



BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN NGÀNH


WEBSITE GIA SƯ



Ngành : Khoa Học Máy Tính

Giảng viên hướng dẫn: Thầy Nguyễn Chí Thanh

Sinh viên thực hiện :

 **Nguyễn Minh Tuệ - 1751010175**

 **Nguyễn Cao Tấn - 1751010131**

Lớp: TH73

TP.HCM 2020

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



WEB GIA SƯ

Đồ Án Ngành

NGÀNH: Khoa Học Máy Tính

 **Nguyễn Minh Tuệ - 1751010175**

 **Nguyễn Cao Tấn - 1751010131**

Giảng viên hướng dẫn: Thầy Nguyễn Chí Thanh

Lớp TH03

TP. HỒ CHÍ MINH, 2020

TRƯỜNG ĐẠI HỌC MỞ CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
THÀNH PHỐ HỒ CHÍ MINH Độc lập – Tự do – Hạnh phúc
KHOA CÔNG NGHỆ THÔNG TIN

GIẤY XÁC NHẬN

Tôi tên là:

Ngày sinh: Nơi sinh:

Chuyên ngành: Mã sinh viên:

Tôi đồng ý cung cấp toàn văn thông tin đồ án/ khóa luận tốt nghiệp hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án/ khóa luận tốt nghiệp vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên

(Ghi rõ họ và tên)

.....

**Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP
CỦA GIẢNG VIÊN HƯỚNG DẪN**

Giảng viên hướng dẫn:

Sinh viên thực hiện: **Lớp:**

Ngày sinh: **Nơi sinh:**

Tên đề tài:

.....

.....

.....

.....

.....

Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án/ khóa luận trước Hội đồng:

.....

.....

.....

.....

.....

.....

.....

.....

Thành phố Hồ Chí Minh, ngày ... tháng ... năm

Người nhận xét

.....

MỤC LỤC

DANH MỤC HÌNH VẼ	12
DANH MỤC BẢNG.....	13
DANH MỤC TỪ VIẾT TẮT	14
TÓM TẮT ĐỒ ÁN	Error! Bookmark not defined.
MỞ ĐẦU	2
Chương 1. TỔNG QUAN VỀ INTERNET, WORLD WIDE WEB, HTML	4
1.1 Khái niệm cơ bản về Internet	4
1.1.1. Giới thiệu Chung	4
1.1.2. Phân Loại.....	4
1.2 World Wide Web	6
1.2.1 Các Khái Niệm Cơ bản Về World Wide Web.....	6
1.2.2 Cách tạo một trang Web	6
1.2.3 Phân Loại Web.....	7
1.2.4 Web Server	9
1.2.5 Trình duyệt Web (Web Client hoặc Web Browser)	10
1.3 HTML	10
1.3.1 Giới thiệu về HTML	10
1.3.2 Cấu trúc Trang	11
1.3.3 Các thẻ HTML cơ bản	12
Chương 2. TÌM HIỂU VỀ NGÔN NGỮ NODEJS	13
2.1 Khái Niệm về NodeJs	13
2.2 Tại sao chúng ta nên dùng NodeJS	14
2.3 Sơ đồ hoạt động của NodeJS	15
2.4 Các kiểu dữ liệu	15
2.5 Biến và Hằng	18

2.5.1	Biến	18
2.5.2	Hằng	19
2.6	Các đặc tính của NodeJS	19
2.7	NPM	20
2.8	Toán tử trong NodeJs.....	22
2.8.1	Toán tử gán	22
2.8.2	Toán tử số học.....	22
2.8.3	Toán tử so sánh	23
2.8.4	Toán tử logic	24
	Bảng 2.3: bảng toán tử logic	25
2.8.5	Toán tử kết hợp.....	25
2.9	Hàm trong NodeJS	26
2.10	Câu lệnh rẽ nhánh	27
2.10.1	Câu Lệnh If-ELSE.....	27
2.10.2	Câu lệnh lồng nhau	27
2.10.3	Switch-Case.....	28
2.11	Đối tượng trong NodeJs (object).....	30
2.11.1	Thuộc tính và phương thức trong đối tượng	30
2.11.2	Gọi thuộc tính và phương thức.....	32
2.11.3	Đối tượng trong đối tượng.....	33
2.11.4	Mảng trong đối tượng.....	34
2.12	Khởi tạo Server trong NodeJs với HTTP module.....	34
2.12.1	Giới thiệu về HTTP module	34
2.12.2	Khởi tạo Server với HTTP module	34
2.12.3	Các phương thức và thuộc tính được dùng trong request và response	35
2.13	Đọc ghi file bằng module fs trong Node.js	37
2.13.1	Giới thiệu về module fs	37
2.13.2	Đọc files với module fs	37

2.13.3	Ghi file.....	38
2.14	Module URL	39
2.14.1	Khai báo sử dụng Module URL	39
2.15	Upload file trong NodeJs	39
2.15.1	Tổng quan.....	39
2.15.2	cài đặt module formidable	40
2.15.3	code upload.....	40
2.16	Sự kiện trong NodeJs	41
2.16.1	Event loop.....	41
2.17	Tạo client request trong NodeJs	42
2.17.1	Cài đặt.....	42
2.17.2	các phương thức mà module request hỗ trợ:	43
Chương 3.	CƠ SỞ DỮ LIỆU MONGODB	44
3.1	Giới thiệu về MongoDB.....	44
3.1.1	Lợi thế của Mongodb.....	44
3.1.2	đặc điểm của Mongodb.....	44
3.2	Mục đích sử dụng cơ sở dữ liệu Mongodb	44
3.3	Thuật ngữ trong Mongodb	45
3.4	Kiểu dữ liệu trong Mongodb	46
3.5	Một số câu lệnh truy vấn cơ bản của Mongodb.....	48
3.6	Truy vấn có điều kiện trong MongoDB	49
3.7	MongoDB Replication	50
3.8	Cài đặt mongodb.....	51
3.9	Kết nối NodeJs với mongodb	51
3.10	Đóng kết nối NodeJs đến Mongodb	51
3.11	Sử dụng Mongodb	52
Chương 4.	PHÂN TÍCH THIẾT KẾ HỆ THỐNG	53
4.1	Tác nhân và use case hệ thống	53
4.1.1	Các tác nhân (Actor)	53

4.2	Các use case	54
4.2.1	Đăng nhập:	54
4.2.2	Đăng ký:	55
4.2.3	Đăng tin tuyển dụng:	56
4.2.4	Tạo hồ sơ gia sư:	58
4.2.5	Nạp tiền vào tài khoản	60
4.2.6	Tra cứu thông tin	62
4.2.7	Hỗ trợ khách hàng	64
4.2.8	Thay đổi quy định	65
4.2.9	Lập báo cáo	66
4.3	Mô hình quy trình nghiệp vụ (Business Process Modeling)	67
4.3.1	Quy trình lập báo cáo:	67
4.3.2	Quy trình tra cứu lớp:	68
4.3.3	Quy trình đăng tin tuyển dụng:	69
4.3.4	Quy trình tạo hồ sơ gia sư:	69
4.3.5	Quy trình đăng ký nhận lớp:	70
4.3.6	Quy trình hỗ trợ khách hàng:	71
4.3.7	Quy trình thay đổi quy định:	71
4.4	mô hình lớp (Class Diagram)	72
4.5	Sơ đồ trình tự (Sequence Diagram)	73
4.5.1	Đăng nhập:	73
4.5.2	Đăng ký:	74
4.5.3	Tạo tin tuyển dụng (lớp mới):	75
4.6	Thiết kế giao diện	75
4.6.1	Giao diện trang chủ	75
4.6.2	Giao diện đăng nhập	76
4.6.3	Giao diện Đăng ký	76
4.6.4	Trang Tin tuyển dụng lớp	77
4.6.5	Trang tạo lớp	78

4.6.6	Quản trị đăng nhập.....	79
4.6.7	Giao diện quản trị	79
4.6.8	Chi tiết người dùng	80
4.6.9	Trang tạo thông tin lớp mới	80
TÀI LIỆU THAM KHẢO		82
PHỤ LỤC.....		82

LỜI CẢM ƠN

Xin tỏ lòng biết ơn sâu sắc đến

Điều đầu tiên em xin phép được gửi lời cảm ơn đến Thầy Nguyễn Chí Thanh Giảng viên môn học đã nhiệt tình giúp đỡ, đóng góp ý kiến tận tình. Mặc dù thầy rất bận rộn nhưng luôn dành thời gian giúp đỡ cũng như chỉ bảo cho em trong khoảng thời gian làm đồ án.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong Khoa Công Nghệ Thông Tin, văn phòng công tác sinh viên nói chung, Ngành Khoa học máy tính nói riêng. Đã dạy và tạo điều kiện giúp đỡ em có kiến thức vững vàng về các môn đại cương cũng như các môn chuyên ngành để chúng em có thể thực hiện bài đồ án thành công tốt đẹp.

Cuối cùng, em xin chân thành gửi lời cảm ơn đến gia đình và bạn bè, luôn tạo điều kiện thuận lợi, giúp đỡ, quan tâm, động viên em trong suốt quá trình hoàn thành bài đồ án.

.....Ngày....Tháng....Năm.....

[illegible]

DANH MỤC HÌNH VẼ

TT	Hình	Tên hình	Trang
1	Hình 1.1	sơ đồ hoạt động của Web Động.	7
2	Hình 1.2	sơ đồ hoạt động của Web tĩnh.	7
3	Hình 1.3	sơ đồ hoạt động của Web Server.	8
4	Hình 2.1	Sơ đồ hoạt động của NodeJs.	13
5	Hình 2.2	Ảnh minh họa khởi tạo một dự án.	17
6	Hình 2.3	Ảnh minh họa cách khởi tạo Package.	18
7	Hình 4.1	Mô hình Use case	45
8	Hình 4.2	hình quy trình lập báo cáo	61
9	Hình 4.3	Quy trình tra cứu lớp	62
10	Hình 4.4	quy trình đăng tin tuyển dụng	62
11	Hình 4.5	Quy trình tạo hồ sơ gia sư	63
12	Hình 4.6	quy trình đăng ký nhận lớp	63
13	Hình 4.7	quy trình hỗ trợ khách hàng	64
14	Hình 4.8	quy trình thay đổi quy định	64
15	Hình 4.9	mô hình lớp	65
16	Hình 4.10	Giao diện trang chủ	62
17	Hình 4.11	Giao diện đăng nhập	62
18	Hình 4.12	Giao diện đăng ký	63
19	Hình 4.13	Giao diện Tuyển dụng lớp	63
20	Hình 4.14	Giao diện tạo lớp	64
21	Hình 4.15	Giao diện quản trị đăng nhập	64
22	Hình 4.16	Giao diện Quản trị	65
23	Hình 4.17	Giao diện chi tiết người dùng	65
24	Hình 4.18	Giao diện tạo thông tin lớp mới	66

DANH MỤC BẢNG

Bảng 2.1: Bảng toán tử số học.	19
Bảng 2.2: Bảng toán tử so sánh.....	20
Bảng 2.3: bảng toán tử logic	21
Bảng 2.4: bảng phương thức module request.	36
Bảng 3.1: bảng thuật ngữ trong Mongoddb	39
Bảng 3.2: bảng kiểu dữ liệu trong MongoDB.....	40
Bảng 3.3: bảng cú pháp truy vấn có điều kiện trong MongoDB	42
Bảng 4.1: mô tả đăng nhập.....	46
Bảng 4.2: Mô tả đăng ký.....	48
Bảng 4.3: mô tả đăng tin tuyển dụng	50
Bảng 4.4: bảng mô tả tạo hồ sơ.....	52
Bảng 4.5: bảng mô tả nạp tiền vào tài khoản	55
Bảng 4.6: Bảng mô tả tra cứu thông tin	57
Bảng 4.7: bảng mô tả hỗ trợ khách hàng	58
Bảng 4.8: Mô Tả thay đổi quy định	60
Bảng 4.9: mô tả quy trình lập báo cáo	61

DANH MỤC TỪ VIẾT TẮT

TT	Từ viết tắt	Ý nghĩa
1	CNTT	Công nghệ thông tin
2	NCKH	Nghiên cứu khoa học
3	DAMH	Đồ án môn học
4	SDT	Số điện thoại
5	CMND	Chứng minh nhân dân
6	CSDL	Cơ sở dữ liệu
7	WWW	World Wide Web
8	Email	ELECTRONIC MAIL
9	HTML	Hypertext Markup Language
10	IIS	Internet Information Services
11	IE	Internet Explorer
12	API	Application Programming Interface
13	HTTP	HyperText Transfer Protocol
14	FS	File system

TỔNG QUAN ĐỒ ÁN

Với mức độ xây dựng phức tạp và quy mô ứng dụng rộng rãi, cho nên vấn đề về thời gian để xây dựng “Website Gia sư” chỉ tạm thời dừng lại ở mức tìm hiểu về ngôn ngữ lập trình NodeJS, và ứng dụng vào thực tế thực nghiệp trang web thuê gia sư này.

Website có mục đích nhằm cung cấp cho các khách hàng (học sinh, sinh viên) các thông tin chính xác về môn học giảng dạy của các gia sư (Tên, môn học giảng dạy, lời giới thiệu về bản thân) và các thức thuê gia sư. Các thông tin của gia sư sẽ được cập nhật thường xuyên và nhanh chóng nhất có thể. Vì vậy có thể tạo điều kiện cho học sinh, sinh viên tiết kiệm được nhiều thời gian tìm người dạy kèm, giá cả theo ý muốn đồng thời kèm theo các thông tin mà khách hàng thực sự cần.

Vì trong thời gian thực hiện còn một số hạn chế cho nên website thuê gia sư chỉ dừng lại ở các chức năng cơ bản như đăng nhập, đăng ký, tạo lớp, tạo hồ sơ gia sư, tra cứu thông tin. Trong tương lai, hệ thống website sẽ tiếp tục phát triển hơn nữa và mở rộng phạm vi chức năng để hoàn thiện được một website chuyên nghiệp.

MỞ ĐẦU

Hiện nay, việc ứng dụng công nghệ thông tin vào các lĩnh vực của xã hội và việc tin học hóa được xem là một yếu tố quan trọng hàng đầu của các tổ chức, các công ty, nhà nước, nó đóng vai trò cốt yếu trong việc tạo ra những bước đột phá trong công nghệ.

Song song với việc phát triển không ngừng nghỉ về khoa học máy tính, kỹ thuật máy tính và mạng truyền thông, thương mại điện tử. đã được chinh phục bởi những công nghệ có đặc tính kỹ thuật tiên tiến, và những công nghệ vượt trội có mặt trên thế giới hiện nay. Ngay tại thời điểm này, internet đã trở thành một sản phẩm có giá trị đặc biệt mà không có đối thủ thay thế và không thể phủ nhận được tầm quan trọng của nó. Internet là nền tảng cốt yếu quan trọng nhất để truyền tải, trao đổi thông tin trên phạm vi toàn cầu.

Ngày nay, chúng ta sẽ nhận thấy mọi việc sẽ dễ dàng hơn khi chúng ta có Internet, mọi việc liên quan đến thông tin tài nguyên sẽ được truyền tải nhanh chóng. Chỉ cần bạn sở hữu một thiết bị thông minh như Laptop hoặc điện thoại có kết nối internet và kèm theo đó là một dòng dữ liệu theo chính nhu cầu người sử dụng muốn, thì lập tức mọi thứ có liên quan đến dữ liệu người dùng sẽ xuất hiện ngay trước mắt chỉ trong vài giây, có đủ các thông tin hình ảnh, âm thanh và các tài nguyên khác mà người sử dụng cần.

Nhờ có Internet mà chúng ta có thể tiện lợi thực hiện được nhiều công việc với tốc độ tối đa và chi phí thấp hơn so với các hoạt động không có internet. Chính vì vậy đã thúc đẩy mạnh sự phát triển thịnh vượng về thương mại điện tử trên khắp thế giới, giúp tăng nhanh năng suất chất lượng cuộc sống con người.

Giờ đây, thương mại điện tử đã khẳng định vị thế và vai trò của mình trong hoạt động sản xuất, kinh doanh thúc đẩy sự phát triển trong lĩnh vực công nghệ. Đối với doanh nghiệp tư nhân và nhà nước, lớn nhỏ, việc quảng bá marketing và giới thiệu sản phẩm đến khách hàng là một vấn đề cần thiết cho doanh nghiệp. vậy việc quảng bá cho doanh nghiệp trước tiên là xây dựng một website để quảng bá tất cả các sản phẩm của doanh nghiệp.

Theo nhu cầu thiết yếu của thị trường cũng như học tập và phát triển hơn về công nghệ này, nhóm em đã quyết định thực hiện đồ án với chủ đề là “xây dựng website gia sư” cho người đang có nhu cầu tuyển gia sư.

Người sử dụng khi truy cập website sẽ được chia làm 2 dạng, khách hàng thuê gia sư và gia sư. Khách hàng truy cập vào website để tìm thông tin gia sư phù hợp. gia sư truy cập website để tìm khách hàng phù hợp. tất cả các thông tin gia sư sẽ được chia sẻ cho khách hàng khi cần.

Với sự chỉ bảo và hướng dẫn tận tình của thầy Nguyễn Chí Thanh em đã thực hiện hoàn thành bài báo cáo đồ án này một cách hoàn thiện. chúng em đã cố gắng rất nhiều để tìm hiểu, phân tích hệ thống và thiết kế nhưng không thể không tránh khỏi những thiếu sót và lỗi sai. Chúng em rất mong quý thầy quý cô cho chúng em lời góp ý qua đó em có thể sửa những lỗi sai và hoàn thành bài báo cáo tốt hơn nữa. chúng em xin chân thành cảm ơn quý thầy cô.

Sinh viên thực hiện:

- Nguyễn Cao Tấn
- Nguyễn Minh Tuệ

Lớp: TH73

.....Ngày.....Tháng.....Năm

Chương 1. TỔNG QUAN VỀ INTERNET, WORLD WIDE WEB, HTML

1.1 Khái niệm cơ bản về Internet

1.1.1. Giới thiệu Chung

- Internet hay Mạng là một hệ thống kết nối thông tin toàn cầu có thể được truy nhập công đồng gồm các mạng máy tính được liên kết lại với nhau. Hệ thống này sẽ truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) và dựa trên một giao thức đó là liên mạng đã được chuẩn hóa (giao thức IP).
- Hệ thống này bao gồm rất nhiều mạng máy tính nhỏ hơn của các doanh nghiệp, của các nhà, viện nghiên cứu và các trường đại học trên thế giới, của người dùng riêng và các chính phủ trên toàn cầu, được liên kết bằng một loạt các công nghệ như mạng điện tử, không dây và mạng quang. Internet đem lại cho chúng ta hàng loạt các tài nguyên và dịch vụ thông tin, chẳng hạn như các tài liệu và ứng dụng siêu văn bản được liên kết với nhau bằng World Wide Web, điện thoại, thư điện tử, và chia sẻ file, vân vân.
- Hệ thống internet được thành lập từ những năm 80 bắt nguồn từ mạng APARRNET của bộ quốc phòng hoa kỳ.
- Khi kết nối mạng internet chúng ta có thể tìm được rất nhiều tài nguyên để nghiên cứu, học tập, trao đổi thông tin, đặt mua, bán hàng một cách rất dễ dàng bằng một vài dòng lệnh và một lần click chuột.
- Từng máy tính cá nhân trên Internet được gọi là một host. Mỗi host có chức năng và khả năng như nhau về truyền thông đến máy khác trên Internet. Một số host khác được nối đến mạng bằng đường dẫn Dial-up tạm thời, còn số khác được nối bằng đường dẫn mạng thật (như Tokenring, Ethernet...).

1.1.2. Phân Loại

Tất cả máy tính trên mạng được chia làm 2 nhóm đó chính là Client và Server:

- Máy Client: là tất cả Máy khách trên mạng và chứa các chương trình của Client
- Máy Server: Máy phục vụ hay được gọi là Máy chủ. Chứa các tài nguyên (tập tin, tài liệu...) và chương trình Server dùng chung cho tất cả máy khách. Server sẽ luôn ở trạng thái chờ để yêu cầu và đáp ứng yêu cầu của Client.
- Internet Server: Là tất cả server cung cấp dịch vụ Internet như FPT Server, Web Server, Mail Server...

Các dịch vụ phổ biến được dùng trên Internet:

- Tổ chức, khai thác nguồn thông tin trên web (WWW – World Wide Web): người dùng, khách hàng có thể dễ dàng truy cập để xem nội dung các trang web khi máy tính được kết nối với Internet
- **Dịch vụ ELECTRONIC MAIL** (được viết tắt là Email và được gọi là thư điện tử) đây là dịch vụ trao đổi một số thông tin trên Internet thông qua các hộp thư điện tử, được sử dụng rộng rãi và rất phổ biến, tiện lợi, nhanh chóng, với chi phí thấp.
- **Dịch vụ FPT**: Để trao đổi và truyền thông với tất cả các máy tính khác, hầu như mọi máy tính trên Internet phải được hỗ trợ giao thức chung đó là TCP/IP (Transmission Control Protocol/Internet Protocol - là giao thức điều khiển truyền, gửi dữ liệu và giao thức Internet), đồng thời một giao thức đồng bộ, cho phép truyền thông tin, thông điệp từ nhiều nguồn khác nhau và đến nhiều đích khác nhau. Một ví dụ nhỏ, ta có thể vừa lấy thư từ trong hộp thư và đồng thời truy cập đến trang Web. TCP mang lại sự đảm bảo tính an toàn tương đối về dữ liệu, IP là giao thức mà nó có thể chi phối phương thức vận chuyển các nguồn dữ liệu trên Internet.

1.2 World Wide Web

1.2.1 Các Khái Niệm Cơ bản Về World Wide Web

- Như đã được đề cập ở trên, “www” chính là tên viết tắt của World Wide Web và được gọi tắt là Web, nơi đây ta có thể được hiểu là một không gian thông tin toàn cầu. Tại đây, mọi người có thể sử dụng để viết, đọc, chia sẻ thông tin thông qua mạng Internet.
- Không gian này được xác định nhờ vào đường dẫn URL được gán và liên kết với nhau bởi các siêu liên kết khi truy cập thông qua Internet. World Wide Web còn là thuật ngữ có nhiệm vụ đề cập tới tất cả các trang HTML kết nối với nhau, có thể truy cập thông qua Internet.
- Web là một dịch vụ được xem là phổ biến nhất hiện nay trên Internet, 85% các giao dịch trên Internet ước lượng thuộc về WWW. Ngày nay số lượng trang Web trên thế giới đã đạt tới con số rất lớn. WWW còn cho phép bạn truy xuất thông tin hình ảnh, văn bản, âm thanh, video trên toàn thế giới. Thông qua các Website, các công ty có thể giảm được thiểu tối đa chi phí in ấn và phân phát tài liệu cho khách hàng ở nhiều nơi.

1.2.2 Cách tạo một trang Web

Chúng ta có rất nhiều cách để tạo một trang Web.

- Chúng ta có thể tạo một trang Web ngay trên bất kì chương trình xử lý văn bản nào.
- Ngoài ra còn có thể thiết kế Web bằng các phần mềm chuyên nghiệp như: DreamWeaver, Visual studio code, Netscape Editor, FrontPage.
- Phần mềm DreamWeaver và Visual Studio code là 2 loại phần mềm chuyên nghiệp sẽ giúp chúng ta thiết kế một trang web nhanh và dễ dàng hơn rất nhiều vì phần lớn các thư viện và mã lệnh được cung cấp sẵn trong code của chúng ta nên khi thao tác chúng ta sẽ thực hiện một cách rất dễ dàng.

