# BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN CƠ SỞ**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**WEBSITE TRÒ CHUYỆN TRỰC TUYẾN**

**Sinh viên: Trần Quốc Toản**

**Mã số: B1606944**

**Khóa: 42**

**Cần Thơ, 06/2020**

# BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN CƠ SỞ**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**WEBSITE TRÒ CHUYỆN TRỰC TUYẾN**

**Người hướng dẫn Sinh viên thực hiện**

**PSG TS.Trần Cao Đệ Trần Quốc Toản**

**Mã số: B1606944**

**Khóa: 42**

***Cần Thơ, 11/2019***

# LỜI CẢM ƠN

*Em xin chân thành cảm ơn PSG TS.Trần Cao Đệ, Bộ môn Công nghệ Thông tin, Khoa Công nghệ Thông tin và Truyền thông, trường Đại học Cần Thơ đã cố vấn, giảng dạy và giúp đỡ sinh viên hoàn thành đề tài này!*

**MỤC LỤC**

[Lời cảm ơn](#_Toc24143963) 1

[Mục lục](#_Toc24143964) 2

[CHƯƠNG I: TỔNG QUAN](#_Toc24143965) 3

* 1. Đặt vấn đề 4
  2. Mục tiêu …………………………………………………………………………...1
  3. Phạm vi và kết quả cần đạt được ………………………………………………….2
  4. Chức năng của chương trình……………………………………………………….2

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT](#_Toc24143967) 5

2.1 Đặt tả yêu cầu……………………………………………………………………...2

2.2 Thiết kế cơ sở dữ liệu……………………………………………………………...2

2.2.1 Mô hình cơ sở dữ liệu………………………………………………....2

2.2.2 Mô tả các lớp………………………………………………………….2

2.3 Mô hình use case…………………………………………………………………..2

2.4 Tổng kết chương……………………………………………………………….......2

[CHƯƠNG III: CÀI ĐẶT VÀ KẾT QUẢ](#_Toc24143967) 5

3.1 Ngôn ngữ lập trình và công nghệ phía Client ………………….............................2

3.1 Ngôn ngữ lập trình và công nghệ phía Server.……………..……………………...2

3.3 Web Socket………………………………………………………………………..2

3.4 Database Mongodb………………………………………………………………...2

3.5 Yêu cầu hệ thống…………………………………………………………………..2

3.5.1 Phần cứng……………………………………………………………..2

3.5.2 Hệ điều hành…………………………………………………………..2

3.6 Kết quả……………………………………………………………………………...2

3.6.1 Chức năng đăng kí tài khoản………………………………………….2

3.6.2 Chức năng đăng nhập tài khoản………………………………………2

3.6.3 Chức năng đăng xuất………………………………………………….2

3.6.4 Chức năng đổi mật khẩu ……………………………………………...2

3.6.5 Chức năng cập nhật ảnh đại diện……………………………………...2

3.6.6 Chức năng tìm kiếm bạn bè…………………………………………...2

3.6.7 Chức năng kết bạn…………………………………………………….2

3.6.8 Chức năng nhắn tin……………………………………………………2

3.6.9 Chức năng nhắn tin với tất cả mọi người……………………………...2

3.6.10 Chức năng tạo nhóm trò chuyện……………………………………..2

3.6.11 Chức năng thêm thành viên vào nhóm………………………………2

3.6.12 Chức năng hiển thị số người online và offline………………………2

[CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN …………………………](#_Toc24143967) 5

4.1 Kết quả đạt được………………………………………………………………......2

4.2 Hạn chế……………………………………………………………………………2

4.3 Hướng phát triển…………………………………………………………………..2

TÀI LIỆU THAM KHẢO…………………………………………………………………..2

**CHƯƠNG I: TỔNG QUAN**

* 1. **Đặt vấn đề**

Ngày nay, với sự phát triển mạnh mẽ của khoa học – công nghệ, nó góp phần thúc đẩy sự tiến bộ của xã hội trong tất cả các lĩnh vực của đời sống xã hội. Đặc biệt với sự bùng nổ của công nghệ thông tin và internet thì tất cả các thông tin luôn được cập nhật, thay đổi một cách liên tục và đầy đủ thông qua mạng truyền thông và internet, mà phổ biến là thông qua các website.

