|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\***    ĐỒ ÁN MÔN HỌC  ĐỀ TÀI | |
| TÌM HIỂU LARAVEL FRAMEWORK  XÂY DỰNG WEBSITE CỔNG THÔNG TIN KHOA CNTT | |
|  | |
|  | |

Sinh Viên thực hiện: **ĐOÀN VĂN THƯƠNG**

MSSV**: 1451010189**

Lớp**:DH14TH02**

GV hướng dẫn: ThS. **Nguyễn Thị Mai Trang**

*Thành phố Hồ Chí Minh, Tháng 10 – Năm 2017*

Lời cảm ơn

Trong suốt quá trình học tập tại *Tr*ườ*ng* Đạ*i H*ọ*c Mở Thành Phố Hồ Chí Minh*, chúng em đã được các thầy cô giảng dạy, giúp đỡ và được truyền đạt nhiều kiến thức vô cùng quý giá. Ngoài ra, chúng em còn được rèn luyện bản thân trong một môi trường học tập đầy sáng tạo và khoa học. Đây là một quá trình hết sức quan trọng giúp em có thể thành công khi bắt tay vào nghề nghiệp tương lai sau này.

Để hoàn thành đồ án chúng em đã nhận được nhiều sự giúp đỡ quý báu của quý thầy cô tại Trường Đại học Mở TP Hồ Chí Minh Khoa Công Nghệ Thông Tin.

Chúng em xin chân thành cám ơn các thầy cô trong Khoa Công Nghệ Thông Tin đã tạo điều kiện cho chúng em thực hiện đề tài đồ án môn học này. Các thầy cô đã tận tình cung cấp, giảng dạy cho chúng em những kiến thức cơ bản, để từ đó có thể triển khai vào đề tài.

Đặc biệt chúng em xin gửi lời cảm ơn sâu sắc tới cô Nguyễn Thị Mai Trang đã nhiệt tình hướng dẫn chúng em hoàn thành đồ án môn học. Mặc dù em đã cố gắng trong suốt quá trình làm đồ án , nhưng do kinh nghiệm thực tế và trình độ chuyên môn chưa được nhiều nên em không tránh khỏi những thiếu sót, em rất mong được sự chỉ bảo, góp ý chân thành từ các thầy, cô cùng tất cả các bạn.

Em xin chân thành cám ơn.

TP. Hồ Chí Minh, 20/10/2017

Sinh viên thực hiện

Đoàn Văn Thương

Lớp DH14TH02 – Trường ĐH Mở TPHCM

Nhận xét của giáo viên hướng dẫn

**MỤC LỤC**

[Lời cảm ơn 1](#_Toc496557924)

[Nhận xét của giáo viên hướng dẫn 1](#_Toc496557925)

[CHƯƠNG I: TỔNG QUAN 3](#_Toc496557926)

[1. Đặt vấn đề 3](#_Toc496557927)

[2. Giới thiệu đề tài 3](#_Toc496557928)

[3. Nội dung đồ án 4](#_Toc496557929)

[**4.** **Mục lục hình ảnh** 5](#_Toc496557930)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 6](#_Toc496557931)

[1. Cổng thông tin điện tử 6](#_Toc496557932)

[1.1. Khái niệm cổng thông tin điện tử 6](#_Toc496557933)

[1.2. Ưu điểm các loại cổng thông tin điện tử 7](#_Toc496557934)

[2. Xây dựng website cổng thông tin khoa công nghệ thông tin: 9](#_Toc496557935)

[2.1. HTML 9](#_Toc496557936)

[2.2. HTML5 10](#_Toc496557937)

[2.3. CSS 12](#_Toc496557938)

[2.4. CSS3 13](#_Toc496557939)

[2.5. CSS Preprocessor 14](#_Toc496557940)

[2.6. Reponsive 18](#_Toc496557941)

[2.7. Bootstrap 18](#_Toc496557942)

[2.8. Jquery 20](#_Toc496557943)

[2.9. Google Map API 21](#_Toc496557944)

[CHƯƠNG III: LARAVEL FRAMEWORK 22](#_Toc496557945)

[1. Giới Thiệu Về PHP Laravel Framework 22](#_Toc496557946)

[1.1 Laravel Framework là gì: 22](#_Toc496557947)

[1.2 Lí do lựa chọn Laravel Framework 23](#_Toc496557948)

[1.3 Cài Đặt Laravel Framework 24](#_Toc496557949)

[1.4 Cấu trúc thư mục Laravel framework 25](#_Toc496557950)

[2. Các thành phần cơ bản của Laravel Framework 27](#_Toc496557951)

[2.1 Mô hình MVC 27](#_Toc496557952)

[2.2 Route 28](#_Toc496557953)

[2.3 Controller 30](#_Toc496557954)

[2.4 View 31](#_Toc496557955)

[2.5 Blade template engine 32](#_Toc496557956)

[2.6 Query Builder và Eloquent ORM 33](#_Toc496557957)

[2.7 Migrations 37](#_Toc496557958)

[2.8 Authentication 39](#_Toc496557959)

[2.9 Middleware 42](#_Toc496557960)

[3. Những lợi ích của Laravel Framework 43](#_Toc496557961)

[CHƯƠNG IV: PHÂN TÍCH, THIẾT KẾ 43](#_Toc496557962)

[1. Sơ đồ các quy trình của website cổng thông tin điện tử 43](#_Toc496557963)

[1.1. phía người sinh viên 43](#_Toc496557964)

[1.2. Phía người quản trị website 44](#_Toc496557965)

[2. Sơ đồ website 45](#_Toc496557966)

[3. Cơ sở dữ liệu 45](#_Toc496557967)

[3.1. Các Thực Thể 45](#_Toc496557968)

[3.2. Sơ Đồ CDM 46](#_Toc496557969)

[3.3. Sơ Đồ PDM 47](#_Toc496557970)

[3.4. Từ điển dữ liệu 47](#_Toc496557971)

[CHƯƠNG IV: HIỆN THỰC 49](#_Toc496557972)

[1. Giới thiệu website cổng thông tin khoa công nghệ thông tin 49](#_Toc496557973)

[2.1. Giao diện người dùng 49](#_Toc496557974)

[Giao diện quản trị 57](#_Toc496557975)

[CHƯƠNG V: TỔNG KẾT 65](#_Toc496557976)

[TÀI LIỆU THAM KHẢO 1](#_Toc496557977)

CHƯƠNG I: TỔNG QUAN

1. Đặt vấn đề

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức cũng như công ty , trường học. Đối với các trường Đại Học việc xây dựng một website cổng thông tin để phục vụ cho nhu cầu của giảng viên, sinh viên, nghiên cứu sinh, người học… là hết sức quan trọng và ngày càng trở nên cấp thiết. Lợi ích của cổng thông tin điện tử là ra đời để giải quyết những khó khăn mà website truyền thống đang gặp phải. Nhờ tính năng quản trị nội dung, tích hợp và liên kết nhiều nguồn thông tin, giải pháp cổng thông tin điện tử bên cạnh phục vụ hoạt động giới thiệu, quảng bá thì lợi ích chính là tạo ra một kênh thông tin phong phú, có khả năng phổ cập thông tin rộng rãi, nhanh chóng; tạo một kênh trao đổi, tương tác thông tin.Thông qua những website này, thông tin về sinh viên, giảng viên cũng như các tin tức, thông báo, tài liệu, tuyển dụng, thực tập…sẽ được cập nhật đến với những người quan tâm một cách nhanh chóng và chính xác.

Bắt nguồn từ ý tưởng này, cùng với những gợi ý của cô Nguyễn Thị Mai Trang em đã thực hiện đồ án “Tìm hiểu Laravel Framework xây dựng website cổng thông tin khoa Công Nghệ Thông Tin” như nội dung em sẽ trình bày sau đây. Em chân thành cảm ơn cô Nguyễn Thị Mai Trang đã tận tình góp ý và hướng dẫn để em có thể hoàn thành đồ án này.

1. Giới thiệu đề tài

Cổng thông tin điện tử chuyên nghiệp là một môi trường giao diện Website tổng hợp, thống nhất có phép mọi người truy cập đơn giản tìm kiếm thông tin, dịch vụ, dữ liệu và các chương trình ứng dụng của người dùng được bảo mật cao. Cổng thông tin điện tử chuyên nghiệp có thể thu  thập, tích hợp nguồn thông tin, dịch vụ theo đặc thù của mỗi người, giúp các khối đơn vị tìm được cách thứ giao tiếp riêng tư và cao cấp. đảm bảo cung cấp đầy đủ thông tin, dịch vụ, giao dịch cần thiết và đúng lúc cho việc ra quyết định, cung cấp thông tin, Cổng thông tin điện tử là một cổng thông tin cung cấp đầy đủ thông tin, đúng thời điểm  cho những người cung cấp thông tin, đó là hiệu quả của Cổng thông tin điện tử

Trong đề tài này em xin phép giới thiệu khái quát khoa công nghệ thông tin trường đại học mở thành phố hồ chí minh. sử dụng PHP framework tin dùng nhất hiện nay là **Laravel** phiên bản 5.xxx với kỹ quy trình **Frontend & Backend** giống làm dự án ngoài thực tế hiện nay.

Đồng thời thiết kế chương trình cho phép người sử dụng có thể xem,tìm kiếm cái bài viết mới nhất,hiển thị bản đồ các cơ sở học ,xem chương trình đào tạo,download các tài liệu cũng như truy cập vào quản lý thông tin , bình luận ở các bài viết….

Website có thế hiển thị trên hầu hết các trình duyệt và hoạt động trên mọi thiết bị từ di động,smart phone đến tablet, desktop .Ngoài ra còn cung cấp môi trường quản trị website, phân quyền, Cung cấp người dùng một quy trình tìm kiếm có hệ thống, quá trình lựa chọn, tổ chức và chắt lọc thông tin và biểu diễn thông tin hiệu quả từ đó quản trị website đơn giản, nhanh chóng, không sợ phức tạp.

1. Nội dung đồ án

Đề tài sẽ khái quát chung về cổng thông tin điện tử từ đó đi sâu vào tìm hiểu những lợi ích và hạn chế khi áp cổng thông tin điện tử vào các hoạt động quản lý học tập như hiện nay

Bên cạnh đó, đề tài sẽ xây dựng website cổng thông tin khoa công nghệ thông tin trực thuộc trường đại học nhiều khoa sử dụng PHP framework Larvel theo mô hình MVC sử dụng ngôn ngữ PHP&MySQL và các ngôn ngữ frontend.

Các chương trình và ngôn ngữ hổ trợ:

**Tool:**

* Source editor:Sublimetext 3
* XAMPP kết hợp sẳn Apache , MySQL và PHP
* CSS Preprocessor : SASS ( phần mềm maping SASS-CSS: koala)

**Frontend**

* HTML/HTML5
* CSS/CSS3
* Javascript/Jquery v3.1/Ajax
* Bootstrap3
* Google Map API

**Backend**

* Laravel Framework v5.4

1. **Mục lục hình ảnh**

[Hình 1:Cổng thông tin điện tử 7](#_Toc496559678)

[Hình 2:Reponsive trên 3 loại thiết bị 18](#_Toc496559679)

[Hình 3:Hiển thị các cơ sở học của trường bằng google map api 22](#_Toc496559680)

[Hình 4: các PHP Framework phổ biến 23](#_Toc496559681)

[Hình 5:Mô hình MVC 27](#_Toc496559682)

[Hình 6:Luồng đi trong mô hình MVC 28](#_Toc496559683)

[Hình 7: Quy trình hoạt động của Route 29](#_Toc496559684)

[Hình 8:Thao tác với database trong Laravel 33](#_Toc496559685)

[Hình 9:Quy trình hoạt động của sinh viên trên website cổng thông tin 44](#_Toc496559686)

[Hình 10:Hoạt động của người quản trị website 45](#_Toc496559687)

[Hình 11:sơ đồ website cổng thông tin 45](#_Toc496559688)

[Hình 12: Sơ đồ CDM 46](#_Toc496559689)

[Hình 13: Sơ đồ PDM 47](#_Toc496559690)

[Hình 14:Giao diện trang chủ 50](#_Toc496559691)

[Hình 15:Giao diện trang giới thiệu 50](#_Toc496559692)

[Hình 16:Giao diện trang tin tức 51](#_Toc496559693)

[Hình 17:Tranh chi tiết tin 51](#_Toc496559694)

[Hình 18:Trang đào tạo 52](#_Toc496559695)

[Hình 19:Trang sách 52](#_Toc496559696)

[Hình 20:Trang tuyển dụng 53](#_Toc496559697)

[Hình 21:Trang liên hệ 53](#_Toc496559698)

[Hình 22:Trang chi tiết tuyển dụng 54](#_Toc496559699)

[Hình 23:Trang đăng nhập 54](#_Toc496559700)

[Hình 24:Trang thông tin sinh viên 55](#_Toc496559701)

[Hình 25:Trang lỗi 404 55](#_Toc496559702)

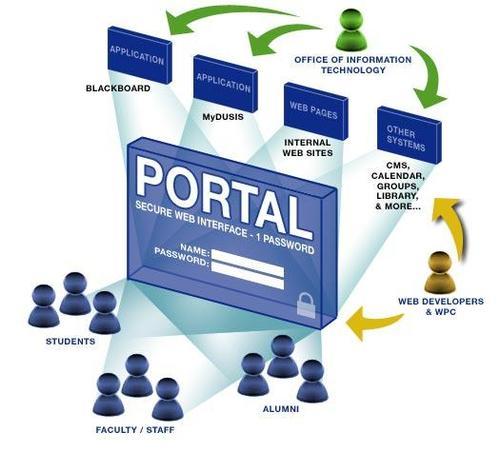
[Hình 26:Trang quản trị 56](#_Toc496559703)

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. Cổng thông tin điện tử
   1. Khái niệm cổng thông tin điện tử

Cổng thông tin điện tử chuyên nghiệp là một môi trường giao diện Website tổng hợp, thống nhất có phép mọi người truy cập đơn giản tìm kiếm thông tin, dịch vụ, dữ liệu và các chương trình ứng dụng của người dùng được bảo mật cao. Cổng thông tin điện tử chuyên nghiệp có thể thu  thập, tích hợp nguông thông tin, dịch vụ theo đặc thù của mỗi người, giúp các khối đơn vị tìm được cách thứ giao tiếp riêng tư và cao cấp..

