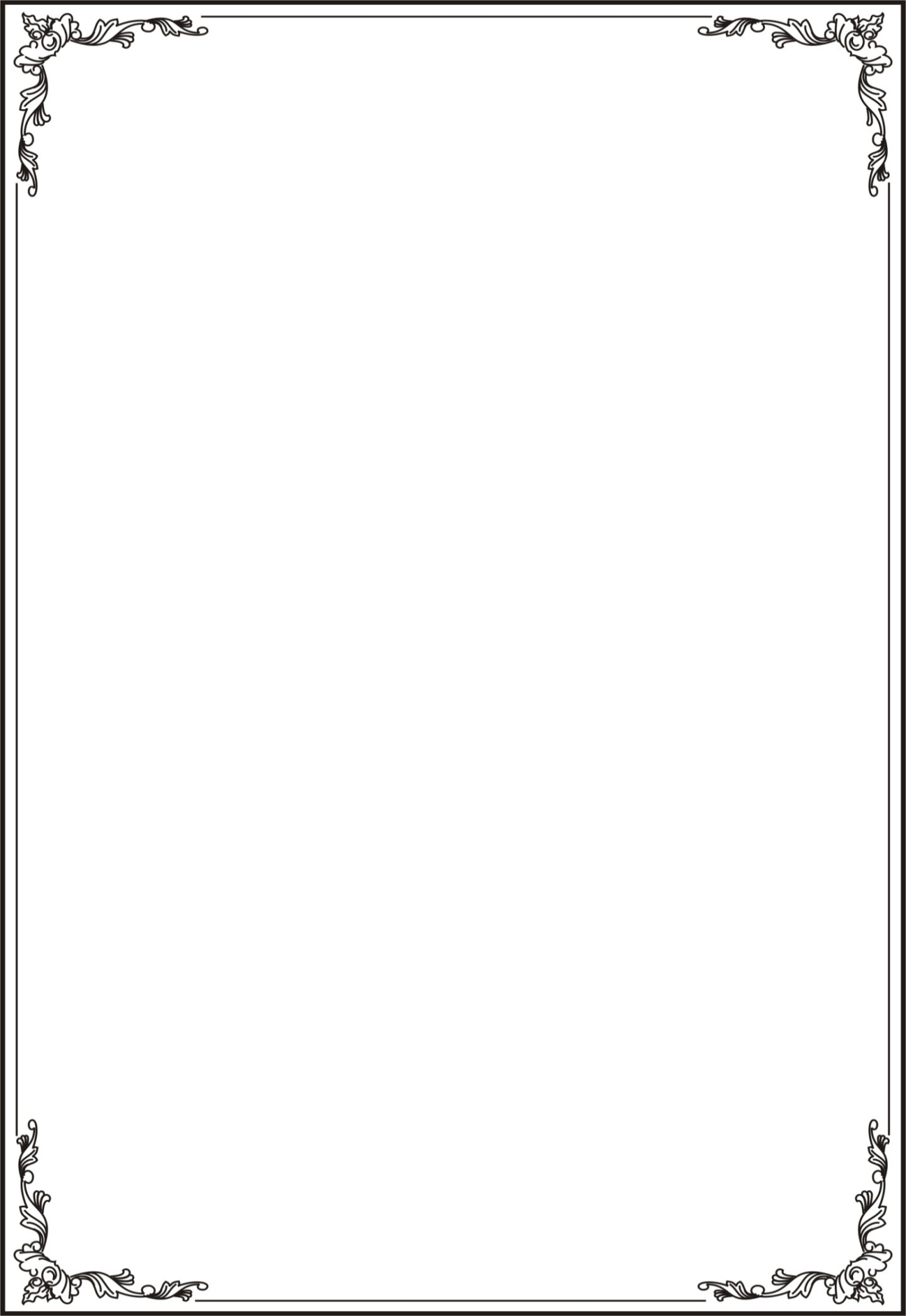
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC THÀNH ĐÔNG**

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



**ĐỒ ÁN TỐT NGHIỆP**

**TÌM HIỂU, NGHIÊN CỨU NỀN TẢNG NODEJS – TRÌNH THÔNG DỊCH THỰC THI MÃ JAVASCRIPT, ỨNG DỤNG TẠO MỘT APP CHATBOX**

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

# 

# LỜI MỞ ĐẦU

## Tên đề tài

***Tìm hiểu, nghiên cứu nền tảng nodejs – Trình thông dịch thực thi mã JavaScript, Ứng dụng tạo một App Chatbox***

1. **Lý do chọn đề tài**

Javascript đang dần trở nên phổ biến, đi kèm với nó là rất nhiều sự thay đổi, khiến cho bộ mặt của ngành phát triển web trở nên lung linh hơn. Javascript bây giờ đã xuất hiện trên cả phía server-side, cùng với sức mạnh của nó ở phía client-side vốn dĩ đã rất mạnh mẽ, tạo nên 1 xu hướng mới, trào lưu mới trong lập trình web. Nếu so với cách đây chỉ vài năm thôi, thật khó để tưởng tượng được rằng thế giới website vẫn còn bị quanh quẩn trong những môi trường sand-box như Flash hay Java Applets

Những thuận lợi của việc sử dụng Javascript xuyên suốt hệ thống web từ server-side đến client-side ([across the stack](https://www.toptal.com/javascript/guide-to-full-stack-javascript-initjs)) với khả năng hợp nhất giữa ngôn ngữ và định dạng dữ liệu (JSON). Khả năng này cho phép các nhà phát triển tái sử dụng tài nguyên 1 cách hiệu quả, cộng thêm hàng tá các tính năng mạnh mẽ khác của Javascript giúp nó, cũng như Node.js, xứng đáng có mặt trong danh sách các kỹ năng của bạn trong tương lai

 Node.js: đi đầu trong việc tạo nên những realtime websites thông qua cơ chế websocket. Cuối cùng thì sau hơn 20 năm của cơ chế web resquest – response, đã xuất hiện ứng dụng realtime web, two way connection (tương tác 2 chiều). Với Node.js, phía client và server có thể bắt đầu giao tiếp với nhau 1 cách đơn giản hơn rất nhiều (do cùng ngôn ngữ), dẫn đến dữ liệu được trao đổi 1 cách thoải mái. Với hệ thống request-response cũ, chỉ phía client là có khả năng gửi thông điệp đến server. Trước đây Flash và Java-Applets thống trị trên nền web, tuy nhiên về cơ bản chúng chỉ là các sandbox-enviroment (môi trường hộp cát), sử dụng trang web như là giao thức truyền tải dữ liệu đến đích là các client. Thêm nữa, việc chúng có cơ chế chạy độc lập và được điều khiển qua các cổng không chính thống (non-standard port) làm cho chúng ngốn nhiều tài nguyên hơn, yêu cầu hệ thống nhiều hơn.

Nhận thức được tầm quan trọng đó em đã tìm hiểu, nghiên cứu và quyết định chọn đề tài “***Tìm hiểu, nghiên cứu nền tảng nodejs – Trình thông dịch thực thi mã JavaScript, Ứng dụng tạo một App Chatbox***” làm đồ án tốt nghiệp.

## Đối tượng nghiên cứu

* Nghiên cứu cách sử dụng và vận hành NodeJs vào ứng dụng ChatBox.
* Nghiên cứu Framework nổi tiếng Express.js được sử dụng phổ biến để phát triển ứng dụng trên nền Nodejs
* Nghiên cứu đối tượng NPM - bộ quản lý đóng gói nổi bật của Node.js
* Nghiên cứu javascript ngôn ngữ của NodeJs.
* Nghiên cứu công cụ Socketio giúp thực hiện những ứng dụng realtime trong NodeJs

## Mục đích nghiên cứu

Em nghiên cứu Nodejs nhằm những mục đích sau:

* Tiếp cận hướng phát triển mới trong việc thiết kế các ứng dụng web đa luồng đa dữ liệu cập nhật realtime
* Nâng cao khả năng tư duy mới cho nền tảng web html realtime
* Sản xuất ứng dụng mang tính thực tế và thân thiện với người dùng trong tương lai
* Tạo thêm nguồn tài liệu cho lập trình ứng dụng realtime website

## Phương pháp nghiên cứu

* Khảo sát thực tế.
* Thu thập các yêu cầu từ phía người dùng.
* Phân tích thiết kế hệ thống theo yêu cầu của người dùng.
* Nghiên cứu cách thức quản lý các soket, các thành phần của nodejs.