Hiện nay nhu cầu giải trí của con người ngày càng cao. Sau những giờ làm việc và học tập mệt mõi, mọi người luôn muốn có một nơi để có thể trò chuyện với bạn bè của mình hoặc có thể trao đổi và làm việc với các đối tác. Đó là lý do mà website trò chuyện trực tuyến ra đời,

Đây là một nơi mà mọi người có thể tương tác trò chuyện với bạn bè của mình trên nền tản website.

* 1. **Mục tiêu**

Website trò chuyện trực tuyến là một nơi mà mọi người từ học sinh, sinh viên đến những người đã đi làm có thể kết bạn và trò chuyện với nhau, trao đổi về học tập hay về công việc. Nhằm mục tiêu gắn kết mọi người lại với nhau. Xây dựng một giao diện đơn giản thân thiện với người dùng và hoàn toàn miễn phí.

* 1. **Phạm vi và kết quả cần đạt được**

Phạm vi mà chương trình muốn hướng đến là tất cả mọi người dùng, nhất là các bạn học sinh sinh viên, nơi mà các bạn có thể trò chuyện, tạo nhóm cho học tập và trao đổi thông tin, tài liệu học tập. Tạo một môi trường hữu ích cho mọi người.

* 1. **Các chức năng chính của chương trình**

Các chức năng chính:

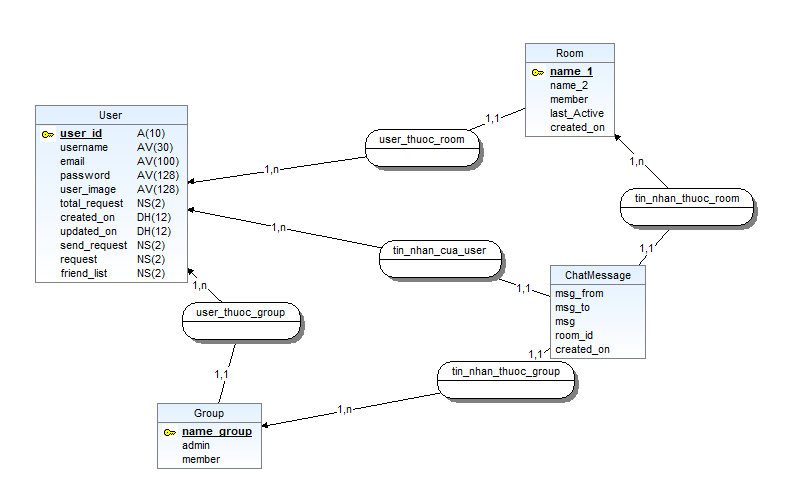
1. Tìm kiếm bạn bè
2. Gửi lời mời kết bạn
3. Đồng ý hoặc hủy lời mời kết bạn
4. Quản lý tài khoản
5. Gửi tin nhắn text
6. Gửi tin nhắn hình ảnh (emoji)
7. Gửi tin nhắn đến tất cả tài khoản trong danh sách bạn bè
8. Tạo nhóm trò chuyện
9. Thêm thành viên vào nhóm
10. Hiển thị số người online và offline
11. Gửi tin nhắn với thời gian thực (real time)

**CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

**2.1 Đặt tả yêu cầu**

**2.2 Thiết kế cơ sở dữ liệu**

**2.2.1 Mô hình cơ sở dữ liệu**



**Hình 2.1** Mô hình CDM

**2.2.1 Mô tả các lớp**

Danh sách các bảng:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Diễn giải** |
| 1 | User | Thông tin tài khoản người dùng |
| 2 | Room | Phòng trò chuyện |
| 3 | ChatMessage | Tin nhắn |
| 4 | Group | Nhóm trò chuyện |