- Tạo Web bằng cách viết mã nguồn bởi trình soạn thảo văn bản như: NodePad, WorldPad. Đây là những chương trình soạn thảo trong văn bản có sẵn trong Window.
- Ngoài ra còn có thể thiết kế bằng cách dùng Web Wizard và những công cụ của Word 1997 hoặc Word 2000 để thiết kế.
- Nhằm xây dựng một trang web có tính thương mại và ứng dụng cao được gọi là hoàn chỉnh. Thì chúng ta phải kết hợp giữa những phương thức sau: Server Script và Client Script (kịch bản trên máy chủ và kịch bản trên máy khách) cộng thêm một trong những trình xử lý cơ sở dữ liệu sau: MS Access, SQL Server, MySQL và Oracle.
- Ngoài ra khi chúng ta muốn triển khai ứng dụng web trên Internet, ngoài các điều kiện về cấu hình phần cứng, thì chúng ta cần có trình chủ Web hay được gọi là Web Server.

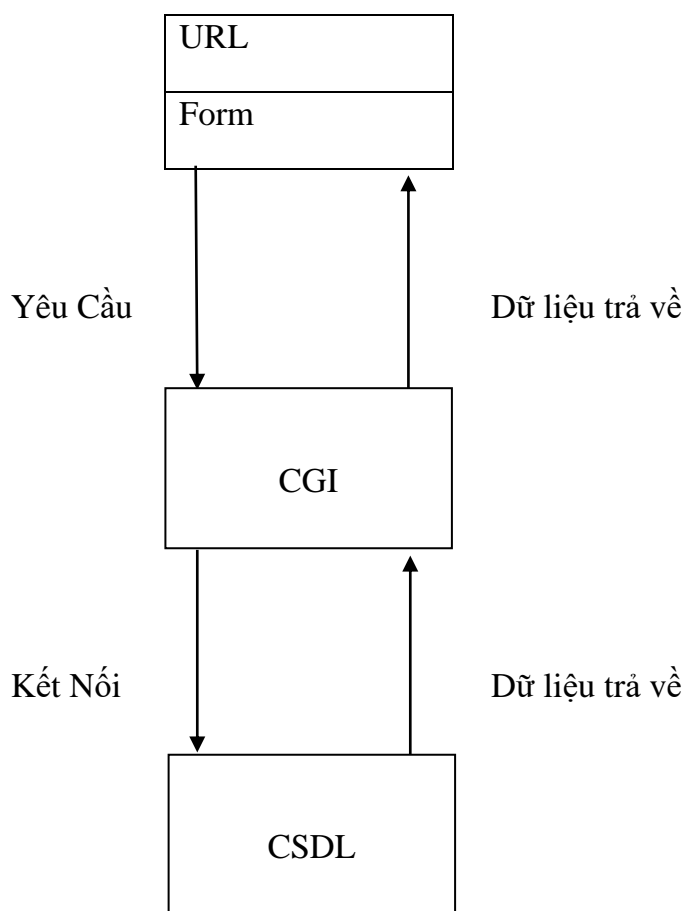
1.2.3 Phân Loại Web

1.2.3.1 Web Động

Và ở đây Web động được hiểu là gì?

- Web Động là một tập hợp các dữ liệu số hóa được tổ chức thành các cơ sở dữ liệu, các dữ liệu được số hóa được gọi ra trình diễn trên các trang web dưới dạng văn bản, âm thanh, hình ảnh. Web còn có thêm các phần truy xuất dữ liệu và xử lý thông tin còn website tĩnh thì không.
- Khác với web tĩnh, web động luôn luôn có thông tin mới do các thông tin này sẽ được cập nhật bởi các phần mềm quản trị website do các công ty thiết kế website cung cấp. Các thông tin mới này sẽ được lưu vào cơ sở dữ liệu (Database) của website và đưa ra sử dụng dựa theo yêu cầu của người dùng.
- Về cơ bản nội dung của trang Web động như một trang Web tĩnh, ngoài ra còn có thể thao tác với CSDL (cơ sở dữ liệu) để đáp ứng nhu cầu phức tạp của một trang Web. Sau khi nhận được yêu cầu từ

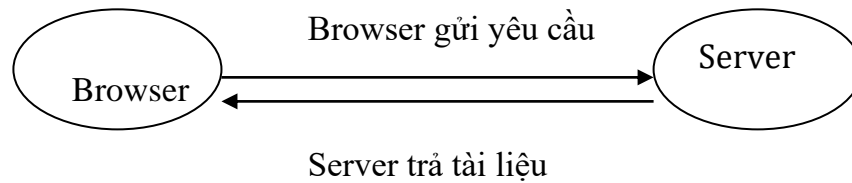
Web Client, chẳng hạn như một truy vấn từ một CSDL đặt trên Server, ứng dụng Internet Server sẽ truy vấn CSDL này, và tạo một trang HTML có chứa kết quả truy vấn rồi sau đó gửi trả cho người dùng.



Hình 1.1: sơ đồ hoạt động của Web Động.

1.2.3.2 Web Tĩnh

Hoạt Động của trang Web tĩnh được hoạt động theo mô hình sau:



Hình 1.2 : sơ đồ hoạt động của Web tĩnh.

Là trang web sử dụng hoàn toàn ngôn ngữ HTML, sau khi trang HTML được tải từ máy chủ xuống, trình duyệt sẽ có nhiệm vụ biên dịch mã và hiển thị nội dung của trang web, và người dùng dường như không thể tương tác với trang web.

Ưu Điểm :

- Tốc độ truy cập nhanh vì đó là những file HTML
- Chi phí đầu tư giá rẻ vì không phải chi trả nhiều tiền cho coder
- Giao diện có thể design theo một hướng đột phá mới lạ
- Thân thiện với môi trường computer vì có thể đặt tên file tùy ý

Nhược Điểm :

- Rất khó để quản lý nội dung.
- Và khó để nâng cấp hoặc bảo trì.

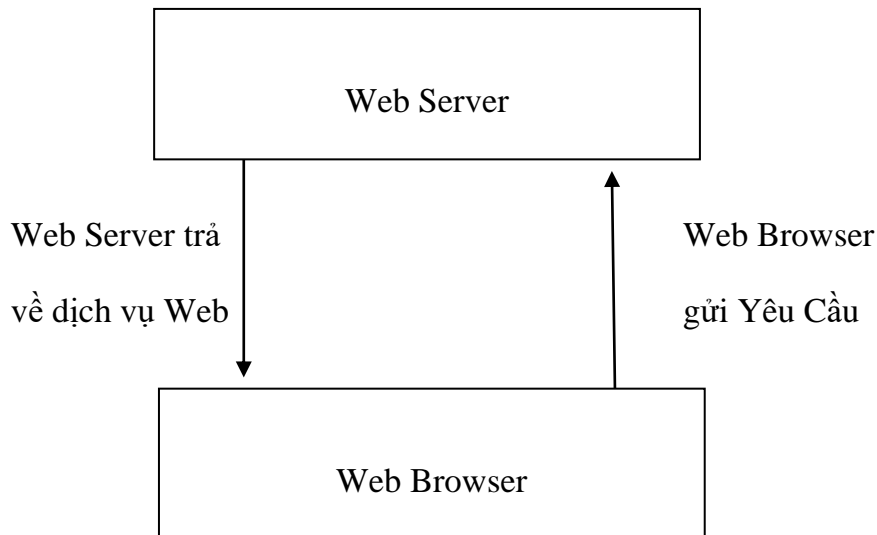
1.2.4 Web Server

Webserver: là một máy tính có kết nối vào Internet và run các phần mềm được thiết kế. Webserver đóng vai trò một chương trình xử lý các nhiệm vụ xác định, xử lý tổ hợp dữ liệu, như tìm trang thích hợp, kiểm tra dữ liệu hợp lệ.... Webserver còn là nơi lưu trữ cơ sở dữ liệu (Database), là phần mềm đảm nhiệm vai trò Server cung cấp dịch vụ Web.

Webserver hỗ trợ các công nghệ nổi tiến khác nhau:

- Apache HTTP Server, Nginx, IIS (Internet Information Services),
lighttpd....

Cách thức hoạt động của một Web Server:



Hình 1.3: sơ đồ hoạt động của Web Server.

1.2.5 Trình duyệt Web (Web Client hoặc Web Browser)

Trình duyệt Web là được xem công cụ truy xuất dữ liệu trên mạng, là phần mềm giao diện trực tiếp với người sử dụng. Nhiệm vụ của Web Browser là nhận các yêu cầu của người sử dụng, gửi tất cả các yêu cầu đó qua mạng tới các Web Server và nhận các dữ liệu cần thiết từ Server để hiển thị lên màn hình. Để vận hành và sử dụng dịch vụ WWW, Client hay người dùng cần có 1 chương trình duyệt Web, kết nối vào Internet thông qua một ISP. Các trình duyệt thông dụng hiện nay là: Microsoft Internet Explorer(IE), Mozilla Firefox và Google Chrome... Trong đó Internet Explorer là một trình duyệt chuẩn cho phép trình bày nội dung được Web server cung cấp, được phép đăng kí tới bất kì Website nào trên Internet, hỗ trợ trình bày trang Web.

1.3 HTML

1.3.1 Giới thiệu về HTML

HTML là chữ viết tắt của HyperText Markup Language được World Wide Web Consortium đặt ra. Một trang web là sự kết hợp giữa các văn bản kèm theo là các thẻ HTML. Các tập tin HTML đều có đuôi là .html hoặc .htm.

HTML giúp định dạng trong văn bản nhờ các thẻ. Các thẻ HTML có thể liên kết từ hoặc cụm từ với các tài liệu khác trên Internet. Hầu hết các thẻ HTML đều có thẻ đóng và thẻ mở. thẻ đóng được kí hiệu kèm dấu “/”

Ví dụ: <head></head>

Ngôn ngữ HTML được qui định cú pháp không phân biệt chữ hoa và chữ thường. và không được chứa các khoảng trắng trong định nghĩa thẻ.

Ví dụ: có thẻ khai báo

<html></HTML>

1.3.2 Cấu trúc Trang

Một trang HTML có cấu trúc trang khá đơn giản và được quy định như sau:

<html>

<head>

<title></title>

</head>

<body>

(các loại thẻ HTML và nội dung hiển thị)

</body>

<footer></footer>

</html>

1.3.3 Các thẻ HTML cơ bản

1. **Thẻ <title>...</title>:** Tạo tiêu đề trang trên thanh tiêu đề, đây là thẻ bắt buộc và nằm trong thẻ <head>.
2. **Thẻ <head>...</head>:** Tạo đầu mục trang
3. **Thẻ <body>...</body>:** Tất cả các thông tin khai báo có trong thẻ <body> đều có thể được xuất hiện trên trang Web. Những thông tin này được nhìn thấy trên trang Web.
4. **Các thẻ định dạng khác.** Thẻ <p>...</p>: Tạo một đoạn hoặc một câu mới. Thẻ ... : Thay đổi kích cỡ, phong chữ và màu kí tự...
5. **Thẻ định dạng bảng:** <table>...</table> Đây là loại thẻ định dạng một bảng trên trang Web. khi khai báo thẻ này, bạn có thể phải khai báo các thẻ cột <td> và thẻ hàng <tr> cùng với các thuộc tính đi kèm của nó.
6. **Thẻ hình ảnh :** Cho phép bạn được chèn hình ảnh vào một trang Web. Thẻ này là loại thẻ thuộc loại không có thẻ đóng.
7. **Thẻ liên kết <a>... :** Là thẻ dùng để liên kết các trang Web hoặc liên kết đến địa chỉ Internet.
8. **Các thẻ Input:** Thẻ Input cho phép người dùng nhập dữ liệu hay chỉ thị thực thi một hành động, thẻ Input bao gồm các loại thẻ như: text, password, submit, button, reset, checkbox, radio, image.
9. **Thẻ Textarea:** < Textarea>.... < \Textarea>: Thẻ Textarea cho phép client (người dùng) nhập liệu văn bản. loại thẻ này không thể giới hạn chiều dài lớn nhất trên một trang Web.
10. **Thẻ Select:** Loại thẻ Select cho phép người được dùng chọn phần tử trong các tập phương thức đã định nghĩa trước. thẻ Select còn cho phép người dùng (Client) chọn một phần tử nào đó trong danh sách phần tử thì thẻ Select sẽ giống như combobox. thẻ Select còn cho

phép người dùng (Client) chọn nhiều phần tử cùng một lần trong danh sách phần tử, thẻ Select đó là dạng listbox.

11. **Thẻ Form:** Khi bạn muốn submit dữ liệu người dùng nhập từ Website phía người dùng (Client) đến phía máy chủ (Server), có hai cách để làm điều này ứng với hai phương thức POST và GET trong thẻ form. Trong một Website còn có thể có nhiều (thẻ Form) khác nhau, nhưng các loại thẻ Form này không được phép lồng nhau, mỗi thẻ form được khai báo hành động (action) chỉ đến một trang khác.

Chương 2. TÌM HIỂU VỀ NGÔN NGỮ NODEJS

2.1 Khái Niệm về NodeJs

- Nodejs là một nền tảng độc lập có mã nguồn mở được phát triển và xây dựng ở trên Javascript Runtime của chrome. NodeJS được viết bởi Ryan Dahl vào năm 2009 dựa trên Javascript engine V8 của Google Qua đó chúng ta có thể xây dựng được các ứng dụng mạng nhanh chóng và rất dễ dàng mở rộng.
- về bản chất NodeJS còn được xem là JavaScript. Trước khi NodeJS ra đời JavaScript chủ yếu được sử dụng để nhúng vào HTML để sử dụng trên các website để người dùng có thể tương tác với trình duyệt. Cho đến hiện nay Javascript đang có một số cải tiến rất đáng kể. trước đó javascript rất thịnh hành được sử dụng đến để tạo một hiệu ứng cho trang Web. Nhưng giờ đây Javascript đã trở thành ngôn ngữ chính và được sử dụng nhiều như các ngôn ngữ khác: C, Ruby, PHP, v.v.
- NodeJS cho phép chúng ta sử dụng JavaScript trên Server và nó còn cho phép chúng ta viết JavaScript bên ngoài trình duyệt.
- NodeJS Khác với hầu hết các chương trình Javascript, Nodejs không chạy trên một trình duyệt nhưng lại chạy trên Server. Nodejs sử dụng nhiều kỹ thuật chi tiết của CommonJS. NodeJS cung cấp một môi trường REPL cho kiểm thử tương tác.

- Tất nhiên Nodejs thừa hưởng hầu như toàn bộ sức mạnh vốn có của Javascript và đã cho chúng ta thấy một cách nhìn toàn diện một cách mới mẻ về việc phát triển web động bằng NodeJS.

2.2 Tại sao chúng ta nên dùng NodeJS

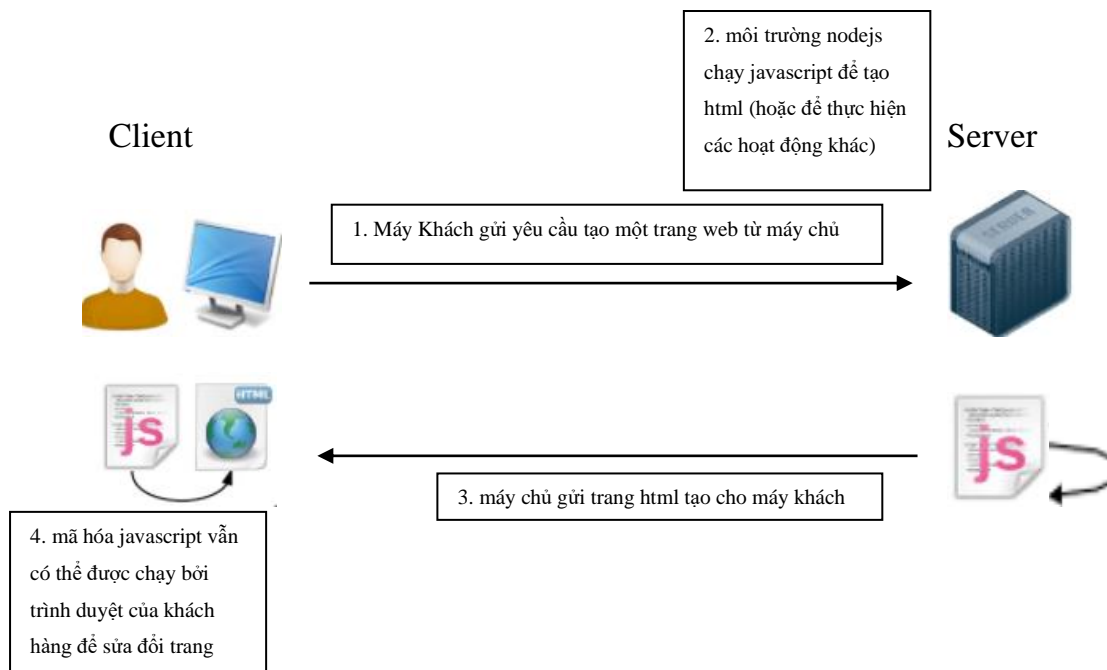
Để thiết kế một website động chúng ta có thể sử dụng rất nhiều ngôn ngữ khác nhau để viết nên một website hoàn chỉnh, mặc dù tính chất, tính năng và dùng các cấu hình khác nhau nhưng tất cả đều đưa ra một kết quả giống nhau. Có khá nhiều ngôn ngữ phổ biến hiện nay như: PHP, Java, Perl, ASP.NET.... và còn rất nhiều ngôn ngữ khác. vậy tại sao chúng ta nên chọn NodeJS để làm ngôn ngữ chính để thực hiện những website này. Và sau đây là những điểm mạnh mà NodeJS đem lại cho các coder một môi trường làm việc thuận lợi, tiết kiệm thời gian và có hiệu quả cao.

- Các ứng dụng của NodeJS được viết hầu hết bằng Javascript, và ngôn ngữ này chính là ngôn ngữ khá thông dụng cho mọi Coder. Trích nguyên văn câu nói của Ryan Dahl người đã tạo ra NodeJS “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động, và cụ thể là javascript không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”
- NodeJs chạy đa nền tảng từ phía Server, sử dụng kiến trúc hướng sự kiện Event – Driven và cơ chế non-Blocking I/O làm cho NodeJS rất nhẹ và có tính hiệu quả.
- Chúng ta có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac, Window, Linux, hơn nữa cộng đồng người sử dụng Nodejs rất lớn và nguồn tài nguyên là hoàn toàn miễn phí. NodeJs hiện đang là một ngôn ngữ thật sự hot trong ngành CNTT, các package đều hoàn toàn free: <https://www.npmjs.com/>
- Các ứng dụng của NodeJS đáp ứng rất tốt về thời gian thực và chạy đa nền tảng, đa thiết bị.

đây là những điểm mạnh mà NodeJS mang lại cho người dùng. Và đó là gì do người dùng nên sử dụng NodeJS để tạo ra những WebSite hoàn thiện.

2.3 Sơ đồ hoạt động của NodeJS

Ngoài ra Node.js cung cấp một môi trường phía máy chủ mà từ trước tới nay Javascript không làm được, ngoài ra NodeJs còn cho phép chúng ta cũng sử dụng ngôn ngữ JavaScript để tạo các trang web. Nói chung nó thay thế các ngôn ngữ máy chủ như Java EE, PHP, v.v...



Hình 2.1: Sơ đồ hoạt động của NodeJs.

Khi người dùng truy cập vào website ở máy khách sẽ gửi yêu cầu về máy chủ, máy chủ sẽ xử lý chúng theo hướng dẫn đã được mã hóa sẵn. ở server môi trường NodeJS sẽ chạy Javascript để tạo ra HTML, sau đó máy chủ gửi một dữ liệu thích hợp (trang HTML đã tạo) cho máy khách. Khi đó máy client vẫn có thể sửa đổi trang vì mã hóa Javascript vẫn có thể được chạy bởi chính trình duyệt của client.

2.4 Các kiểu dữ liệu

Vì được viết dựa trên JavaScript nên rất nhiều tính năng trong NodeJS được kế thừa từ ngôn ngữ JavaScript. chúng ta sẽ tìm hiểu kiểu dữ liệu, cú pháp, các vòng lặp và cấu trúc điều khiển trong NodeJS.

Gồm các kiểu dữ liệu:

- Kiểu Undefined:

`Typeof instance == "undefined"`

(Một biến chưa được gán giá trị có giá trị undefined.)

Ví dụ:

`var x;(tạo một biến nhưng không gán giá trị)`

`console.log("x's value is", x)`

- Kiểu Boolean:

`Typeof instance == "boolean"`

(Kiểu boolean mang hai giá trị logic là: true, và false.)

Ví dụ:

```
if (điều kiện boolean) {  
    console.log(" trả về giá trị là true");  
} else {  
    console.log(" trả về giá trị là false");  
}
```

- Kiểu số:

`Typeof instance == "number"`

(Không có kiểu số nguyên. Ngoài việc chứa giá trị là dấu phẩy động, kiểu số mang ba giá trị biểu tượng: +Infinity, -Infinity, and NaN (not-a-number).)

- Kiểu chuỗi:

`Typeof instance == "string"`

(Tạo một chuỗi con của chuỗi ban đầu bằng cách ghép từng ký tự hoặc dùng String.substr().)

- Kiểu BigInt:

Typeof instance == "bigint"

(đại diện cho các giá trị số nguyên với độ chính xác (precision) bất kỳ. BigInt cho phép ta có thể lưu và tính toán trên các số nguyên lớn mà nó có thể lớn hơn cả giới hạn an toàn của kiểu Number. Số bigint không thể chung số với số Number để tính toán. Khi đó, lỗi TypeError sẽ xảy ra.)

- Kiểu Symbol:

Typeof instance == "symbol"

Mỗi Symbol là một giá trị đơn nhất và có thể được dùng như một khóa của một Object và bất biến.

- Kiểu Mảng một chiều:

Array.Length (chiều dài của Array(mảng) trong constructor có giá trị là 1)

Array.prototype(được bổ sung các thuộc tính cho hầu hết tất cả các đối tượng Array(mảng).)

Array.from(): (Tạo một Mảng mới từ đối tượng giống iterable hoặc mảng.)

Array.isArray(): (Trả về giá trị true nếu một biến là mảng, và false nếu không phải là mảng.)

Array.of(): (Tạo một Array(mảng) mới với bất kể số hoặc loại của các đối số và một số lượng các đối số.)

Ví dụ:

```
var msgArray = [];
```

```
msgArray [0] = "Hello";
```

```
msgArray [99] = 'World';
```

```
if (msgArray.length === 100) {  
    console.log('Chiều dài là 100');  
}
```

- Mảng 2 chiều:

Ví dụ:

```
const allClasses = [  
    ['nguyen v a', 'nguyen v b'],  
    ['nguyen v c', 'nguyen v d']  
];
```

```
console.log(allClasses[0][0]); //print 'nguyen v a'  
console.log(allClasses[1][1]); //print 'nguyen v d'
```

2.5 Biến và Hằng

2.5.1 Biến

Tên biến và tên hằng chỉ được bắt đầu bằng dấu “_” và chữ và không được bắt đầu bằng số, không được chứa các kí tự đặc biệt.