## 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 lập trình ứng dụng web socket với Nodejs.
* Nghiên cứu và làm ứng dụng chatbox realtime.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Trong chương này em sẽ trình bày cơ sở lý thuyết của 4 nền tảng NPM, Javascript socketio, Express, giúp cho việc tạo một App ChatBox trong NodeJs.

* 1. **NPM**
     1. **NPM là gì?**

 NPM viết tắt của từ Node Package Manager là một công cụ tạo và quản lý các thư viện javascript cho Nodejs. Những ai làm việc với javascript chắc hẳn ít nhiều cũng đã từng nghe về nó.

NPM cung cấp 2 chức năng chính bao gồm:

* Là kho lưu trữ trực tuyến cho các package/module. Chúng ta có thể tìm kiếm các package trên [search.nodejs.org](http://search.nodejs.org/).
* Quản lý các module javascript và phiên bản của chúng trong các dự án của chúng ta đơn giản hơn, dễ dàng hơn, tiết kiệm thời gian hơn.
  + 1. **NPM hoạt động như thế nào?**

Nó hoạt động dựa trên hai vai trò:

* Nó là repository được sử dụng rộng rãi để publish project Node.js nguồn mở. Nghĩa là đây là nền tảng trực tuyến – nơi mọi người có thể publish và chia sẻ công cụ được viết bằng JavaScript.
* npm là công cụ dòng lệnh giúp tương tác với các nền tảng trực tuyến, như trình duyệt và máy chủ. Tiện ích này hỗ trợ cài đặt và gỡ cài đặt gói, quản lý phiên bản và quản lý dependency cần thiết để chạy dự án.

Để sử dụng, bạn phải cài đặt node.js – vì chúng được đóng gói cùng với nhau.

Tiện ích dòng lệnh npm cho phép node.js hoạt động chính xác.

Để sử dụng gói, dự án của bạn phải chứa file tên là package.json. Trong gói đó, bạn sẽ tìm thấy metadata cụ thể cho project.

Metadata cho thấy vài điều liên quan đến dự án theo thứ tự sau:

* Tên dự án
* Phiên bản ban đầu
* Mô tả
* Điểm vào
* Kiểm tra các lệnh
* [git](https://www.hostinger.vn/huong-dan/github-la-gi/) respiratory
* Từ khóa
* Giấy phép
* Những phụ thuộc
* devDependencies

Metadata giúp xác định project và hoạt động như baseline để người dùng nhận thông tin về nó.

* + 1. **Tính năng của NPM**

Để chạy được phần mềm NPM hệ thống của bạn cần phải cài đặt NodeJS. Phần mềm NPM được tích hợp sẵn trong bộ cài NodeJS, vì vậy khi cài đặt xong NodeJS bạn sẽ có cả hai.

Vì NPM là một phần mềm cài đặt trên máy tính của bạn nên bạn có thể sử dụng nó để cài đặt các thư viện Javascript từ trên Internet. Để cài đặt một thư viện nào đó, chỉ cần mở cửa sổ CMD và thực thi

npm giúp chúng ta quản lý các thư viện javascript đơn giản hơn, dễ dàng hơn, nhanh chóng hơn, giúp chúng ta tiết kiệm được rất nhiều thời gian trong việc xây dựng nền móng của dự án. Giúp chúng ta tập trung trong việc phát triển những phần chính của dự án

* 1. **Javascript**

**1.2.1 Javascript là gì?**

**JavaScript** là một ngôn ngữ lập trình của [HTML](https://quantrimang.com/html) và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

[JavaScript](https://quantrimang.com/JavaScript) được biết đến đầu tiên với tên Mocha, và sau đó là LiveScript, nhưng công ty Netscape đã đổi tên của nó thành JavaScript, bởi vì sự phổ biến như là một hiện tượng của Java lúc bấy giờ. JavaScript xuất hiện lần đầu trong Netscape 2.0 năm 1995 với tên LiveScript. Core đa năng của ngôn ngữ này đã được nhúng vào Netscape, IE, và các trình duyệt khác.

**ECMA-262** **Specification** định nghĩa một phiên bản chuẩn của ngôn ngữ JavaScript như sau:

* JavaScript là một ngôn ngữ chương trình thông dịch, nhẹ.
* Được thiết kế để tạo các ứng dụng mạng trung tâm.
* Bổ sung và tích hợp với Java.
* Bổ sung và tích hợp với HTML.
* Mở và đa nền tảng.

**1.2.2. Client-side JavaScript**

Client-Side JavaScript là Form phổ biến nhất của ngôn ngữ này. Script nên được bao gồm trong một tài liệu HTML cho việc mã hóa để được thông dịch bởi trình duyệt.

Nghĩa là một trang web không cần là một HTML tĩnh, nhưng có thể bao gồm các chương trình mà tương tác với người dùng, điều khiển trình duyệt, và tạo nội dung HTML động.

Kỹ thuật Client-Side JavaScript cung cấp nhiều lợi thế hơn các CGI Server-Side Script. Ví dụ, bạn có thể sử dụng JavaScript để kiểm tra nếu người sử dụng đã nhập một địa chỉ Email hợp lệ trong một trường Form.

JavaScript code được thực thi khi người sử dụng đệ trình Form, và chỉ nếu tất cả đầu vào là hợp lệ, chúng sẽ được đệ trình tới Web Server.

JavaScript có thể được sử dụng để nắm bắt các sự kiện được khởi tạo từ người sử dụng như nhấp chuột, điều hướng link, và các hoạt động khác mà người dùng khởi tạo.

**1.2.3. Các lợi thế của JavaScript**

Các lợi thế của việc sử dụng JavaScript là:

**Sự tương tác Server ít hơn**: Bạn có thể xác nhận đầu vào (input) người sử dụng trước khi gửi trang tới Server. Điều này làm tiết kiệm lưu lượng tải ở Server, nghĩa là Server của bạn tải ít hơn.

**Phản hồi ngay lập tức tới khách truy cập**: Họ không phải chờ cho một trang web tải lại để thấy xem nếu họ đã quên nhập cái gì đó.

**Khả năng tương tác tăng lên**: Bạn có thể tạo các giao diện mà phản ứng lại khi người sử dụng rê chuột qua chúng hoặc kích hoạt chúng thông qua bàn phím.

**Giao diện phong phú hơn**: Bạn có thể sử dụng JavaScript để bao gồm những mục như các thành phần Drag và Drop (DnD) và các con trượt (Slider) để cung cấp một Rich Interface (Giao diện giàu tính năng) tới site khách truy cập của bạn.

**1.2.4. Hạn chế của JavaScript**

Chúng ta không thể đối xử JavaScript như là một ngôn ngữ chương trình chính thức (full-fledged). Nó thiếu các tính năng quan trọng sau:

* Client-side JavaScript không cho phép đọc và ghi các file, bởi vì lý do bảo mật.
* JavaScript không được sử dụng cho việc kết nối mạng các ứng dụng bởi vì không có những hỗ trợ có sẵn.

JavaScript không có bất kỳ khả năng đa luồng hoặc đa xử lý.

Một lần nữa, JavaScript là một ngôn ngữ chương trình thông dịch, nhẹ mà cho phép bạn xây dựng khả năng tương tác trong các trang HTML tĩnh.

**1.2.5. Các công cụ phát triển JavaScript**

Một trong những điểm mạnh lớn nhất của JavaScript là nó không yêu cầu các công cụ phát triển tốn kém. Bạn có thể bắt đầu với một bộ biên soạn (Editor) văn bản đơn giản như Notepad. Khi nó là một ngôn ngữ thông dịch bên trong context của một trình duyệt web, bạn không cần phải mua một Complier (bộ phiên dịch).

Để làm cuộc sống của bạn đơn giản hơn, nhiều nhà cung cấp đã cho ra đời các công cụ chỉnh sửa JavaScript đẹp, được liệt kê dưới đây:

**Microsoft FrontPage**Microsoft đã phát triển một HTML Editor phổ biến được gọi là FrontPage. FrontPage cũng cung cấp cho nhà lập trình các công cụ JavaScript để giúp đỡ tạo các Website có tính tương tác.

**Macromedia Dreamweaver MX** Macromedia Dreamweaver MX là một HTML và JavaScript Editor rất phổ biến trong cộng đồng lập trình web chuyên nghiệp. Nó cung cấp các thành phần JavaScript đã xây dựng trước mà thuận tiện, tích hợp tốt với cơ sở dữ liệu, và theo các chuẩn mới như XHTML và XML.

**Macromedia HomeSite 5** HomeSite 5 là một HTML và JavaScript Editor từ Macromedia mà có thể được sử dụng để quản lý các Website cá nhân một cách hiệu quả.