Bảng 2.1 Danh sách các bảng trong mô hình CDM

* Bảng User: Thông tin tài khoản người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | User\_id | PK | Varchar(10) | Id tài khoản |
| 2 | Username |  | Varchar(30) | Tên tài khoản |
| 3 | Email |  | Varchar(100) | Email tài khoản |
| 4 | Password |  | Varchar(30) | Mật khẩu |
| 5 | User\_image |  | Varchar(100) | Hình ảnh |
| 6 | Total\_request |  | Int | Tổng yêu cầu |
| 7 | Created\_on |  | Datetime | Ngày tạo |
| 8 | Updated\_on |  | Datetime | Ngày cập nhật |
| 9 | Send\_request |  | Array | Gửi lời mời |
| 10 | request |  | Array | Lời mời |
| 11 | Friend\_list |  | Array | Danh sách bạn |

Bảng 2.2 User: Thông tin tài khoản người dùng

* Bảng Room: phòng trò chuyện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | Room\_id | PK | Varchar(10) | Id phòng |
| 2 | Name\_1 |  | Varchar(30) | Tên User thứ nhất |
| 3 | Name\_2 |  | Varchar(30) | Tên User thứ hai |
| 4 | Last\_active |  | Datetime | Lần hoạt động |
| 5 | Created\_on |  | Datetime | Ngày tạo |

Bảng 2.3 Room: phòng trò chuyện

* Bảng ChatMessage: tin nhắn

Bảng 2.4 ChatMessage: tin nhắn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | Msg\_from |  | Varchar(30) | Người gửi |
| 2 | Msg\_to |  | Varchar(30) | Người nhận |
| 3 | Msg |  | Varchar(100) | Tin nhắn |
| 4 | Room\_id |  | Varchar(10) | Id phòng |
| 5 | Created\_on |  | Datetime | Ngày tạo |

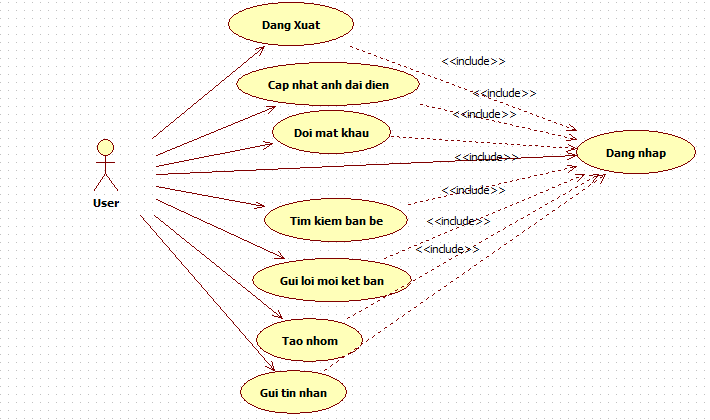
* Bảng Group: nhóm trò chuyện

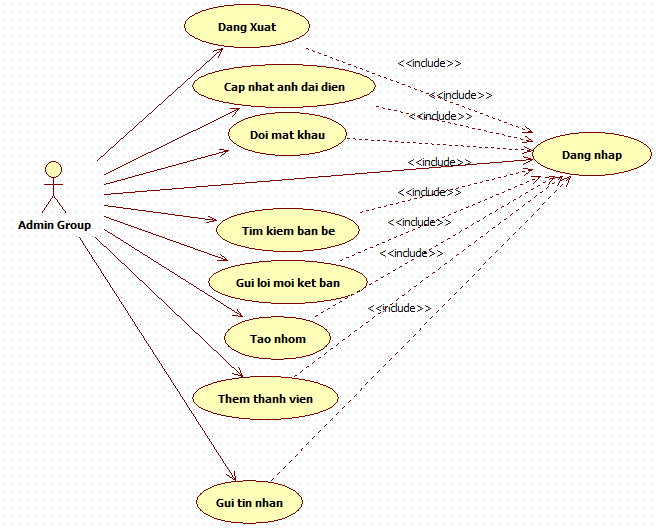
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | Group\_id | PK | Varchar(10) | Id nhóm |
| 2 | Name\_group |  | Varchar(30) | Tên nhóm |
| 3 | Admin |  | Varchar(30) | Chủ sở hủ |
| 4 | Member |  | Array | Thành viên |
| 5 | Created\_on |  | Datetime | Ngày tạo |