Cổng thông tin điện tử chuyên nghiệp còn được gọi là Portal hay Cổng giao tiếp điện tử hoặc Cổng giao dịch điện tử. Cả 3 tên gọi này có thể hiểu là Portal – Cổng thông tin điện tử chuyên nghiệp, chúng thể hiện 3 cấp độ phát triển khách nhau từ thấp đến cao. Tùy theo nhu cầu thiết kế Web Cổng thông tin của khối đơn vị, lãnh đạo quyết định tên gọi phù hợp..Cổng thông tin điện tử thường được phân thành 3 loại dựa vào những tính năng đặc trưng của chúng có những ưu nhược điểm hoàn toàn khác nhau



Hình 1*:Cổng thông tin điện tử*

* 1. Ưu điểm các loại cổng thông tin điện tử

Ưu điểm của cổng thông tin điện tử cao cấp

* là kênh thông tin hiệu quả: Nhờ tính năng quản trị nội dung, tích hợp và liên kết nhiều nguồn thông tin, giải pháp cổng thông tin điện tử bên cạnh phục vụ hoạt động giới thiệu, quảng bá thì lợi ích chính là tạo ra một kênh thông tin phong phú, có khả năng phổ cập thông tin rộng rãi, nhanh chóng; tạo một kênh trao đổi, tương tác thông tin.  
  -  Nhờ việc tạo lập một điểm quy tụ thông tin nhất quán, cho phép tích hợp các phần mềm để xử lý công việc chuyên môn và hỗ trợ tác nghiệp giữa các cơ quan đơn vị, cổng thông tin điện tử tạo ra một môi trường làm việc cộng tác đơn giản và hiệu quả:  
      +  Lãnh đạo có thể nhanh chóng truyền đạt các thông điệp, các chủ trương chính sách tới nhiều đối tượng. Ngược lại, có thể dễ dàng nắm bắt diễn biến, tiến độ thực hiện công việc, các đề xuất kiến nghị của các đơn vị.  
        +  Các nhà quản lý có thể thực hiện đánh giá, giám sát các hoạt động thông qua lập kế hoạch, lịch làm việc, chia sẻ tài liệu, lập và gửi báo cáo đồng thời tới các đối tượng.  
        +  Các cán bộ chuyên môn thuộc các cơ quan khác nhau có thể cùng phối hợp xử lý các nghiệp vụ liên ngành thông qua các phần mềm được tích hợp sẵn trên cổng.  
        +  CBCNV cùng thực hiện các thủ tục liên quan công việc, quy trình quy định của cơ quan qua các dịch vụ trực tuyến.  
        +  Hệ thống cho phép quản lý nhiều website (ví dụ website con của các đơn vị trực thuộc) trên một nền tảng chung.  
  -  Đăng nhập một lần (Single Sign On): tạo sự đồng nhất, thuận tiện và bảo mật khi sử dụng.  
  -  Việc tuân thủ các tiêu chuẩn chung và thống nhất sẽ làm cho các ứng dụng có thể giao tiếp và trao đổi thông tin lẫn nhau; Cung cấp nền tảng giúp duy trì, quản lý, mở rộng, tích hợp ứng dụng, nâng cấp dễ dàng, tiết kiệm chi phí đầu tư để xây dựng lại hệ thống.  
  -  Cung cấp cho người dùng khả năng truy cập thông tin hiệu quả hơn nhờ truy cập bảo mật, khả năng tương tác thông tin nhiều chiều, cũng như truy cập tổng thể tới tất cả các cá nhân, thông tin, tổ chức.  
  -  Hỗ trợ xây dựng chính sách quản lý, phân phối, chia sẻ thông tin trên web, giúp kiểm soát tình trạng, phiên bản của tài nguyên.  
  -  Cung cấp người dùng một qui trình tìm kiếm có hệ thống, quá trình lựa chọn, tổ chức và chắt lọc thông tin và biểu diễn thông tin hiệu quả.

**Ưu điểm của cổng thông tin điện tử chuyên nghiệp:**

Giá thành rẻ hơn giải pháp cao cấp.

- Là kênh thông tin hiệu quả cho các đơn vị, có thể phổ cập được thông tin một cách rộng rãi, nhanh chóng.

- Người dùng có thể tạo sự đồng nhất và thuận tiện khi sử dụng nhờ tính năng chỉ cần đăng nhập một lần vào hệ thống cổng thông tin.

- Việc tuân thủ các tiêu chuẩn chung và thống nhất sẽ làm cho các ứng dụng có thể giao tiếp và trao đổi thông tin lẫn nhau; Cung cấp nền tảng giúp duy trì, quản lý, mở rộng, tích hợp ứng dụng, nâng cấp dễ dàng, tiết kiệm chi phí đầu tư để xây dựng lại hệ thống.

- Cung cấp cho người dùng khả năng truy cập thông tin hiệu quả hơn nhờ truy cập bảo mật, khả năng tương tác thông tin nhiều chiều, cũng như truy cập tổng thể tới tất cả các cá nhân, thông tin, tổ chức.

- Hỗ trợ xây dựng chính sách quản lý, phân phối, chia sẻ thông tin trên web, giúp kiểm soát tình trạng, phiên bản của tài nguyên.

- Cung cấp người dùng một quy trình tìm kiếm có hệ thống, quá trình lựa chọn, tổ chức và chắt lọc thông tin và biểu diễn thông tin hiệu quả.

**Ưu điểm của cổng thông tin điện tử phổ thông**

* Giá thành rẻ hơn hai giải pháp cao cấp và chuyên nghiệp ở trên.
* Là kênh thông tin hiệu quả cho các đơn vị, có thể phổ cập được thông tin một cách rộng rãi, nhanh chóng.
* Người dùng có thể dễ dàng quản lý cổng thông tin mà không sợ nhầm lẫn với các website trực thuộc
* Quản trị website đơn giản, nhanh chóng, không sợ phức tạp.
* Hỗ trợ xây dựng chính sách quản lý, phân phối, chia sẻ thông tin trên web, giúp kiểm soát tình trạng, phiên bản của tài nguyên.
* Cung cấp người dùng một quy trình tìm kiếm có hệ thống, quá trình lựa chọn, tổ chức và chắt lọc thông tin và biểu diễn thông tin hiệu quả.

1. Xây dựng website cổng thông tin khoa công nghệ thông tin:
   1. HTML
2. Giới thiệu

Ngôn ngữ HTML (*Hyper Text Markup Language* – Ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ được thiết kế nhằm mục đích chính là tạo ra và hiển thị thông tin của các trang web. Đến nay, HTML đã trở thành một chuẩn Internet và được tổ chức W3C (*World Wide Web Consortium*) duy trì.

Một trang web được viết bằng ngôn ngữ HTML bao gồm nhiều lệnh, mỗi lệnh được gọi là một thẻ (tag). Mỗi thẻ quy định một cách thức hiển thị riêng của dữ liệu trong trang web đó, dễ nhận thấy nhất là các thẻ chữ in đậm, in nghiêng, màu chữ, v.v… Các thẻ nói trên sẽ định dạng hình thức hiển thị của dữ liệu, do đó người dùng chỉ nhìn thấy các dữ liệu đã được định dạng mà không thấy các thẻ do người thiết kế tạo ra. Toàn bộ nội dung dữ liệu và các thẻ này sẽ được lưu trong một file text với tên mở rộng thường gặp là .html hoặc .htm, các file text này là các trang web được hiển thị trên các trình duyệt (như Google Chrome, Internet Explorer, Mozilla Firefox…).

Cấu trúc một trang web:

* Một trang web thường được mở đầu và kết thúc bởi cặp thẻ <html></html>.
* Cặp thẻ <head></head> chứa các thông tin quản lý hoạt động bên trong trang web. Bên trong thẻ head là thẻ title, chứa tiêu đề trang web.
* Cặp thẻ <body></body> chứa những dữ liệu sẽ hiển thị trong trang web.

Soạn thảo một trang web:

Có nhiều phương pháp để soạn thảo một trang web, trong đó cách đơn giản nhất là dùng chương trình soạn thảo văn bản để gõ. Các chương trình soạn thảo thông dụng có thể kể đến là Notepad, Wordpad và MS Word. Trong đó Notepad được ưu tiên sử dụng do tính đơn giản, tiện dụng và dễ thao tác.Ngoài ra còn phải kể đến sublimeText 3, tool viết HTML mạnh mẽ nhất hiện nay.(Trong đề tài này em sử dụng SublimeText3).

* 1. HTML5

1. Giới thiệu

HTML5 là ngôn ngữ đánh dấu dùng cho cấu trúc và trình bày nội dung của trang web. HTML5 là phiên bản thứ năm của ngôn ngữ HTML tiếp theo sau thế hệ HTML4. HTML5 đang được cộng tác nghiên cứu và phát triển bởi tổ chức W3C (*World Wide Web Consortium*) và WHATWG (*Web Hypertext Application Technology Working Group*) hứa hẹn sẽ là công nghệ cốt lõi của Internet trong tương lai không xa.

HTML5 được thiết kế dựa trên mục tiêu chính là cải thiện khả năng hỗ trợ đa phương tiện mới nhất đồng thời vẫn thể hiện nội dung website rõ ràng, dễ hiểu với người dùng cũng như với các thiết bị điện tử khác. Do đó, HTML5 là một ứng viên tiềm năng cho nền tảng ứng dụng trên thiết bị di động.

HTML5 vẫn chưa là chuẩn chính thức vì vậy chưa có trình duyệt nào hỗ trợ HTML5 đầy đủ. Tuy nhiên các trình duyệt phổ biến như Firefox, Chrome, Opera, Safari,… vẫn tiếp tục cập nhật thêm nhiều chức năng của HTML5 trong các phiên bản mới nhất.

Các ưu điểm:

* Điều khiển bắt lỗi tốt hơn.
* Nhiều thẻ mới hơn để thay thế các ngôn ngữ kịch bản.
* Độc lập hơn nhờ giản lược sự cần thiết của các plug-in bên ngoài.
* Các thẻ video và audio hỗ trợ media.
* Hỗ trợ lưu trữ offline tốt hơn.

DOCTYPE:

DOCTYPE là một khai báo kiểu tài liệu giúp các trình duyệt web xác định nội dung của HTML đó thuộc phiên bản nào. Trong HTML5, việc khai báo DOCTYPE trở nên đơn giản hơn với một dòng câu lệnh đơn giản (ví dụ: <!DOCTYPE html>).

1. Cấu trúc HTML5: Trang web thông thường có nội dung (body), cùng các header, footer và các thẻ hỗ trợ các yếu tố trang của HTML5:

* <section> xác định phần nội dung của trang.
* <header> định nghĩa tiêu đề của một trang.
* <footer> xác định chân của một trang.
* <nav> dùng cho vùng điều hướng (menu, bar,… ).
* <article> định nghĩa nội dung chính trên một trang.
* <aside> định nghĩa thêm nội dung bên trong một nôi dung khác.
* <figure> định nghĩa hình ảnh chú thích cho một bài viết.

Ngoài ra còn có các thẻ <form> mới và các thuộc tính đầu vào như:

* <datalist> Danh sách lựa chọn cho giá trị đầu vào (tạo danh sách với thẻ option nằm ngoài datalist).
* <keygen> Tạo khóa (cung cấp cách thức bảo mật) để chứng thực người dùng. Thẻ keygen là một cặp sinh mã. Khi một form được submit, có 2 khóa được tạo, một khóa dạng private và một khóa dạng public.
* <output> Định nghĩa các dạng đầu ra khác nhau (ví dụ: các phép tính hoặc dạng mã lệnh).
* tel : Giá trị số điện thoại.
* search : Trường tìm kiếm.
* url : Giá trị đầu vào là một đường dẫn.
* email: Giá trị đầu vào là một hay nhiều emails.
* datetime : Giá trị đầu vào là ngày, tháng.
* month: Giá trị đầu vào là tháng.
* week : Giá trị đầu vào là tuần.
* time : Giá trị đầu vào là giờ
* number : Giá trị đầu vào là số
* range : Giá trị đầu vào là dãy giá trị số.
* color : Giá trị đầu vào là dạng màu hexa như #FFFFFF.
* …

Các thẻ dùng cho Media

* <audio> Dùng cho nội dung multimedia như sound, music hay các dạng audio khác.
* <video> Dùng cho nội dung video như clip hoặc các dạng video khác.
* <source> Dùng cho các nguồn media trong các thẻ media, định nghĩa bên trong video hoặc audio.
* <embed> Nhúng nội dung như các plugin.
  1. CSS

1. Giới thiệu

CSS (*Cascading Style Sheet* – các tập tin dạng tầng) được dùng để định dạng cách trình bày của các đối tượng trong trang web. CSS cho phép trình bày các thuộc tính mở rộng của đối tượng mà các thẻ trong file HTML (hoặc XHTML) thông thường không có.