**1.2.6. JavaScript ngày nay?**

Chuẩn ECMAScript Edition 5 sẽ là cập nhật đầu tiên được công bố trong 4 năm qua. JavaScript 2.0 tuân theo chuẩn ECMAScript Edition 5, và sự khác nhau giữa chúng là ít.

Chi tiết kỹ thuật cho JavaScript 2.0 có thể được tìm thấy trong site sau: **http://www.ecmascript.org/**

Ngày nay, JavaScript của Netscape và JScript của Microsoft tuân theo chuẩn ECMAScript, mặc dù cả hai ngôn ngữ vẫn hỗ trợ các tính năng mà không là một phần của chuẩn này.

**1.2.7. Các thuộc tính của Navigator**

Có một số thuộc tính liên quan đến Navigator mà bạn có thể sử dụng trong trang web của mình. Dưới đây là những thuộc tính của Navigator và mô tả chi tiết về chúng.

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| **appCodeName** | Thuộc tính này là một chuỗi mà chứa tên code của trình duyệt, như Netscape cho Netscape và Microsoft Internet Explorer cho IE. |
| **appVersion** | Thuộc tính này là một chuỗi mà chứa phiên bản của trình duyệt cũng như thông tin hữu ích khác như ngôn ngữ và khả năng tương thích của nó. |
| **language** | Thuộc tính này chứa hai chữ viết tắt cho ngôn ngữ mà được sử dụng bởi trình duyệt. Chỉ dành cho Netscape. |
| **mimTypes[]** | Thuộc tính này là một mảng chứa tất cả kiểu MIME được hỗ trợ bởi Client. Chỉ dành cho Netscape. |
| **platform[]** | Thuộc tính này là một chuỗi chứa platform mà trình duyệt được biên dịch. "Win32" cho Hệ điều hành Windows 32 bit. |
| **plugins[]** | Thuộc tính này là một mảng chứa tất cả plug-in đã được cài đặt trên client. Chỉ dành cho Netscape. |
| **userAgent[]** | Thuộc tính này là một chuỗi mà chứa tên code và phiên bản của trình duyệt. Giá trị này được gửi tới Server nguồn để nhận diện Client. |

**1.2.8. Các phương thức Navigator**

Dưới đây là danh sách các phương thức Navigator và mô tả chi tiết về chúng:

**javaEnabled():**Phương thức này quyết định nếu JavaScript được kích hoạt trong Client. Nếu JavaScript được kích hoạt, phương thức này trả về true, nếu không thì nó trả về false.

**plugings.refresh:**Phương thức này tạo plug-in được cài đặt mới nhất và đặt các mảng plugins với tất cả tên plug-in mới. Chỉ dành cho Netscape.

**preference(name,value):**Phương thức này cho phép một script đã đánh dấu để nhận và thiết lập một số quyền ưu tiên Netscape. Nếu tham số thứ hai bị bỏ qua, phương thức này sẽ trả về giá trị của quyền ưu tiên đã xác định; nếu không thì, nó thiết lập giá trị đó. Chỉ dành cho Netscape.

**taintEnabled():**Phương thức này trả về true nếu dữ liệu hỏng (data tainting) được kích hoạt; nếu không là false.

**1.2.9. Cách để debug một Script.**

Có nhiều cách để debug các lỗi trong JavaScript của bạn:

* **Sử dụng JavaScript Validator**

Một cách để kiểm tra code có các bug lạ là để chạy nó thông qua một chương trình mà kiểm tra nó để đảm bảo rằng nó là hợp lệ và nó tuân theo qui tắc cú pháp chính thức của ngôn ngữ. Những chương trình này được gọi là validating parsers hoặc validators, và thường đi với các bộ soạn HTML và JavaScript thương mại.

Validator tiện lợi nhất cho JavaScript là Douglas Crockford của JavaScript Lint, mà có sẵn miễn phí tại: Douglas Crockford's JavaScript Lint.

Bạn ghé thăm trang web này, paste các JavaScript code vào trong khu vực văn bản đã cho, và nhấn nút jslint. Chương trình này sẽ phân tích JavaScript code của bạn, đảm bảo rằng tất cả định nghĩa biến và hàm tuân theo cú pháp chính xác. Nó cũng sẽ kiểm tra các lệnh JavaScript, như if và while, để đảm bảo chúng cũng tuân theo định dạng chính xác.

* **Sử dụng một JavaScript Debugger**

Một Debugger là một ứng dụng mà đặt tất cả các khía cạnh của việc thực thi script dưới sự điều khiển của một nhà lập trình. Debugger cung cấp một điều khiển Fine-Grained (Điều khiển phân định tinh tế) qua trạng thái của script thông qua một giao diện mà cho phép bạn kiểm tra và thiết lập các giá trị cũng như điều khiển luồng của việc thực thi.

Một khi một script đã được tải vào trong một debugger, nó có thể được chạy một dòng tại một thời điểm hoặc được chỉ dẫn để dừng tại điểm dừng (Breakpoint). Khi việc thực thi bị dừng, nhà lập trình có thể kiểm tra trạng thái của script và các biến của nó để quyết định làm điều gì nếu có gì đó bị lỗi. Bạn có thể cũng quan sát các biến với các thay đổi trong các giá trị của chúng.

* **Chỉ dẫn hữu ích cho nhà lập trình**

Bạn có thể nên nhớ các**tip** sau để giảm số lượng các lỗi trong các script của bạn và làm đơn giản hóa tiến trình debug.

Sử dụng nhiều **comments**. Các comment cho bạn khả năng giải thích tại sao bạn viết script, cách bạn thực hiện và để giải thích các khu vực cụ thể phức tạp trong code.

Luôn luôn sử dụng**indentation** **(sự thụt code)** để làm code của bạn dễ dàng hơn khi đọc. Các lệnh thụt vào này cũng làm nó dễ hơn với bạn để so khớp các thẻ bắt đầu và kết thúc, các dấu ngoặc, và các phần tử HTML và script khác.

Viết **modular code**. Bất cứ khi nào có thể, nhóm các lệnh vào trong các hàm. Các hàm giúp bạn tạo nhóm các lệnh liên quan nhau, và kiểm tra và tái sử dụng các phần của code mà không mất nhiều công sức.

Nhất quán trong cách đặt tên biến và hàm. Nói cách khác, giữ tất cả chúng ở dạng chữ thường hoặc chữ hoa; nếu bạn ưa thích Camel-Back Notation, sử dụng nó một cách nhất quán.

Sử dụng tên biến đủ dài và có ý nghĩa mà miêu tả nội dung của hàm và biến.

**Kiểm tra các script dài trong Modular Fashion**. Nghĩa là, đừng cố gắng viết toàn bộ script trước khi kiểm tra bất cứ phần nào của nó. Viết một phần và thực thi nó trước khi thêm phần tới vào code.

Sử dụng **các tên biến và hàm có tính miêu tả cao** và tránh sử dụng các tên đơn ký tự.

**Quan sát các dấu trích dẫn**. Nhớ rằng những dấu trích dẫn này được sử dụng trong các cặp bao quanh các chuỗi và nó phải sử dụng trong cùng một kiểu giống nhau (hoặc đơn hoặc kép).

**Kiểm tra các dấu bằng**. Bạn không nên sử dụng một dấu bằng đơn (=) cho mục đích so sánh.

Khai báo **các biến một cách rõ ràng** bởi sử dụng từ khóa **var**.

**1.2.10. JavaScript trong Internet Explorer**

# Tất cả các trình duyệt hiện đại đi kèm với hỗ trợ có sẵn cho[JavaScript](https://quantrimang.com/JavaScript). Thường thì, bạn có thể cần thao tác để enable (kích hoạt) hoặc disable (vô hiệu hóa) hỗ trợ này. Chương này giải thích thủ tục để kích hoạt và vô hiệu hóa hỗ trợ JavaScript trong trình duyệt của bạn: ****Internet Explorer,****[Firefox](https://quantrimang.com/mozilla-firefox)****,****[Chrome](https://quantrimang.com/google-chrome)****, và****[Opera](https://quantrimang.com/opera).

## JavaScript trong Internet Explorer

Dưới đây là các bước đơn giản để kích hoạt hoặc vô hiệu hóa JavaScript trong IE:

Chọn **Tools -> Internet Options** từ menu.

Chọn **Security** tab từ hộp thoại đó.

Nhấp chuột chọn **Custom Level**.

Kéo bánh xe chuột tới khi bạn thấy **Scripting option**.

Chọn Enable ở dưới **Active scripting**.

Cuối cùng nhấn OK và thoát ra.