Trong NodeJS ta có 2 cách để khai báo biến và giá trị. một là dùng từ khóa Var để khai báo, hai là không cần dùng từ khóa.

```
var a = "helloworld";  
  
// khai bao bien a su dung tu khoa var  
b = "xinchaothegioi";  
console.log("bien a=" + a);  
console.log("bien b=" + b);  
kết quả là:
```

```
bien a= helloworld
```


biến `b=xinchaothegioi`.

2.5.2 Hằng

Hằng trong NodeJs sẽ được khai báo cùng từ khóa `const`.

Trong nodejs thì khi khai báo hằng và biến thì hoàn toàn sẽ không cần phải khai báo kèm theo kiểu dữ liệu của chúng mà ngôn ngữ nó tự nhận diện kiểu cho chúng ta.

Ví dụ cụ thể:

```
const x = "welcome ";
```

khai báo hằng `x` cùng kiểu dữ liệu là string

```
const y = 10000;
```

khai báo hằng `y` cùng kiểu dữ liệu là số nguyên(int)

```
const z = 10.6;
```

khai báo hằng `z` cùng kiểu dữ liệu số thực(float)

```
const d = true;
```

khai báo hằng `d` cùng kiểu dữ liệu logic(boolean)

2.6 Các đặc tính của NodeJS

- Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), NodeJs chủ yếu dựa vào nền của NodeJS Server và chờ 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 NodeJS giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước
- Chạy rất nhanh: NodeJS được xây dựng và hình thành dựa vào nền tảng V8 Javascript Engine, cho nên việc thực thi chương trình rất nhanh.
- Đơn luồng nhưng khả năng mở rộng cao: NodeJS 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 có khả năng mở rộng như trái ngược với hầu hết các máy chủ truyền thống mà tạo ra hạn chế để xử lý yêu cầu. Node.js sử dụng một

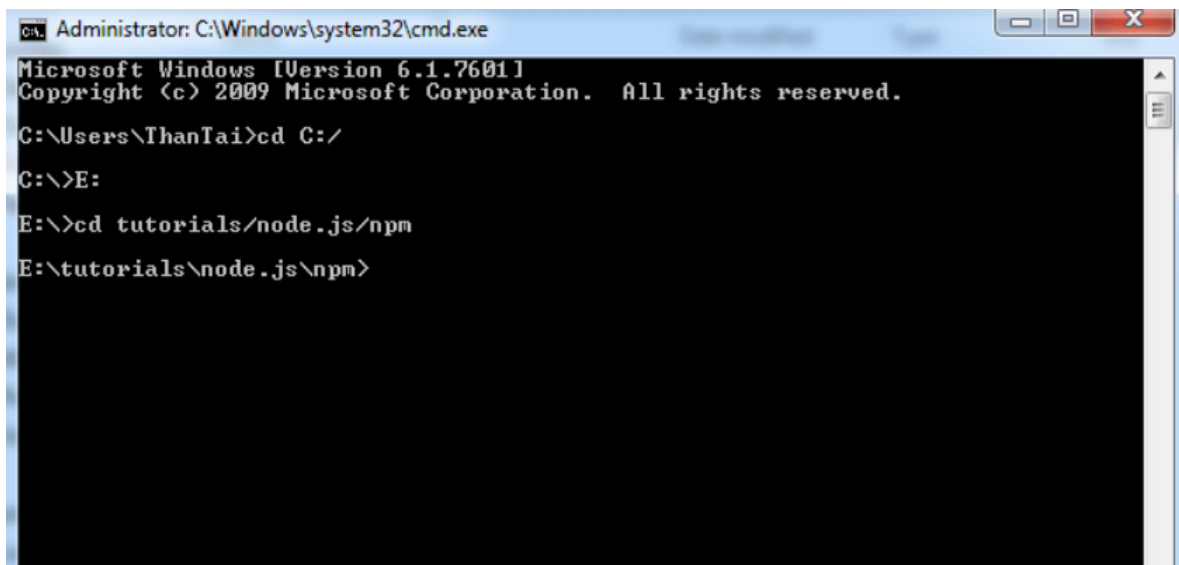
chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ đến số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.

- Không đệm: NodeJS không đệm bất kì kiểu 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.

2.7 NPM

Npm là một chương trình quản lý thư viện, source của NodeJs được tích hợp sẵn trong NodeJs. khi cài đặt node.js thì cũng đồng nghĩa với việc đã cài npm. Npm giống như composer trong PHP, NPM sử dụng command line để quản lý (download, update, delete) các package cho node.js.

- Để khởi tạo một dự án bằng nodeJS với npm chúng ta sử dụng lệnh
npm init
- Đầu tiên đến thư mục cần khởi tạo dự án.

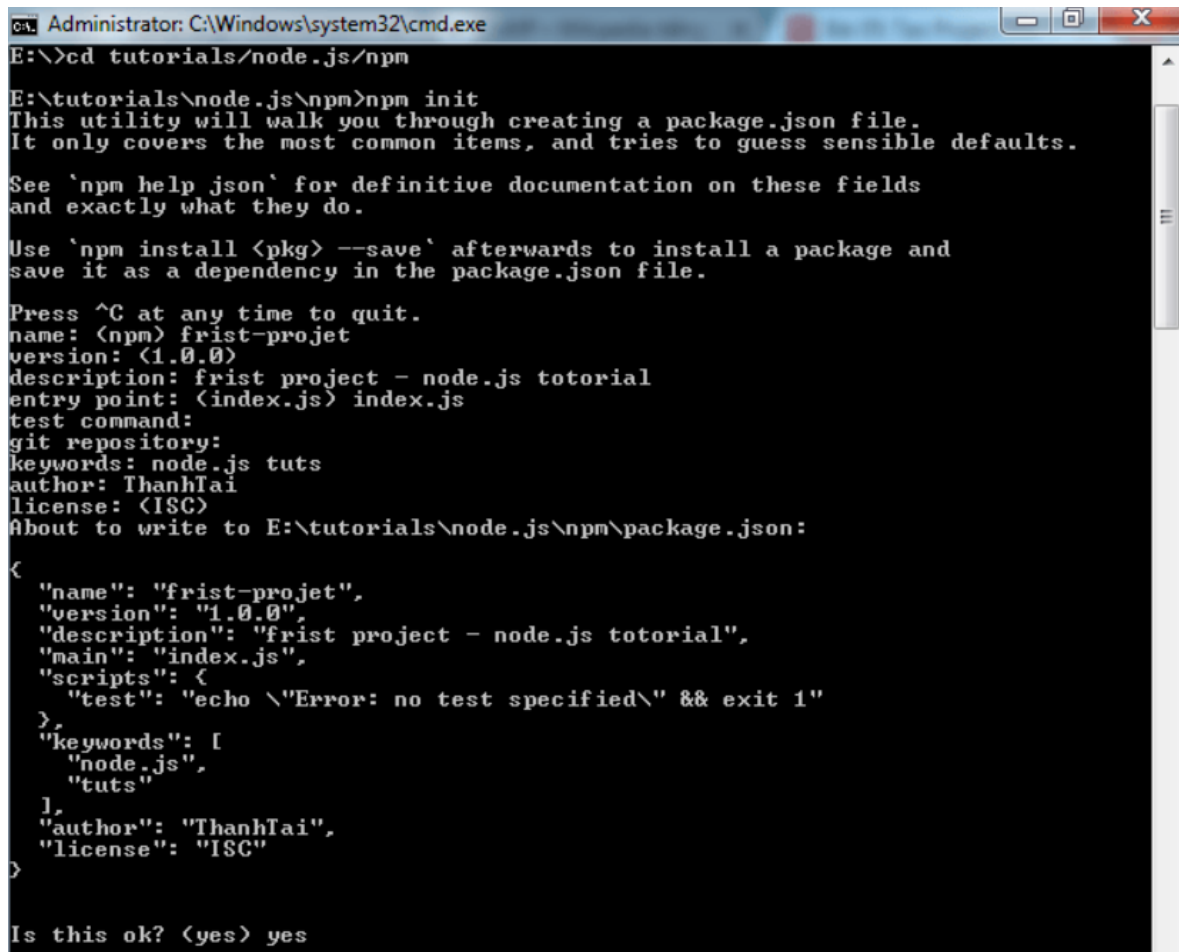


```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ThanTai>cd C:/
C:\>E:
E:\>cd tutorials/node.js/npm
E:\tutorials\node.js\npm>
```

Hình 2.2: Ảnh minh họa khởi tạo một dự án.

- Sau đó sử dụng lệnh (npm init) để bắt đầu khởi tạo project



```
Administrator: C:\Windows\system32\cmd.exe
E:\>cd tutorials/node.js/npm
E:\tutorials\node.js\npm>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help json' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg> --save' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (npm) frist-projet
version: (1.0.0)
description: frist project - node.js totorial
entry point: (index.js) index.js
test command:
git repository:
keywords: node.js tuts
author: ThanhTai
license: (ISC)
About to write to E:\tutorials\node.js\npm\package.json:
{
  "name": "frist-projet",
  "version": "1.0.0",
  "description": "frist project - node.js totorial",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "node.js",
    "tuts"
  ],
  "author": "ThanhTai",
  "license": "ISC"
}
Is this ok? (yes) yes
```

Hình 2.3: Ảnh minh họa cách khởi tạo Package.

- Sau khi chạy xong các bạn sẽ thấy một file package.json vừa được sinh ra. File này sẽ lưu trữ hầu hết các thông tin của dự án.

Xây dựng package là một điều quan trọng trong NodeJS mà chúng ta không thể bỏ qua. Điều này cũng tương tự như Ruby-Gems: đây là tập hợp hàm có sẵn và có thể sử dụng được, thành phần tái sử dụng, và tập hợp các cài đặt dễ dàng thông qua kho lưu trữ trực tuyến.

Một số các Modul npm phổ biến:

- ExpressJS là Sinatra-inspired web framework rất phát triển của NodeJS, chứa hầu hết các ứng dụng chuẩn của NodeJS hiện nay.
- Connect: là phần mở rộng của HTTP server framework cho Node.js, cung cấp “plugins” được xem là trung gian, phục vụ như một nền tảng cơ sở cho Express.

- Socket.io và sockJS được xem là hai thành phần Server-side websockets components nổi tiếng nhất cho đến bây giờ.
- Jade: Một trong những engines mẫu, có cảm hứng từ HAML, còn là một phần mặc định trong Express.js
- mongo and mongojs: MongoDB hàm bao để cung cấp các API cho cơ sở dữ liệu đối tượng trong MongoDB NodeJS
- redis: là thư viện Redis client.
- Underscore: Thư viện tiện ích phổ biến nhất trong JavaScript, package được sử dụng với NodeJS.

2.8 Toán tử trong NodeJs

2.8.1 Toán tử gán

Toán tử gán là lấy giá trị của toán hạng bên phải gán nó vào toán hạng bên trái.

Ví dụ:

```
var name="HelloWorld";
```

2.8.2 Toán tử số học

Toán tử số học là 1 dạng phép tính giản đơn cộng, trừ, nhân, chia trong số học. Ngoài ra còn có phép chia lấy dư (%) sử dụng để lấy ra đơn vị dư của 1 phép toán.

Toán Tử	Lý giải	Ví dụ	Kết quả
+	Cộng hai số hạng	$26 + 7$	33
-	Trừ hai số hạng	$26 - 7$	19
*	Nhân hai số hạng	$26 * 7$	182

/	Chia hai số hạng	26 / 7	3
%	Trả về số dư	26 % 7	5

Bảng 2.1: Bảng toán tử số học.

2.8.3 Toán tử so sánh

Là toán tử được sử dụng nhằm thực hiện các phép toán so sánh giữa hai số hạng

Phép toán	Tên	Giải Thích	Ví dụ
==	Bằng	Hai số hàng bằng nhau	A==5
!=	Không bằng	Hai số hạng không bằng nhau	A!=5
===	Đồng nhất	Hai số bằng nhau và cùng kiểu	A===5
>	Lớn hơn	Vế trái lớn hơn vế phải	A>5

>=	Lớn hơn hoặc bằng	Vế trái lớn hơn hoặc bằng vế phải	$A \geq 5$
<	Nhỏ hơn	Vế trái nhỏ hơn vế phải	$A < 5$
<=	Nhỏ hơn hoặc bằng	Vế trái nhỏ hơn hoặc bằng vế phải	$A \leq 5$

Bảng 2.2: Bảng toán tử so sánh.

2.8.4 Toán tử logic

Toán tử logic chính là tổ hợp các giá trị boolean.

Ví dụ: bài toán tử dấu && trả giá trị true nếu toán tử ở vế trái và toán tử ở vế phải là true.

TRUE && TRUE

//kết quả là true

Toán tử	Tên	Trả về True nếu	Ví dụ	Kết quả
---------	-----	-----------------	-------	---------

 	Or	Vế trái hoặc vế phải là True	True False	True
Or	Or	Vế trái hoặc vế phải là True	True False	True
Xor	Xor	Vế trái hoặc vế phải là True nhưng không phải cả hai.	True False	False
&&	And	Vế trái và vế phải là True	True && False	False
And	And	Vế trái và vế phải là True	True && False	False
!	Not	Không phải true	!True	False

Bảng 2.3: bảng toán tử logic

2.8.5 Toán tử kết hợp

- Tăng

```
var a=10;
var a++;
trả về a=11
```

- Giảm

```
var a=10;
var a--;
trả về a=9
```

- Cộng

```
var a=10;
var b=15;
a+=b;
trả về a=25
```

- Trừ

```
var a=20;
var b=15;
a-=b;
trả về a=5
```

- Nhân

```
var a=10;  
var b=15;  
a*=b;  
trả về a=150
```

- Chia

```
var a=30;  
var b=15;  
a/=b;  
trả về a=2
```

2.9 Hàm trong NodeJS

Dùng từ khóa function để khai báo hàm trong NodeJs

Ví dụ:

```
function hello()  
{  
    console.log('hello world');  
}
```

ở đây chúng ta đã khai báo một hàm có tên là hello và để sử dụng chúng ta cần phải gọi lại tên hàm là hello kèm với cặp dấu ().

Ví dụ:

```
hello();
```

Sử dụng hàm có tham số:

Ví dụ:

```
function hello(name)  
{  
    console.log("chao " + name);  
}  
khởi tạo hàm  
hello('tuan');
```



```
gọi hàm hello với tham số truyền vào là tuan
hello('nam');
gọi hàm hello với tham số truyền vào là nam
```

Kết quả:

```
chao tuan
chao nam
```

2.10 Câu lệnh rẽ nhánh

2.10.1 Câu Lệnh If-ELSE

If-else là điều kiện được sử dụng khá nhiều trong hầu hết các ngôn ngữ lập trình.

Ví dụ:

```
var diem = 7.5;
if (diem < 5) {
    console.log("yeu");
} else
if (diem >= 5 && diem < 7) {
    console.log("TB");
} else
if (diem >= 7 && diem < 8) {
    console.log("kha");
} else if (diem > 8 && diem <= 9)
{
    Console.log("gioi")
}
{
    console.log("XuatXac");
}
```

Kết quả: kha

2.10.2 Câu lệnh lồng nhau

Ví dụ:

```
var diem = 6.5;
if (diem < 5) {
    console.log("yeu");
} else
if (diem >= 5 && diem < 7) {
    console.log("TB");
} else
if (diem >= 7 && diem < 8) {
    console.log("kha");
} else if(diem >8 && diem<=9)
{
    Console.log("gioi")
}
{
    console.log("XuatXac");
}
```

Kết quả: TB

2.10.3 Switch-Case

Cấu trúc của Switch-Case như sau:

```
Switch (bien)
{
    Case giá trị 1:
        //hanh dong;
    Break;
    Case giá trị 2:
        //hanh dong;
    Break;
```

```

    Case giá trị N:
        //hành động;
    Break;
    Default:
        //hành động;
    Break;
}
Ví dụ:
var now = 5;
switch (now)
{
    Case1:
        case1:
            break;
        case 2:
            console.log("Thứ 2");
            break;
        case 3:
            console.log("Thứ 3");
            break;
        case 4:
            console.log("Thứ 4");
            break;
        case 5:
            console.log("Thứ 5");
            break;
        case 6:
            console.log("Thứ 6");

```

```

        break;
    case 7:
        console.log("Thứ 7");
        break;
    case 8:
        console.log("Chủ nhật");
        break;
    default:
        console.log("Sai");
        break;
}

```

Kết quả: Thứ 5

2.11 Đối tượng trong NodeJs (object)

Trong nodejs đối tượng là một khái niệm trừu tượng thể hiện cho một đối tượng cụ thể, và có thể tự tạo một đối tượng theo ý của mình dựa vào yêu cầu của ứng dụng.

Ví dụ:

Chúng ta tạo ra một đối tượng chuyên xử lý vấn đề về sinh viên thì ta sẽ đặt tên cho nó là Student.

Sử dụng từ khóa new Object (lưu ý chữ “O” trong Object phải ghi hoa)

Ví dụ: var Students = new Object();

- Ngoài ra ta cũng có thể dùng từ khóa {}.

Ví dụ: var Students = {};

2.11.1 Thuộc tính và phương thức trong đối tượng

2.11.1.1 Thuộc tính

- Thuộc tính là tất cả những đặc điểm (có thể hiểu là biến) cần lưu trữ trong đối tượng.

Ví dụ: với đối tượng Students ta cần có các thuộc tính sau:

Name: tên

Age: tuổi

Class: lớp

Pointavg: điểm TB

ví dụ về khai báo:

cách 1: khai báo new Object().

```
var Students = new Object();  
Students.name = 'Nguyễn Cao Tấn';  
Students.age = 18;  
Students.class = 'Sinh viên';  
Students.pointAvg = 7.0;
```

Cách 2: khai báo nhóm

```
var Students = { name: 'Nguyễn Cao Tấn', age: 18, class:  
'Sinh viên', pointAvg: 7.0};
```

Cách 3: thêm thuộc tính vào sau.

```
var Students = { };  
Students.name = 'Nguyễn Cao Tấn';  
Students.age = 18;  
Students.class = 'Sinh viên';  
Students.pointAvg = 7.0;
```

2.11.1.2 Phương thức

Phương thức trong NodeJs cũng có các tính chất như các ngôn ngữ khác, có thể hiểu và gọi là hàm.

Ví dụ: với Object Students ở phần thuộc tính ta sẽ tạo thêm phương thức addstudents() bằng 3 cách sau:

Cách 1: Khởi tạo xong khai báo với new Object().

```
var Students = new Object();  
Students.AddStudents = function (ten, tuoi, lop, DiemTb) {  
    this.name = ten;  
    this.age = tuoi;
```

```

        this.class = lop;
        this.pointAvg = Diemtb;
    }

```

Cách 2: Khai báo ngay trong khi khởi tạo.

```

var Students = {
    AddStudents: function (ten, tuoi, lop, DiemTb)
    {
        this.name = ten;
        this.age = tuoi;
        this.class = lop;
        this.pointAvg = Diemtb;
    }
};

```

Cách 3: Khởi tạo xong rồi khai báo ({}).

```

var Students = {};
Students.AddStudents = function ((ten, tuoi, lop, DiemTb) {
    this.name = ten;
    this.age = tuoi;
    this.class = lop;
    this.pointAvg = Diemtb;
}

```

2.11.2 Gọi thuộc tính và phương thức

2.11.2.1 gọi thuộc tính

Cú pháp:

```
doituong.thuoc tinh;
```

hoặc

```
doituong['thuoc tinh'];
```

Ví dụ:

```
Students.name;
```

Hoặc

```
Students['name'];
```

2.11.2.2 gọi phương thức

Cú pháp:

```
doituong.phuongthuc;
```

ví dụ:

```
Students.AddStudents('Tấn', 20, 'ABC', 2.5);
```

2.11.3 Đối tượng trong đối tượng

Cú pháp:

```
doituong = {  
  doituong1: {  
    doituong2: {  
      //code  
    }  
  }  
}
```

Ví dụ:

```
var Person = {  
  name: {  
    firstName: "Cao",  
    lastName: "Tấn"  
  },  
  birthday: {  
    day: 26,  
    month: 7,  
    year: 1999  
  }  
}
```

```
}
```

2.11.4 Mảng trong đối tượng

Cú pháp:

```
arr = [doituong1, doituong2];
```

hoặc

```
arr = [];
```

```
arr[0] = doituong;
```

```
arr[1] = doituong1;
```

```
arr[..] = doituong...;
```

Ví dụ:

```
var Person = {  
  name: {  
    firstName: "Cao",  
    lastName: "Tân"  
  },  
  birthday: {  
    day: 26,  
    month: 07,  
    year: 1999  
  }  
};  
  
var arr = [Person];  
  
console.log(arr[0].name.firstName);
```

2.12 Khởi tạo Server trong NodeJs với HTTP module

2.12.1 Giới thiệu về HTTP module

HTTP là một module được tích hợp sẵn vào trong NodeJs, module này có nhiệm vụ khởi tạo một cổng kết nối HTTP server trả về client.

2.12.2 Khởi tạo Server với HTTP module

Để sử dụng được bất kỳ module nào trong NodeJs thì chúng ta cần phải require module đó. Và để có thể require một module trong ngôn ngữ NodeJs, sử dụng từ khóa “require”.

Ví dụ: khởi tạo server chạy port 3000

```
var http = require('http');
```

```
http.createServer().listen(3000);
```

trong phương thức createServer sẽ chứa một phương thức ẩn danh có 2 tham số truyền vào.

Tham số thứ nhất: là biến chứa đựng các thông số có liên quan đến request mà người dùng (client) gửi lên.

Tham số thứ hai là biến chứa đựng các thông số có liên quan đến response mà máy chủ Server muốn trả về client.

Ví dụ:

tạo một tệp tin index.js trong thư mục node-tutorial

([khai báo sử dụng module HTTP](#))

```
var http = require('http');
```

([Khởi tạo server chạy cổng 3000](#))

```
http.createServer(function (request, response) {  
    response.write('hello world! toidicode.com');  
    response.end();  
}).listen(3000);
```

Sau đó run file index.js và mở trình duyệt truy cập link sau:

<http://localhost:3000>

2.12.3 Các phương thức và thuộc tính được dùng trong request và response

2.12.3.1 Writehead

Hàm `writehead()` là hàm thiết lập kiểu dữ liệu mà máy chủ server muốn trả về.

Ví dụ: thiết lập server trả về là một trang HTML

(khai báo sử dụng module HTTP)

```
var http = require('http');
```

(Khởi tạo server chạy cổng 3000)

```
http.createServer(function (request, response) {  
    response.writeHead(200, {"Content-Type": 'text/html'});  
    response.write('Hello world');  
    response.end();  
}).listen(3000);
```

2.12.3.2 write

Hàm `write()` thiết lập nội dung mà server muốn trả về trình duyệt, nội dung này có thể là text có thể là HTML code.

Ví dụ: Trả về một trang HTML có thẻ H1 chứa dòng chữ hello world.

(khai báo module HTTP)

```
var http = require('http');
```

(Khởi tạo server chạy cổng 3000)

```
http.createServer(function (request, response) {  
    response.writeHead(200, {'Content-Type': 'text/html'});  
    response.write('<html>');  
    response.write('<head>');  
    response.write('<title>Hello world </title>');  
    response.write('</head>');  
    response.write('<body> <h1>Hello world<h1> </body>');  
    response.write('</html>');  
    response.end();  
});
```

```
}).listen(3000);
```

2.12.3.3 URL

URL chứa các paramter trong URL mà client gửi lên server.