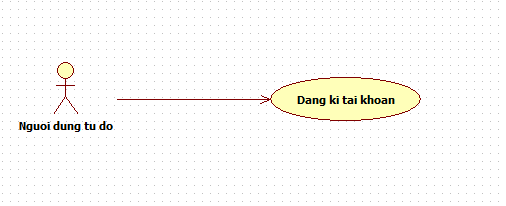
Bảng 2.5 Group: nhóm trò chuyện

**2.4 Mô hình use case**

**Use case tổng quát**



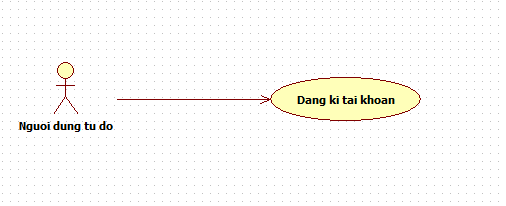




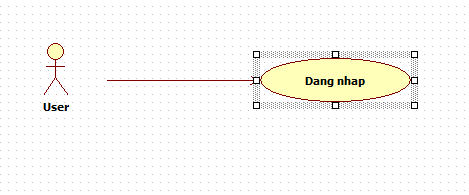
|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN USE CASE** | **DIỄN GIẢI** |
| 1 | Đăng kí tài khoản | Người dùng tự do đăng kí tài khoản để có thể đăng nhập vào hệ thống |
| 2 | Đăng nhập | Đăng nhập vào hệ thống |
| 3 | Đăng xuất | Thoát tài khoản ra khỏi hệ thống |
| 4 | Cập nhật ảnh đại diện | User cập nhật ảnh đại diện |
| 5 | Đổi mật khẩu | Đổi mật khẩu tài khoản |
| 6 | Tìm kiếm bạn bè | User tìm kiếm bạn bè thông qua username |
| 7 | Gửi lời mời kết bạn | User gửi lời mời kết bạn đến bạn bè |
| 8 | Phản hồi lời mời kết bạn | User có thể đồng ý |
| 9 | Tạo nhóm | Tạo nhóm với những người bạn để có thể trò chuyện với nhau |
| 10 | Thêm thành viên | Thêm thành viên mong muốn vào nhóm chat |
| 11 | Gửi tin nhắn | Gửi tin nhắn với mọi người |