Để vô hiệu hóa hỗ trợ JavaScript trong IE của bạn, bạn cần chọn nút **Disable** ở dưới **Active scripting**.

## JavaScript trong Firefox

Sau đây là các bước để kích hoạt hoặc vô hiệu hóa JavaScript trong Firefox:

Mở một tab mới, soạn**about: config** trong thanh địa chỉ.

Sau đó bạn sẽ thấy một hộp thoại cảnh báo. Chọn**I'll be careful, I promise!**

Sau đó bạn sẽ thấy danh sách**configure options** (các tùy chọn định cấu hình) trong trình duyệt.

Trong thanh tìm kiếm, soạn **javascript.enabled**.

Tại đây, bạn sẽ tìm thấy tùy chọn để kích hoạt hoặc vô hiệu hóa JavaScript bằng việc kích chuột phải trên giá trị tùy chọn đó, chọn **toggle**.

Nếu javascript.enabled là true, nó biến đổi thành false với việc nhấn**toggle**. Nếu javascript bị vô hiệu hóa, nó kích hoạt khi nhấn toggle.

## JavaScript trong Chrome

Sau đây là các bước để kích hoạt hoặc vô hiệu hóa JavaScript trong Chrome:

Nhấp chuột vào menu tại phía trên cùng bên phải của trình duyệt..

Chọn **Settings**.

Nhấp chuột chọn **Show advanced settings** ở cuối trang.

Dưới khu vực **Privacy**, nhấn nút Content settings.

Trong khu vực "JavaScript", chọn "Do not allow any site to run JavaScript" hoặc "Allow all sites to run JavaScript (recommended)".

## JavaScript trong Opera

Dưới đây là các bước để kích hoạt hoặc vô hiệu hóa JavaScript trong Opera:

Từ menu, bạn chọn **Tools -> Preferences**.

Chọn tùy chọn **Advanced** từ hộp thoại.

Chọn **Content** từ các mục được liệt kê.

Chọn **Enable JavaScript** checkbox.

Cuối cùng nhấn Ok và thoát ra.

Để vô hiệu hóa JavaScript trong Opera, bạn không nên chọn **Enable JavaScript checkbox**.

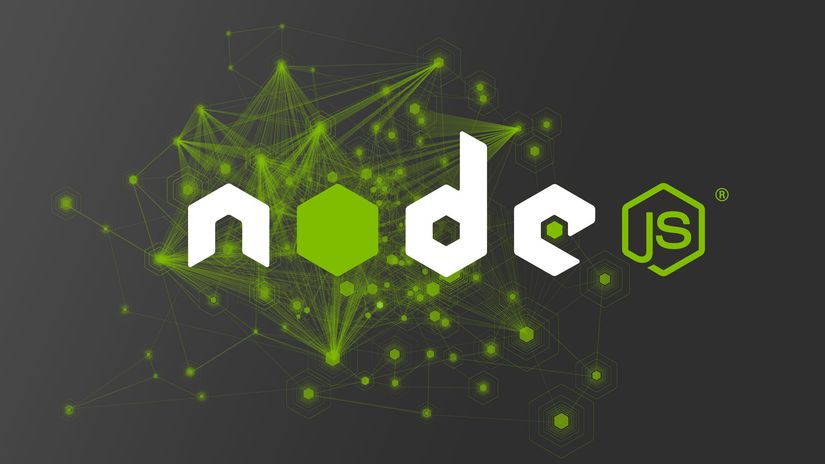
## Cảnh báo về các trình duyệt Non-JavaScript

Nếu bạn phải thực hiện cái gì đó quan trọng sử dụng JavaScript, thì khi đó bạn có thể hiển thị một thông báo cảnh báo tới người dùng bởi sử dụng các thẻ**<noscript>**.

nếu trình duyệt của người dùng không hỗ trợ JavaScript hoặc JavaScript không được kích hoạt, thì thông báo từ </noscript> sẽ được hiển thị trên màn hình.

* 1. **Nodejs**

## Nodejs là gì?



* Nodejs là một **nền tảng** (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
* Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Dù sao thì chúng ta cũng nên biết qua một chút chút lịch sử của thứ mà chúng ta đang học một chút chứ nhỉ? =))
* Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.
* Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

## Tại sao sử dụng Nodejs?

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn là một startup đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Node.js có thể xử lý hàng ngàn kết nối đồng thời trong khi PHP sẽ chỉ có nước sụp đổ. Bên cạnh các lợi ích về tốc độ thực thi và khả năng mở rộng, có thể bạn cũng đã biết một chút về JavaScript, vì vậy tại sao lại phải phiền toái để học thêm về một ngôn ngữ lập trình hoàn toàn mới như PHP? Và sau đó bạn sẽ có một sự phấn khích khi học về một cái gì đó mới mẻ và gần như chưa được khám phá. Bạn còn nhớ cái cảm giác khi mà một cái gì đó mới xuất hiện và sau đó trở thành phổ biến khắp mọi nơi mà bạn hối tiếc đã không học về nó sớm hơn, và mãi mãi chỉ là người đến sau? Đừng phạm phải sai lầm như vậy lần này nữa. Node.js đang ngày càng trở nên lớn mạnh hơn.

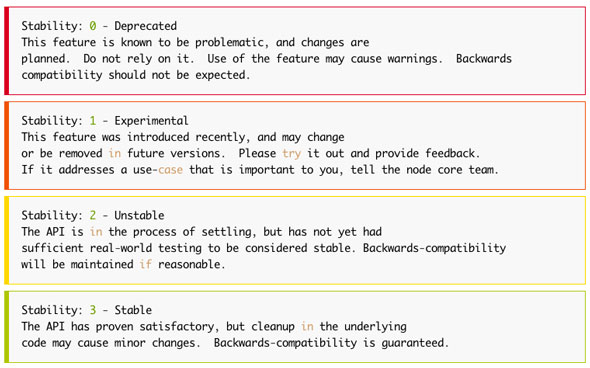
* Các ứng dụng Nodejs được viết bằng **javascript**, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: *“Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”*
* Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free: <https://www.npmjs.com/>
* Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

## Nhược điểm của Nodejs

Giống như hầu hết các công nghệ mới, việc triển khai Node.js trên host không phải là điều dễ dàng. Nếu bạn có một web hosting xài chung, bạn không thể đơn giản tải lên một ứng dụng Node.js và mong chờ nó hoạt động tốt. VPS và dedicated server là một sự lựa chọn tốt hơn - bạn có thể cài đặt Node.js trên chúng. Thậm chí dễ hơn là sử dụng một dịch vụ có khả năng mở rộng như là Heroku, và bạn có thể hoàn toàn an tâm để phát triển trang web của mình trên đó - bạn chỉ cần trả tiền khi cần thêm nhiều tài nguyên hơn.

Mặt khác, chúng ta rất dễ cài đặt Node.js chạy cục bộ trên máy tính của bạn sử dụng các hệ điều hành như Windows, Mac hoặc Linux và bắt đầu phát triển ứng dụng ngay lập tức - chỉ việc tải phiên bản Node.js tương ứng [tại đây](http://nodejs.org/download/). Một điều quan trọng nên chú ý là Node.js không chỉ đơn giản là một sự thay thế cho Apache - các ứng dụng web đang tồn tại sẽ không có khả năng tương thích, và bạn sẽ làm việc hiệu quả với những ứng dụng phát triển từ đầu (***mặc dù có rất nhiều framework ngoài kia để giúp đỡ bạn với nhiều đặc trưng phổ biến***).

Một nhược điểm lớn khác của Node.js đó là nó vẫn đang trong giai đoạn phát triển ban đầu, điều này có nghĩa là một số đặc trưng sẽ thay đổi trong quá trình phát triển tiếp theo. Trong thực tế, nếu bạn đọc các [tài liệu đi kèm](http://nodejs.org/api/all.html#all_stability_index), thì nó bao gồm một chỉ số ổn định (stability index), chỉ số này cho thấy mức độ rủi ro khi bạn sử dụng các đặc trưng hiện có.

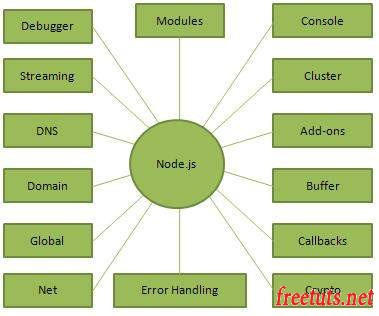


## Các đặc tính của NodeJS

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](https://raw.githubusercontent.com/joyent/node/v0.12.0/LICENSE).