Các ưu điểm:

* Rút ngắn thời gian định dạng trang web.
* Hạn chế việc làm rối mã HTML của trang web bằng các thẻ quy định kiểu dáng (ví dụ: chữ đậm, chữa màu,… ).
* Dễ dàng cập nhật thuộc tính trang web.
* Tạo ra sự nhất quán về cách trình bày các trang web.
* Tách biệt công việc trình bày trang web với thiết kế cấu trúc trang web, thích hợp với các dự án làm việc nhiều người.

1. Các kiểu style:

* Style định dạng cho một thẻ trong toàn trang (inline style).  
  Ví dụ: style a { font-size:16 } dùng để định dạng tất cả các thẻ a trong trang web có cỡ chữ là 16.
* Style định dạng cho một đối tượng cụ thể có tên.

Ví dụ: style #title { color: magenta } để định dạng đối tượng title có màu chữ là màu magenta.

* Style định dạng cho một thẻ dùng cho một vùng nào đó trong trang web.  
  Ví dụ: style #title a { font-size:18 } để định dạng tất cả các thẻ a trong đối tượng title có cỡ chữ là 18.
* Class: một class trong CSS là tập hợp các định dạng được khai báo (ví dụ: màu nền, màu chữ, khung viền,… ) để hiển thị cho một hay nhiều đối tượng được chỉ định.

Ví dụ: .title { font-size:16; font-weight:bold }

* 1. CSS3

1. Giới thiệu

Một trang web HTML hoàn chỉnh không thể thiếu CSS. Do đó, khi HTML5 được ra đời, CSS cũng được nâng cấp lên một phiên bản mới là CSS3 với nhiều cải tiến đáng kể. Một trong những thuộc tính mới trong CSS3 là multi-column với khả năng hiển thị nội dung dưới dạng cột đơn giản hơn bao giờ hết. Multi-column hoạt động như chức năng chia cột trong MS Word.

1. Các ứng dụng CSS3

Border Radius: thuộc tính tạo góc cạnh bo tròn bằng cách khai báo border-radius trong tập tin CSS.

Transform:

* Rote: xoay vòng đối tượng.
* Scale: phóng to, thu nhỏ đối tượng.
* Skew: độ nghiêng (trái, phải) của đối tượng.
* Translate: di chuyển đối tượng theo trục x.

Shadow: tạo hiệu ứng bóng đổ cho đối tượng:

* Horizontal Length: bóng đổ theo chiều ngang.
* Vertical Length: bóng đổ theo chiều dọc.
* Inset: bóng đổ vào bên trong.

Text Shadow: tạo bóng đổ cho text:

* Horizontal Length: bóng đổ theo chiều ngang.
* Vertical Length: bóng đổ theo chiều dọc.
* Blur Radius: độ nhòe.

Text Rotation: xoay đối tượng text.

Gradient: tạo hiệu ứng chuyển sắc:

* Type: radial (trong, ngoài), linear (trái, phải).
* Start Color: màu bắt đầu.
* Start H Position: bắt đầu theo chiều ngang.
* Start V Position: bắt đầu theo chiều dọc.
* Angle: hướng chuyển màu.
* End Polor: màu kết thúc.
* End H Position: kết thúc theo chiều ngang.
* End V Position: kết thúc theo chiều dọc.
  1. CSS Preprocessor

1. Giới thiệu

CSS Preprocessor được hiểu là ngôn ngữ tiền xử lý CSS, hay hiểu nôm na nó là phiên bản mở rộng của ngôn ngữ CSS. Nó có nhiệm vụ giúp bạn logic hóa và cấu trúc các đoạn mã CSS để cho CSS nó đến gần hơn với một ngôn ngữ lập trình. Ngoài ra, nó còn có một số lợi ích sau:

* Tiết kiệm thời gian viết CSS
* Dễ dàng bảo trì và phát triển
* Tính linh hoạt và tái sử dụng
* Các tập tin CSS được tổ chức một cách rõ ràng

SASS (**Syntactically Awesome StyleSheets**) là một phần mở rộng của CSS, nó giúp chúng ta sử dụng biến (variables), quy tắc xếp chồng (nested rules), mixins, thừa kế (selector inheritance), hàm (functions), ... và hoàn toàn tương thích với cú pháp của CSS.Các phần mềm biên dịch SASS nổi tiếng hiện nay như Compass, Koala, Prepros ( trong đồ án này em chọn Koala).

Ví dụ về cách sử dụng biến trong SASS:

$font-famiy: Helvetica**,** sans-serif;

$primary-color: #333;

body {

**font-family**:$font- famiy;

**color**: $primary-color;

}

Ví dụ về cách sử viết theo quy tắc xếp chồng (nested)

nav {

ul {

**margin**: 0;

**padding**: 0;

**list-style**: none;

}

li { **display**: inline-block; }

a {

**display**: block;

**padding**: 6px 12px;

**text-decoration**: none;

}

}

Sau khi biên dịch

nav ul {

margin: 0;

padding: 0;

list-style: none;

}

nav li {

display: inline-block; }

nav a {

display: block;

padding: 6px 12px;

text-decoration: none;

}

1. Các đặc tính của SASS

* Hoàn toàn tương thích với CSS
* Mở rộng ngôn ngữ như các biến (variable), mixins, hàm (function), ...
* Nhiều function hữu ích cho các thao tác với màu sắc và các giá trị khác
* Các đặc tính nâng cao như các control directive
* Có cấu trúc, tùy biến đầu ra

SASS có hai định dạng file là \*.sass và \*.scss. Và cách viết của hai định dạng này cũng là khác nhau (nhưng các control directive, function thì có cùng một ý nghĩa). Điểm qua một số sự khác biệt trong cách viết của hai định dạng file:

\*.sass:

* + Sử dụng indent để thể hiện quy tắc xếp chồng (nested rules)
  + Không cần sử dụng ; khi kết thúc một property
  + Khai báo mixins bằng ký tự =
  + Sử dụng mixins bằng ký tự +

\*.scss:

* + Sử dụng dấu { và } để thể hiện quy tắc xếp chồng (nested rules)
  + Sử dụng ; để kết thúc một property
  + Khai báo mixins bằng directive @mixin
  + Sử dụng mixins bằng directive @include
  1. Reponsive

Reponsive là kỹ thuật biến đổi trang web sao cho chúng có thể hiển thị tốt trên mọi thiết bị bằng cách sử dụng Css hoặc Bootstap:



Hình 2*:Reponsive trên 3 loại thiết bị*

cú pháp sự dụng như sau :

**Màn hình tablet nằm ngang (Landscape Tablet)**

@media only screen and (min-width: 960px) and (max-width: 1240px) {

//code here

}

**Màn hình tablet đứng (Portrait Tablet)**

@media only screen and (min-width: 768px) and (max-width: 959px) {

//code here

}

**Màn hình điện thoại nằm ngang (Landscape Mobile)**

@media only screen and (min-width: 480px) and (max-width: 767px) {

//code here

}

**Màn hình điện thoại đứng (Portrait Mobile)**

@media only screen and (min-width: 100px) and (max-width: 479px) {

//code here

}

* 1. Bootstrap

Bootstrap là một framework HTML, CSS, và JavaScript cho phép thiết kế phát triển trên webmobile.

Bootstrap là một **Front-end Framework** được viết bằng SASS và biên dịch thành CSS, nó là một bộ thư viện mạnh mẽ được tích hợp nhiều CSS có sẵn giúp cho việc lập trình HTML & CSS trở nên đơn giản và dễ dàng hơn. Thông thường khi chúng ta lập trình bằng CSS thuần thì công việc kiểm tra tính tương thích trên các trình duyệt khác nhau và trên các thiết bị di động khác nhau  rất là khó khăn, nhưng với Bootstrap thì lập trình viên không cần phải tốn nhiều công sức nữa vì mọi thứ đã có người tạo nên Bootstrap lo. Ngoài ra Bootstrap tương thích với tất cả các trình duyệt hiện đại hiện nay như (Chrome, Firefox, Internet Explorer, Safari, và Opera).

Bootstrap chia layout của một trang web ra thành 12 cột, mỗi cột 9 Grid và đây chính lả chuẩn thiết kế HTML & CSS của Bootstrap. Nó có hỗ trợ hầu hết các module của một trang web như menu, tabs, tooltip, popup, ... Ngoài ra nó còn sử dụng thêm Javascript để xử lý các hiệu ứng cấp cao, code javascript của nó sử dụng Jquery nên để sử dụng được bắt buộc bạn phải bổ sung thêm thư viện jQuery nữa. Nội dung trong trang web sử dụng Boostrap phải đặt trong một trong hai lớp sau:

* lớp *container* : cho chiều rộng cố định
* lớp *container-fluid*: cho chiều rộng cố định full màn hình của từng khung hình

Hiện nay Version mới nhất là Version 4.0.có 2 cách để sự dụng Bootstrap

* Download tại getbootstrap.com
* Dùng CDN

1. Trong bootstrap sẽ có 4 loại màn hình chính:

* Đầu tiên là màn hình *xs* dành cho điện thoại
* Tiếp đến là màn hình *sm* dành cho tablets
* Sau đấy là màn hình *md* dành cho máy tính
* Cuối cùng là màn hình *lg* dành cho màn hình lớn widescreens

Để sự dụng hệ thông lưới trong Bootstrap : đầu tiên phải tạo ra môt lớp *row,*

Ví dụ về một giao diện gồm 2 cột bằng nhau ở màn hình desktop và chia thành một cột 3 và một cột 9 ở màn hình máy tính bảng và một cột 4, một cột 8 ở màn hình kích thước lớn.

<div class=col-sm-3 col-md-6 col-lg-4">....</div>

<div class="col-sm-9 col-md-6 col-lg-8">....</div>

* 1. Jquery

Jquery là một thư viện được tạo ra từ JavaScript, Ajax hay nói cách khác đó là một bước phát triển mới của JavaScript. Jquery làm cho chúng ta dễ dàng thao tác hơn và đỡ mất thời gian hơn đối với khi thao tác trên Ajax cũng như JavaScript. Query làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax

* + jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery:
  + Thao tác DOM − jQuery giúp dễ dàng lựa chọn các phần tử DOM để *traverse* một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle.
  + Xử lý sự kiện − jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler.
  + Hỗ trợ AJAX − jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.
  + Hiệu ứng − jQuery đi kèm với rất nhiều các hiệu ứng đa dạng và đẹp mắt mà bạn có thể sử dụng trong các Website của mình.
  + Gọn nhẹ − jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped).
  + Được hỗ trợ hầu hết bởi các trình duyệt hiện đại − jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+
  + Cập nhật và hỗ trợ các công nghệ mới nhất − jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

Có hai cách để sử dụng jQuery:

* Cài đặt nội bộ **−** Bạn có thể tải jQuery Library trên thiết bị nội bộ của bạn và include nó trong HTML code.
* Sử dụng từ CDN (CDN Based Version) **−** Bạn có thể include thư viện jQuery vào trong HTML code một cách trực tiếp từ Content Delivery Network (CDN).

Ví dụ về một đoạn Jquery đơn giản. Chương trình làm ẩn đi phần tử có class là *text*

$(document).ready(function(){  
 $('.text').hide();  
});

Từ ví dụ trên ta thấy rằng thao tác cơ bản nhất của jQuery là chọn một phần nào đó của tài liệu HTML. Bạn tiến hành nó bằng cách sử dụng hàm $(). Thường thì nằm giữa dấu ngoặc () là một chuỗi dưới dạng tham số, nó có thể là những CSS Selectors.. Hàm $() chính là một jQuery Object, đây là nên móng cho tất cả những gì chúng ta sẽ học từ bây giờ. Jquery Object bao gồm không hoặc nhiều thành phần DOM và cho phép chúng ta tương tác với chúng bằng nhiều cách.

* 1. Google Map API

Google Maps là một ứng dụng bản đồ trực tuyến trên web được cung cấp miễn phí bởi Google. Nó có thể dùng để nhúng vào các trang web của bên thứ ba thông qua Google Maps API. Nó cung cấp rất nhiều những tiện ích nâng cao giúp người dùng tìm kiếm đường đi,các địa điểm, các địa danh... Google phát triển Google Maps API dành cho 4 mảng chính đó là:

* Các ứng dụng trên Android
* Các ứng dụng trên IOS
* Các ứng dụng web
* Các ứng dụng webservice

Hàng triệu trang web và ứng dụng sử dụng Google Maps API để cung cấp kinh nghiệm tìm kiếm vị trí cho người dùng của họ.Ngoài ra google map API còn hoạt động tốt trên các thiết bị di động như IOS, Android..