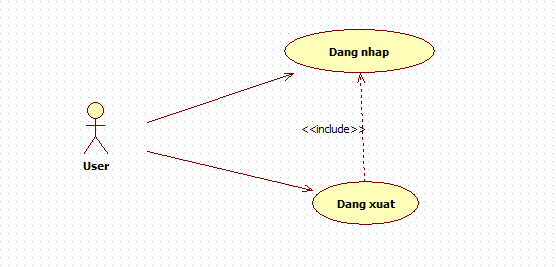
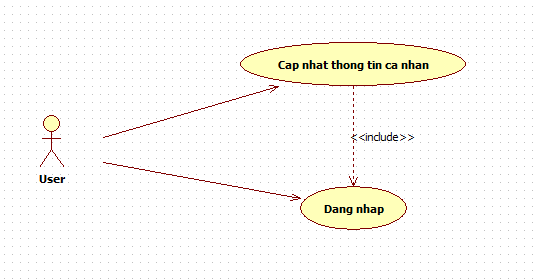
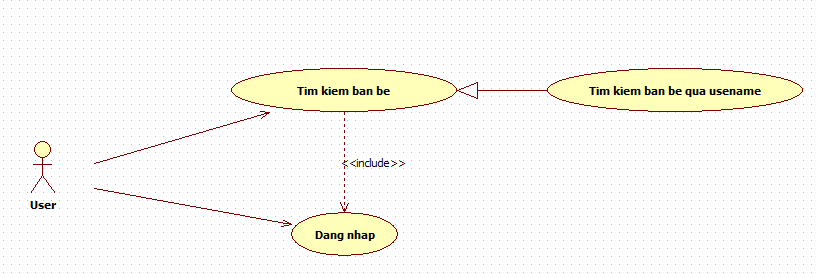
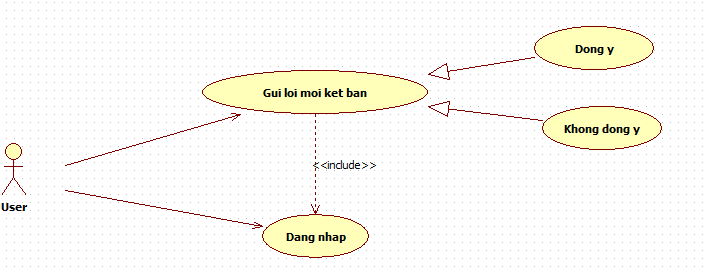
Bảng 2.6 Diễn giải mô tả các use case

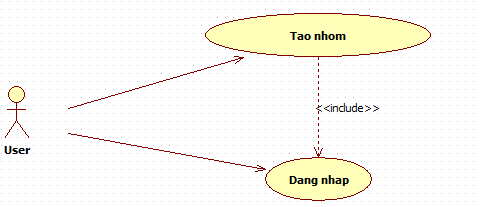
* **Use case Đăng kí tài khoản**



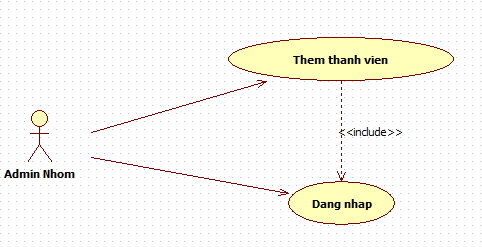
* **Use case Đăng nhập**

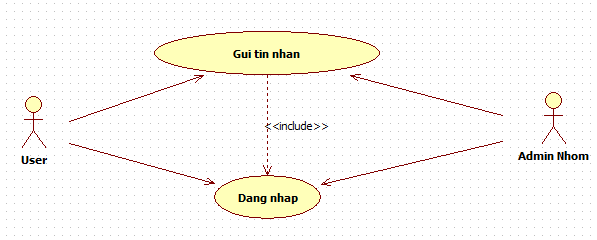


* **Use case Đăng xuất**
* **Use case Cập nhật thông tin cá nhân**
* **Use case Tìm kiếm bạn bè**
* **Use case Gửi lời mời kết bạn**
* **Use case Tạo nhóm**



* **Use case Thêm thành viên**

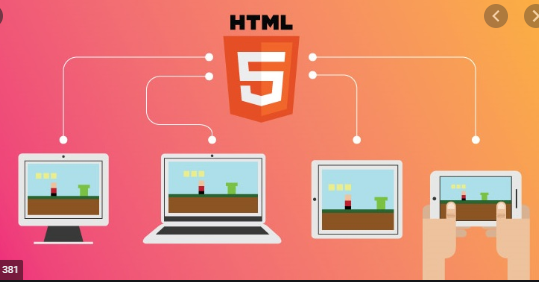


* **Use case Gửi tin nhắn**

**CHƯƠNG III: CÀI ĐẶT VÀ KẾT QUẢ**

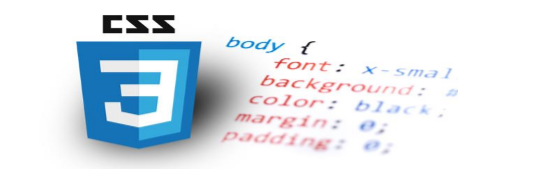
**3.1 Ngôn ngữ lập trình và công nghệ phía Client**