Sau đây là sơ đồ về các thành phần quan trọng trong NodeJS mà trong series này chúng ta sẽ tìm hiểu dần (hình sưu tầm từ tutorialspoint).



## Những ứng dụng nên viết bằng Nodejs

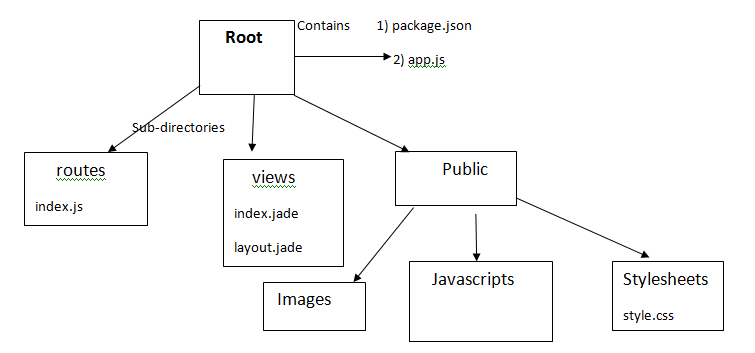
Rõ ràng, không phải cứ hot và mới là Nodejs làm gì cũng tốt, ví dụ như một ứng dụng cần tính ổn định cao, logic phức tạp thì các ngôn ngữ PHP hay Ruby… vẫn là sự lựa chọn tốt hơn. Còn dưới đây là những ứng dụng có thể và nên viết bằng Nodejs:

* **Websocket server:** Các máy chủ web socket như là Online Chat, Game Server…
* **Fast File Upload Client:** là các chương trình upload file tốc độ cao.
* **Ad Server:** Các máy chủ quảng cáo.
* **Cloud Services:** Các dịch vụ đám mây.
* **RESTful API:** đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* **Any Real-time Data Application:** bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

Express

* 1. **Các Framework nodejs: Express và SocketIO.**
     1. **Framework Express**
  2. ***Khái niệm Express Framework là gì?***

**Expressjs** là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.



**Tổng hợp một số chức năng chính của Express như sau:**

* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.
  1. ***Các thành phần cơ bản của Express***

## Request & response

Request - Biểu diễn một HTTP request và có các thuộc tính cho các request như các chuỗi truy vấn, tham số, body, HTTP header và những phần khác.

Response - Biểu diễn một HTTP response được ứng dụng Express gửi đi khi nó nhận về một HTTP request.

## Route

**Router là một Object** (khác Routing nhé), nó là một instance riêng của middleware và routes (Hai khái niệm này là gì thì chúng ta sẽ tìm hiểu sau nhé).

* Router là một instance của middleware và route nên nó có các chức năng của cả hai. Chúng ta có thể gọi nó là một mini-application
* Các Application dùng ExpressJS làm core đều có phần Router được tích hợp sẵn trong đó.
* Router hoạt động như một middleware nên chúng ta có thể dùng nó như một arguments. Hoặc dùng nó như một arguments cho route khác.

## File tĩnh trong Express

Express cung cấp tiện ích **express.static** để phục vụ cho các file tĩnh như hình ảnh, css, js, ...Về cơ bản, bạn chỉ cần truyền tên thư mục nơi bạn giữ các file này, express.static sẽ sử dụng file đó một cách trực tiếp

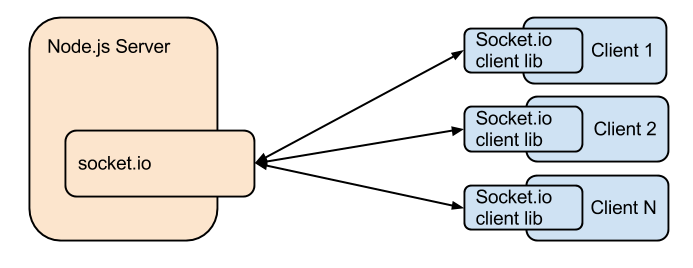
* + 1. **Framework SocketIO**
  1. ***SocketIO là gì?***

Là một module của NodeJs

Được xây dựng nhằm mục đích tạo ra real time NodeJS application. [Socket.io](http://socket.io/) cung cấp cho lập trình viên các đặc trưng như event, room và tự động phục hồi lại kết nối.

Khi chúng ta include [Socket.io](http://socket.io/) module vào trong ứng dụng của mình nó sẽ cung cấp cho chúng ta hai object đó là: socket server quản lý functionality phía server và socket client điều khiển funtionality phía client.

Khi client muốn kết nối tới [Socket.io](http://socket.io/) server, nó sẽ gửi cho server một “handshake HTTP request”. Server sẽ phân tích request đó với những thông tin cần thiết trong suốt quá trình kết nối. Nó sẽ tìm cấu hình của middleware mà đã được đăng ký với server và thực thi chúng trước khi đưa ra sự kiện kết nối. Khi kết nối thành công thì connection event listener được thực thi, tạo ra một instance mới của socket có thể coi như định danh của client mà mỗi một client kết nối tới sẽ có 1 định danh. Các bạn có thể thấy rõ khi xem hình dưới đây



Một module khác của Node.js là LightStreamer-adapter cũng có tạo các kết nối từ client tới server nhưng không trực tiếp mà thông qua LightStreamer Server, đó là các máy chủ theo thời gian thực và nằm ngoài tiến trình của Node.js Server

* 1. ***Tại sao phải sử dụng SocketIO***

Các socket theo truyền thống là giải pháp mà hầu hết các hệ thống thời gian thực được kiến trúc, cung cấp một kênh liên lạc hai chiều giữa máy khách và máy chủ. Điều này có nghĩa là máy chủ có thể đẩy thông điệp đến máy khách. Bất cứ khi nào một sự kiện xảy ra, ý tưởng là máy chủ sẽ lấy nó và đẩy nó đến các máy khách được kết nối có liên quan.

Socket.IO khá phổ biến, nó được Microsoft Office, Yammer, Zendesk, Trello và nhiều tổ chức khác sử dụng để xây dựng các hệ thống thời gian thực mạnh mẽ. Là một trong những JavaScript framework mạnh nhất trên GitHub và mô-đun NPM (Node Package Manager). Socket.IO có một cộng đồng lớn để hỗ trợ các vướng mắc của bạn khi sử dụng.

Socketio không được xem là ngôn ngữ mà chỉ là một trong những công cụ có thể hỗ trợ thực hiện cho một số ứng dụng dạng realtime. Chính vì vậy, bạn thể sử dụng socketio để thay thế hoàn toàn cho một ngôn ngữ khác và các loại ngôn ngữ đó có thể là: php, asp.net, nodejs,…

* 1. ***Làm thế nào để sử dụng SocketIO***

Cấu trúc một ứng dụng realtime sử dụng socket bao gồm 2 phần: phía server, phía client.

Phía server Đây là nơi sẽ cài đặt socket io. Ngôn ngữ để dựng server có thể là php, [asp.net](http://asp.net/), nodejs,... Tuy nhiên, tùy vào ngôn ngữ lựa chọn mà cách cấu trúc server khác nhau. Ở đây, nếu được thì khuyến khích sử dụng nodejs để dựng server, vì như vậy có thể cài trực tiếp socketio vào cùng một server. Nếu sử dụng php thì phải cài thêm những package khác, hoặc phải chuẩn bị riêng server để chạy socketio.

Phía client: Ở phía client sẽ xây dựng giao diện người dùng. Ở đây có thể sử dụng js, hoặc các thư viện của js như jquery,... Nói chung là ngôn ngữ gì cũng được.

* 1. ***Cơ chế hoạt động của socketIO***

Cơ chế hoạt động của một ứng dụng realtime đó là thông qua server để lắng nghe (listen) data và truyền data về các máy client. Vì vậy cần cài khai báo sử dụng socketio ở cả phía server và client.

Để lắng nghe data, ta sử dụng câu lệnh socket.on(), để phát dữ liệu thì sử dụng lệnh socket.emit() .

Lưu ý: socket.on và socket.emit có parameter thứ 1 là tên đường truyền. Tên đường truyền có thể là tên bất ký, tuy nhiên đễ truyền và nhận dữ liệu của chung 1 đường truyền thì tên đường truyền phải giống nhau.

* 1. ***Những đặc điểm nổi bật của SocketIO là gì?***

Socket.io hiện đang được cộng đồng các lập trình viên dùng ngày càng nhiều bởi sự tiện lợi của nó. Hiện nay, Socket.io giúp ta sở hữu nhiều tính năng nổi bật như: bảo mật, binary, kết nối tự động, ghép kênh, tạo phòng,..