Ví dụ:

(khai báo sử dụng module HTTP)

```
var http = require('http');
```

(Khởi tạo server chạy cổng 3000)

```
http.createServer(function(request, response)
```

```
{
```

```
var params = request.url;
```

```
response.write(param);
```

```
response.end();
```

```
}).listen(3000);
```

Nếu các tên miền hiện ra phía sau domain thì sẽ là url

Ví dụ: <http://localhost:3000/WebDoAn> thì url sẽ là WebDoAn

2.13 Đọc ghi file bằng module fs trong Node.js

2.13.1 Giới thiệu về module fs

Module fs (viết tắt của file system) là module được tích hợp sẵn trong NodeJs có chức năng xử lý file, thư mục trong NodeJs

Khi sử dụng module chúng ta phải require() nó. Để require module fs sử dụng cú pháp:

```
Require('fs');
```

Để xem phương thức trong module ta dùng lệnh console.log().

Ví dụ:

```
var fs = require('fs');
```

```
console.log(fs);
```

2.13.2 Đọc files với module fs

Để đọc file ta sử dụng phương thức readFile() cú pháp là:

```
readFile('pathFile', [option], function(err, data) {  
    //  
});
```

pathFile: đường dẫn File cần đọc

option: có thể bỏ trống, có thể chứa các chuỗi hoặc mảng

err: biến lỗi

data: dữ liệu đọc được từ file.

Các bước thực hiện:

Bước đầu tiên sẽ tạo ra một file code.html

Bước 2: tạo một file server.js cùng cấp với file code.html để khởi tạo server và xử lý

Bước 3: chạy file server là nhận lại kết quả.

2.13.3 Ghi file

Dùng phương thức WriteFile() để ghi nội dung vào một file.

Cú pháp:

```
wireFile('filepath', 'noidung', options, callback);
```

trong đó:

filepath: đường dẫn file cần ghi.

content: nội dung muốn ghi vào file.

option: chuỗi hoặc mảng, và có thể để trống.

callback: xử lý sau khi ghi file.

Ví dụ: ghi nội dung vào file write.html

```
var fs = require('fs');
```

```
var content = 'ghi vào file writer.html';
```

```
fs.writeFile('writer.html', content, 'utf8', function (err) {
```

```
    (Kiểm tra)
```

```
    if (err)
```

```
        throw err;
```

```
    else
```

```
console.log('Ghi file thanh cong!');  
});
```

2.14 Module URL

Module url dùng để xử lý và phân tích chuỗi URL qua đó biết được thông số URL

2.14.1 Khai báo sử dụng Module URL

Cú pháp:

```
require('url');
```

các thuộc tính của module giống như đối tượng location trong javascript.

Ví dụ:

```
var url = require('url');  
var website = "http://webgiasu?a=5";  
var parse = url.parse(website, true);  
console.log('auth: ' + parse.auth);  
console.log('hash: ' + parse.hash);  
console.log('host: ' + parse.host);  
console.log('hostname: ' + parse.hostname);  
console.log('href: ' + parse.href);  
console.log('path: ' + parse.path);  
console.log('pathname: ' + parse.pathname);  
console.log('port: ' + parse.port);  
console.log('protocol: ' + parse.protocol);  
console.log('query: ' + parse.query.a);  
console.log('search: ' + parse.search);  
console.log('slashes: ' + parse.slashes);
```

2.15 Upload file trong NodeJs

2.15.1 Tổng quan

Chức năng này được thực hiện khi tích hợp các module sau

- Module http

- Module fs
- Module formidable (module phân tích dữ liệu từ form gửi lên)

2.15.2 cài đặt module formidable

cú pháp:

```
npm install -s formidable
```

2.15.3 code upload

- viết code khởi tạo server (chạy port 3000)
- tạo file chứa code html form (đặt ở file index.html cùng cấp với server.js)
- dùng module fs để render file view đó ra trên server.
- chạy server và nhận kết quả.
- trên file server.js chúng ta sẽ tiếp tục require module formidable
- xuất kết quả khi chạy server.

Ví dụ:

```
var http = require('http');
var fs = require('fs');
var formidable = require('formidable');
http.createServer(function (req, res) {
  if (req.url === '/upload' && req.method.toLowerCase() ===
    'post') {
    var form = new formidable.IncomingForm();
    form.uploadDir = "uploads/";
    form.parse(req, function (err, fields, file) {
      var path = file.files.path;
      var newpath = form.uploadDir + file.files.name;
      fs.rename(path, newpath, function (err) {
        if (err) throw err;
        res.end('Upload Thanh cong!');
      });
    });
  }
});
```

```

    });
    return;
  }
  res.writeHead('200', {'Content-Type': 'text/html'});
  fs.readFile('index.html', 'utf8', function (err, data) {
    if (err) throw err;
    res.end(data);
  })
}).listen(3000);

```

2.16 Sự kiện trong NodeJs

2.16.1 Event loop

NodeJs được thực thi nhanh hơn rất nhiều so với ngôn ngữ khác là nhờ có rất nhiều sự kiện ở trong NodeJs, đó là các yếu tố giúp cho Nodejs cải thiện được tốc độ mà ngôn ngữ khác không có được.

- Thêm sự kiện: để sử dụng được sự kiện trong NodeJs bắt buộc phải require module event và khởi tạo eventEmitter

Cú pháp:

[Require module event](#)

```

var event = require('events');
var eventEmitter = new event.EventEmitter();

```

và muốn thêm sự kiện dùng phương thức on theo cú pháp sau:

```
on(eventName, handleEvent);
```

trong đó:

eventName: tên sự kiện muốn thêm

handleEvent: hàm xử lý khi eventName được gọi

- Thực thi sự kiện: dùng phương thức emit

Cú pháp:

```
emit(eventName[,...args]);
```

trong đó:

eventNam: sự kiện muốn thực thi

args: là tham số truyền vào dữ liệu.

Ví dụ:

```
var event = require('events');
```

```
var EventEmitter = new event.EventEmitter();
```

```
var connectionHandle = function (data) {
```

```
    console.log('connection successful' + data);
```

```
}
```

```
eventEmitter.on('connection', connectionHandle);
```

[thực thi sự kiện](#)

```
eventEmitter.emit('connection', new Date());
```

2.17 Tạo client request trong NodeJs

2.17.1 Cài đặt

module request là một module phát triển dựa trên module http của NodeJs

-cú pháp:

```
npm install request
```

- require vào dự án sau khi cài đặt xong.

```
let requests = require('requests');
```

gửi request trong module request sử dụng cú pháp sau:

```
request(url, function (err, res, body));
```

-khi đó:

request: biến ánh xạ module request.

err: chứa lỗi nếu có.

res: chưa thông tin server trả về.

body: chưa body mà server trả về.

2.17.2 các phương thức mà module request hỗ trợ:

Method	Mô tả
Request.get()	Gửi request với phương thức GET.
Request.post()	Gửi request với phương thức POST.
Request.put()	Gửi request với phương thức PUT .
Request.patch()	Gửi request với phương thức PATCH.
Request.del()	Gửi request với phương thức DEL.
Request.head()	Gửi request với phương thức HEAD.
Request.options()	Gửi request với phương thức OPTIONS.

Bảng 2.4: bảng phương thức module request.

Ví dụ:

```
let requests = require('requests');
request.post('https://webgiasu', function (err, res, body) {
  nếu có lỗi
  if (err)
    throw err;
  in ra header
  console.log(res);
  in ra body nhận được
  console.log(body);
})
```

Chương 3. CƠ SỞ DỮ LIỆU MONGODB

3.1 Giới thiệu về MongoDB

Mongodb là một chương trình cơ sở dữ liệu có mã nguồn mở được thiết kế theo hướng đối tượng, mongodb được viết bằng C++ trong đó các bảng được cấu trúc một cách cực kì linh hoạt, cho phép dữ liệu lưu trên bảng không cần phải tuân theo một quy định cấu trúc nào. Chính vì vậy mà mongodb được sử dụng để lưu trữ các dữ liệu có cấu trúc phức tạp, đa dạng và không cố định. Ngoài ra mongodb còn là cơ sở dữ liệu đa nền tảng, có hiệu suất cao, tính khả dụng cao và mở rộng rất dễ dàng.

3.1.1 Lợi thế của Mongodb

- Cấu trúc đối tượng rõ ràng.
- Không có Join phức tạp.
- Khả năng truy vấn sâu, mongodb hỗ trợ truy vấn động trên các Document bởi sử dụng ngôn ngữ truy vấn, nhưng rất mạnh mẽ như SQL.
- Ít schema: Collection giữ các Document khác nhau, dữ liệu không đồng nhất, không bị ràng buộc.
- Mongodb dễ dàng mở rộng hơn.
- Giúp truy cập dữ liệu nhanh hơn, sử dụng bộ nhớ nội tại để lưu giữ công việc.

3.1.2 đặc điểm của Mongodb

- định hướng của Document: dữ liệu được lưu trong các tài liệu JSON.
- Có thể lập chỉ mục trên bất kì thuộc tính nào.
- Truy vấn của mongodb đa dạng hơn
- Tốc độ cập nhật nhanh hơn.

3.2 Mục đích sử dụng cơ sở dữ liệu Mongodb

Cơ sở dữ liệu mongodb gồm các chức năng như: lưu trữ (storage), truy cập(accessibility), phân tán dữ liệu(distribution)

- Lưu trữ: MongoDB lưu trữ dữ liệu định hướng document một cách linh hoạt, được định dạng JSON, điều này có nghĩa các trường có thể thay đổi từ document này sang document khác và cấu trúc dữ liệu cũng được thay đổi theo thời gian, cập nhật nhanh chóng, document model ảnh xạ đến các object trong code giúp dễ dàng thao tác với dữ liệu.
- Truy cập: truy vấn đặc biệt, đa dạng, lập chỉ mục và tổng hợp theo thời gian thực, còn cung cấp các phương thức mạnh mẽ giúp phân tích dữ liệu và truy cập.
- Phân tán dữ liệu: MongoDB là một cơ sở dữ liệu phân tán, tính sẵn sàng cao, kèm theo là nhân rộng tích hợp theo chiều ngang và phân tán theo vị trí địa lý, rất dễ dàng xây dựng và sử dụng. tự động sharding, protect end-to-end và chuyển đổi dự phòng, được quản lý tự động hóa, giám sát và sao lưu và hoàn toàn miễn phí

3.3 Thuật ngữ trong MongoDB

Thuật Ngữ	Diễn giải
_id	Trường bắt buộc phải có trong document, đại diện cho giá trị duy nhất, được hiểu khóa chính trong document.
Collection	Nhóm của nhiều document trong MongoDB, là một bảng tương ứng trong csdl relational database management system, nằm trong một csdl duy nhất, không định nghĩa các cột, hàng hoặc kiểu dữ liệu trước
Cursor	là một con trỏ đến tập kết quả của một truy vấn, máy khách có thể lặp qua con trỏ để lấy kết quả.
Database	Chứa các collection, chứa các bảng, có tập tin riêng lưu trên bộ nhớ vật lý, máy chủ có thể chứa nhiều database
Document	Là một bản ghi thuộc một collection, bao gồm trường tên và giá trị.

Field	Là cặp name-value trong document, một document có thể có hoặc không có trường. các trường giống các cột csdl quan hệ.
JSON	Viết tắt của JavaScript Object Notation, dữ liệu có cấu trúc và hỗ trợ rất nhiều ngôn ngữ lập trình.
Index	Cấu trúc dữ liệu đặc biệt, quét một phần nhỏ các tập dữ liệu dễ dàng, hỗ trợ phân tích hiệu quả các truy vấn, nếu không có chỉ mục, mongodb sẽ quét tất cả các documents của collection để tìm document phù hợp với câu truy vấn.

Bảng 3.1: bảng thuật ngữ trong Mongoddb.

3.4 Kiểu dữ liệu trong Mongoddb

Kiểu dữ liệu	Number	Alias
Double	1	“double”
String	2	“string”
Object	3	“object”
Array	4	“array”
Binary data	5	“binData”
Underfined	6	“underfined”

ObjectId	7	“objectId”
Boolean	8	“bool”
Date	9	“date”
Null	10	“null”
Regular expression	11	“regex”
Dbpointer	12	“dbpointer”
Javascript	13	“javascript”
Symbol	14	“symbol”
JavaScript(with scope)	15	“javascriptWithScope”
32-bit integer	16	“int”
Timestamp	17	“timestamp”
64-bit integer	18	“long”
Decimal 128	19	“decimal”
Min key	-1	“minkey”
Max key	127	“maxkey”

Bảng 3.2: bảng kiểu dữ liệu trong MongoDB.

- Chuỗi (string): Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải được định dạng là UTF-8 hợp lệ.
- Số nguyên (integer): Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên được quy định là 32 bit hoặc 64 bit phụ thuộc vào máy chủ Server.
- Boolean: Kiểu dữ liệu này được dùng với mục đích lưu giữ một giá trị Boolean (true/false).

- Double: Kiểu dữ liệu này được dùng với mục đích lưu các giá trị số thực dấu chấm động.
- Min/ Max keys: Kiểu dữ liệu này được dùng với mục đích so sánh một giá trị với các phần tử BSON thấp nhất và cao nhất.
- Mảng (array): Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.
- Timestamp: Đánh dấu thời điểm nào đó mà một Document được sửa đổi hoặc được thêm vào.
- Object: Kiểu dữ liệu này được dùng với mục đích cho các Document được nhúng vào.
- Null: Kiểu dữ liệu này được dùng với mục đích lưu một giá trị Null.
- Symbol: Kiểu dữ liệu này được sử dụng giống như một chuỗi
- Date: Loại dữ liệu này được dùng để lưu giữ date và time hiện tại trong định dạng UNIX time.
- Object ID: Kiểu dữ liệu này được dùng với mục đích lưu giữ ID của Document.
- Binary data: Kiểu dữ liệu này được dùng với mục đích lưu giữ dữ liệu nhị phân.
- Code: Kiểu dữ liệu này được dùng với mục đích lưu giữ JavaScript code vào trong Document.
- Regular expression: Kiểu dữ liệu này được dùng với mục đích lưu giữ Regular Expression.

3.5 Một số câu lệnh truy vấn cơ bản của MongoDB

- CREATE TABLE (Tạo bảng):

`db.createCollection('t_name');` (t_name: là tên bảng)

- ADD TABLE (chèn bảng):

`db.t_name.insert({c1_ten: c1_giatri, c2_ten: c2_type,...});`

- CREATE DATABASE (Tạo Database):

`use db_name;`

- DELETE DATABASE (Xóa Database):

`db.dropDatabase();`

- DELETE TABLE (Xóa Bảng):

`db.t_name.drop();`

- UPDATE (Cập nhật):

`db.t_name.update({ _id: id_giatri }, { $set: { c_name: c_new_giatri } });`

- DELETE (Xóa):

`db.t_name.remove({_id: id_value});`

- DELETE ALL (Xóa tất cả):

`db.t_name.deleteMany();`

- FIND ALL (Tìm tất cả):

`db.t_name.find();`

- FIND (tìm theo trường):

`db.t_name.find({c_name: c_value});`

3.6 Truy vấn có điều kiện trong MongoDB

Có thể thêm điều kiện vào trong hàm `find()` với cú pháp:

`db.collectionName.find(condition)`

Liệt kê cú pháp của mệnh đề điều kiện:

Phép Toán	Cú pháp
Bằng (equality)	{key: value}
Nhỏ hơn (less than)	{key: {\$lt: value}}
Nhỏ hơn bằng (less than equals)	{key: {\$lte: value}}
Lớn hơn (greater than)	{key: {\$gt: value}}
Lớn hơn bằng (greater than equals)	{key: {\$gte: value}}
Khác (not equals)	{key: {\$ne: value}}
Thuộc (in)	{key: {\$in: [gtri1, gtri2,...]}}
Không thuộc vào(Not in)	{key: {\$nin: [gtri1, gtri2,...]}}

Bảng 3.3: bảng cú pháp truy vấn có điều kiện trong MongoDB

3.7 MongoDB Replication

Replication là dữ liệu được đồng bộ hóa trên nhiều server. Nó sẽ cung cấp rất nhiều tính năng và tăng tính sẵn sàng của data với nhiều bản sao dữ liệu trên nhiều server DB, replication bảo vệ DB khỏi việc mất dữ liệu trên một server đơn lẻ. replication sẽ cho phép bạn khôi phục dữ liệu từ các ổ cứng lỗi và việc service lỗi và hay gặp vấn đề. Các dữ liệu có thể sao chép rất nhiều và có thể chỉnh một bản cho vấn đề khôi phục, report

Vậy tại sao lại cần REPLICATION:

- Điều đầu tiên là dữ liệu data rất an toàn.
- Có tính năng luôn luôn sẵn sàng cho dữ liệu.
- Có thể khôi phục dữ liệu khỏi lỗi.
- Không bị downtime khi xảy ra maintenance.
- Tăng khả năng đọc và xử lý dữ liệu được cải thiện.

3.8 Cài đặt mongodb

- Để có thể cài đặt Mongodb ta sử dụng lệnh:

Npm install mongodb

- Để sử dụng module mongodb ta phải require nó vào file chúng ta sử dụng (chỉ sử dụng module MongoClient)

Cú pháp:

Var mongoClient = require(mongodb).MongoClient;

3.9 Kết nối NodeJs với mongodb

Muốn khởi tạo kết nối từ NodeJs đến mongodb chúng ta sẽ sử dụng phương thức connect trong đối tượng MongoClient

Cú pháp:

mongoClient.connect(url, function (error, dbs) {

});

Khi đó:

- url là đường dẫn kết nối đến mongodb
- error: biến chứa lỗi nếu có.
- db: là object chứa data kết nối.

ví dụ:

```
var mongoClients = require('mongodb').MongoClient;  
mongoClients.connect('mongodb://127.0.0.1:3000', function (err, db) {  
  (neu ket noi khong thanh cong thi in ra loi)  
  if (err) throw err;  
  console.log('Ket noi thanh cong');  
});
```

3.10 Đóng kết nối NodeJs đến Mongodb

để đóng kết nối bằng NodeJs ta dùng cú pháp sau:

db.close();

trong đó, db là object chứa data kết nối.

ví dụ:

```
var mongoClient = require('mongodb').MongoClient;
mongoClient.connect('mongodb://127.0.0.1:3000', function (err, db) {
  (neu ket noi khong thanh cong thi in ra loi)
  if (err) throw err;
  console.log('Ket noi thanh cong');
  db.close();
  console.log('close thanh cong');
});
```

3.11 Sử dụng Mongodb

Các bước thực thi:

1. Hiện thị tất cả các databases

Cú pháp: show dbs

2. Tạo database

Sử dụng cú pháp: use 'database_name'

3. Tạo một collection và insert dữ liệu vào collection

Cú pháp: db.collection_name.insert({'key_name': 'value'})

4. Get tất cả object của một collection

Cú pháp: db.collection_name.find({})

5. Drop một collection

Cú pháp: db.collection_name.drop()

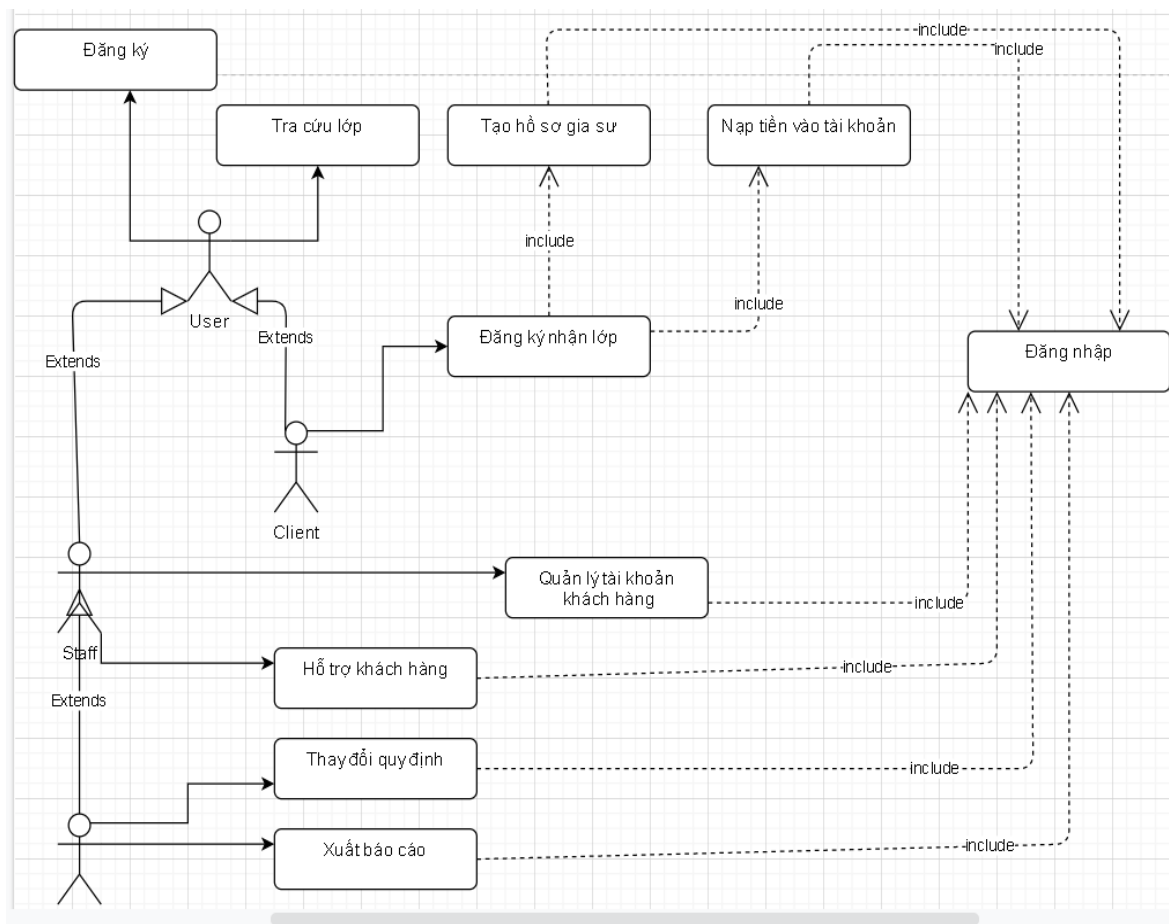
Chương 4. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1 Tác nhân và use case hệ thống

4.1.1 Các tác nhân (Actor)

- Khách hàng (Client)
- Người quản trị (Admin).
- Nhân viên (Staff)

4.1.2 Mô hình use case:



Hình 4.1 Mô hình Use case

4.2 Các use case

4.2.1 Đăng nhập:

- Mô tả: Người dùng đăng nhập vào hệ thống có thể thay đổi mật khẩu, cập nhật thông tin cá nhân, đăng ký tài khoản mới và các chức năng trong menu chính được người quản trị phân quyền.