* **HTML 5**



HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

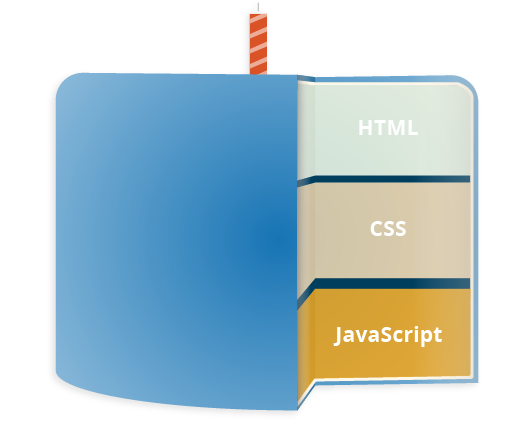
* **CSS 3**



Trong [tin học](https://vi.wikipedia.org/wiki/Tin_h%E1%BB%8Dc), các tập tin định kiểu theo tầng – dịch từ [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh) là Cascading Style Sheets (CSS) – được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ [HTML](https://vi.wikipedia.org/wiki/HTML) và [XHTML](https://vi.wikipedia.org/wiki/XHTML).[[1]](https://vi.wikipedia.org/wiki/CSS#cite_note-1) Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho [XML](https://vi.wikipedia.org/wiki/XML), [SVG](https://vi.wikipedia.org/wiki/SVG), [XUL](https://vi.wikipedia.org/w/index.php?title=XUL&action=edit&redlink=1). Các đặc điểm kỹ thuật của CSS được duy trì bởi [World Wide Web Consortium](https://vi.wikipedia.org/wiki/W3C) ([W3C](http://www.w3c.org/)). Thay vì đặt các thẻ quy định kiểu dáng cho văn bản [HTML](https://vi.wikipedia.org/wiki/HTML) (hoặc [XHTML](https://vi.wikipedia.org/wiki/XHTML)) ngay trong [nội dung](https://vi.wikipedia.org/w/index.php?title=N%E1%BB%99i_dung&action=edit&redlink=1) của nó, bạn nên sử dụng CSS.

**CSS3 là phiên bản thứ 3 và cũng là mới nhất của CSS, CSS3 được bổ sung thêm nhiều tính năng mới tiện lợi hơn CSS cho người dùng.**

* **JAVASCRIPT**



JavaScript là một ngôn ngữ lập trình hoặc ngôn ngữ kịch bản cho phép triển khai những chức năng phức tạp trên trang web như hiển thị các cập nhật nội dung kịp thời, tương tác với bản đồ, hoạt cảnh 2D/3D vv... - điều có sự hỗ trợ của JavaScript. Nó là lớp thứ ba của chiếc bánh tiêu chuẩn của các công nghệ web

* **JQUERY**

**jQuery** là thư viện javascript nổi bật nhất, phổ biến nhất trong [lập trình web](https://topdev.vn/blog/lap-trinh-web/). Ra đời vào năm 2006 bởi John Resig. Nó đã trở thành một thành phần không thể thiếu trong các website có sử dụng Javascript. Với slogan “Write less – Do more” (viết ít hơn – làm nhiều hơn) nó đã giúp lập trình viên tiết kiệm được rất nhiều thời gian và công sức trong việc thiết kế website.