### *Khả năng bảo mật*

Ngay khi Socket.io xuất hiện thì nó có thể tự động tạo ra những kết nối dạng bảo mật như:

* **Thực hiện Proxy và cân bằng cho tải.**
* **Tạo tường lửa cá nhân và các phần mềm chống lại virus.**
* **Socket.io thực hiện xây dựng dựa vào Engine.IO như sau: nó sẽ khởi chạy theo  phương thức long-polling để tự động kết nối, rồi dùng những phương thức giao tiếp khác để giao tiếp được tốt hơn.**

### *Kết nối lại tự động + phát hiện tình trạng ngắt kết nối*

Khi chạy, nếu như client bị ngắt kết nối thì nó có thể kết nối tự động cho đến khi server xác nhận đã được phản hồi. Đây là tính năng cho phép thiết bị có thể tự động kết nối lại mãi mãi cho đến khi server phản hồi. Socket.io còn cung cấp cho bạn những event có thể phát hiện ngắt kết nối giữa client và server.

### *Hỗ trợ nhị phân*

Socket.io hỗ trợ bạn một số kiểu mã nhị phân:

* **ArrayBuffer cùng Blob nằm trên trình duyệt**
* **ArrayBuffer cùng Buffer nằm trong Node.js**

### *Hỗ trợ tạo phòng và kênh*

Socket.io cho phép người dùng tạo các kênh riêng biệt, từ đó tạo ra những mối quan hệ riêng giữa các phần tương tự như module riêng lẻ hoặc dựa vào một số quyền khác nhau. Ngoài ra, nó còn hỗ trợ bạn tạo các phòng khác nhau cùng những clients được tham gia vào những phòng khác.

# CHƯƠNG 2: CÀI ĐẶT VÀ CẤU HÌNH

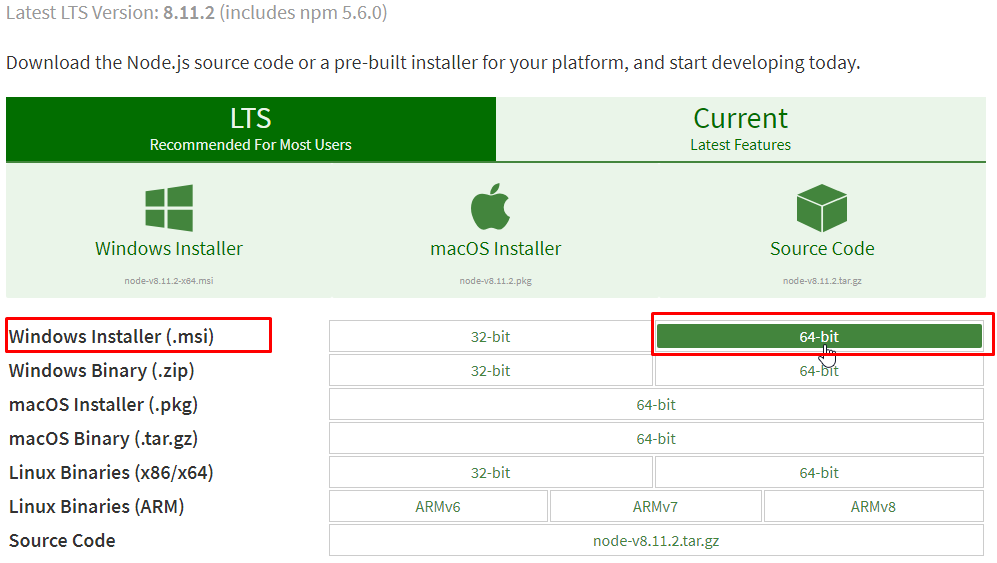
## 

**2.1. Cài đặt NodeJS, NPM**

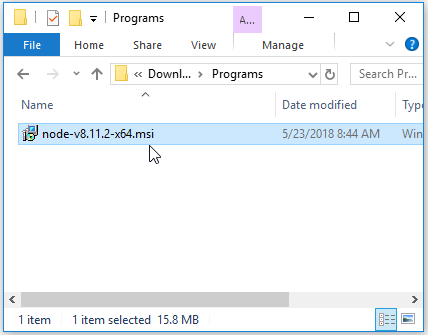
- Bước 1: Download NodeJS

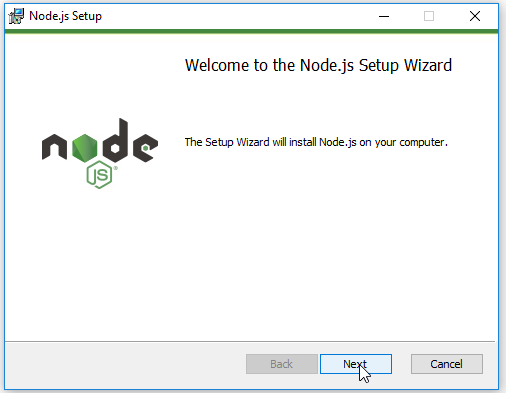
Truy cập trang <https://nodejs.org/en/download/> để download nodejs

Có khá nhiều bản nodejs khác nhau (32bit/64bit) cho windows, linux hay macOS. Ở đây mình dùng windows 64bit nên sẽ download bản .msi 64bit để cài.

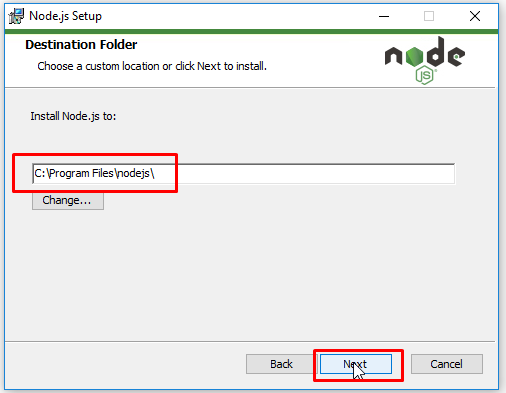
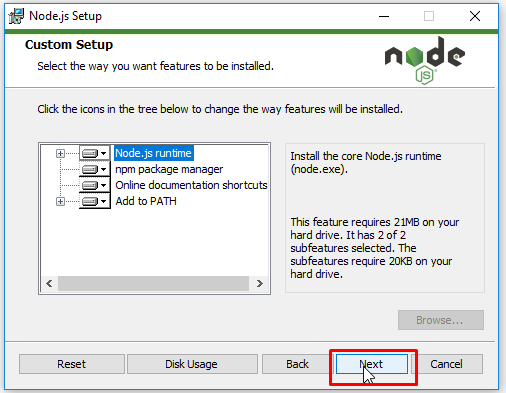
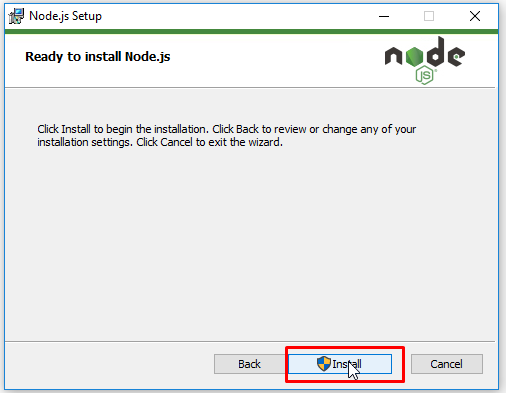
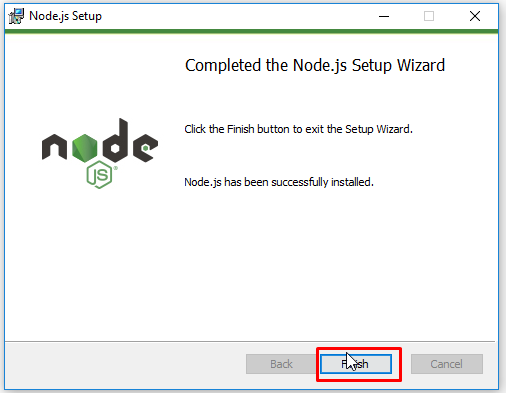


- Bước 2: Click đúp vào file .msi vừa tải về để tiến hành cài đặt.