Description	Người dùng đăng nhập.
Actors	Nhân viên, admin, khách hàng.
Trigger	Người dùng muốn đăng nhập sử dụng dịch vụ web.
Pre-Conditions	Tài khoản đã được người dùng tạo sẵn và được phân quyền và người dùng sử dụng đúng tài khoản, mật khẩu đã được tạo.
Post-Conditions	Người dùng đăng nhập thành công, hệ thống ghi nhận phiên hoạt động.
Basic Flow	<ol style="list-style-type: none">1. Người dùng truy cập trang web.2. Người dùng muốn đăng nhập hoặc muốn sử dụng dịch vụ bắt buộc đăng nhập.3. Người dùng sử dụng tài khoản đã được tạo từ trước và chọn lệnh đăng nhập.4. Hệ thống kiểm tra thông tin đăng nhập thành công và cho phép người dùng sử dụng dịch vụ.5. Hệ thống ghi nhận phiên hoạt động của tài khoản.
Exception Flow	- Thông tin người dùng cung cấp không

	<p>hợp lệ:</p> <p>Hệ thống thông báo ra màn hình thông tin không đúng quy định.</p> <p>Người dùng nhập lại thông tin khác hợp lệ.</p> <p>- Tài khoản đã bị vô hiệu hóa hoặc bị khóa. Người dùng phải liên hệ bộ phận hỗ trợ để có thể tiếp tục sử dụng tài khoản.</p>
--	---

Bảng 4.1 mô tả đăng nhập

4.2.2 Đăng ký:

- Mô tả: Người dùng đăng ký tài khoản để nhận lớp hoặc đăng thông tin tuyển dụng gia sư.

Description	Người dùng đăng ký tài khoản.
Actors	Khách hàng, admin.
Trigger	Người dùng yêu cầu đăng ký tài khoản mới.
Pre-Conditions	Thông tin tài khoản chưa được sử dụng ở tài khoản khác.
Post-Conditions	Người dùng đăng ký thành công. Hệ thống ghi nhận tài khoản mới và lưu vào cơ sở dữ liệu người dùng.
Basic Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập trang web. 2. Người dùng muốn đăng ký hoặc muốn sử dụng dịch vụ bắt buộc có tài khoản người dùng. 3. Người dùng điền thông tin hợp lệ vào biểu mẫu đăng ký và chọn lệnh đăng

	<p>ký</p> <p>4. Hệ thống kiểm tra thông tin đăng ký thành công và cho phép người dùng sử dụng dịch vụ.</p> <p>5. Hệ thống ghi nhận phiên hoạt động của tài khoản và lưu dữ liệu tài khoản mới vào dữ liệu người dùng.</p>
Exception Flow	<p>- Thông tin người dùng cung cấp không hợp lệ :</p> <p>Hệ thống thông báo ra màn hình thông tin không đúng quy định.</p> <p>2. Người dùng nhập lại thông tin khác hợp lệ.</p> <p>- Thông tin người dùng đã được sử dụng ở tài khoản khác:</p> <p>Hệ thống thông báo ra màn hình tài khoản đã tồn tại.</p> <p>Người dùng nhập lại thông tin khác.</p>

Bảng 4.2 Mô tả đăng ký

4.2.3 Đăng tin tuyển dụng:

- Mô tả: Người dùng điền thông tin cần tuyển dụng (ngày, giờ dạy, học phí, môn,...). Sau khi đăng tin các gia sư sẽ tiến hành thương lượng và đề nghị nhận lớp. Sau khi nhà tuyển dụng chấp nhận sẽ trừ phí hoa hồng của gia sư (theo quy định) và thông tin tuyển dụng sẽ ẩn đi. Khi người tuyển dụng hoặc gia sư hủy lớp tùy theo quy định sẽ được hoàn tiền lại tài khoản.

Description	Người dùng đăng tin tìm gia sư.
--------------------	---------------------------------

Actors	Khách hàng.
Trigger	Người dùng chọn chức năng đăng tin tuyển dụng gia sư mới.
Pre-Conditions	<ul style="list-style-type: none"> Tài khoản người dùng không bị vô hiệu hóa. Số tin tuyển dụng đang được đăng phải phù hợp với số tin được quy định.
Post-Conditions	Người dùng đăng tin tuyển thành công, hệ thống ghi nhận thông tin lớp mới, cập nhập lên bản tin tuyển dụng và thông báo đăng tin thành công.
Basic Flow	<ol style="list-style-type: none"> Người dùng đã đăng nhập và chọn chức năng tạo tin tuyển dụng mới. Người dùng điền thông tin cần tuyển vào biểu mẫu lớp mới và sau đó chọn chức năng đăng tin. Hệ thống kiểm tra thông tin tài khoản, biểu mẫu hợp lệ và tiến hành ghi nhận. Hệ thống ghi nhận dữ liệu lớp mới và cập nhập lên bản tin. Hệ thống ghi nhận phiên hoạt động của tài khoản.
Exception Flow	<p>- Thông tin người dùng cung cấp không hợp lệ :</p> <p>Hệ thống thông báo ra màn hình thông tin không đúng quy định.</p>

	<p>Người dùng nhập lại thông tin khác hợp lệ.</p> <p>- Tài khoản đã quá số lần tạo tin tuyển dụng:</p> <p>Hệ thống thông báo ra màn hình tài khoản quá số lần tạo tin.</p> <p>Người dùng xử lý các tin tuyển dụng đã đăng để có thể đăng tin mới.</p> <p>- Tài khoản đã bị khóa chức năng đăng tin:</p> <p>Hệ thống thông báo ra màn hình lí do bị vô hiệu hóa của tài khoản người dùng.</p> <p>Người dùng phải xử lý vấn đề bị vô hiệu hóa hoặc gửi mail về bộ phận hỗ trợ để được hướng dẫn.</p>
--	--

Bảng 4.3 mô tả đăng tin tuyển dụng

4.2.4 Tạo hồ sơ gia sư:

- Mô tả: Người dùng muốn nhận lớp sẽ phải điền đầy đủ thông tin(địa chỉ, email, sdt, hình đại diện, số CMND,...). Sau khi xác thực qua tin nhắn hoặc email sẽ được quyền nhận lớp.

Description	Người dùng tạo hồ sơ gia sư.
Actors	Khách hàng.
Trigger	Người dùng chọn chức năng tạo hồ sơ gia sư.
Pre-Conditions	<ul style="list-style-type: none"> • Tài khoản người dùng chưa tạo hồ sơ gia sư. • Tài khoản không bị vô hiệu hóa hoặc khóa.

	<ul style="list-style-type: none"> • Người có độ tuổi phù hợp với quy định lao động.
Post-Conditions	Người dùng tạo hồ sơ thành công và hệ thống cập nhập hồ sơ vào dữ liệu của người dùng. Người dùng có thể sử dụng hồ sơ để ứng tuyển các tin tuyển dụng.
Basic Flow	<ol style="list-style-type: none"> 1. Người dùng đã đăng nhập và chọn chức năng bổ sung hồ sơ gia sư. 2. Người dùng điền thông tin gia sư(hình bằng cấp, CMND, hình bản thân, giới thiệu bản thân,...) vào biểu mẫu hồ sơ gia sư và sau đó chọn chức năng đăng tin. 3. Hệ thống kiểm tra các thông tin biểu mẫu hợp lệ và tiến hành ghi nhận. 4. Hệ thống ghi nhận thành công và chuyển hồ sơ về cho nhân viên duyệt hồ sơ. 5. Nhân viên duyệt hồ sơ hợp lệ và phê duyệt hồ sơ cho người dùng. 6. Hệ thống ghi nhận bổ sung hồ sơ cho tài khoản người dùng và gửi thông báo qua mail hoặc sms.
Exception Flow	<p>- Thông tin người dùng cung cấp không hợp lệ:</p> <ol style="list-style-type: none"> 1. Hệ thống thông báo ra màn hình thông tin không đúng quy định. 2. Người dùng nhập lại thông tin khác hợp lệ.

	<ul style="list-style-type: none"> - Nhân viên kiểm tra hình bằng cấp không hợp lệ (rách, mờ, quá hạn...) 1. Hệ thống thông báo ra màn hình tài khoản quá số lần tạo tin. 2. Người dùng thực hiện chụp lại hoặc tải lại hình phù hợp hơn. -Nhân viên kiểm tra số tuổi của tài khoản không phù hợp luật lao động: 1. Nhân viên gửi phản hồi về cho tài khoản người dùng. 2. Người dùng nhận được phản hồi.
--	---

Bảng 4.4 bảng mô tả tạo hồ sơ

4.2.5 Nạp tiền vào tài khoản

- Mô tả: Người dùng có thể nạp tiền vào tài khoản trang web để sử dụng 1 số chức năng. Phía người tuyển dụng có thể sử dụng tiền nạp vào để đẩy trang web mình lên đầu trong 1 khoảng thời gian. Phía gia sư mỗi lần đề nghị nhận 1 lớp sẽ mất 1 khoảng phí nhỏ và sau khi nhà tuyển dụng đồng ý gia sư sẽ mất 1 khoảng tiền hoa hồng. Phí có thể xem phần quy định.

Description	Người dùng nạp tiền trả phí sử dụng dịch vụ gia sư.
Actors	Khách hàng.
Trigger	Người dùng chọn chức năng nạp tiền vào tài khoản web gia sư.

Pre-Conditions	<p>Tài khoản người dùng có liên kết với các nguồn ví hoặc thẻ (Thẻ nội địa, ví điện tử,...).</p> <p>Tài khoản không bị vô hiệu hóa hoặc khóa.</p>
Post-Conditions	<p>Người dùng nạp tiền vào tài khoản thành công tiến hành chuyển phí về cho website.</p> <p>Người dùng sử dụng những dịch vụ phí như phí hoa hồng sau khi nhận lớp hoặc phí yêu cầu dạy của tin tuyển.</p>
Basic Flow	<p>Người dùng đã đăng nhập và chọn chức năng nạp tiền vào tài khoản.</p> <p>Người dùng được chuyển sang trang thanh toán qua bằng các cổng thanh toán (thẻ thanh toán nội địa hoặc các hình như thanh toán theo như người quản trị đăng ký.)</p> <p>Người dùng cung cấp thông tin thẻ hoặc ví và chọn mức giá muốn nạp.</p> <p>Hệ thống kiểm tra xác thực thông tin thanh toán của khách hàng thành công và ghi nhận giao dịch thành công.</p> <p>Hệ thống thông báo nhận được phí thành công và tiến hành chuyển số tiền tương ứng vào tài khoản người dùng.</p> <p>Người dùng có thể sử dụng số tiền điểm này để thanh toán các dịch vụ cung cấp trên web.</p>

Exception Flow	<p>- Thông tin thanh toán người dùng cung cấp không hợp lệ: Hệ thống thông báo ra màn hình thông tin thanh toán không chính xác. Người dùng nhập lại chính xác thông tin khác hợp lệ.</p> <p>- Số tiền trong ví hoặc trong thẻ của khách hàng không đủ số dư để thực hiện giao dịch: Hệ thống thông báo ra màn hình tài khoản có số dư không đủ. Người dùng đổi nguồn thanh toán khác hoặc nạp tiền vào thẻ để có thể tiến hành thanh toán.</p>
-----------------------	---

Bảng 4.5 bảng mô tả nạp tiền vào tài khoản

4.2.6 Tra cứu thông tin

- Mô tả: Người dùng có thể tra cứu thông tin về việc làm, hoặc các gia sư để xem chất lượng giảng dạy.

Description	Người dùng tra cứu thông tin tuyển dụng gia sư (lớp).
Actors	Khách hàng.
Trigger	Người dùng chọn chức năng tìm kiếm thông tin lớp.
Pre-Conditions	<ul style="list-style-type: none"> • Người dùng truy cập vào đường dẫn đến trang web gia sư.
Post-Conditions	<ul style="list-style-type: none"> - Người dùng tra cứu thông tin lớp có trong dữ liệu tin tuyển. - Người dùng xem xét thông tin tuyển dụng phù hợp để ứng tuyển.

<p>Basic Flow</p>	<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang web và chọn chức năng tìm kiếm lớp. 2. Người dùng cung cấp các từ khóa để lọc thông tin lớp tuyển dụng (theo vị trí, học phí, môn học,...). 3. Người dùng chọn lệnh tìm kiếm và dữ liệu được hệ thống chuyển về server. 4. Server tiếp nhận thông tin cần tìm kiếm và tiến hành lọc danh sách kết quả lớp theo từ khóa. 5. Sau khi tìm thấy các lớp phù hợp với các từ khóa sẽ trả kết quả phản hồi lại cho phía client. 6. Kết quả được browser phía client tiếp nhận và hiển thị thông tin danh sách các lớp lên màn hình khách hàng. 7. Khách hàng nhận được danh sách các lớp theo yêu cầu và lựa chọn.
<p>Exception Flow</p>	<p>- Thông tin tìm kiếm không hợp lệ: Hệ thống thông báo lỗi về các thông tin không hợp lệ. Người dùng tiến hành nhập lại các từ khóa hợp lệ.</p> <p>- Thông tin lớp theo các từ khóa tìm kiếm không tồn tại:</p> <ol style="list-style-type: none"> 1. Hệ thống thông báo ra màn hình không có tin tuyển dụng nào phù hợp với tiêu chí tìm kiếm. 2. Người dùng tìm kiếm các lớp khác.

Bảng 4.6 Bảng mô tả tra cứu thông tin

4.2.7 Hỗ trợ khách hàng

- Mô tả: Người dùng khi muốn hỗ trợ 1 vấn đề sẽ gửi mail về địa chỉ GaSuSupport@gmail.com để nhận được sự hỗ trợ từ nhân viên tư vấn và hỗ trợ. (hủy lớp, hoàn tiền,...)

Description	Hỗ trợ khách hàng.
Actors	Khách hàng.
Trigger	Người dùng gửi mail về bộ phận hỗ trợ khách hàng của web.
Pre-Conditions	<ul style="list-style-type: none"> • Người dùng gặp các sự cố liên quan tới vấn đề về dịch vụ web(hủy lớp, khiếu nại,...).
Post-Conditions	<ul style="list-style-type: none"> - Người dùng truy cập trang web. - Người dùng chọn mục hỗ trợ khách hàng. - Người dùng đọc các điều khoản được khiếu nại theo quy định. - Người dùng gửi mail kèm vấn đề cần hỗ trợ về bộ phận chăm sóc khách hàng.
Basic Flow	<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang web và chọn mục hỗ trợ khách hàng. 2. Hệ thống hiển thị mục hỗ trợ khách hàng kèm các điều khoản và hướng giải quyết vấn đề. 3. Người dùng xem xét và thực hiện theo các vấn đề phù hợp để được hỗ trợ. 4. Khách hàng gửi mail kèm thông tin chi tiết vấn đề về mail hỗ trợ khách hàng của bộ phận hỗ trợ. 5. Sau khi nhận được mail hỗ trợ và xem

	<p>xét phù hợp điều kiện hỗ trợ, nhân viên sẽ phản hồi cách thức để giải quyết vấn đề và hướng dẫn từng bước về khách hàng.</p> <p>6. Khách hàng nhận được mail phản hồi và thực hiện theo sự hướng dẫn của nhân viên hỗ trợ.</p>
Exception Flow	<p>- Thông tin hỗ trợ không hợp lệ với quy định về hỗ trợ của dịch vụ web gia sư:</p> <ol style="list-style-type: none"> 1. Nhân viên gửi phản hồi về việc vấn đề không hợp lệ để được giải quyết và chú thích lí do. 2. Người dùng nhận và gửi mail hỗ trợ khác (nếu có).

Bảng 4.7 bảng mô tả hỗ trợ khách hàng

4.2.8 Thay đổi quy định

- Mô tả: Người quản trị (admin) có thể thay đổi các quy định trên trang web (mức phí, số lớp có thể nhận...).

Description	Thay đổi quy định.
Actors	Quản trị viên.
Trigger	Quản trị viên chọn mục quy định và thực hiện chức năng thay đổi quy định.
Pre-Conditions	Công ti có các hướng phát triển mới cần thay đổi hoặc cập nhập các quy định mới.
Post-Conditions	Người dùng thay đổi quy định thành công và hệ thống ghi nhận kết quả thay đổi kèm phiên hoạt động của khách hàng .
Basic Flow	Người dùng truy cập vào trang quản trị của

	<p>hệ thống.</p> <p>Người dùng chọn mục quy định.</p> <p>Người dùng thay đổi hoặc thêm quy định mới vào theo biểu mẫu thiết kế sẵn.</p> <p>Người dùng chọn lưu kết quả và kết quả được kiểm tra thông tin hợp lệ.</p> <p>Hệ thống kiểm tra thông tin hợp lệ và cập nhập kết quả thay đổi quy định này vào dữ liệu và gửi phản hồi thành công về client.</p> <p>Hiện thị thông cập nhập quy định thành công lên phía màn hình quản trị viên.</p> <p>Các quy định sau khi cập nhập sẽ được áp dụng lên các chức năng dịch vụ web. (phí nhận lớp, tuổi tối đa,...).</p>
Exception Flow	<p>- Thông tin không hợp lệ:</p> <ol style="list-style-type: none"> 1. Thông báo lỗi ra màn hình của người quản trị. 2. Người dùng sửa lại nội dung quy định phù hợp

Bảng 4.8 Mô Tả thay đổi quy định

4.2.9 Lập báo cáo

- Mô tả: Người quản trị có thể xem và xuất báo cáo doanh thu, số lượng người dùng....

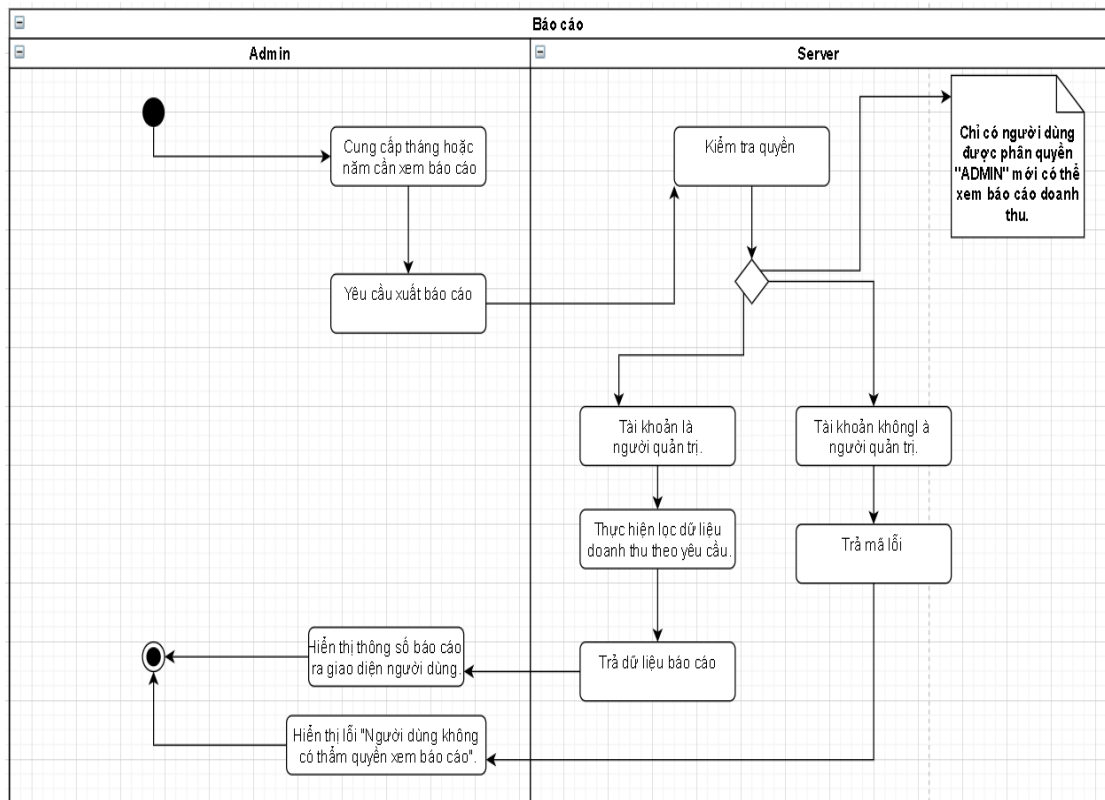
Description	Lập báo cáo
Actors	Quản trị viên.
Trigger	Quản trị viên chọn chức năng xem báo cáo trên trang quản trị. (Báo cáo mức hoạt động, doanh thu,...)
Pre-Conditions	Người dùng được phân quyền là quản trị

	viên. Tài khoản quản trị phải còn hiệu lực. Mốc thời gian báo cáo phải phù hợp.
Post-Conditions	Báo cáo sau khi được xuất sẽ được phân tích và định hướng hướng phát triển của công ti.
Basic Flow	Người dùng truy cập vào trang quản trị của hệ thống. Người dùng chọn mục báo cáo. Người dùng chọn mốc thời gian để xem báo cáo. Người dùng có thể xuất báo cáo sau khi xem. Báo cáo được tải về theo định dạng được quy định sẵn (Excel...) Hiện thị thông báo xuất báo cáo thành công.
Exception Flow	- Thông tin mốc thời gian xem báo cáo không hợp lệ: Thông báo lỗi ra màn hình của người quản trị. Người dùng sửa lại mốc thời gian phù hợp.