* **BOOSTRAP 4**



Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn. Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tao ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive của bạn dễ dàng hơn và nhanh chóng hơn. Những điểm thuận lợi khi bản sử dụng bootstrap: Rất dễ để sử dụng đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt. Tính năng Responsive Bootstrap’s xây dựng sẵn reponsive css trên các thiết bị phones, tablets, và desktops. Trong Bootstrap 3 mobile-first styles là một phần của core framework. Và tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera)

**Bootstrap 4**(viết tắt là BS4), là phiên bản mới nhất của Bootstrap.

* **TEMPLATES EJS**



EJS là viết tắt của “Embedded JavaScript templating”, đây là một thư viện, được sử dụng để phân tích các tập tin ejs, và tạo ra HTML trả về cho client (trình duyệt).

**3.2 Ngôn ngữ lập trình và công nghệ phía Server**

* **NODE.JS**



Node.js là một [hệ thống phần mềm](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_ph%E1%BA%A7n_m%E1%BB%81m) được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là [máy chủ web](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7_web).[[1]](https://vi.wikipedia.org/wiki/Node.js#cite_note-1) Chương trình được viết bằng [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), sử dụng kỹ thuật điều khiển theo sự kiện, [nhập/xuất không đồng bộ](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADp/xu%E1%BA%A5t_kh%C3%B4ng_%C4%91%E1%BB%93ng_b%E1%BB%99&action=edit&redlink=1) để tối tiểu [tổng chi phí](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%95ng_chi_ph%C3%AD&action=edit&redlink=1) và tối đại [khả năng mở rộng](https://vi.wikipedia.org/w/index.php?title=Kh%E1%BA%A3_n%C4%83ng_m%E1%BB%9F_r%E1%BB%99ng&action=edit&redlink=1).[[2]](https://vi.wikipedia.org/wiki/Node.js#cite_note-2) Node.js bao gồm có [V8 JavaScript engine](https://vi.wikipedia.org/wiki/V8_(JavaScript_engine)) của Google, [libUV](https://github.com/joyent/libuv), và vài thư viện khác.

Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của [Joyent](https://vi.wikipedia.org/w/index.php?title=Joyent&action=edit&redlink=1).[[3]](https://vi.wikipedia.org/wiki/Node.js#cite_note-3)[[4]](https://vi.wikipedia.org/wiki/Node.js#cite_note-Node.js_pushes_JavaScript_to_the_server-side-4)

Mục tiêu ban đầu của Dahl là làm cho trang web có [khả năng push](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_ngh%E1%BB%87_push&action=edit&redlink=1) như trong một số ứng dụng web như [Gmail](https://vi.wikipedia.org/wiki/Gmail). Sau khi thử với vài ngôn ngữ Dahl chọn Javascript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.[[5]](https://vi.wikipedia.org/wiki/Node.js#cite_note-nodeupandrunning-5)

Vài môi trường tương tự được viết trong các ngôn ngữ khác bao gồm [Twisted](https://vi.wikipedia.org/w/index.php?title=Twisted_(ph%E1%BA%A7n_m%E1%BB%81m)&action=edit&redlink=1) cho [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Perl Object Environment](https://vi.wikipedia.org/w/index.php?title=Perl_Object_Environment&action=edit&redlink=1) cho [Perl](https://vi.wikipedia.org/wiki/Perl), [libevent](https://vi.wikipedia.org/w/index.php?title=Libevent&action=edit&redlink=1) cho [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và EventMachine cho [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). Khác với hầu hết các chương trình Javascript, Nodejs không chạy trên một trình duyệt mà chạy trên Server. Node.js sử dụng nhiều chi tiết kỹ thuật của [CommonJS](https://vi.wikipedia.org/w/index.php?title=CommonJS&action=edit&redlink=1).[[6]](https://vi.wikipedia.org/wiki/Node.js#cite_note-6) Nó cung cấp một môi trường [REPL](https://vi.wikipedia.org/w/index.php?title=Read-eval-print_loop&action=edit&redlink=1) cho kiểm thử tương tác.