Để sử dụng ta nhúng thư viên file script vào có đính kèm API key

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC2Qcxw\_0TWYNdIrbrsuju7kjCWxlR5uGc&callback=initMap" async defer>

</script>

Định nghĩa 1 hàm javascript để tạo map

var map;

function initMap() {

map = new google.maps.Map(document.getElementById('google-map'), {

center: {lat: 21.017030, lng: 105.783902},

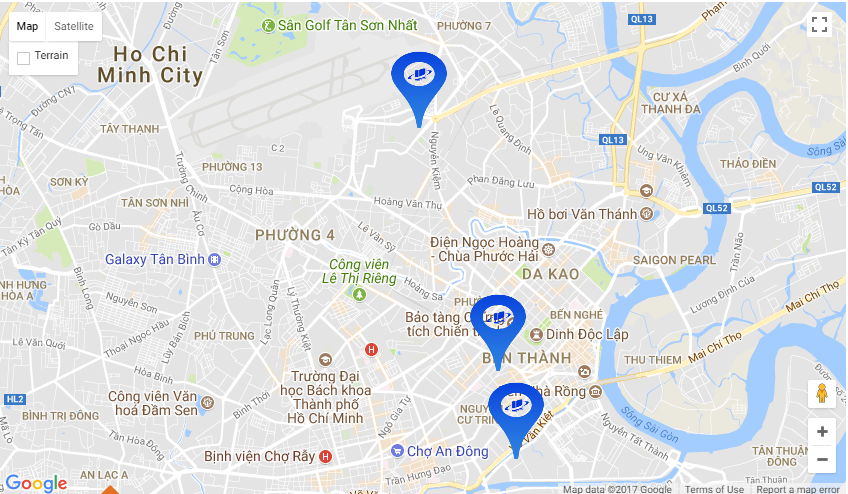
zoom: 13

});

}

//với lat,lng tương ứng với vĩ độ, và độ

ví dụ: cơ sở chính khoa công nghệ thông tin nằm ở vĩ độ 10.7611666, và kinh độ 106.6925135

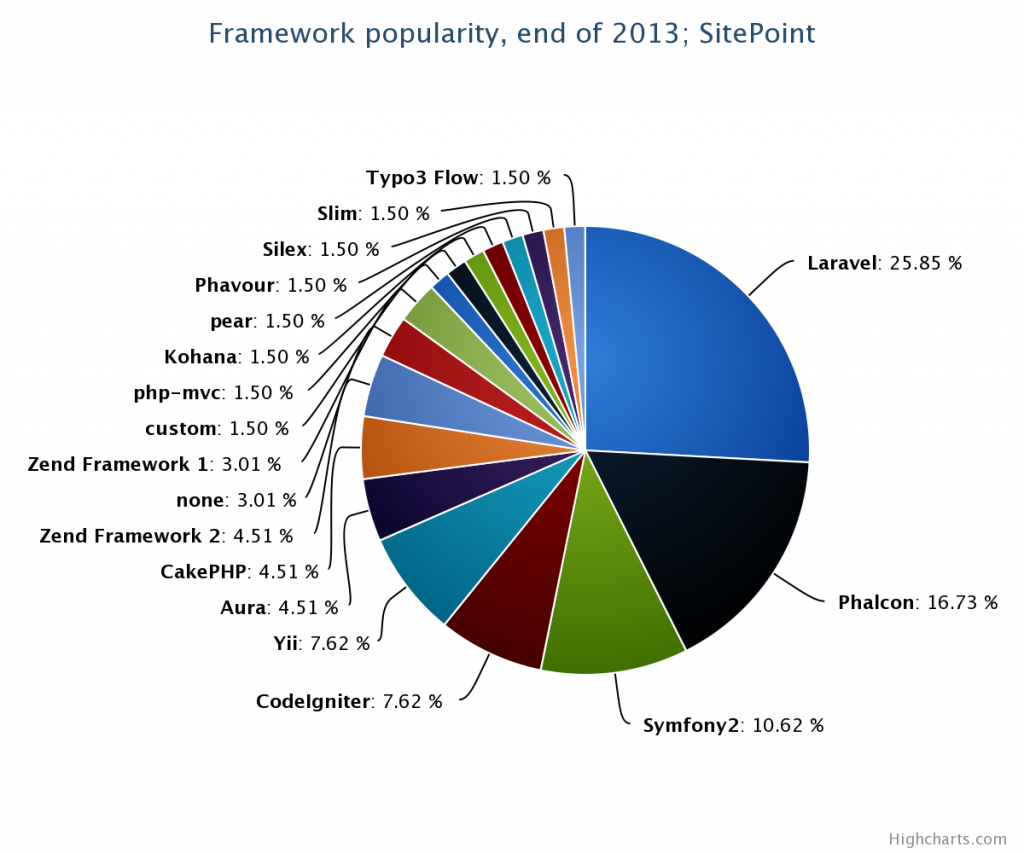


Hình 3*:Hiển thị các cơ sở học của trường bằng google map api*

CHƯƠNG III: LARAVEL FRAMEWORK

* + 1. Giới Thiệu Về PHP Laravel Framework
  1. Laravel Framework là gì:

Laravel là một Open source PHP Framework miễn phí được phát triển bởi Taylor Otwell ra mắt lần đầu vào tháng 6 năm 2011. Phiên bản hiện tại là 5.5, Nó là một framework dùng để xây dựng web application, được thiết kế dựa trên mô hình MVC( Model, View, Controller,) , toàn bộ source của laravel được đặt trên github (<https://github.com/laravel/laravel>), nó được hàng ngàn lập trình viên trên khắp thế giới ưu chuộng và phát triển. Cho tới năm 2015, thì Framework laravel hiện đang đứng top 1 thế giới về mức độ phổ biến và ưu dùng. Được cộng đồng hỗ trợ phát triển mạnh mẽ. . Theo kết qủa khảo sát của các Developer vào tháng 12 năm 2013, thì Laravel Framework đứng top 1 một trong những framework phổ biến nhất, tiếp sau là [Phalcon](http://en.wikipedia.org/wiki/Phalcon_%28framework%29), [Symfony2](http://en.wikipedia.org/wiki/Symfony2), [CodeIgniter](http://en.wikipedia.org/wiki/CodeIgniter) và các framework khác. Tháng 8 năm 2014, Laravel Framework được xem như là một dự án PHP phổ biến nhất trên Github.



Hình 4: các PHP Framework phổ biến

* 1. Lí do lựa chọn Laravel Framework

Như chúng ta đã biết ,PHP đang là 1 trong những ngôn ngữ phổ biến trong việc thiết kế và phát triển web hiện nay. Với các ưu điểm như mã nguồn mở, linh hoạt, dễ sử dụng, cộng đồng lớn… nó đã và đang trở thành sự lựa chọn số 1 của các lập trình viên. Tuy nhiên, để viết một website thì việc dùng code thuần để viết quả không hề đơn giản, chưa kể đến việc bảo trì, nâng cấp hay lý do bảo mật.  Chính vì lý do này nên các [laravel framework](http://blogit.vn/web-development/php/php-framework) đã ra đời, nhằm hỗ trợ cho các lập trình viên phát triển ứng dụng 1 cách dễ dàng hơn.Việc sử dụng laravel framework thích hợp phát triển các ứng dụng web một cách nhanh chóng, tiết kiệm được thời gian  
Em chọn đề tài này cũng bởi vì em muốn tìm hiểu một framework mới nhưng vô cùng mạnh mẽ này để trau dồi kiến thức , kỹ năng tự học ,tự nghiên cứu góp phần nâng cáo kiến thức cho bản thân sau khi ra trường, phù hợp với tính hình công nghệ, việc làm bên ngoài thực tế.

* 1. Cài Đặt Laravel Framework

Các công cụ cần chuẩn bị:

* Phiên bản PHP lớn hơn 5.3
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension
* chương trình Xampp (hoặc Wampp)

Xampp là chương trình tạo máy chủ Web (Web Server) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Không như Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

* Composer

Là một Dependency Management trong PHP, công cụ quản lý các thư viện mà project của bạn sử dụng, ta chỉ cần khai báo nó, composer sẽ tự động tải code của các thư viện, tạo ra các file cần thiết vào project của bạn, và update các thư viện nếu cần.Tải composer tại đây <https://getcomposer.org/Composer-Setup.exe>

Cài đặt Laravel Project:

Sau khi cài đặt Xampp và composer

* cú pháp để tạo project laravel mới

composer create-project --prefer-dist laravel/laravel blog

* để chạy laravel

php artisan serve

Laravel chạy thành công ở địa chỉ : localhost:8000

* 1. Cấu trúc thư mục Laravel framework

khoacntt/

├── app/

│ ├── Console

│ ├── Exceptions

│ ├── Http

│ │ ├── Controllers

│ │ ├── Middleware

│ │ └── Kernel.php

│ └── Providers

├── bootstrap/

├── config/

├── database/

├── public/

├── resources/

├── storage/

├── routes/

├── tests/

├── vendor/

├── .env

├── .env.example

├── composer.json

├── composer.lock

├── package.json

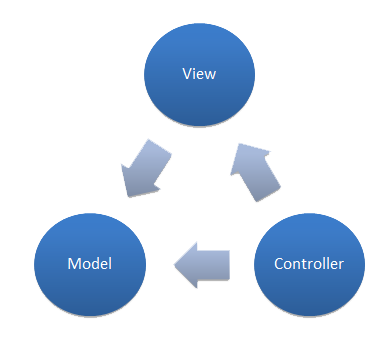
├── gulpfile.js

├── server.php

└── artisan

* app là thư mục được tự động load bởi controller, nó chức tất cả các thưc mục ,  các tập tin php, các class php, thư viện, models để xây dựng project như console,Http và Providers :
  + console: thư mục chứa tất các tập tin định nghĩa lệnh artisan
  + Exception: thư mục chứa tập tin quản lý ,điều hướng lỗi
  + Http: thư mục chứa models,controller,middleware và request,hầu hết các xử lý logic để xử lý các yêu cầu nhập ứng dụng sẽ được đặt tại đây,bao gồm:
    - Controllers là thư mục chứa các tập tin Controller.
    - middleware là thư mục chứ tập tin lọc và ngăn chặn request cũng như xác thực,cấp quyền.
    - Models là thư mục chứa các model thao tác với cơ sở dữ liệu.
    - Kernel.php là tập tin cấu hình, định nghĩa middleware hoặc nhóm middleware.
  + Providers chứa tất cả các tập tin là các lớp dịch vụ,cấu trúc dữ liệu .
* bootstrap là thư mục chứa tập tin điều hướng để khới động hệ thống,thư mục này có chứa một thư cache để tối ưu hóa hiệu năng.
* config chứa mọi tập tin cấu hình của laravel.
* database chứa các tập tin về cơ sở dữ liệu như:
  + migrations chứa các tập tin định nghĩa việc tạo và sử dụng bảng.
  + seeds chứa các tập tin định nghĩa insert dữ liệu mẫu vào database.
  + factories  chứa các tập tin định nghĩa các cột bảng dữ liệu để tạo ra các dữ liệu ảo phục vụ cho tests.
* public chính là webroot người dùng sẽ truy cập vào đây,đây cũng là nơi chứa các tập tin css,js,image,library,font…
* resources chứa các tập tin giao diện (frontend) ,views, ngôn ngữ.
* storage chứa các tập tin hệ thống như cookie,session,cache,log…
* routes là thư mục chứa các tập tin định nghĩa,xử lý hoặc điều hướng các đường dẫn bao gôm web,api và console.
* tests chứa các tập tin định nghĩa tests.
* vendor thư mục của composer.
* .env và .env.example là 2 tập tin cấu hình chính của laravel như keyapp,tên app,url app,email cơ sở dữ liệu bật hay tắt debug.
* composer.json, composer.lock là 2 tập tin của composer.
* package.js tập tin cấu hình của nodejs chứa các package cần thiết cho project.
* **gulpfile.js** là tập tin gulp builder**.**
* phpunit.xml là tập tin xml của phpunit dùng để testing project.
* server.php là tập tin để artisan trỏ đến tạo server khi gõ lệnh php artisan serve
* artisan tập tin thực thi lệnh của Laravel, cũng là tập tin mà chúng ta tương tác nhiều nhất.
  + 1. Các thành phần cơ bản của Laravel Framework

2.1 Mô hình MVC

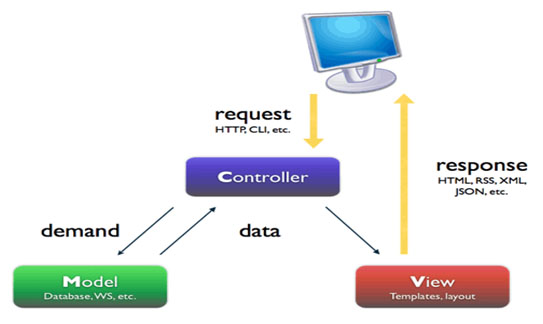


Hình 5*:Mô hình MVC*

MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm

* M là Model: cấu trúc dữ liệu theo cách tin cậy và chuẩn bị dữ liệu theo lệnh của controller. Đây là thành phần chứa tất cả nghiệp vụ logic, phường thức xử lý, truy xuất database , đối tượng mô tả dữ liệu như các Class, hàm xử lý…
* V là View: đảm nhận việc hiển thị thông tin, tương tác với người dùng nhận dữ liệu từ thao tác của người dùng.
* C là Controller: giữ nhiệm vụ điều hướng từ các lệnh của người dùng và gọi đúng những phương thức xử lý thích hợp, chẳng hạn thành phần này sẽ nhận Request từ url và form để thao tác trực tiếp ,gửi đến cho Model để cập nhật dữ liệu rồi sau đó truyền lệnh đến View để hiển thị ra giao diện.