Bảng 4.9: mô tả quy trình lập báo cáo

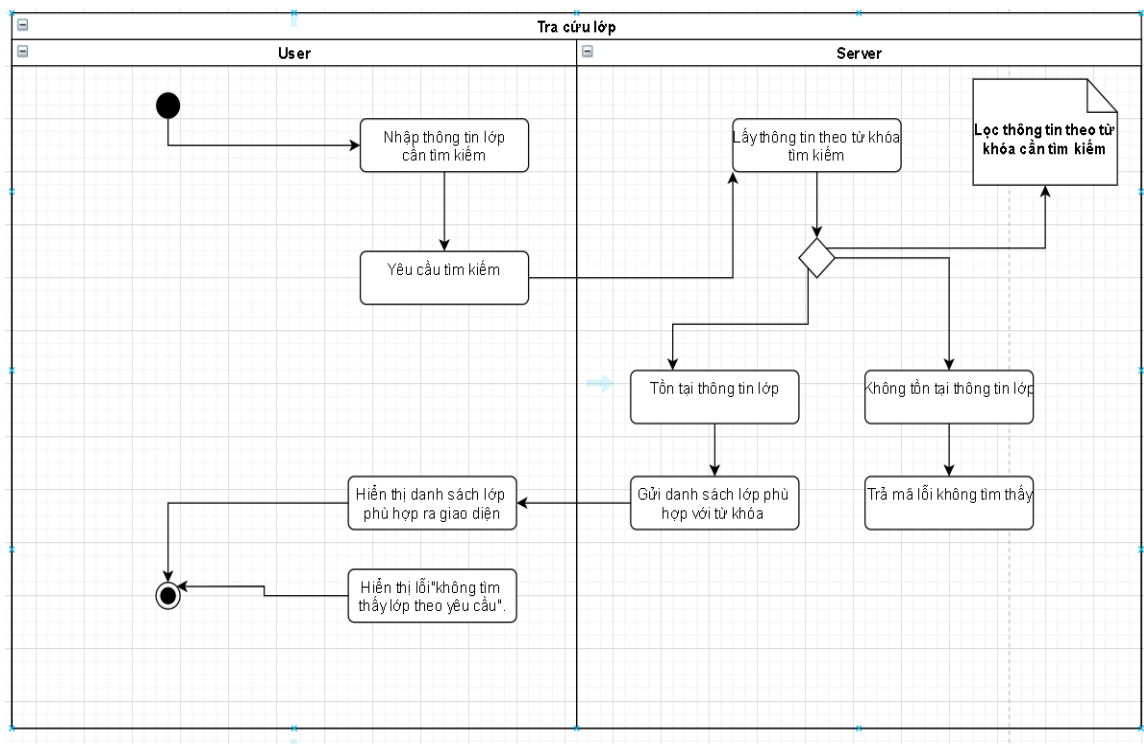
4.3 Mô hình quy trình nghiệp vụ (Business Process Modeling)

4.3.1 Quy trình lập báo cáo:



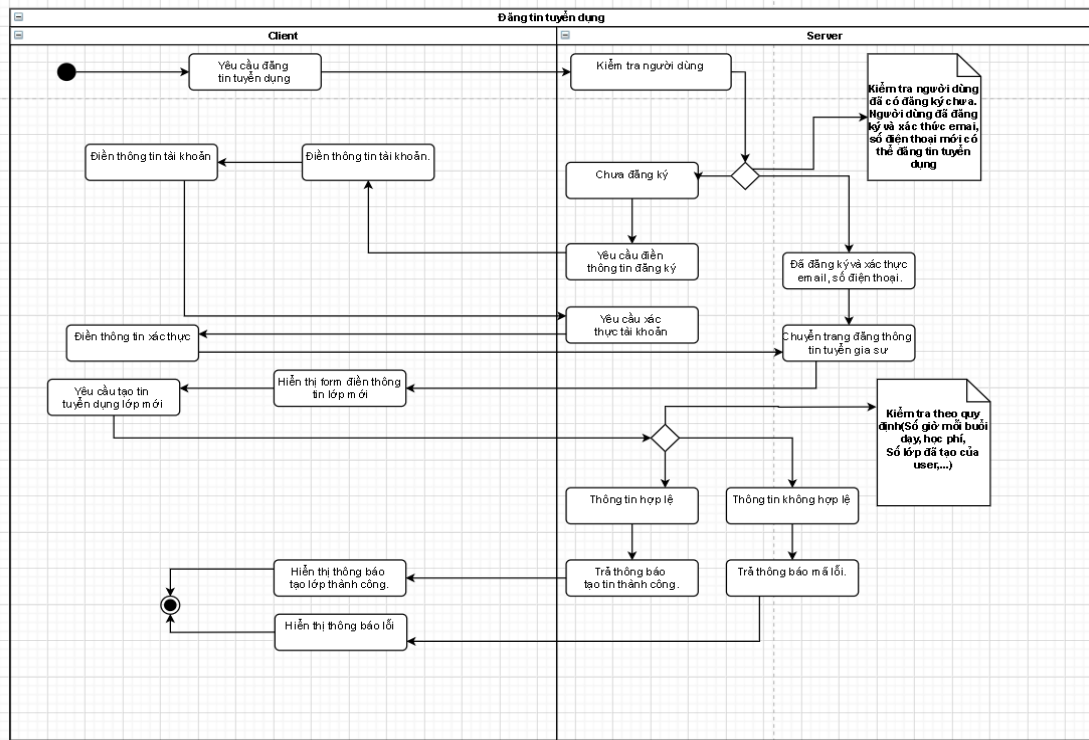
Hình 4.2 hình quy trình lập báo cáo

4.3.2 Quy trình tra cứu lớp:



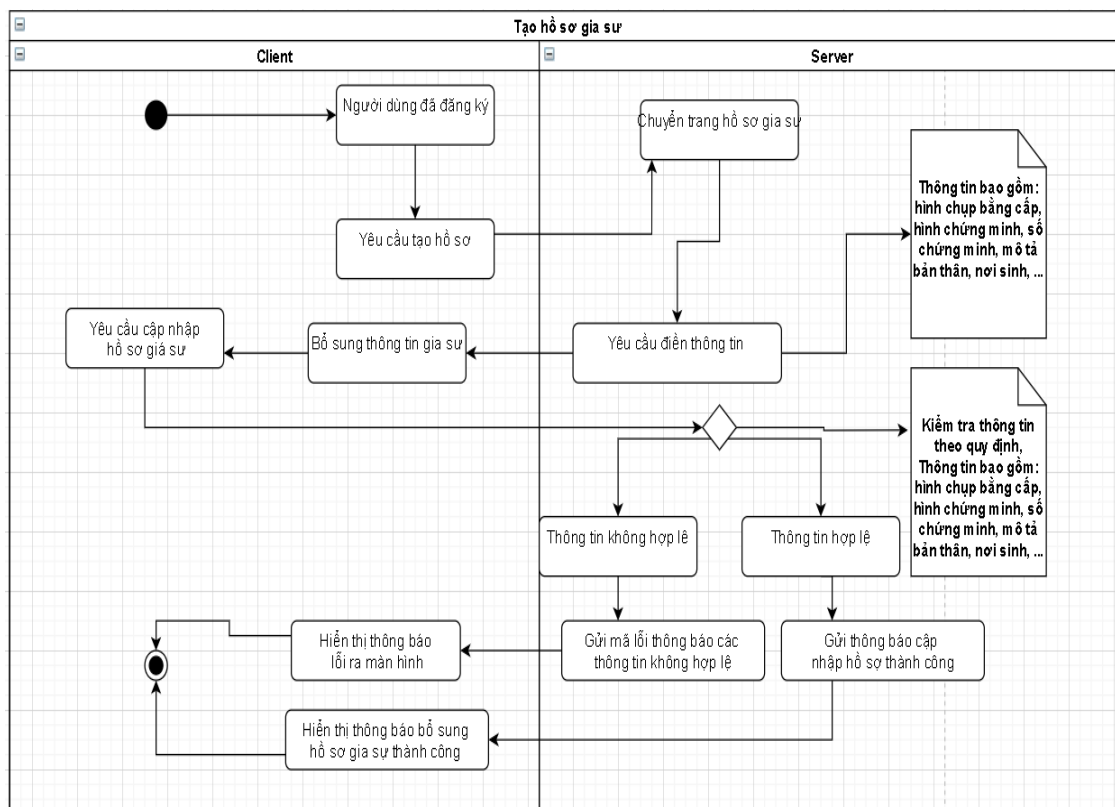
Hình 4.3 Quy trình tra cứu lớp

4.3.3 Quy trình đăng tin tuyển dụng:



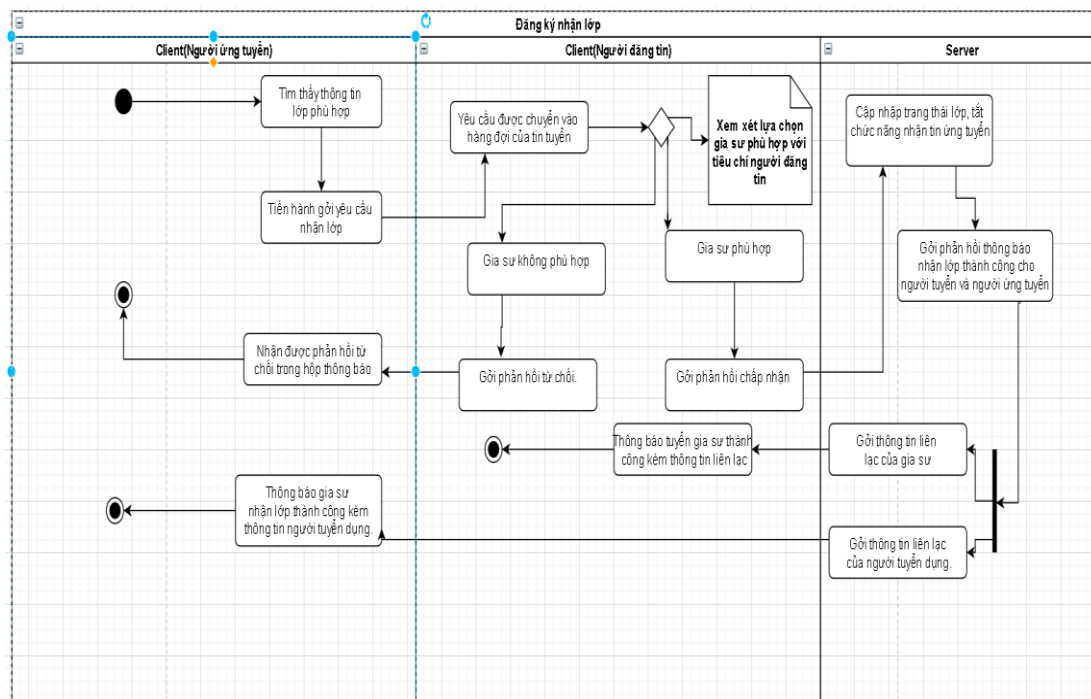
Hình 4.4 quy trình đăng tin tuyển dụng

4.3.4 Quy trình tạo hồ sơ gia sư:



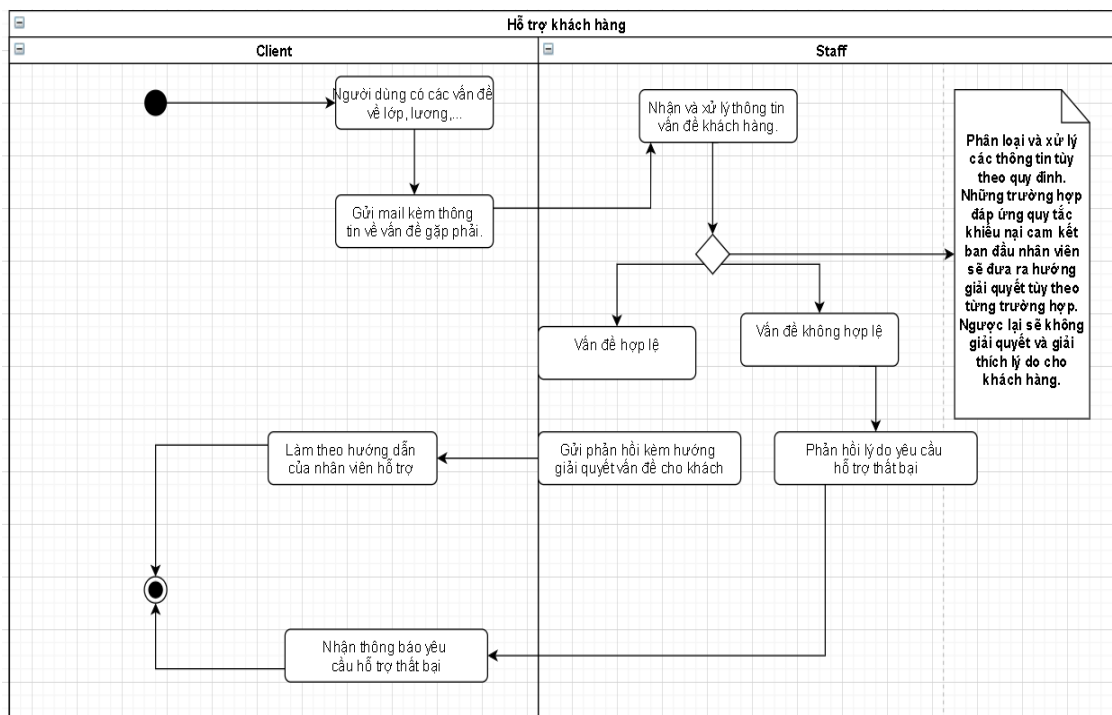
Hình 4.5 Quy trình tạo hồ sơ gia sư

4.3.5 Quy trình đăng ký nhận lớp:



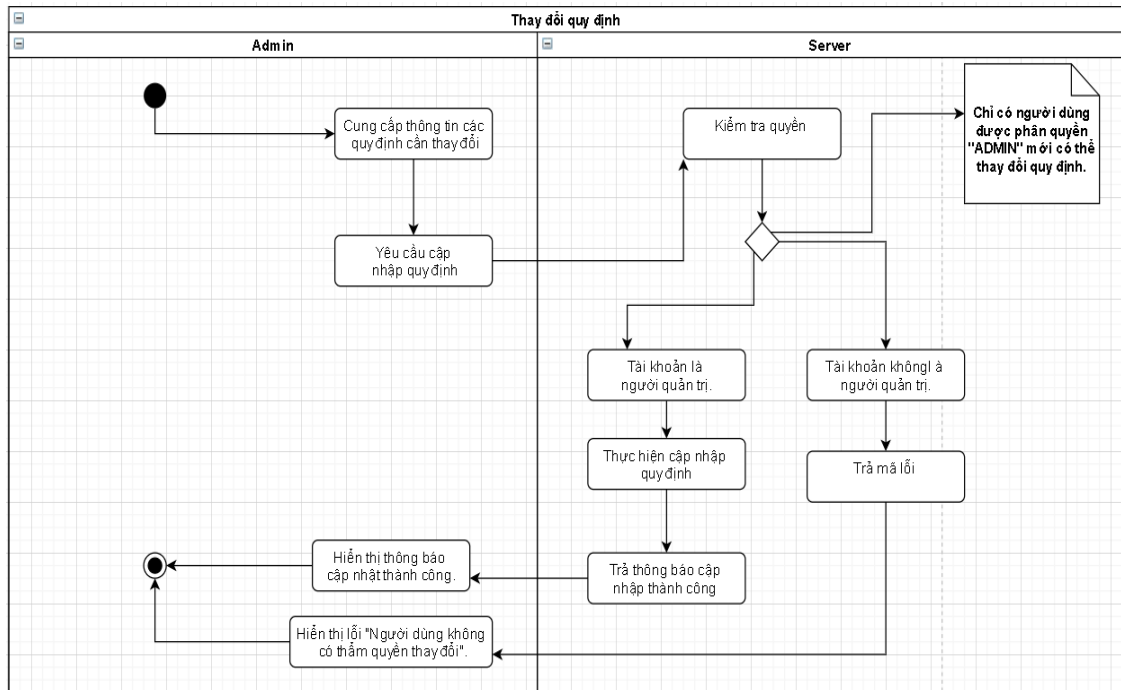
Hình 4.6 quy trình đăng ký nhận lớp

4.3.6 Quy trình hỗ trợ khách hàng:



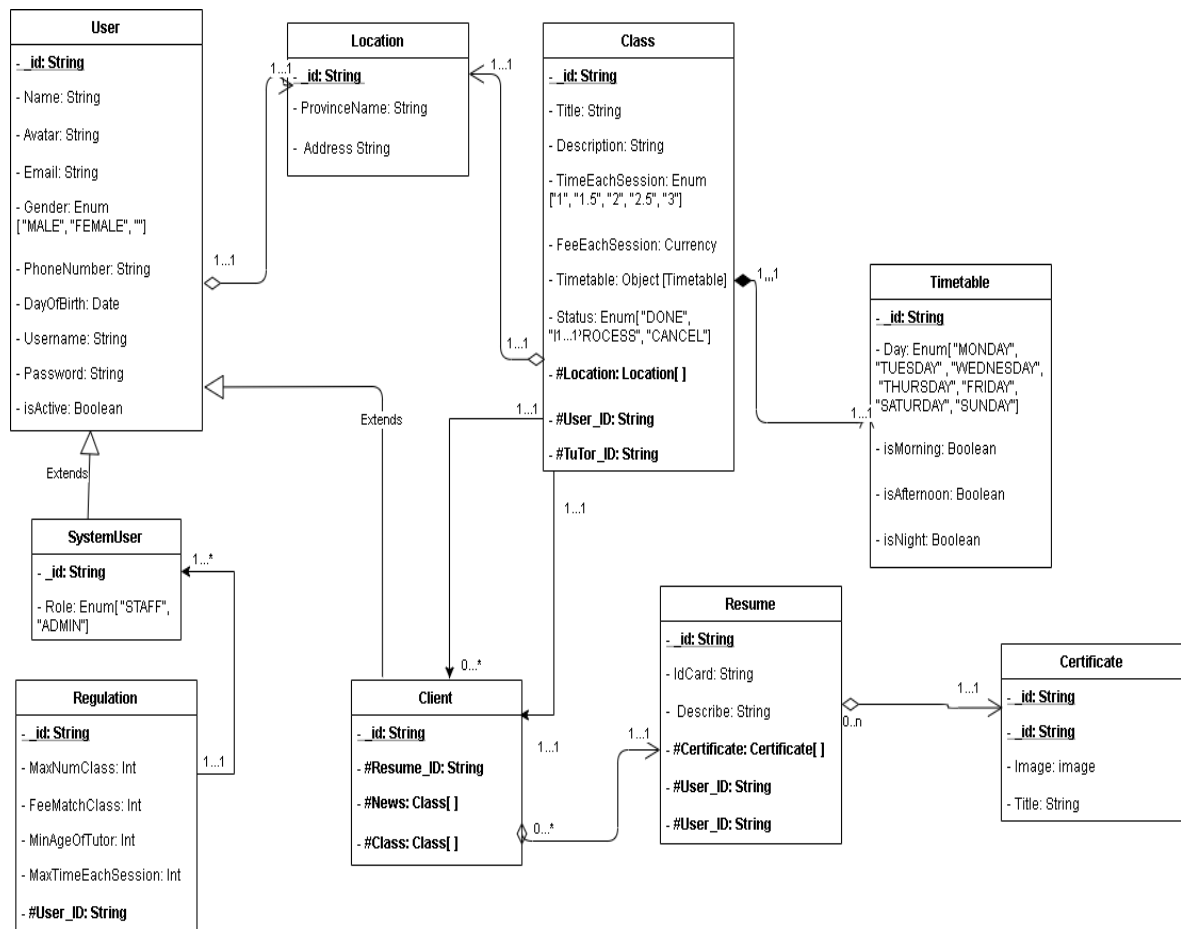
Hình 4.7 quy trình hỗ trợ khách hàng

4.3.7 Quy trình thay đổi quy định:



Hình 4.8 quy trình thay đổi quy định

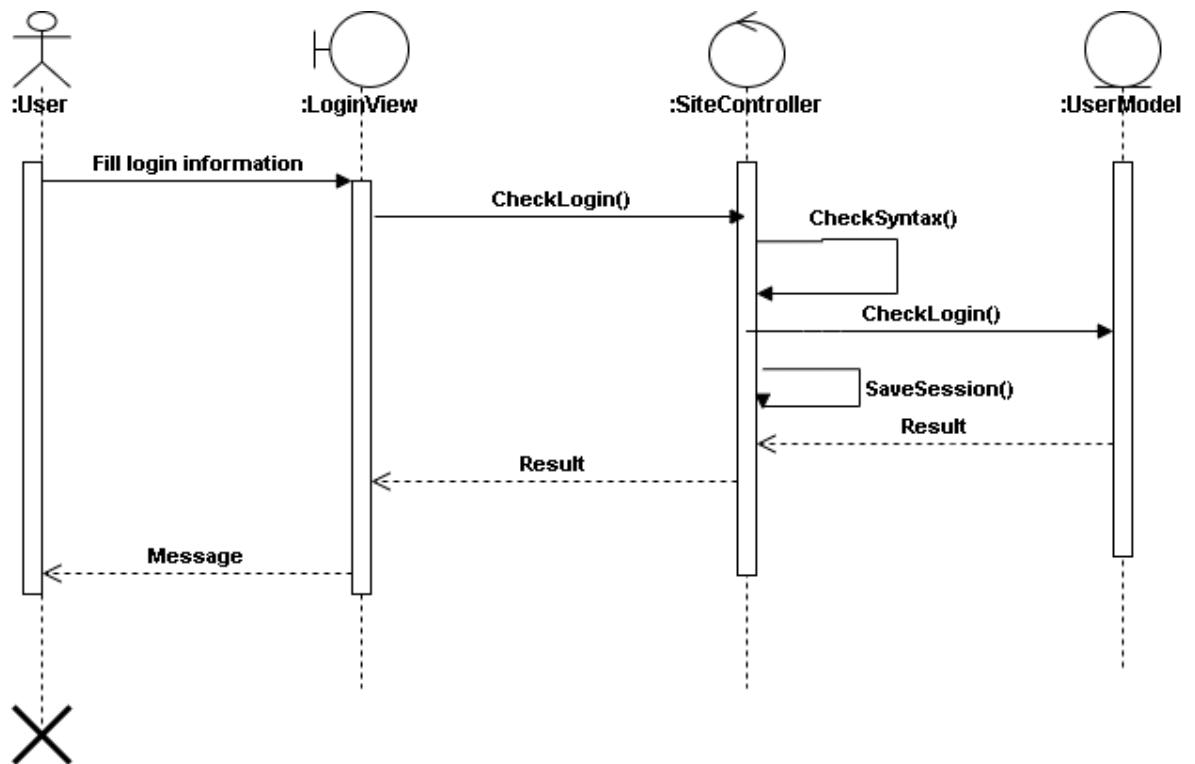
4.4 mô hình lớp (Class Diagram)



Hình 4.9 mô hình lớp

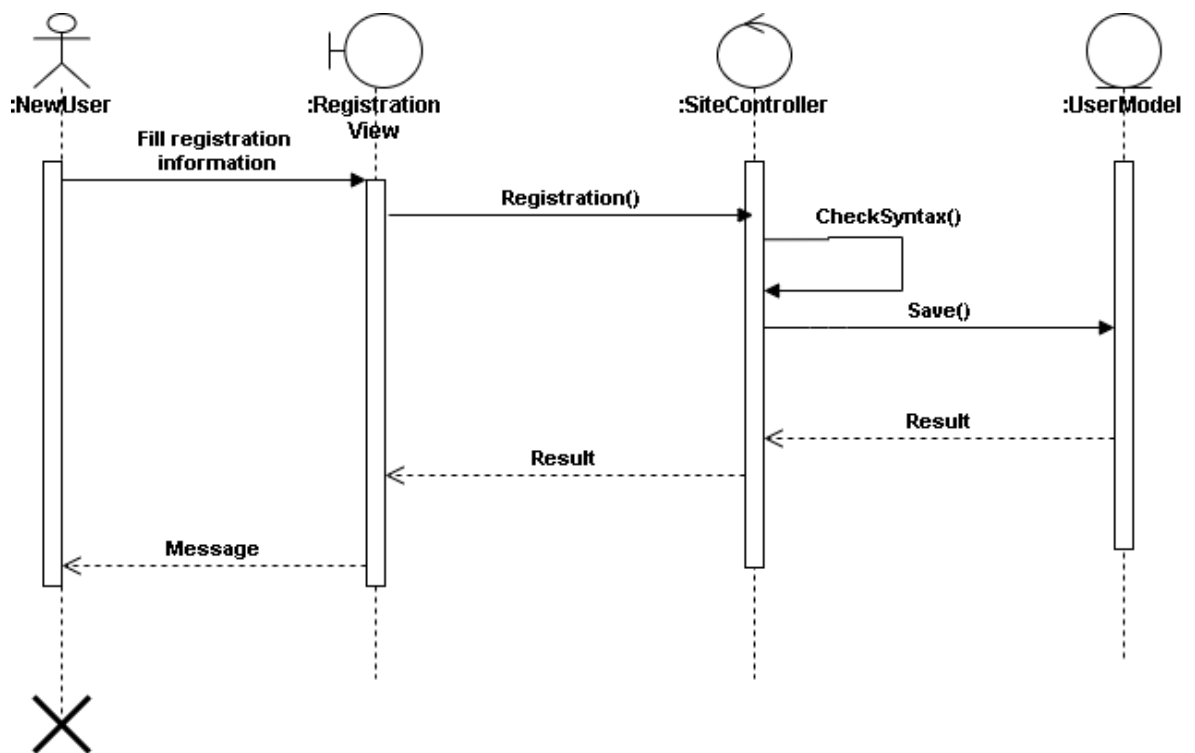
4.5 Sơ đồ trình tự (Sequence Diagram)