[](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-3.png)

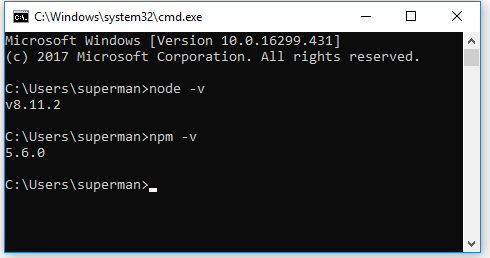
[](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-4.png)

Bươc 3: Chọn thư mục để cài đặt nodejs (mấy cái này cứ để mặc định cũng được) [[](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-5.png)](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-5.png) [](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-7.png) [](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-8.png) [](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-9.png)

Sau khi cài đặt thành công NodeJs chúng ta cần kiểm tra version của nodejs và npm vừa cài đặt bằng cách mở cửa sổ cmd và nhập 2 lệnh sau:

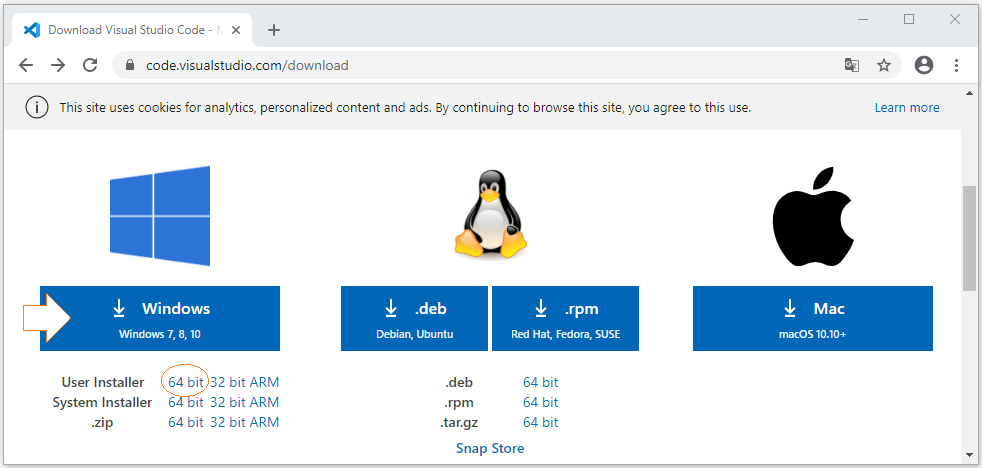
node -v (kiểm trả version nodejs)

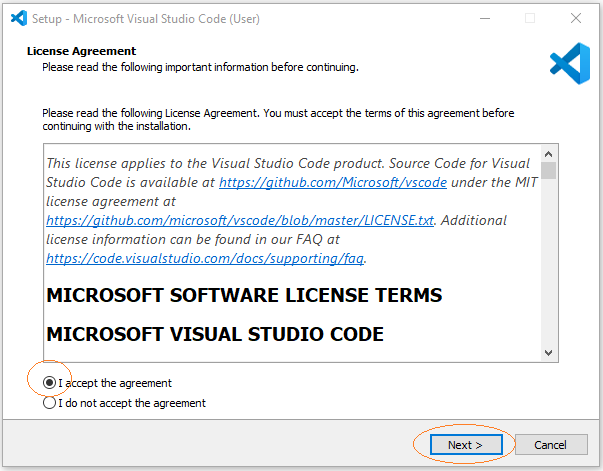
npm -v (kiểm tra version npm)

[](https://stackjava.com/wp-content/uploads/2018/05/install-nodejs-10.png)

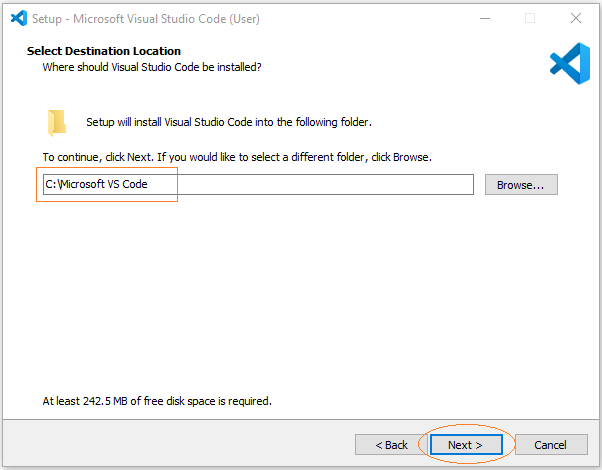
**2.2. Cài đặt trình soạn thảo “Visual Studio Code” để viết code**

Vào trang chủ vscode, chọn phiên bản phù hợp với thiết bị của các bạn và tải về.

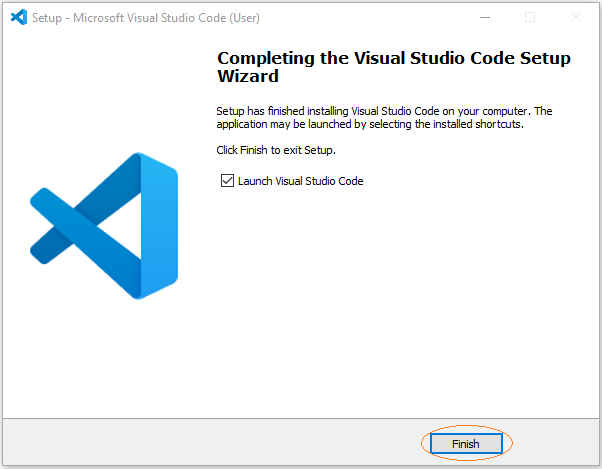




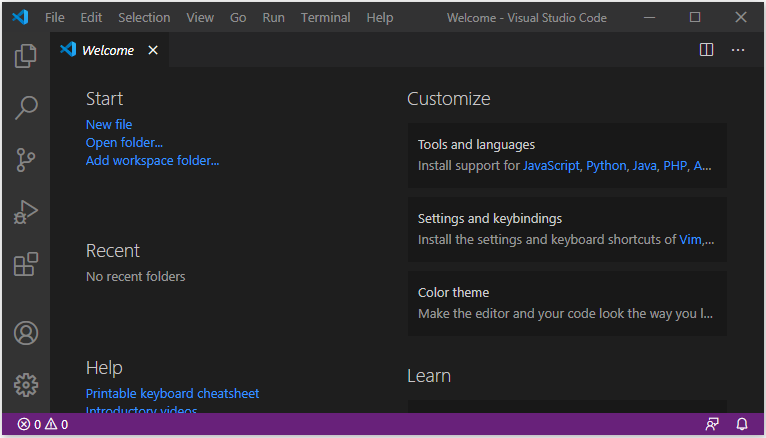
Chọn file cài đặt vừa tải về nhấn Next



Chọn đường dẫn cài đặt file



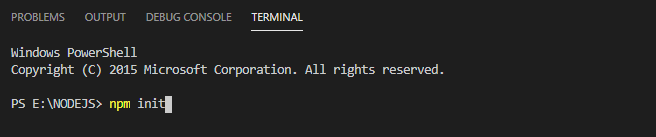
Nhấn finish để hoàn thành quá trình cài đặt



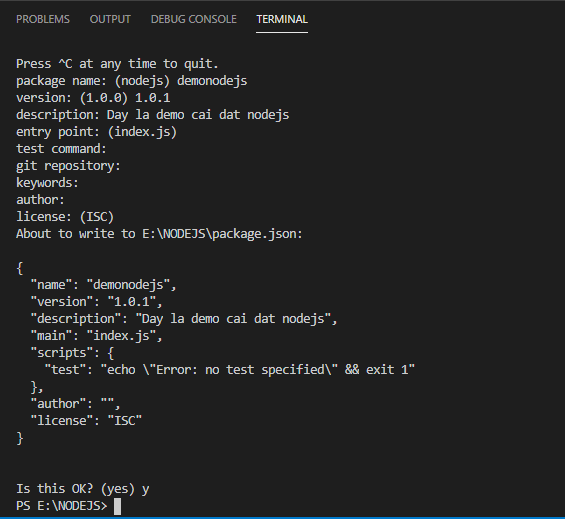
**Visual Studio Code** đã được cài đặt thành công, và đây là hình ảnh của nó trong lần chạy đầu tiên

**2.3 . Tạo ứng dụng đầu tiên bằng NodeJs qua NPM**

Bước 1: Để tạo một ứng dụng nào đó, chỉ cần mở cửa sổ **Terminal (hoặc CMD)** và thực thi lệnh



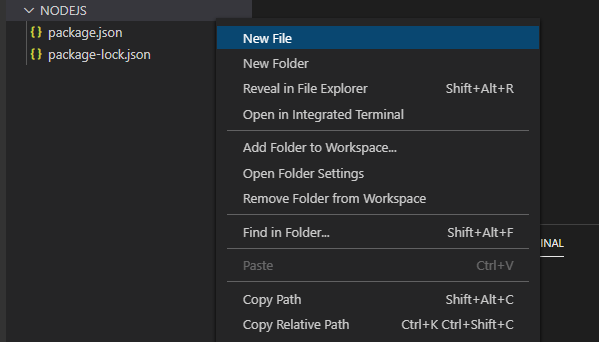
* Bước 2: Lần lượt khai báo các thuộc tính trong package.json:

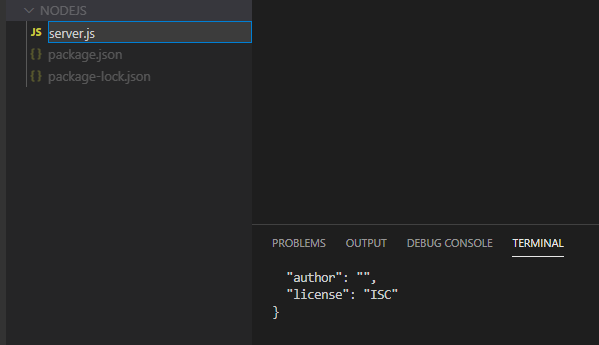


* **package name**: tên gói thư viện
* **version**: phiên bản gói
* **description**: phần mô tả về gói thư viện
* **entry point**: đường dẫn trở tới file mã nguồn, điểm chạy đầu tiền của package.
* **author**: tác giả
* **keywords**: các từ khóa
* Bước 3: Sau khi khai báo thuộc tính đầy đủ chúng ta gõ lệnh sau để cài đặt toàn bộ thư viện( dependencies) cần thiết

npm install

* Bước 4: Trong thư mục đường dẫn vừa khởi tạo chúng ta tạo một file lấy tên là server.js





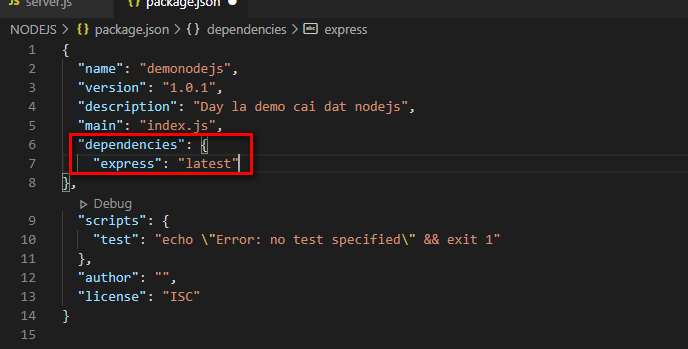
Như vậy là quá trình khởi tạo một ứng dụng đầu tiên bằng nodejs đã hoàn thành

**2.4. Thêm package Express vào project NodeJs**

- Cài đặt bằng command line ta dung lệnh:

npm install express *--save*

- Cài đặt bằng tạo file package.json  
Trong file package.json ta khai báo express

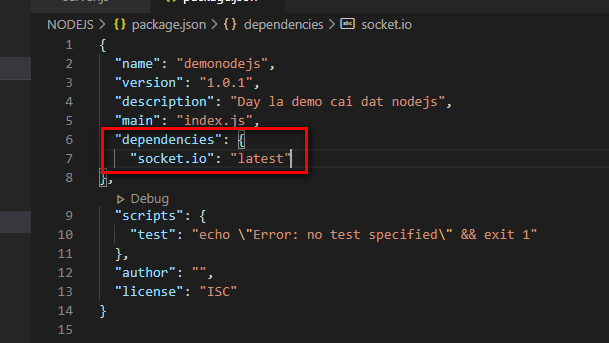


**2.5. Thêm package SocketIO vào project NodeJs**

- Cài đặt bằng command line ta dung lệnh:

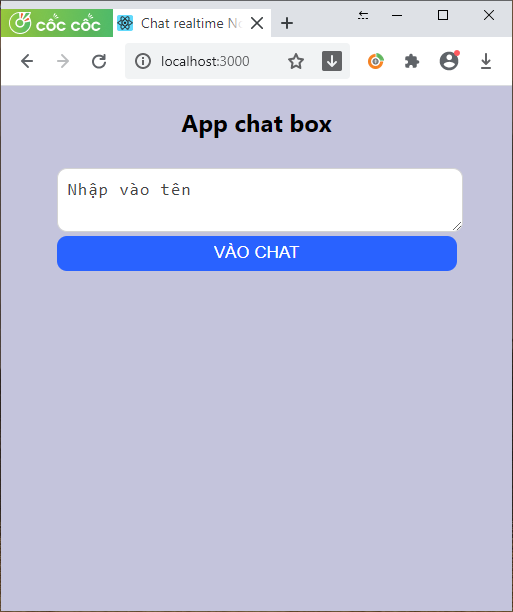
npm install socket.io

- Cài đặt bằng tạo file package.json  
Trong file package.json ta khai báo socket.io

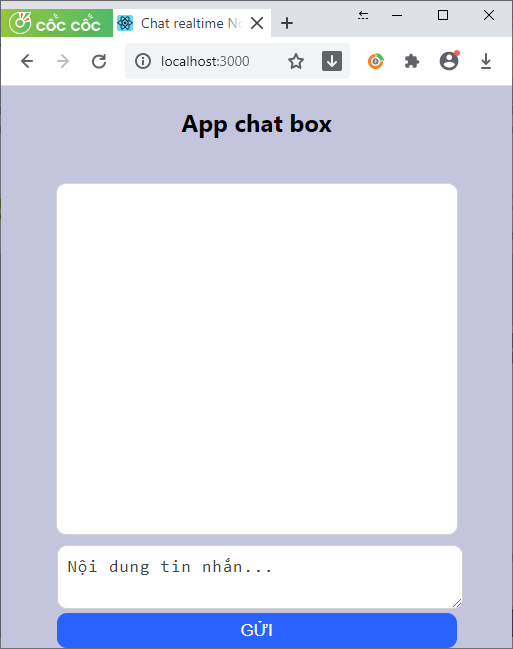


# CHƯƠNG 3: KẾT QUẢ THỰC HIỆN

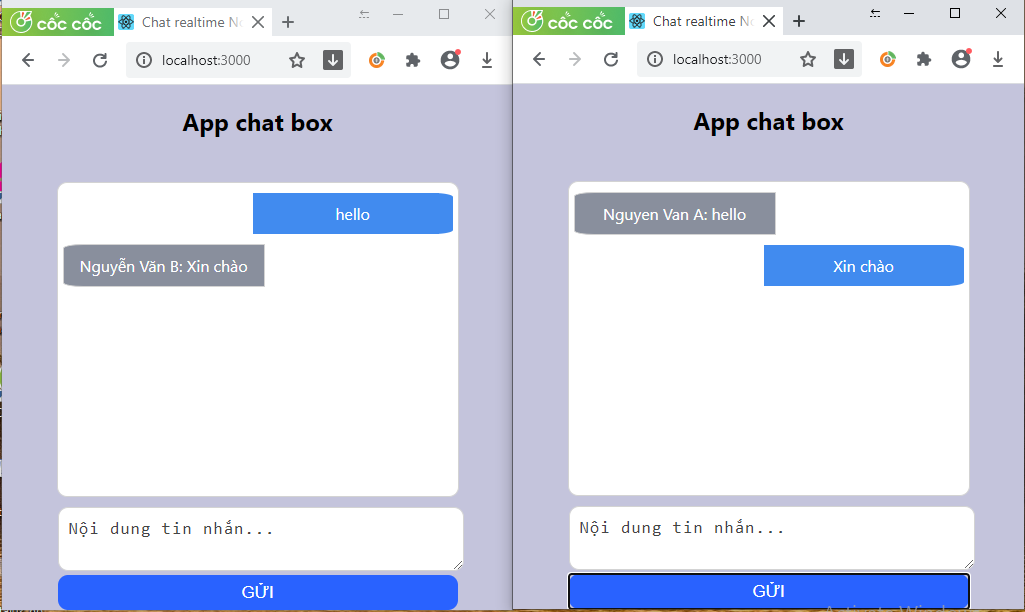
3.1. Form khai báo tên vào app



3.2. Form chat box



3.2. Form tương tác giữa các tài khoản trong chat box



# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Kết quả đạt được**
* Đã thực hiện thành công việc tạo ra một App chat box sử dụng nodejs cơ bản
* Nắm bắt được lập trình socket realtime
* **Hạn chế**
* Do chưa có thời gian nên chưa nghiên cứu hết các tính năng khác liên quan tới nodejs.
* **Hướng phát triển**
* Mong muốn tạo được app chat box mạnh mẽ ứng dụng thực tế trong tương lai
* Nghiên cứu thêm mở rộng về các packer trong nodejs

# 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.