Luồng đi trong MVC



Hình 6:*Luồng đi trong mô hình MVC*

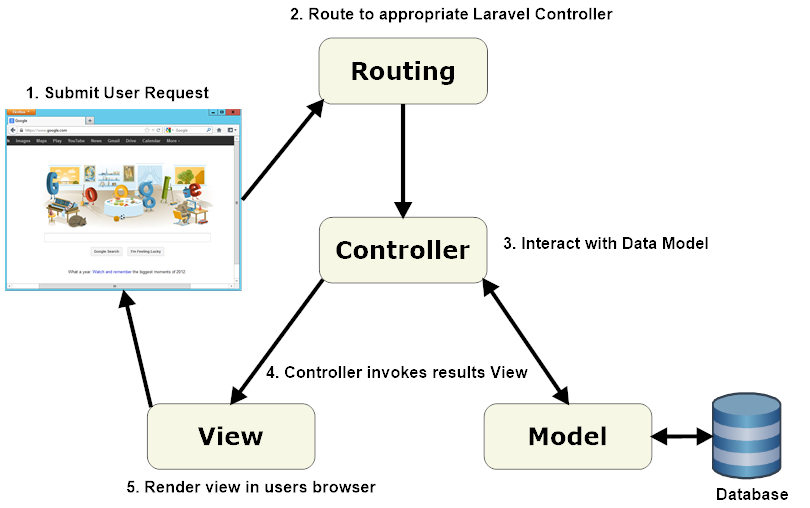
Khi có một yêu cầu từ phía client gửi đến server, Bộ phận controller có nhiệm vụ nhận yêu cầu, xử lý yêu cầu đó. Và nếu cần, nó sẽ gọi đến phần model, vốn là bộ phần làm việc với Database..

Sau khi xử lý xong, toàn bộ kết quả được đẩy về phần View. Tại View, sẽ render ra mã HTML tạo nên giao diện, và trả toàn bộ html về trình duyệt để hiển thị.

### 2.2 Route

Với các ứng dụng web ngày nay, việc làm cho ứng dụng có chức năng tốt – giao diện đẹp là một chuyện, nhưng để có một trang web thực sự tốt thì “đường dẫn thân thiện” là không thể thiếu.

**Route** có nghĩa là định tuyến, nó là một trong những thành phần quan trọng nhất của laravel ,có chức năng là định ra các dạng request (hay url) định sẳn ,dẫn đường cho HTTP request được gởi đến nơi mà chúng ta muốn đến.Hay nó là nơi nhận request từ người dùng( bằng phương thức get hoặc post) sau đó có thể chỉ định 1 action của 1 Controller thực hiện request đó. Ví dụ khi người dùng truy cập vào website với đường dẫn là khoacntt/news, thì lúc này Route sẽ nhận được một phần trong URL là /news và nó sẽ xử lý request đó gọi đến Controller News hoặc là một thao tác khác  (*xử lý thế nào thì tùy vào coder*). Đặc biệt hơn các Framework trước đây như Codeigniter  là trong Laravel Route có rất nhiều loại khác nhau như Route dành cho phương thức POST, Route dành cho phương thức GET ở cùng một URL.



Hình 7*: Quy trình hoạt động của Route*

Ví dụ bạn có 1 trang web thể hiện danh sách sinh viên của một khoa cntt:  
[*http://khoacntt.com/lop.php?vendor=cntt&type=sinhvien*](http://khoacntt.com/lop.php?vendor=cntt&type=sinhvien)

Thay vì để một đường dẫn ngớ ngẩn như vậy, chúng ta cần tạo ra 1 trang web có thể chạy dưới đường dẫn “đẹp” kiểu như:

<http://khoacntt.com/sinhvien/danhsach>

Rõ ràng đường dẫn đẹp thì có lợi cho người dùng, rất hiệu quả về mặt marketing cũng như SEO. Route trong Laravel Framework cho phép ta thực hiện được điều này.  
Hầu hết các route của laravel định nghĩa trong thư mục  routes. Laravel là một framework hỗ trợ chúng ta rất đa dạng về route, và cũng rất dễ sử dụng.

**Cấu Trúc**

*Route::get(‘index/tintuc’,action(){} );*

Trong đó:

* Route: tên từ khóa khai báo 1 route trong laravel
* get: phương thức sử dụng
* index/tintuc : đường dẫn url
* action: hành động của route, có thể gọi đến một hàm, hoặc một controller

Ví dụ: *Route::get('/dao-tao' ,'PageController@TrangDaoTao');*

*Route::post('/dang-nhap','UserController@post\_Danhnhap');*

Ngoài ra Laravel còn bao gồm có các loại như sau:

* **Route::get**nhận resquest với phương thức GET.
* **Route::post** nhận resquest với phương thức POST.
* **Route::put** nhận resquest với phương thức PUT.
* **Route::delete** nhận resquest với phương thức DELETE.
* **Route::match** kết hợp nhiều phương phức như POST,GET,PUT,..
* **Route::any** nhận tất cả các phương thức.
* **Route::group**tạo ra các nhóm route để dễ quản lý và cấp quyền cho một nhóm các route.
* **Route::controller**gọi đến controller tương ứng mà chúng ta tự định.
* **Route::resource**sử dụng với resource controller.

### 2.3 Controller

Controller là chữ C trong mô hình MVC, nó là một lớp gồm nhiều hành động được định nghĩa bởi người lập trình có thể thao tác với Model và View.

Sau khi Route nhận được request thì sẽ chuyển hướng request đến cho controller và giao cho controller này xử lý.  Controller sẽ gọi tới Model để tương tác với cơ sở dữ liệu (database) và  đưa dữ liệu tới View và View sẽ nhận dữ liệu và chuyển thành những đoạn mã HTML rồi gửi về cho Client.

để tạo controller trong Laravel cũng phải có một số ràng buộc như sau:

* Controller phải được đặt trong đường dẫn *App\Http\controllers*
* Tên của controller phải giống với tên class trong file controller đó.
* Class controller (do ta tạo) phải extends(kế thừa) từ Controller (chú ý chữ C phải viết hoa).

**Cách tạo một Controller:**

* Tạo trực tiếp bằng file  có tên trùng với tên Class của Controller, Ví dụ *MyController.php*

*namespace App\Http\Controllers;*

*class MyController extends Controller*

*{*

*//Thực hiện các công việc*

*}*

* Tạo bằng dòng lệnh

*php artisan make:controller MyController*

Khi đó Laravel sẽ tự động tạo ra một Controller như sau :

*namespace App\Http\Controllers;*

*use App\User;*

*use App\Http\Controllers\Controller;*

*class controllername extends Controller*

*{*

*//các hành động.*

*}*

2.4 View

View là một tầng logic trong mô hình MVC, mục đích của sử dụng view là tách phần hiển thị nội dung ra khỏi phần xử lý dữ liệu của PHP. Do đó, đúng với tiêu chuẩn thì View chỉ chịu trách nhiệm nhận dữ liệu đầu cuối và hiển thị ra giao diện mà hầu như rất ít dùng PHP.

Trong Laravel framework, View có nhiệm vụ nhận dữ liệu từ Controller và sau đó dựa vào layout của giao diện nó sẽ hiển thị ra cho người dùng. Trong laravel, mặc định tất cả các file view sẽ được lưu trong thư mục app/views với đuôi file là .php hoặc .blade.php. Trong View ta có thể dùng tất cả các ngôn ngữ mà PHP hỗ trợ như :HTML,CSS,Javascript,Bootstrap… Để gọi view trong laravel ta dùng cú pháp sau: *Route::get('duogndanroute',function(){*

*return view('tenview','dataneuco');*

*});*

Trong đó:

* *'duogndanroute'* là tên Route mà ta đã tạo.
* *'tenview' là tên view*
* $*'dataneuco '*là mảng dữ liệu sẽ được merge với tên view bằng hàm array\_merge.

Ví dụ:

View không có data   
return view('pages.about');

View có data   
  
*$news =NewsQModel::get\_news(4);*

*return view('pages.news', [*

*'news' =>$news*

*]);*

2.5 Blade template engine

Hiện nay cũng có khá nhiều các PHP framework nổi tiếng như Codeigniter,zend,CAKE,... Nhưng chỉ một số framework có tích hợp template engine , vì đơn thuần khi sử dụng template engine thì nó sẽ làm giảm hiệu năng của trang web. Nhưng với Laravel thì lại khác, Laravel có tích hợp sẵn cho chúng ta blade template với tốc độ xử lý rất nhanh. Template engine có tác dụng giúp sạch đi những đoạn code PHP nằm trong View nên tách biệt hoàn toàn giữa người cắt CSS và người code PHP.

blade template bản chất nó là một view trong Laravel nên các bạn phải đặt nó trong thư mục resources/views. Và tên file của blade template phải kết thúc bằng .blade.php

Hiển thị dữ liệu trong blade template

Trong PHP khi muốn in dữ liệu ra màn hình các bạn có thể dụng echo,printf,... Nhưng ở đây ta sử dụng blade template thì sẽ có 2 cách để in dữ liệu ra màn hình khác.

* Sử dụng {{}}xuất toàn bộ nội dung kể cả tag HTML,Javascript. ví dụ: Hôm nay là ngày: {{$noidung}}
* sử dụng {!!$noidung !!} biên dịch mã HTML,Javascript.
* Cấu trúc điều khiển if else : @if () @else @endif
* Vòng lặp : @for() @endfor hoặc @foreach() @endforeach

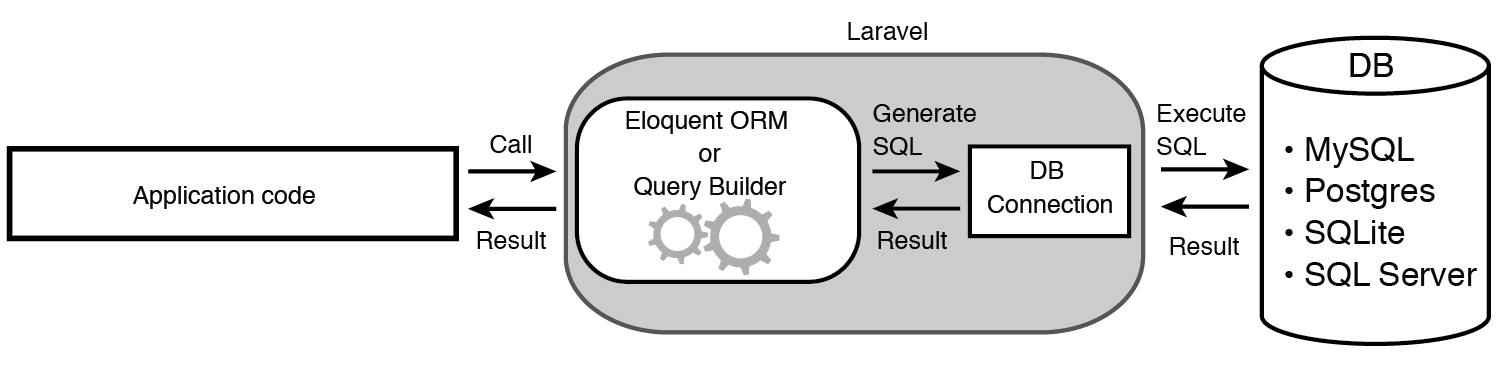
Một trong những chức năng hữu ích của blade template là kế thừa giao diện(Template inheritance). Balde template sử dụng các từ khóa cho việc kế thừa lại một view giao diện như:@yield(),@section(),@extends() :

* + @yield(),@section: ó tác dụng như một biến nó được tạo ra nhằm báo cho blade template  biết vị trí ý sẽ dùng để chèn nội dung cho biến đó.
  + @extends(): Có tác dụng khai báo cho blade template biết là file hiện tại đang kế thừa từ file nào

2.6 Query Builder và Eloquent ORM

Trước đây khi làm việc với PHP thuần, để tương tác với database ta sẽ cần phải kết nối với database và viết những câu truy vấn hết sức là phức tạp để gọi được dữ liệu ra. Nhưng khi sử dụng các famework đặc biệt là Laravel thì ta sẽ không phải lo về điều đó nữa, bởi Laravel có cung cấp cho chúng ta một chức năng rất hay là Query Builder.và Model. các bước thực hiện tuần tự như sau:

* Cấu hình database
* Tạo Model
* Viết Query builder hoặc ORM



*Hình 8:Thao tác với database trong Laravel*

2.6.1. Query Builder

Trong Laravel, Query Build cho phép ta thao tác với Cơ sở Dữ liệu 1 cách đơn giản và thuận tiện nhất. Nó được mọi người sử dụng rất nhiều truy xuất DB, và nó hoạt động trên tất cả hệ thống CSDL mà Laravel hỗ trợ (mysql, sqlite, pgsql ...). Query builder của Laravel cung cấp cú pháp ngắn gọn để thực thi câu lệnh truy vấn dữ liệu, nó là công cụ vô cùng mạnh mẽ có thể sự dụng để thao tác với database như:

* + - Lấy ra các dòng trong bảng
    - Thêm, Xóa, sửa một record mới
    - Kết bảng
    - Lọc, nhóm, hay sắp xếp các record.

Query Builder sử dụng kết nối PDO (PHP Data Object), bản thân API PDO đã bảo vệ bạn trước các tấn công SQL Injection, do đó khi xử dụng Query Builder bạn không cần lo lắng khi thao tác nhập xuất dữ liệu với DB.