4.5.1 Đăng nhập:



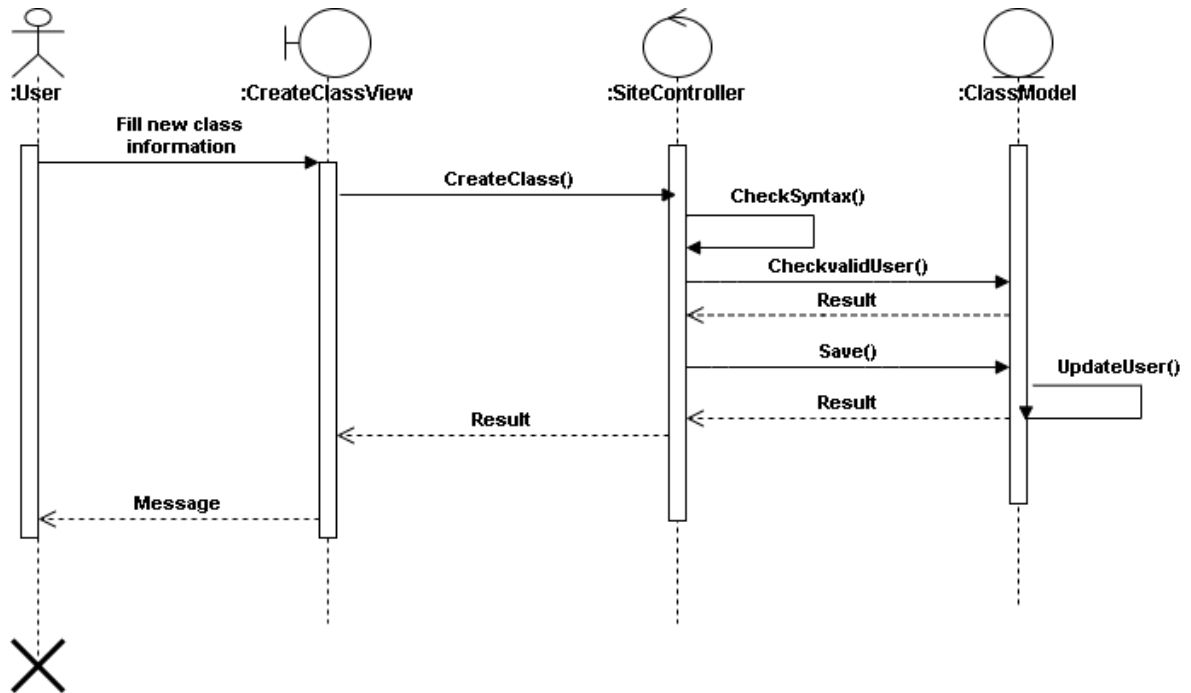
Hình 4.19: Sơ đồ trình tự đăng nhập

4.5.2 Đăng ký:



Hình 4.20: Sơ đồ trình tự đăng ký

4.5.3 Tạo tin tuyển dụng (lớp mới):



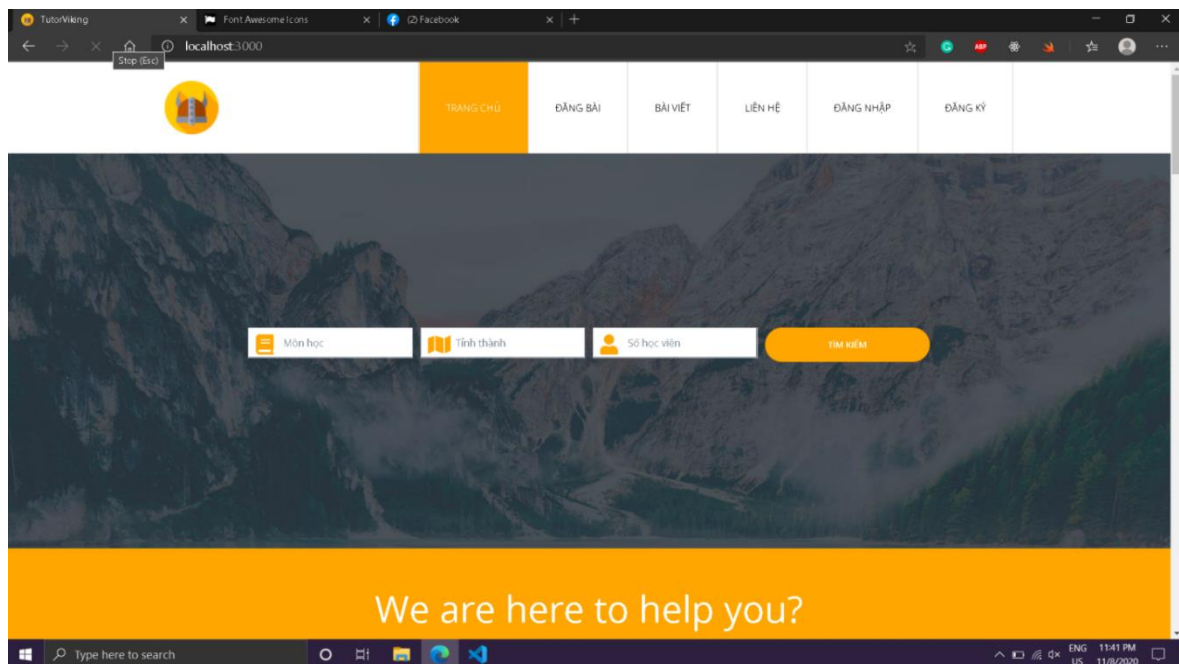
Hình 4.21: Sơ đồ trình tự tạo tin tuyển dụng

4.6 Thiết kế giao diện

4.6.1 Giao diện trang chủ

Trang chủ “Index” là trang chính, chứa tất cả các mục để lựa chọn đăng nhập, đăng ký, bài viết, tạo lớp...

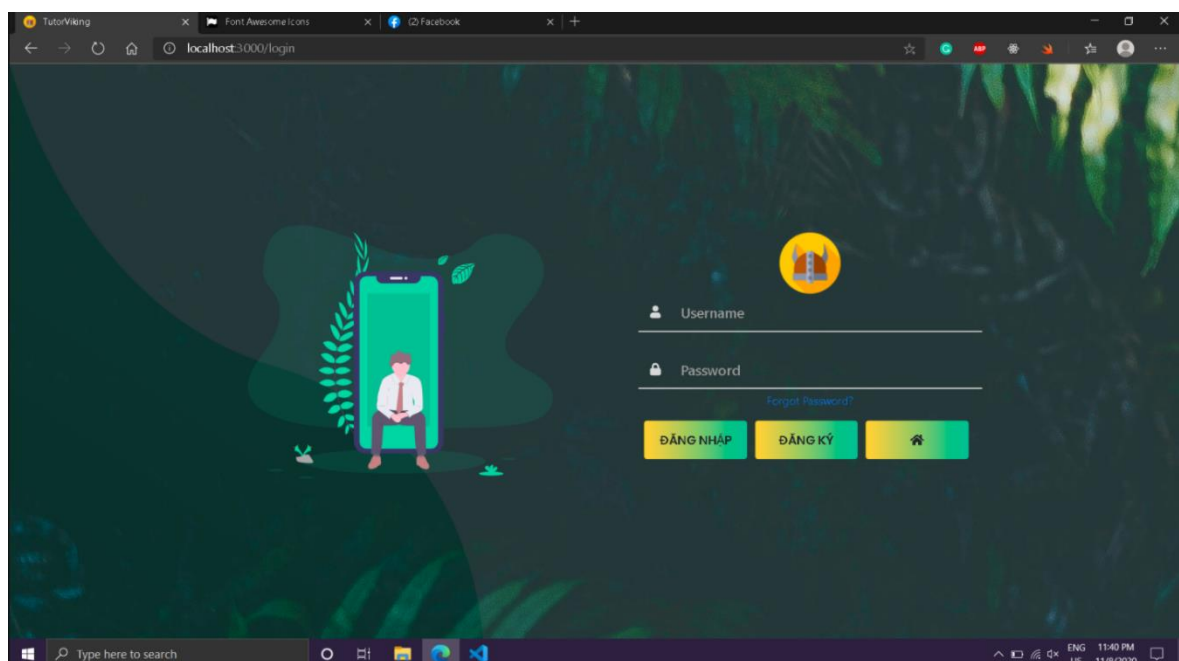
- Giao diện thiết kế lần này màu sắc có một chút sắc sỡ, mạnh mẽ nhưng lại thân thiện và gần gũi và dễ sử dụng cho khách hàng.



Hình 4.10 Giao diện trang chủ

4.6.2 Giao diện đăng nhập

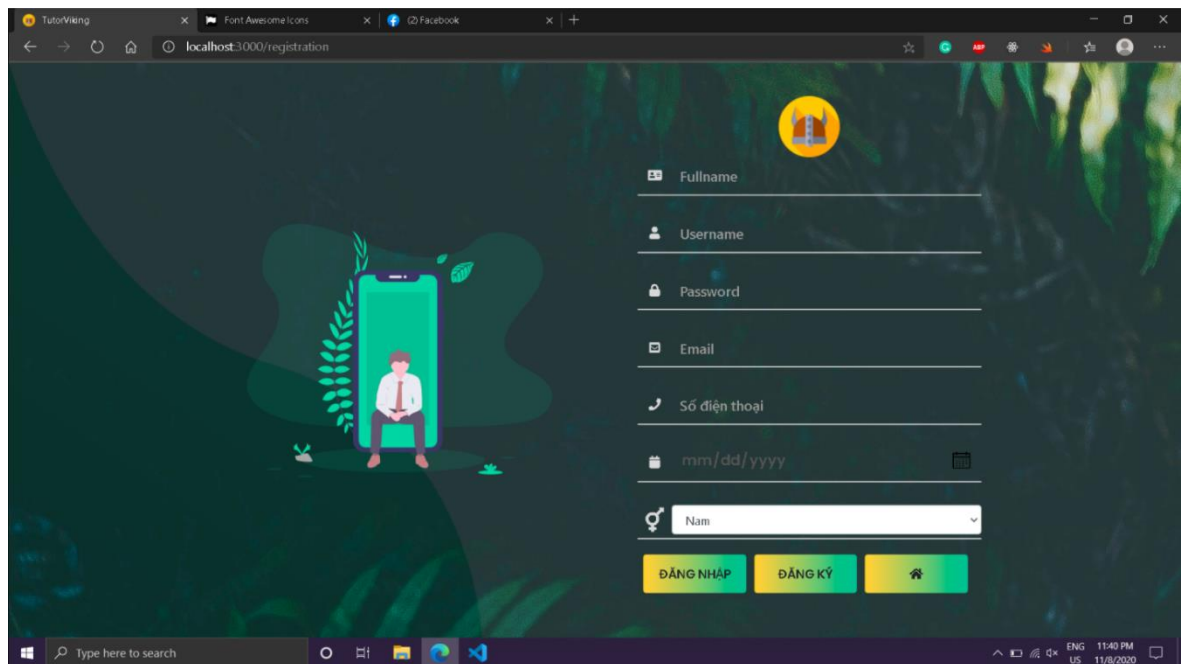
Trang đăng nhập là trang mà khách hàng sẽ nhập User và Pass của mình để đăng nhập vào website, nếu khách hàng quên mật khẩu có thể nhấn vào nút Forgot Password để tìm lại mật khẩu.



Hình 4.11 Giao diện đăng nhập

4.6.3 Giao diện Đăng ký

Trang đăng ký. Tại đây khách hàng sẽ điền đầy đủ thông tin cá nhân và đăng ký cho mình một tài khoản.



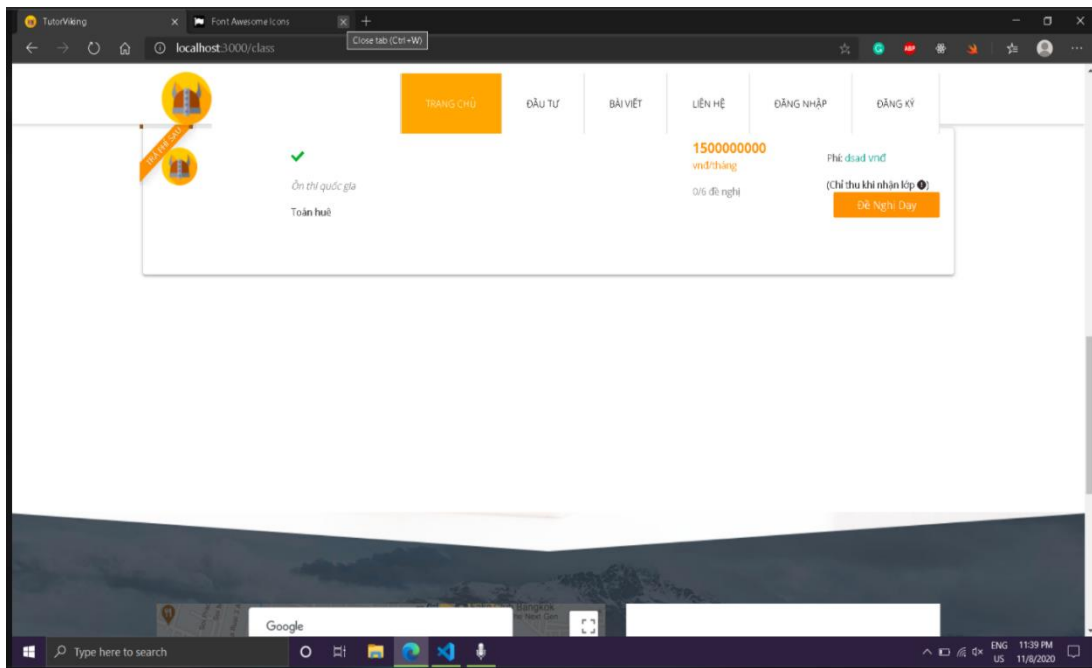
The screenshot displays a web browser window with the address bar showing 'localhost:3000/registration'. The page has a dark green background with a stylized illustration of a person sitting on a large smartphone screen. The registration form on the right side includes the following fields and controls:

- Fullname
- Username
- Password
- Email
- Số điện thoại
- mm/dd/yyyy (date picker)
- Gender dropdown menu with 'Nam' selected
- Three buttons at the bottom: 'ĐĂNG NHẬP' (Login), 'ĐĂNG KÝ' (Register), and a home icon.

Hình 4.12 Giao diện đăng ký

4.6.4 Trang Tin tuyển dụng lớp

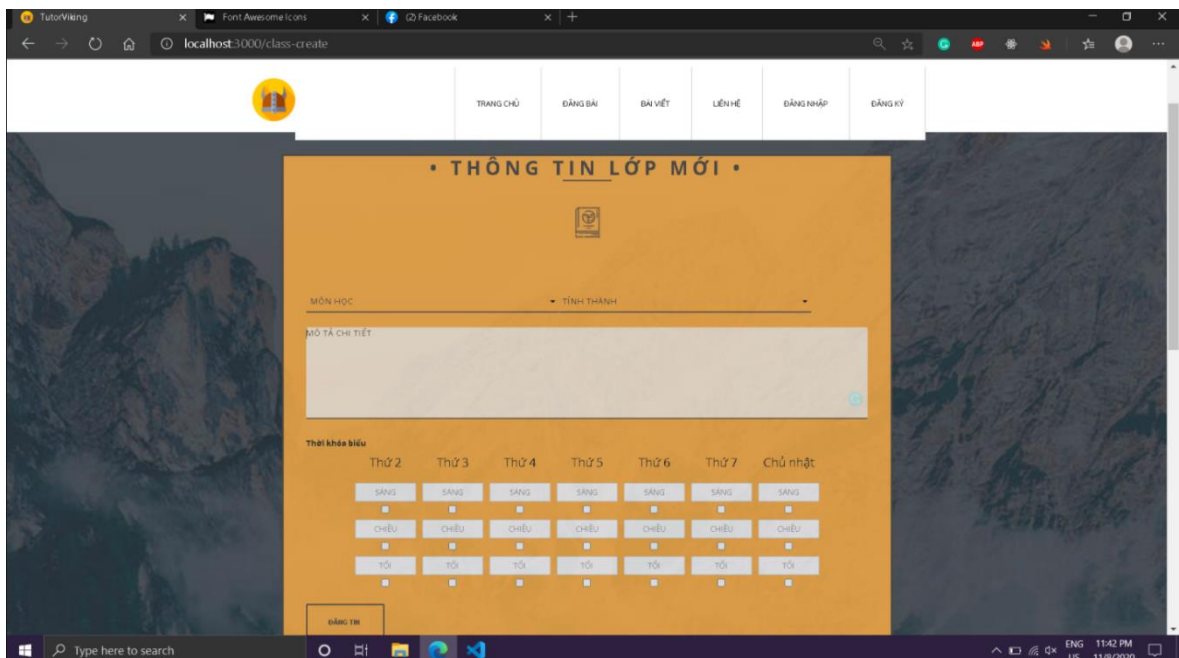
Tại trang đăng tin tuyển dụng này gia sư có thể xem và nhận các lớp học phù hợp với mình.



Hình 4.13 Giao diện Tuyển dụng lớp

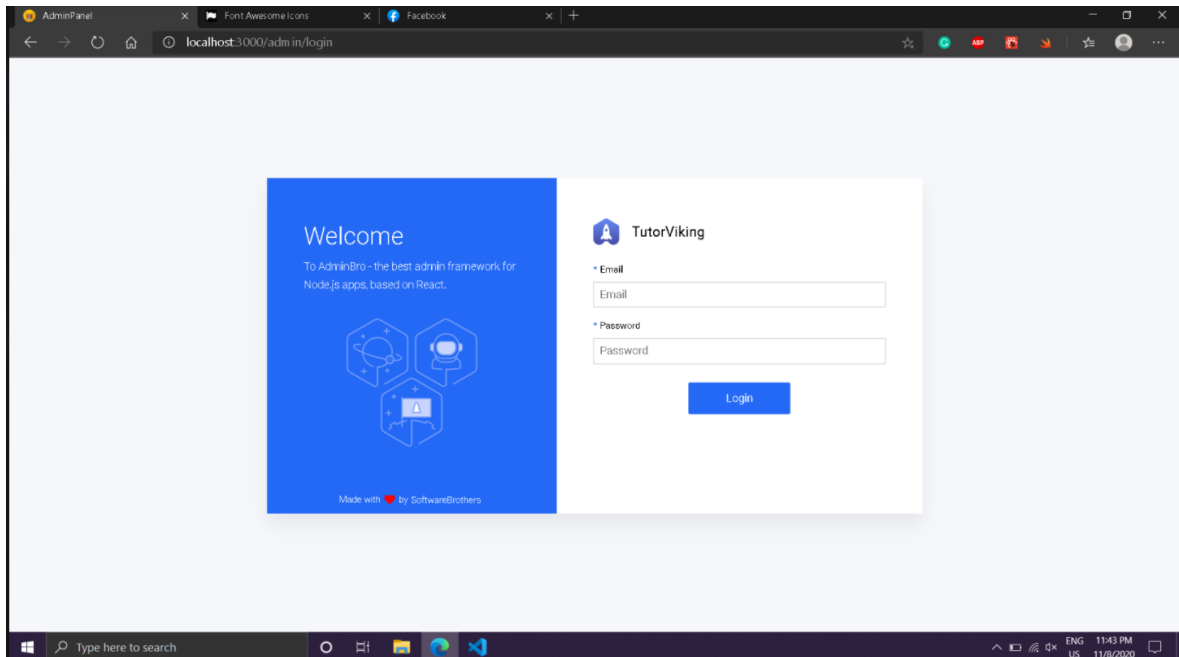
4.6.5 Trang tạo lớp

Trang tạo lớp nhằm tạo một lớp mới theo thông tin mà người tạo muốn.