* **FRAMEWORK EXPRESS.JS**

**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.

* **SOCKET.IO**



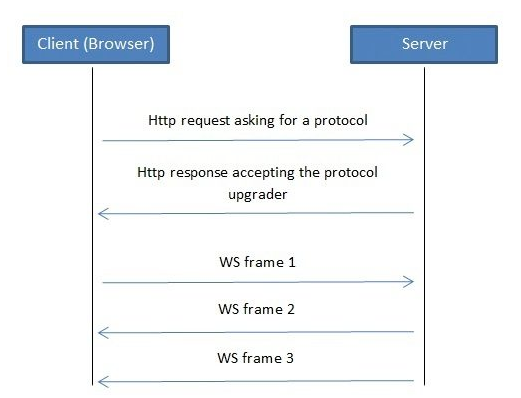
Socket.IO là một thư viện JavaScript cho các ứng dụng web thời gian thực. Nó cho phép giao tiếp hai chiều thời gian thực giữa máy khách và máy chủ web. Nó có hai phần: thư viện phía máy khách chạy trong trình duyệt và thư viện phía máy chủ cho Node.js. Cả hai thành phần có một API gần như giống hệt nhau.

**3.3 WEB SOCKET**

**WebSoket** là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

WebSockets mới xuất hiện trong HTML5, là một kỹ thuật Reverse Ajax. WebSockets cho phép các kênh giao tiếp song song hai chiều và hiện đã được hỗ trợ trong nhiều trình duyệt (Firefox, Google Chrome và Safari). Kết nối được mở thông qua một HTTP request (yêu cầu HTTP), được gọi là liên kết WebSockets với những header đặc biệt. Kết nối được duy trì để bạn có thể viết và nhận dữ liệu bằng JavaScript như khi bạn đang sử dụng một TCP socket đơn thuần.

Dữ liệu truyền tải thông qua giao thức HTTP (thường dùng với kĩ thuật Ajax) chứa nhiều dữ liệu không cần thiết trong phần header. Một header request/response của HTTP có kích thước khoảng 871 byte, trong khi với WebSocket, kích thước này chỉ là 2 byte (sau khi đã kết nối). Vậy giả sử bạn làm một ứng dụng game có thể tới 10,000 người chơi đăng nhập cùng lúc, và mỗi giây họ sẽ gửi/nhận dữ liệu từ server



Giao thức có hai phần: Bắt tay và truyền dữ liệu Ban đầu client sẽ gửi yêu cầu khởi tạo kết nối websocket đến server, server kiểm tra và gửi trả kết quả chấp nhận kết nối, sau đó kết nối được tạo và quá trình gửi dữ liệu có thể được thực hiện, dữ liệu chính là các Ws frame

**3.4 DATABASE MONGODB**



MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng

So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

**3.5 YÊU CẦU HỆ THỐNG**

**3.5.1 PHẦN CỨNG**

- Bộ xử lý: Pentium III 1.8 Ghz trở lên

- RAM: 2 GB trở lên

- HDD: 80 GB trở lên - Cấu hình mạng: Có Internet

- Độ phân giải màn hình: 960 x 640

Phải đảm bảo rằng các thiết bị phần cứng có đủ khả năng phục vụ hệ thống hoạt động hiệu quả và nhanh chóng

**3.5.2 HỆ ĐIỀU HÀNH**

Hệ thống phải tương tác tốt với các phần mềm sau:

- Hệ điều hành: Mọi hệ điều hành.

- Webserver: Express js.

- Cơ sở dữ liệu NoSql Mongodb

- Công nghệ: HTML5, CSS3, JQUERY, EJS, JAVASCRIPT

- Hệ thống có thể chạy tốt với các trình duyệt: Chrome, Cốc Cốc, Firefox...