Cú pháp sử dụng:

*DB::table($ten\_bang)->bieuthuc->laydulieu();*

Trong đó:

* $ten\_bang là đặt bảng mà trong truy vấn sẽ truy vấn tới.
* bieuthuc là các biểu thức mà trong câu truy vấn sẽ thực hiện như: where, orWhere, orderBy, groupBy, ...
* laydulieu() là phương thức để thực thi câu truy vấn bao gồm các biểu thức trước đó, cơ bản nhất là get() ngoài ra còn có, first(), pluck(), lists(), ...

Ví dụ: Lấy ra sinh viên có mssv là 1451010189 hoặc tên là Thương.

$product = DB::table(sinhvien)->where(mssv,1451010189)

->orWhere(ten,Thuong)

->get();

Ngoài ra còn có những cú pháp sau :

* Select cột trong bảng:

DB::table(tenabng)->select(cot1,cot2)->get();

* Join bảng

$users = DB::table('users')->join('contacts', 'users.id', '=', 'contacts.user\_id')->get();

* OrderBy.

$users = DB::table('users')

->orderBy('name', 'desc')

->get();

GroupBy

$users = DB::table('users')

->groupBy('account\_id')

->having('account\_id', '>', 100)

->get();

…

2.6.2. ORM

Eloquent ORM đi kèm với Laravel cung cấp ActiveRecord đầy đủ, đẹp đẽ và đơn giản để làm việc với database. Mỗi bảng của database sẽ được ánh xạ qua ‘Model’, và model này được sử dụng để tương tác với bảng. Do trong Eloquent ORM, mỗi bảng của database tương ứng với 1 model nên để bắt đầu sử dụng được nó, ta cần tạo Eloquent model trong thư mục app. Lưu ý là mỗi Eloquent model này đều phải extend Illuminate\Database\Eloquent\Model class. Về cơ bản, các câu lệnh query của Eloquent ORM có 1 chút thay đổi so với Query Builder, khiến câu lệnh trông ngắn gọn và đẹp đẽ hơn:

Ví dụ:

*namespace App;*

*use Illuminate\Database\Eloquent\Model;*

*class User extends Model*

*{*

*protected $table = 'users';*

*}*

từ đó ta có $users = User::all();Kết quả trả về là 1 eloquent collections, trong đó mỗi kết quả là 1 User object.

Ngoài ra Eloquent còn có 1 phần mình rất thích là Relationships với các mối quan hệ giữa các bảng như:

* One To One
* One To Many
* One To Many (Inverse)
* Many To Many
* Has Many Through
* Polymorphic Relations
* Many To Many Polymorphic Relations

2.6.7. So sánh Query builder với ORM

*Cú pháp*

|  |  |  |
| --- | --- | --- |
|  | Query builder | ORM |
| Lấy tất cả các record | $users = DB::table('users')  ->get(); | $users = User::all(); |
| Lấy record theo id | $user = DB::table('users')  ->where('id', 1)->first(); | $user = User::find(1); |
| Lấy 1 record theo 1 trường xác định | $user = DB::table('users')  ->where('name', Thuong)  ->first(); | $user= User::where('name', 'Framgia')  ->first(); |
| Lấy 1 record nhất định của tất cả các bản ghi | $names = DB::table('users')->lists('name'); | $names = User::lists('name'); |
| Aggregates | $users = DB::table('users')->count(); | $users = User::count(); |
| Insert | DB::table('users')  ->insert(['email'=>'test@framgia.com']  ); | $user= new User;$user  ->email= test@framgia.com';  $user->save(); |
| Update | DB::table('users')->where('id',1)  ->update(['name' => 'Framgia Test']); | $user=User::find(1);  $user->name = “DoanVan”  $user->save(); |
| Delete | DB::table('users')->where('id','=',1)->delete(); | $user = User::find(1);  $user->delete(); |

* Tính bảo mật
  + Query Builder sử dụng  nhằm bảo vệ ứng dụng và tránh các lỗi về SQL injection trong khi Eloquent ORM chưa làm được điều đó.
* Tính tương tác
  + Bạn có thể sử dụng tất cả các function của Query Builder trong Eloquent nhưng không thể sử dụng các funcation của Eloquent trong Query Builder.
* Tính dễ sử dụng
  + Các câu lệnh của Eloquent ORM là ngắn gọn, dễ hiểu và dễ sử dụng hơn so với các câu lệnh khá dài dòng của Query Builder.
  + Hơn nữa, sử dụng Eloquent ORM cũng dễ dàng hơn trong việc kết nối giữa các bảng với nhau.
  + Tuy nhiên, với 1 số truy vấn phức tạp, không sử dụng được Eloquent ORM thì vẫn cần sử dụng Query Builder để cho kết qủa chính xác nhất.
* Hiệu suất
  + So sánh 1 số query test, thì cho thấy tốc độ của Eloquent ORM bị chậm hơn so với Query Builder, nhất là ở những truy vấn cần thao tác với dữ liệu lớn.

2.7 Migrations

Migration trong Laravel giống như một control database có tác dụng quản lý cũng như lưu trữ lại cấu trúc của database giúp cho việc sửa đổi database trở lên dễ dàng hơn.

Điều kiện để sử dụng Migations:

* Phải có kết nối với database .
* Migrations muốn sử dụng được thì phải nằm trong thư mục App\database\migrations.

Lệnh tạo migration trong composer:

* php artisan make:migration TenMigrate  : Tạo migrations thông thường.
* php artisan make:migration TenMigrate --create=TableName  : Tạo migrations cho bảng.
* php artisan make:migration TenMigrate --table=TableName  : Tạo migrations chỉnh sửa bảng.

Ví dụ: tạo migration TaoBangTable :*php artisan migration TaobangTable*

Sau khi tạo thành công, file migration có dạng: *TaobangTable yyyy\_mm\_dd\_TaoBangTale.php*

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('users', function (Blueprint $table) {

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

//

}

}

* Hàm up trog migration có tác dụng thực thi migration
* Hàm down có tác dụng khi thực thi rollback(trở về trước đó)
* Dưới đây là một số các câu lệnh tạo bảng hay dùng trong Migrations.

*Schema::create('students', function (Blueprint $table) {*

*$table->increments('id');*

*$table->string('name');*

*$table->string('sex');*

*$table->string('birth');*

*$table->string('class');*

*$table->string('year');*

*$table->string('subject');*

*$table->string('study');*

*$table->string('type');*

*$table->string('phone');*

*$table->string('email');*

*});*

chạy lênh php artisan migrate sẽ tạo một bảng students trong database với các cột như khai báo.

2.8 Authentication

Authentication hay còn được gọi là xác thực là hành động nhằm thiếp lập hoặc chứng thực một cái gì đó đáng tin cậy. Khái niệm Authentication hay bị nhầm lẫn với Authorization (sự ủy quyền) tuy nhiên việc phân biệt 2 khái niệm này đơn giản ta có thể hiểu là Authentication là quá trình xác minh nhận dạng của người dùng còn Authorization là cho người dùng quyền để thực thi thao tác trên hệ thống.

Việc xác thực sẽ đảm bảo được thông tin được gửi tới đúng người cần gửi và đảm bảo người dùng đang thi hành các chức năng trên hệ thống là người được ủy quyền làm việc đó. Laravel giúp việc triển khai Authentication trở nên rất đơn giản, hầu hết tất cả mọi thứ đã được cấu hình trước trong file config/auth.php.

Để tạo Auth trong Laravel thì cũng hết sức đơn giản. Các bạn chỉ cần dùng lệnh: *php artisan make:auth*

Bây giờ bạn có các route và view chuẩn bị cho các authentication controllers, bạn đã sẵn sàng để đăng kí và xác nhận những user mới cho ứng dụng. Bạn chỉ đơn giản truy cập ứng dụng thông qua trình duyệt. Các authentication controller đã sẵn sàng gồm các logic (thông qua trait của chúng) để xác nhận những user đã tồn tại và lưu những user mới vào database. Khi một user được xác nhận thành công, họ sẽ được chuyển sang URI '/'. Bạn có thể tùy biến địa chỉ chuyển hướng post-authentication bằng cách định nghĩa thuộc tính redirectTo trong AuthController:

*protected $redirectTo = '/home';*

Khi một user không được xác nhận thành công, họ sẽ tự động chuyển hướng quay lại form đăng Ta cũng có thể tùy biến "guard" cái mà sử dụng để xác thực user. Để bắt đầu, định nghĩa một thuộc tính guard trong AuthController. Giá trị của thuộc tính này nên tương ứng với một trong những guard đã được cấu hình trong file auth.php.

*protected $guard = 'admin';*

Ta có thể truy cập người dùng đã được xác thực thông qua facade Auth:

*$user = Auth::user();*

Để xác định user đã đăng nhập vào ứng dụng của taa hay chưa, ta có thể sử dụng phương thức check trên face Auth, sẽ trả về true nếu user đã được xác thực:

*if (Auth::check()) {*

*//user đã đăng nhập thành công*

*}*

Ta cũng có thể dùng cách này để bảo vệ các Route, có nghĩa là khi người dùng chưa đăng nhập thì ta không cho họ vào Route trang admin. Route middleware có thể được sử dụng để cho phép chỉ những user đã được xác thực truy cập vào các route đã cho. Laravel mang tới middleware auth, cái mà được định nghĩa trong app\Http\Middleware\Authenticate.php. Toàn bộ những gì ta cần là đính kèm middleware vào định nghĩa (khai báo) của route.

*Route::get('profile', ['middleware' => 'auth', function() {*

*// Only authenticated users may enter...*

*}]);*

Tất nhiên, ta không bắt buộc phải sử dụng các authentication controller trong Laravel. Nếu ta lựa chọn xóa những controller này, ta sẽ cần phải quản lí việc xác thực user bằng cách sử dụng các class Laravel xác thực trực tiếp.

Chúng ta sẽ truy cập vào các Laravel's authentication services thông qua facade Auth, vì vậy chúng ta cần đảm bảo import facade Auth tại đầu class. Tiếp theo, hãy kiểm tra phương thức attempt:

*public function authenticate()*

*{*

*if (Auth::attempt(['email' => $email, 'password' => $password])) {*

*// Authentication passed...*

*return redirect()->intended('dashboard');*

*}*

*}*

Phương thức attempt chấp nhận một mảng các cặp key / value như là tham số đầu tiên. Các giá trị trong mảng sẽ được dùng để tìm user trong database. Vì vậy trong ví dụ trên, user sẽ được lấy ra bởi giá trị của cột email. Nếu tìm thấy user, hashed password được lưu trong database sẽ được dùng để so sánh với giá trị hashed password mà được truyền vào phương thức thông qua mảng. Nếu 2 hashed passowrd trùng hợp, một session sẽ được bắt đầu cho user.Phương thức attemp sẽ trả về true nếu xác thực thành công. Ngược lại là false.

Phương thức intended trên redirector sẽ chuyển hướng user tới URL họ vừa cố gắn truy cập trước khi bị bắt bởi authentication filter. Một fallback URI có thể được cho trước vào phương thức này trong trường hợp đích đến dự kiến không có.

Để đăng xuất người dùng khỏi ứng dụng của ta, ta có thể sử dụng phương thức logout trên facade Auth. Việc này sẽ xóa toàn bộ thông tin xác thực trong session của user:

*Auth::logout();*

Ngoài ra nếu ta muốn cung cấp chức năng "remember me" trong ứng dụng, ta có thể truyền một giá trị boolean như tham số thứ 2 vào phương thức attempt, cái mà sẽ giữ cho người dùng đã được xác thực vô thời hạn, hoặc tới khi họ đăng xuất thủ công. Tất nhiên, table users phải có một cột string remember\_token, cái mà sẽ được dùng để lưu token "remember me". Nếu bạn "remembering" người dùng, bạn có thể dùng phương thức viaRemember để xác định nếu user đã được xác thực bằng cách dùng cookie "remember me":

*if (Auth::viaRemember()) {*

*//*

*}*

*Nguồn tham khảo: https://laravel.com/docs/5.4/authentication*

2.9 Middleware

HTTP middleware cung cấp một giải pháp tiện ích cho việc lock các HTTP request vào ứng dụng. Ví dụ, Laravel có chứa một middleware xác thực người dùng đăng nhập vào hệ thống. Nếu user chưa đăng nhập, middleware sẽ chuyển hướng user tới màn hình login. Còn nếu user đã đăng nhập rồi, thì middleware sẽ cho phép request được thực hiện tiếp tiến trình xử lý.

Tất nhiên là có thể viết thêm middleware để thực hiện nhiều tác vụ nữa ngoài việc kiểm tra đăng nhập vào hệ thống. Middleware CORS chịu trách nhiệm cho việc thêm các header hợp lý vào trong tất cả các response gửi ra ngoài. Middleware log có thể thực hiện ghi log cho tất cả các request tới chương trình.

Vài middleware đã có sẵn trong Laravel framework, bao gồm middleware cho bảo trì, xác thực, phòng chống CSRF và còn nữa. Tất cả những middleware này nằm trong thư mục app/Http/Middlware.