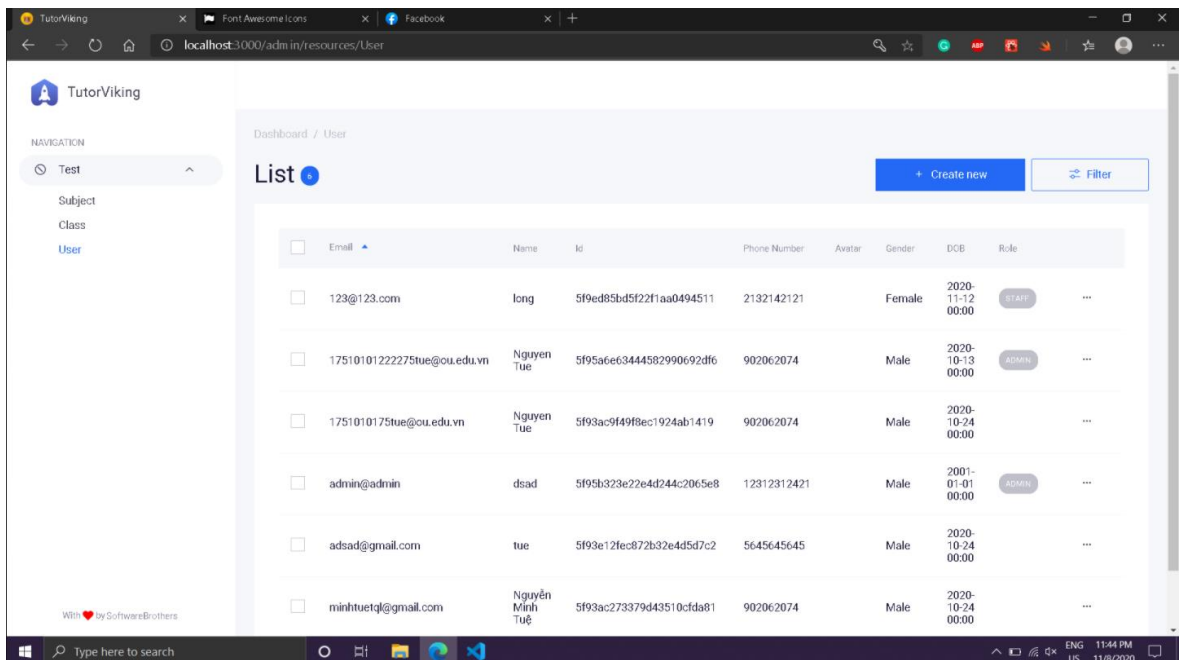
Hình 4.14 Giao diện tạo lớp

4.6.6 Quản trị đăng nhập



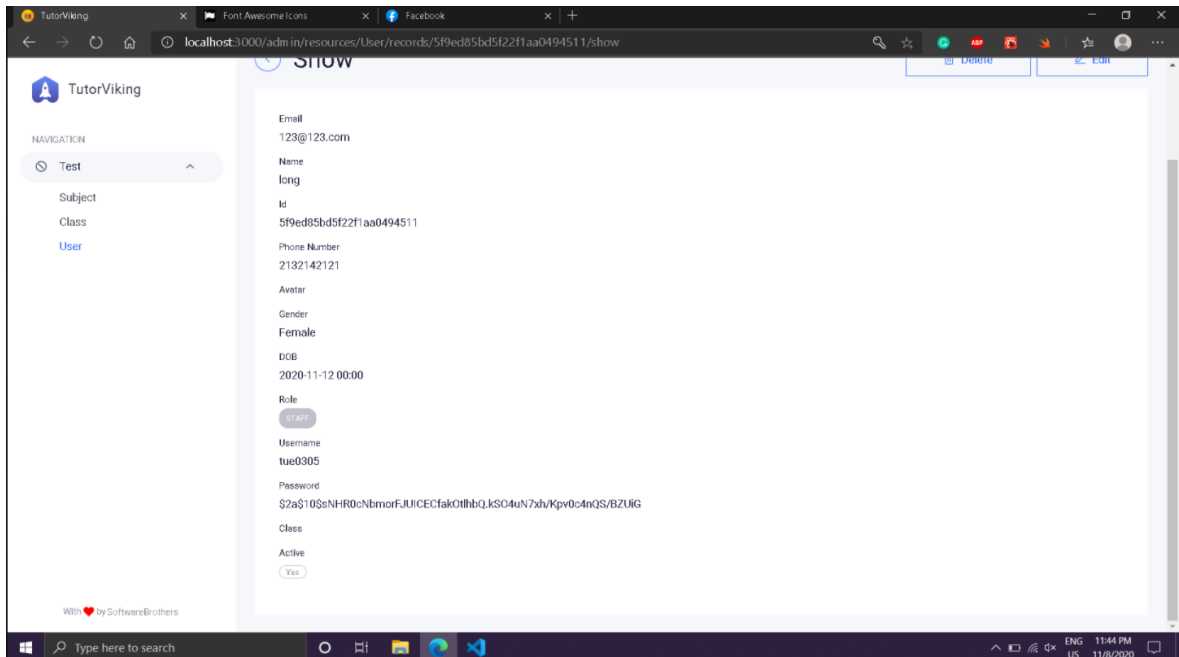
Hình 4.15 Giao diện quản trị đăng nhập

4.6.7 Giao diện quản trị



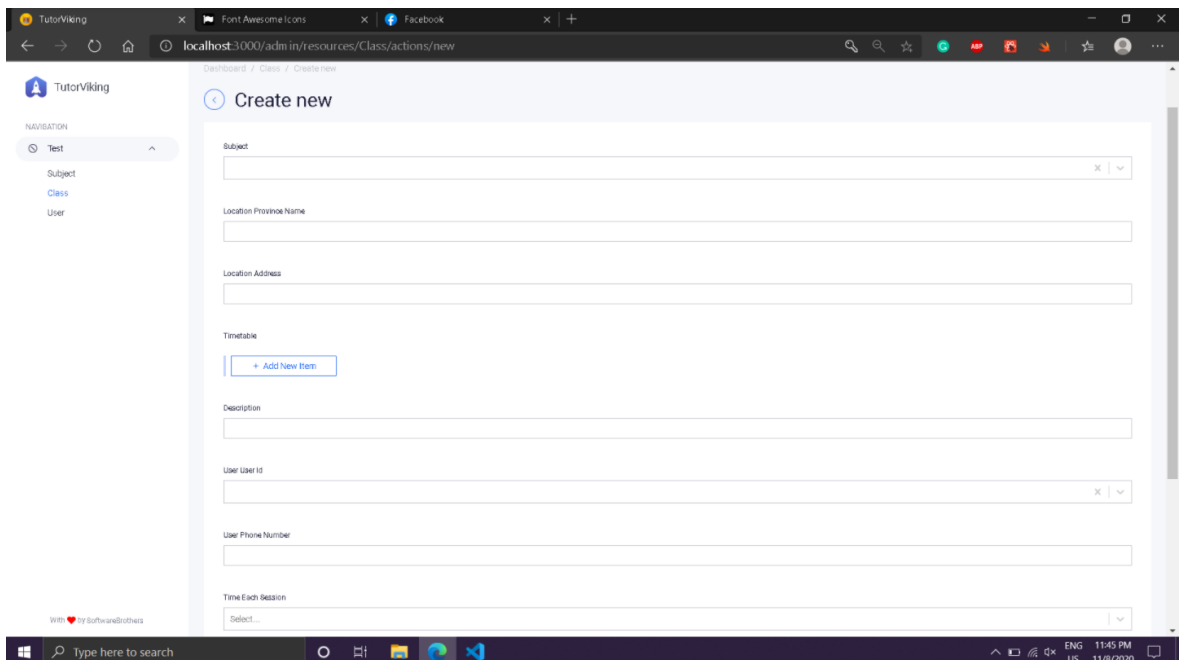
Hình 4.16 Giao diện Quản trị

4.6.8 Chi tiết người dùng



Hình 4.17 Giao diện chi tiết người dùng

4.6.9 Trang tạo thông tin lớp mới



Hình 4.18 Giao diện tạo thông tin lớp mới

Lời Kết

Đề tài “Xây dựng Website Gia sư” với ngôn ngữ Nodejs đối với chúng em tương đối khó khăn, cũng xuất phát từ thực tế ngày nay nhằm tạo ra một nền tảng thuê gia sư tiếp cận đến vấn đề học tập một cách nhanh chóng, hỗ trợ cho tất cả mọi người.

Mặc dù chúng em đã có nhiều cố gắng, tìm hiểu tương đối nhiều kiến thức về ngôn ngữ này, cùng kết hợp với tra cứu tài liệu chuyên ngành nhưng vì hạn chế về nhiều mặt thời gian, kinh nghiệm nên chúng em không thể tránh khỏi những thiếu sót về bài làm cũng như trình bày. Chúng em đã cố gắng hết sức và đã hoàn thành những mục độ sau:

- Tìm hiểu chuyên sâu về ngôn ngữ lập trình NodeJs và Mongodb áp dụng và thực hành vào bài làm “Website Gia sư” .
- ứng dụng được kinh nghiệm đã học vào bài làm một cách hiệu quả.

Hướng nghiên cứu và phát triển:

- chúng em sẽ tiếp tục tìm hiểu chuyên sâu về ngôn ngữ Nodejs này để có thể áp dụng nhiều hơn nữa vào thực tiễn đời sống của chúng ta hiện nay.
- Tìm hiểu thêm về một số ngôn ngữ để có thể nâng cao phần giao diện và các thao tác xử lý
- Xây dựng trang web với quy mô lớn hơn và nhiều chức năng hơn.

Và cuối cùng là để hoàn thành đề tài này một cách hoàn chỉnh, một lần nữa chúng em xin được phép gửi lời cảm ơn chân thành đến thầy **Nguyễn Chí Thanh** là người đã quan tâm, luôn giúp đỡ chúng em trong khoảng thời gian vừa qua.

Trân trọng!

- Nguyễn Cao Tấn
- Nguyễn Minh Tuệ

TÀI LIỆU THAM KHẢO

- [1]. Thanh tai, vu (2016). Học Lập Trình Nodejs Cơ Bản[Online], viewed 19 October 2020, form :< <https://toidicode.com/nodejs-co-ban>>.
- [2]. Viblo Team(2014). Tự học nodejs từ đầu[Online], viewed 21 October 2020, Form :< <https://viblo.asia/p/tu-hoc-nodejs-tu-dau-6J3ZgnGPKmB>>.
- [3]. NguyenCongHa (2016). Tổng quan về Internet[Online], view 15 October 2020, Form:<<https://hoc247.net/tin-hoc-dai-cuong/bai-1-tong-quan-ve-internetl9055.html>> .
- [4]. McPherson, Stephanie Sammartino(ed) (2009). Tim Berners-Lee: Inventor of the World Wide Web. Twenty-First Century Books . ISBN 978-0-8225-7273-2. viewed 28 October 2020, Form :<https://en.wikipedia.org/wiki/World_Wide_Web>
- [5]. MatBao(2002). MongoDB: Tính năng nổi bật từ MongoDB bạn cần biết[Online], viewed 1 November 2020, Form:< <https://wiki.matbao.net/mongodb-la-gi-tinh-nang-noi-bat-tu-mongodb-ban-can-biet/>>.

PHỤ LỤC

Phụ Lục 1: Bìa Đồ Án



**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN NGÀNH

WEBSITE GIA SƯ



Ngành : Khoa Học Máy Tính

Giảng viên hướng dẫn: Thầy Nguyễn Chí Thanh

Sinh viên thực hiện :

✚ Nguyễn Minh Tuệ - 1751010175
✚ Nguyễn Cao Tấn - 1751010131

Lớp: TH73

TP.HCM 2020

Phụ lục 2: Phụ Bìa Đồ Án



**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



WEB GIA SƯ

Đồ Án Ngành

NGÀNH: Khoa Học Máy Tính

 **Nguyễn Minh Tuệ - 1751010175**
 **Nguyễn Cao Tấn - 1751010131**

Giảng viên hướng dẫn: Thầy Nguyễn Chí Thanh

Lớp TH03

TP. HỒ CHÍ MINH, 2020

Phụ Lục 3: Mục Lục

Mục Lục	
DANH MỤC HÌNH VẼ	13
DANH MỤC BẢNG	14
DANH MỤC TỪ VIẾT TẮT	15
TÓM TẮT ĐỒ ÁN	1
MỞ ĐẦU	2
Chương 1.TỔNG QUAN VỀ INTERNET, WORLD WIDE WEB, HTML	4
1.1 Khái niệm cơ bản về Internet	4
1.1.1. Giới thiệu Chung	4
1.1.2. Phân Loại	4
1.2 World Wide Web	5
1.2.1 Các Khái Niệm Cơ bản Về World Wide Web	5
1.2.2 Cách tạo một trang Web	5
1.2.3 Phân Loại Web	6
1.2.4 Web Server	8
1.2.5 Trình duyệt Web (Web Client hoặc Web Browser)	9
1.3 HTML	9
1.3.1 Giới thiệu về HTML	9
1.3.2 Cấu trúc Trang	9
1.3.3 Các thẻ HTML cơ bản	10
Chương 2.TÌM HIỂU VỀ NGÔN NGỮ NODEJS	11
2.1 Khái Niệm về NodeJs	11
2.2 Tại sao chúng ta nên dùng NodeJS	11
2.12 Khởi tạo Server trong NodeJs với HTTP module	29
2.12.1 Giới thiệu về HTTP module	29
2.12.2 Khởi tạo Server với HTTP module	29
2.12.3 Các phương thức và thuộc tính được dùng trong request và response	30
2.13 Đọc ghi file bằng module fs trong Node.js	31
2.13.1 Giới thiệu về module fs	31
2.13.2 Đọc files với module fs	31
2.13.3 Ghi file	32
2.14 Module URL	32
2.14.1 Khái báo sử dụng Module URL	33
2.15 Upload file trong NodeJs	33
2.15.1 Tổng quan	33
2.15.2 cài đặt module formidable	33
2.15.3 code upload	33
2.16 Sự kiện trong NodeJs	34
2.16.1 Event loop	34
2.17 Tạo client request trong NodeJs	35
2.17.1 Cài đặt	35
2.17.2 các phương thức mà module request hỗ trợ:	36
Chương 3.CƠ SỞ DỮ LIỆU MONGODB	38
3.1 Giới thiệu về MongoDB	38
3.1.1 Lợi thế của MongoDB	38
3.1.2 đặc điểm của MongoDB	38

2.3 Sơ đồ hoạt động của NodeJS	12
2.4 Các kiểu dữ liệu	13
2.5 Biến và Hằng	15
2.5.1 Biến	15
2.5.2 Hằng	15
2.6 Các đặc tính của NodeJS	16
2.7 NPM	17
2.8 Toán tử trong NodeJs	19
2.8.1 Toán tử gán	19
2.8.2 Toán tử số học	19
2.8.3 Toán tử so sánh	20
2.8.4 Toán tử logic	21
2.8.5 Toán tử kết hợp	21
2.9 Hàm trong NodeJS	22
2.10 Câu lệnh rẽ nhánh	23
2.10.1 Câu Lệnh If-ELSE	23
2.10.2 Câu lệnh lồng nhau	23
2.10.3 Switch-Case	24
2.11 Đối tượng trong NodeJs (object)	25
2.11.2 Thuộc tính và phương thức trong đối tượng	25
2.11.4 Phương thức	26
2.11.5 Gọi thuộc tính và phương thức	27
2.11.6 Đối tượng trong đối tượng	27
2.11.7 Mảng trong đối tượng	28
3.2 Mục đích sử dụng cơ sở dữ liệu MongoDB	38
3.3 Thuật ngữ trong MongoDB	39
3.4 Kiểu dữ liệu trong MongoDB	40
3.5 Một số câu lệnh truy vấn cơ bản của MongoDB	41
3.6 Truy vấn có điều kiện trong MongoDB	42
3.7 MongoDB Replication	43
3.8 Cài đặt mongodb	43
3.9 Kết nối NodeJs với mongodb	43
3.10 Đóng kết nối NodeJs đến MongoDB	44
3.11 Sử dụng MongoDB	44
Chương 4.PHÂN TÍCH THIẾT KẾ HỆ THỐNG	45
4.1 Tác nhân và use case hệ thống	45
4.1.1 Các tác nhân (Actor)	45
4.2 Các use case	45
4.2.1 Đăng nhập:	45
4.2.2 Đăng ký:	45
4.2.3 Đăng tin tuyển dụng:	45
4.2.4 Tạo hồ sơ gia sư:	45
4.2.5 Nạp tiền vào tài khoản	45
4.2.6 Tra cứu thông tin	46
4.2.7 Hỗ trợ khách hàng	46
4.2.8 Thay đổi quy định	46
4.2.9 Lập báo cáo	46
4.2.10 Use Case	46

4.2.11	Active Diagram.....	46
4.2.12	Sequence Diagram.....	46
4.2.13	Class Diagram.....	46
4.2.14	Mô Hình Database.....	46
PHỤ LỤC		48
TÀI LIỆU THAM KHẢO.....		49

Phụ Lục 4: Trang cảm ơn

LỜI CẢM ƠN

Xin tỏ lòng biết ơn sâu sắc đến

Điều đầu tiên em xin phép được gửi lời cảm ơn đến Thầy Nguyễn Chí Thanh Giảng viên môn học đã nhiệt tình giúp đỡ, đóng góp ý kiến tận tình. Mặc dù thầy rất bận rộn nhưng luôn dành thời gian giúp đỡ cũng như chỉ bảo cho em trong khoảng thời gian làm đồ án.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong Khoa Công Nghệ Thông Tin, văn phòng công tác sinh viên nói chung, Ngành Khoa học máy tính nói riêng. Đã dạy và tạo điều kiện giúp đỡ em có kiến thức vững vàng về các môn đại cương cũng như các môn chuyên ngành để chúng em có thể thực hiện bài đồ án thành công tốt đẹp.

Cuối cùng, em xin chân thành gửi lời cảm ơn đến gia đình và bạn bè, luôn tạo điều kiện thuận lợi, giúp đỡ, quan tâm, động viên em trong suốt quá trình hoàn thành bài đồ án.

.....Ngày....Tháng....Năm.....

Phụ Lục 5: Danh Mục Hình

TT	Hình	Tên hình	Trang
1	Hình 1.1	sơ đồ hoạt động của Web Động.	7
2	Hình 1.2	sơ đồ hoạt động của Web tĩnh.	7
3	Hình 1.3	sơ đồ hoạt động của Web Server.	8
4	Hình 2.1	Sơ đồ hoạt động của NodeJs.	13
5	Hình 2.2	Ảnh minh họa khởi tạo một dự án.	17
6	Hình 2.3	Ảnh minh họa cách khởi tạo Package.	18
7	Hình 4.1	Mô hình Use case	45
8	Hình 4.2	hình quy trình lập báo cáo	61
9	Hình 4.3	Quy trình tra cứu lớp	62
10	Hình 4.4	quy trình đăng tin tuyển dụng	62
11	Hình 4.5	Quy trình tạo hồ sơ gia sư	63
12	Hình 4.6	quy trình đăng ký nhận lớp	63
13	Hình 4.7	quy trình hỗ trợ khách hàng	64
14	Hình 4.8	quy trình thay đổi quy định	64
15	Hình 4.9	mô hình lớp	65
16	Hình 4.10	Giao diện trang chủ	62
17	Hình 4.11	Giao diện đăng nhập	62
18	Hình 4.12	Giao diện đăng ký	63
19	Hình 4.13	Giao diện Tuyển dụng lớp	63
20	Hình 4.14	Giao diện tạo lớp	64
21	Hình 4.15	Giao diện quản trị đăng nhập	64
22	Hình 4.16	Giao diện Quản trị	65
23	Hình 4.17	Giao diện chi tiết người dùng	65
24	Hình 4.18	Giao diện tạo thông tin lớp mới	66

Phụ Lục 6: Danh Mục Bảng

DANH MỤC BẢNG

Bảng 2.1: Bảng toán tử số học.....	19
Bảng 2.2: Bảng toán tử so sánh.	20
Bảng 2.3: bảng toán tử logic.....	21
Bảng 2.4: bảng phương thức module request.	36
Bảng 3.1: bảng thuật ngữ trong MongoDB.....	39
Bảng 3.2: bảng kiểu dữ liệu trong MongoDB	40
Bảng 3.3: bảng cú pháp truy vấn có điều kiện trong MongoDB	42
Bảng 4.1: mô tả đăng nhập.	47
Bảng 4.2: Mô tả đăng ký.	48
Bảng 4.3: mô tả đăng tin tuyển dụng.....	50
Bảng 4.4: bảng mô tả tạo hồ sơ	52
Bảng 4.5: bảng mô tả nạp tiền vào tài khoản	55
Bảng 4.6: Bảng mô tả tra cứu thông tin.....	57
Bảng 4.7: bảng mô tả hỗ trợ khách hàng	58
Bảng 4.8: Mô Tả thay đổi quy định.....	60
Bảng 4.9: mô tả quy trình lập báo cáo.....	61

Phụ Lục 7: Danh Mục Viết Tắt

DANH MỤC TỪ VIẾT TẮT

TT	Từ viết tắt	Ý nghĩa
1	CNTT	Công nghệ thông tin
2	NCKH	Nghiên cứu khoa học
3	DAMH	Đồ án môn học
4	SDT	Số điện thoại
5	CMND	Chứng minh nhân dân
6	CSDL	Cơ sở dữ liệu
7	WWW	World Wide Web
8	Email	ELECTRONIC MAIL
9	HTML	Hypertext Markup Language
10	IIS	Internet Information Services
11	IE	Internet Explorer
12	API	Application Programming Interface
13	HTTP	HyperText Transfer Protocol
14	FS	File system

Phụ Lục 8: Lời mở đầu

MỞ ĐẦU

Hiện nay, việc ứng dụng công nghệ thông tin vào các lĩnh vực của xã hội và việc tin học hóa được xem là một yếu tố quan trọng hàng đầu của các tổ chức, các công ty, nhà nước, nó đóng vai trò cốt yếu trong việc tạo ra những bước đột phá trong công nghệ.

Song song với việc phát triển không ngừng nghỉ về khoa học máy tính, kỹ thuật máy tính và mạng truyền thông, thương mại điện tử đã được chinh phục bởi những công nghệ có đặc tính kỹ thuật tiên tiến, và những công nghệ vượt trội có mặt trên thế giới hiện nay. Ngay tại thời điểm này, internet đã trở thành một sản phẩm có giá trị đặc biệt mà không có đối thủ thay thế và không thể phủ nhận được tầm quan trọng của nó. Internet là nền tảng cốt yếu quan trọng nhất để truyền tải, trao đổi thông tin trên phạm vi toàn cầu.

Ngày nay, chúng ta sẽ nhận thấy mọi việc sẽ dễ dàng hơn khi chúng ta có Internet, mọi việc liên quan đến thông tin tài nguyên sẽ được truyền tải nhanh chóng. Chỉ cần bạn sở hữu một thiết bị thông minh như Laptop hoặc điện thoại có kết nối internet và kèm theo đó là một dòng dữ liệu theo chính nhu cầu người sử dụng muốn, thì lập tức mọi thứ có liên quan đến dữ liệu người dùng sẽ xuất hiện ngay trước mắt chỉ trong vài giây, có đủ các thông tin hình ảnh, âm thanh và các tài nguyên khác mà người sử dụng cần.

Nhờ có Internet mà chúng ta có thể tiện lợi thực hiện được nhiều công việc với tốc độ tối đa và chi phí thấp hơn so với các hoạt động không có internet. Chính vì vậy đã thúc đẩy mạnh sự phát triển thịnh vượng về thương mại điện tử trên khắp thế giới, giúp tăng nhanh năng suất chất lượng cuộc sống con người.

Giờ đây, thương mại điện tử đã khẳng định vị thế và vai trò của mình trong hoạt động sản xuất, kinh doanh thúc đẩy sự phát triển trong lĩnh vực công nghệ. Đối với doanh nghiệp tư nhân và nhà nước, lớn nhỏ, việc quảng bá marketing và giới thiệu sản phẩm đến khách hàng là một vấn đề cần thiết cho doanh nghiệp. vậy việc quảng

bá cho doanh nghiệp trước tiên là xây dựng một website để quảng bá tất cả các sản phẩm của doanh nghiệp.

Theo nhu cầu thiết yếu của thị trường cũng như học tập và phát triển hơn về công nghệ này, nhóm em đã quyết định thực hiện đồ án với chủ đề là “xây dựng website gia sư” cho người đang có nhu cầu tuyển gia sư.

Người sử dụng khi truy cập website sẽ được chia làm 2 dạng, khách hàng thuê gia sư và gia sư. Khách hàng truy cập vào website để tìm thông tin gia sư phù hợp. gia sư truy cập website để tìm khách hàng phù hợp. tất cả các thông tin gia sư sẽ được chia sẻ cho khách hàng khi cần.

Với sự chỉ bảo và hướng dẫn tận tình của thầy Nguyễn Chí Thanh em đã thực hiện hoàn thành bài báo cáo đồ án này một cách hoàn thiện. chúng em đã cố gắng rất nhiều để tìm hiểu, phân tích hệ thống và thiết kế nhưng không thể không tránh khỏi những thiếu sót và lỗi sai. Chúng em rất mong quý thầy quý cô cho chúng em lời góp ý qua đó em có thể sửa những lỗi sai và hoàn thành bài báo cáo tốt hơn nữa. chúng em xin chân thành cảm ơn quý thầy cô.

Sinh viên thực hiện: Nguyễn Cao Tấn

Nguyễn Minh Tuệ

Lớp: TH73

.....Ngày.....Tháng.....Năm

Phụ Lục 9: Tài liệu Tham Khảo

TÀI LIỆU THAM KHẢO

- [1]. [ThanhTai.Vu](http://thanhται.vu) (2016). Học Lập Trình Nodejs Cơ Bản[Online], viewed 19 October 2020, form :< <https://toidicode.com/nodejs-co-ban>>.
- [2]. Viblo Team(2014). Tự học nodejs từ đầu[Online], viewed 21 October 2020, Form :< <https://viblo.asia/p/tu-hoc-nodejs-tu-dau-6J3ZgnGPKmB>>.
- [3]. NguyenCongHa (2016). Tổng quan về Internet[Online], view 15 October 2020, Form :< <https://hoc247.net/tin-hoc-dai-cuong/bai-1-tong-quan-ve-internet-19055.html>> .
- [4]. McPherson, Stephanie Sammartino(ed) (2009). Tim Berners-Lee: Inventor of the World Wide Web. Twenty-First Century Books . ISBN 978-0-8225-7273-2. viewed 28 October 2020, Form :<https://en.wikipedia.org/wiki/World_Wide_Web>
- [5]. [MatBao](http://matbao.net)(2002). MongoDB: Tính năng nổi bật từ MongoDB bạn cần biết[Online], viewed 1 November 2020, Form:< <https://wiki.matbao.net/mongodb-la-gi-tinh-nang-noi-bat-tu-mongodb-ban-can-biet/>>.