Tạo middleware:

Để tạo mới một middleware, sử dụng câu lệnh artisan *make:middleware*

*php artisan make:middleware admin*

Thiết lập middleware cho Route

Nếu bạn muốn thiết lập middleware cho một số route cụ thể, bạn đầu tiên cần phải thêm middleware vào trong biến $routeMiddleware trong file app/Http/Kernel.php và đặt cho nó một key:

Sau khi đã được khai báo trong HTTP Kernel, bạn có thể sử dụng khoá middleware để thiết lập thông số cài vào trong route:

*Route::get('admin/profile', ['middleware' => 'auth', function () {*

*//*

*}]);*

để gán nhiều Middleware ta dùng 1 mảng như sau:

*Route::get('/', ['middleware' => ['first', 'second'], function () {*

*//*

*}]);*

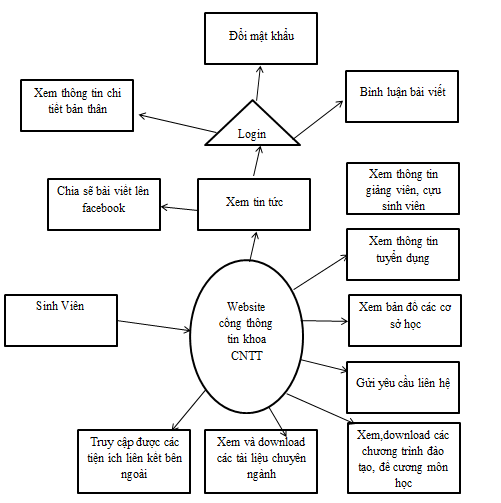
Nguồn tham khảo: https://laravel.com/docs/5.4/middleware

* + 1. Những lợi ích của Laravel Framework
* ổ chức file và cấu trúc code rõ ràng
* Được thừa hưởng những ưu điểm và thế mạnh của các framework khác.
* Hệ thống thư viện và công cụ phong phú
* Form Validation
* Code ít hơn, nhưng làm được nhiều việc hơn
* Mô hình MVC rõ ràng, dễ bảo trì
* Thích hợp làm việc nhóm và phân chia công việc
* Chức năng hỗ trợ bảo mật, validate dữ liệu, SQL Injection,…
* Template phong phú, thư viện form helper giúp xây dựng form nhanh và bảo mật
* Hỗ trợ đường dẫn thân thiện (Friendly Url)
* Bảo mật session, cookie,…
* sử dụng mô hình ORM rất đơn giản khi thao tác với DB
* Các lệnh tương tác với cơ sở dữ liệu cực kỳ ngắn gọn và thân thiện.
* Việc quản lý layout thật sự giản đơn với Balade Templating .
* Dễ dàng tích hợp các thư viện khác vào dự án, và được quản lý với Composer
* Phần route rất mạnh!!!

Điểm yếu của Laravel là không hỗ trợ các phiên bản PHP cũ vì thế các website có nền tảng PHP version thấp hơn khá là khó khi có ý định chuyển sang Laravel và phải luôn cập nhật các thông tin mới nhất về PHP để áp dụng vào Laravel. Phiên bản của Laravel hiện tại chỉ hỗ trợ phiên bản 5.5.9 trở lên.

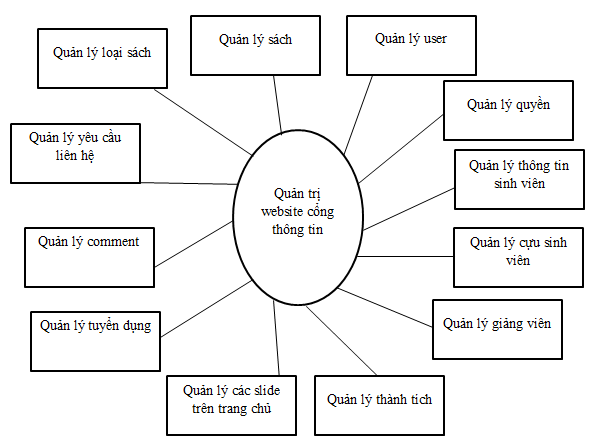
CHƯƠNG IV: PHÂN TÍCH, THIẾT KẾ

1. Sơ đồ các quy trình của website cổng thông tin điện tử
   1. phía người sinh viên



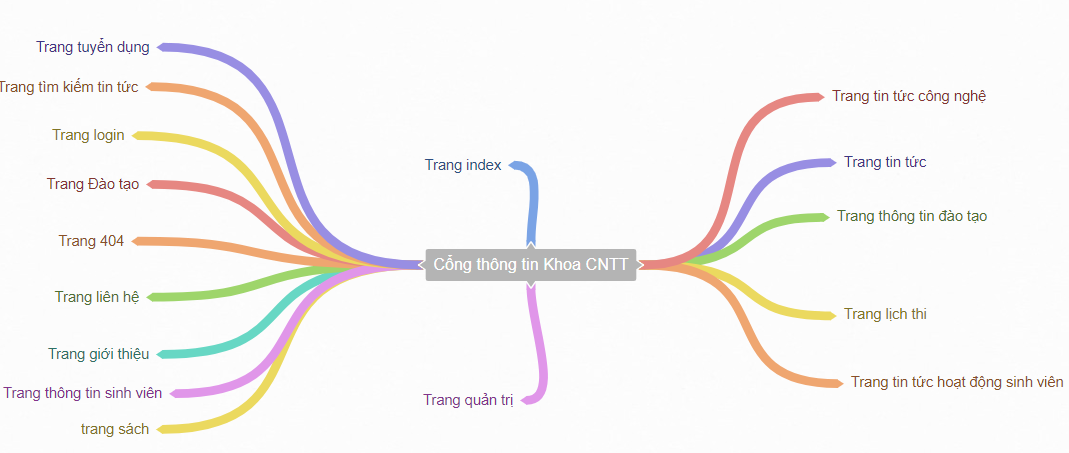
Hình 9*:Quy trình hoạt động của sinh viên trên website cổng thông tin*

* 1. Phía người quản trị website



*Hình 10:Hoạt động của người quản trị website*

1. Sơ đồ website



Hình 11:sơ đồ website cổng thông tin

1. Cơ sở dữ liệu
   1. Các Thực Thể

* Sinh Viên
* Giảng Viên
* Cựu Sinh Viên
* Users
* Việc Làm
* Liên Hệ
* Thành Tích
* Visitor
* Sách
* Loại Sách
* Tin Tức
* Loại Tin Tức
* Bình Luận
* Slide
  1. Sơ Đồ CDM



Hình 12: Sơ đồ CDM

* 1. Sơ Đồ PDM



Hình 13: Sơ đồ PDM

### 3.4. Từ điển dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| Tên dữ liệu | Diễn giải | Loại dữ liệu | Ghi chú |
| Sinh Viên | Sinh Viên | Table |  |
| IdStudent | ID sinh viên | Int | Sinh tự động |
| Name | Họ tên sinh viên | Char(30) |  |
| Sex | Giới tính | Text |  |
| Birth | Ngày sinh | Datetime |  |
| Class | Lớp | Text |  |
| Year | Khóa học | Date |  |
| Study | Khoa | Text |  |
| Type | Loại đào tạo | Text |  |
| Phone | Số điện thoại | Text |  |
| Created\_at | Ngày tạo | Datetime |  |
| Updated\_at | Ngày sửa | Datetime |  |
| IDGiangVien | ID giảng viên | Int | Sinh tự động |
| Image | Hình ảnh | Text |  |
| Alias | Chức danh | Text |  |
| Degree | Học vị | Text |  |
| Subject | Phụ Trách | Text |  |
| Visitor | Khách truy cập | Table | Lưu số lượng truy cập |
| IP | Địa chỉ IP | Text |  |
| Country | Địa chỉ quốc gia | Text |  |
| Click | Tổng số lượt click | Int |  |
| Việc Làm | Bảng tin tuyển dụng | Table |  |
| IDViecLam | ID Việc Làm | Int | Sinh tự động |
| Company | Công ty tuyển | Text |  |
| Type | Loại việc làm | Text | Ví dụ (full time,partime) |
| Start\_date | Ngày đăng | Datetime |  |
| End\_date | Ngày hết hạn | Datetime |  |
| Tag | Tag liên quan | Text |  |
| Address | Địa chỉ | Text |  |
| Salary | Lương | Int |  |
| Content | Nội dung | Text |  |
| Slug | Tiêu đề không dấu | Text |  |
| Liên Hệ | Bảng liên Hệ | Table |  |
| IDLienHe | ID Liên Hệ | Int | Sinh tự động |
| File | File | Text | File gửi lên |
| Award | Bảng thành tích | Table |  |
| IDAward | ID Thành tích | Int | Sinh tự động |
| Competition | Cuộc thi | Text |  |
| Gain | Thành tích | Text |  |
| Bình Luận | Bảng bình luận | Table |  |
| IDBinhLuan | ID Bình Luận | Int | Sinh tự động |
| IDTinTuc | ID Tin Tức | Int | Khóa ngoại |
| IDUser | ID user | int | Khóa ngoại |
| IDLoaitin | Id loại tin | Int | Khóa ngoại |
| Author | Tác giả | Text |  |
| Sumary | Tóm tắt | Text |  |
| Business | Vị trí công tác | Int |  |
| Info | Thông tin | Text |  |
| Slide | Bảng slide | Table |  |
| IDSlide | ID Slide | Int | Sinh tự động |
| TypeSlide | Loại Slide | Int | Ví dụ header,bottom,logo |
| Password | Mật khẩu | Text | Mã hóa |
| Remember\_token | Token\_Csrf | Text | Bảo mật csrf |
| Role | Quyền | Int | Ví dụ guest,admin,super admin |

CHƯƠNG IV: HIỆN THỰC

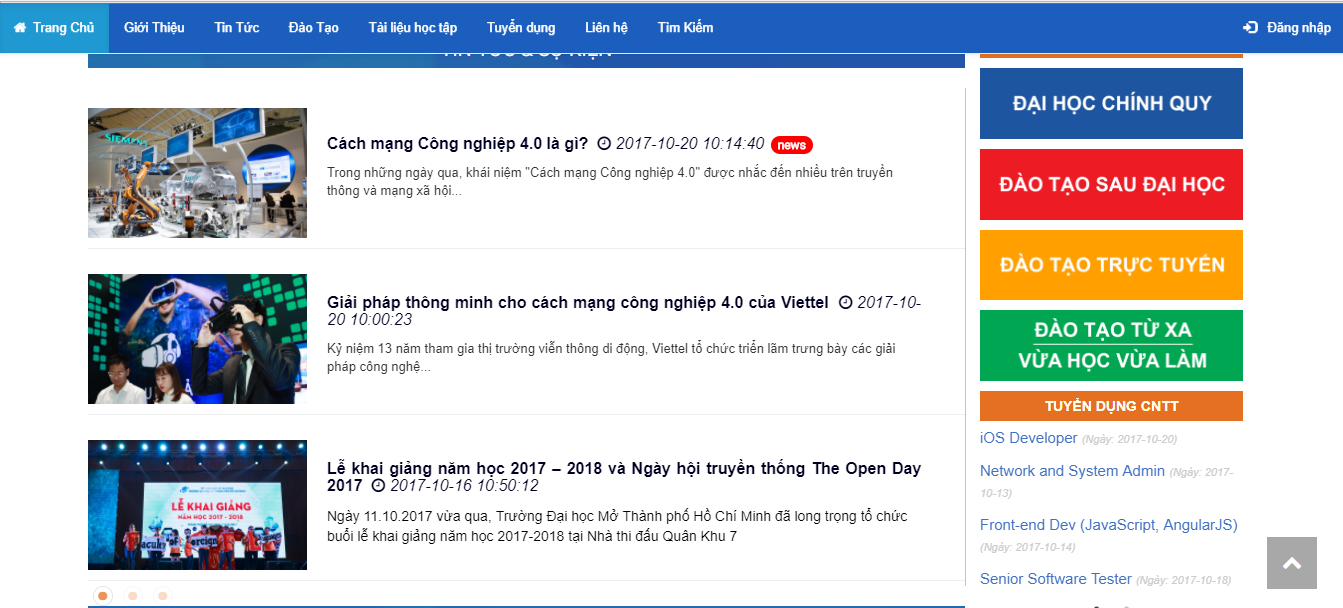
1. Giới thiệu website cổng thông tin khoa công nghệ thông tin

Website có tên Cổng thông tin khoa CNTT demo khoa công nghệ thông tin trường đại học mở TP.HCM là website cổng thông tin với giao diện thân thiện, đẹp mắt, đơn giản hóa mọi chi tiết để dễ tiếp cận với người dùng.đặc biệt tương tích với nhiều loại,kích cỡ các thiết bị như Iphone,Ipad,tablet…

Các chức năng chính của website gồm:

* Xem thông tin chi tiết về nhiều loại sách(lập trình, giải thuật,tiếng anh chuyên ngành.. ),
* Xem thông tin chi tiết tin tức,lịch thi, hoạt động sinh viên,thông tin đào tạo, tin tức công nghệ.
* Đăng nhập.
* Tìm kiếm tin tức theo từ khóa
* Bình luận bằng ajax sau mỗi bài viết
* Chia sẽ tin tức mạng xã hội
* Xem thông tin chi tiết tuyển dụng,thực tập
* Xem bản đồ các cơ sở học
* Gửi thông tin yêu cầu liên hệ lên văn phòng khoa có đính kèm file
* Xem thông tin sinh viên
* Đổi mật khẩu
* Xem,download chương trình đào tạo, đề cương môn học
* Xem thông tin giảng viên, bàng vàng thành tích sinh viên,cưu sinh viên
* Hiển thị thống kê số lượt truy cập website.
* quản trị tin tức, loại tin, tin tuyển dụng,sách,loại sách,liên hệ,slide, cấp quyền cho user,
* Quản trị thông tin user,sinh viên,cựu sinh viên,giảng viên,lượt truy cập.

1. **Các trang của website**
   1. **các trang giao diện người dùng**
      1. Trang Chủ



Hình 14*:Giao diện trang chủ*

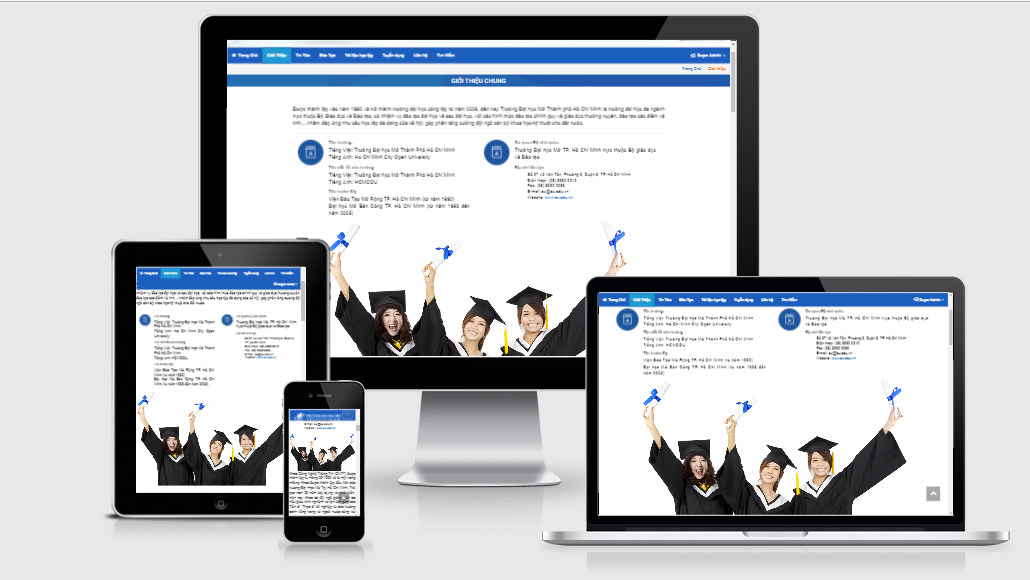
Hiển thị các tin tức mới nhất về lịch thi, thông tin đào tạo, tuyển dụng . bên cạnh đó hiển thị các liên kết bên ngoài, các tiện ích giành cho sinh viên, liên kết các khoa khác, phần cuối thống kê lượt truy cập,tổng số lượt click.có form tìm kiếm từ khóa co người dùng nhập vào, phái dưới footer là thông tin liên hệ cơ bản về khoa cũng như nút cuộn trang trở về đầu khi người dùng keo xuống dưới cùng.



Hình *:Trang chủ hiển thị trên nhiều loại màn hình thiết bị*

* + 1. Trang Giới thiệu

Trang giới thiệu, hiển thị chung về khoa, các ngành học.



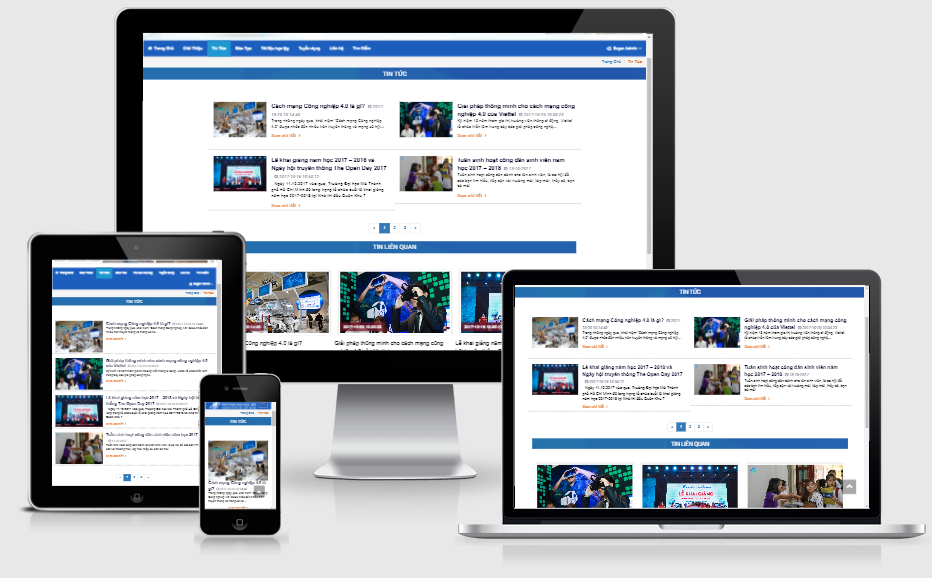
Hình 16:Trang giới thiệu

* + 1. Trang tin tức

Trang tin tức ,hiển thị các tin theo loại:

* Tin tức tổng hợp
* Lịch thi
* Thông tin đào tạo
* Hoạt động sinh viên
* Tin tức khoa học và công nghệ

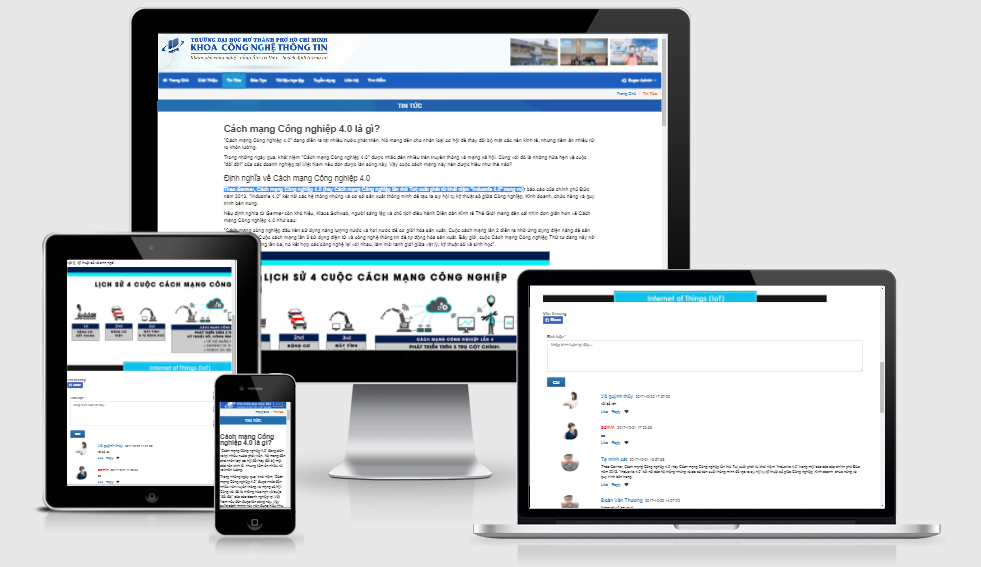
Ngoài ra còn có phân trang, và hiện các tin liên quan theo loại phía dưới, có chức năng rê chuột vào ảnh sẽ phóng to ảnh, giúp người dùng có thể xem rõ hình ảnh



Hình 17*:Trang tin tức*

* + 1. Trang chi tiết tin

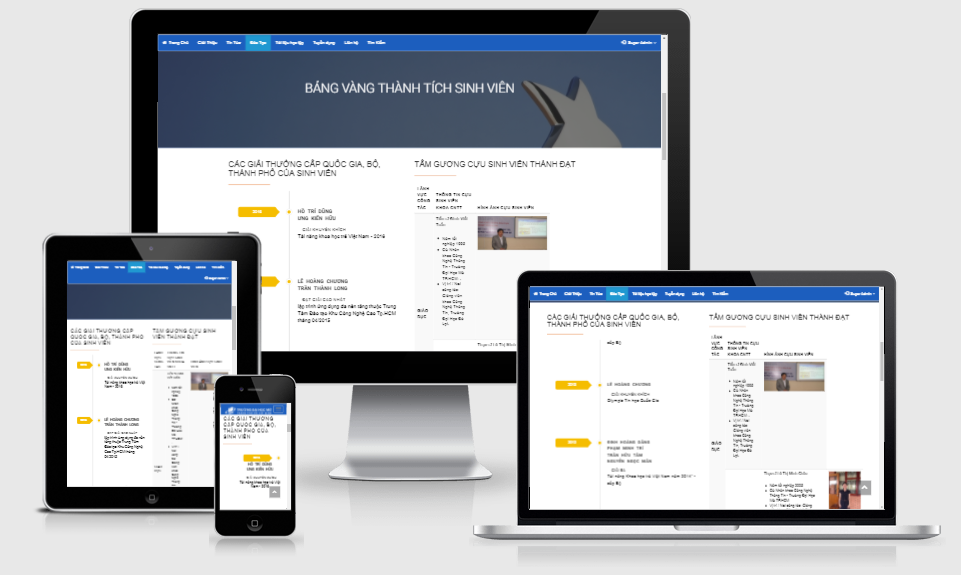
Trang chi tiết tin, hiển thị chi tiết một tin , ngoài ra cho phép user bình luận và nút share lên mạng xã hội facebook.khi người dùng nhấp vào bình luận, website sẽ kiểm tra người dùng đã đăng nhập hay chưa, nếu chưa sẽ hiện ra thông báo nhắc nhỡ người dùng phải đăng nhập. Phía dưới hiển thị danh sách các tin liên quan.



Hình 18*:Trang chi tiết tin trên nhiều loại màn hình thiết bị*

* + 1. Trang đào tạo

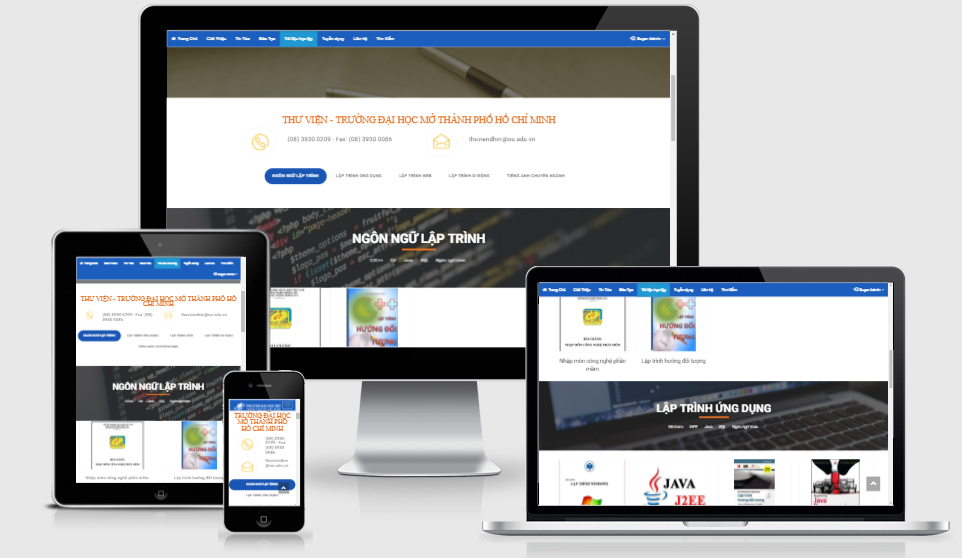
Trang đào tạo hiễn thi chương trình đào tạo, đề cương môn học,biểu mẫu, chuẩn đầu ra . Ngoài ra còn có bảng vàng thành tích được thường xuyên cập nhật,thông tin cựu sinh viên thành đạt, cũng như danh sách giảng viên.



Hình 19*:Trang đào tạo*

* + 1. Trang sách

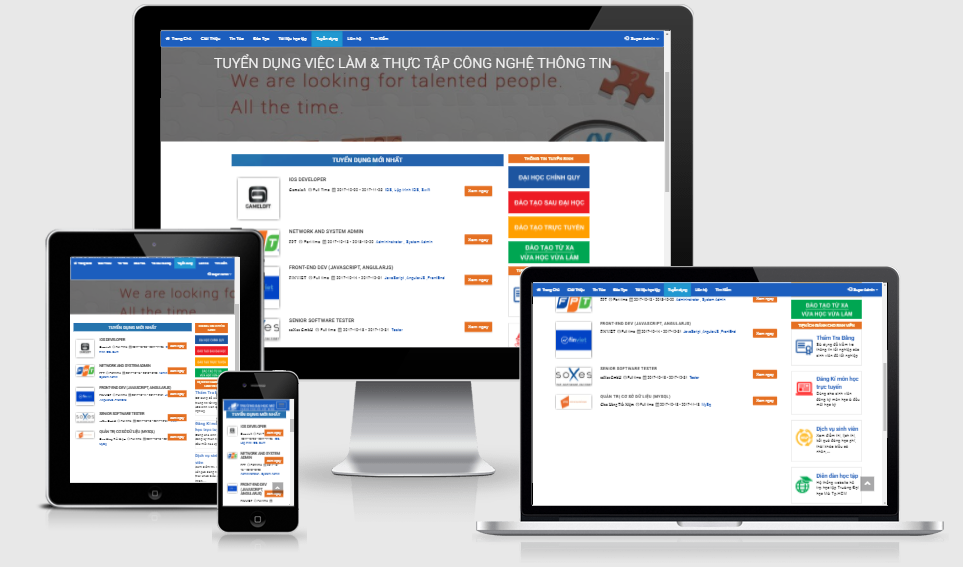
Trang sách, sách được lọc theo từng loại như ngôn ngữ lập trình,Lập trình web, lập trình trên di động,tài liệu tiếng anh chuyên ngành.. Có chức năng phóng to hình khi di chuyển chuột vào hình giúp người dùng xem rõ hình của quyển sách hơn.



Hình 20:Trang sách

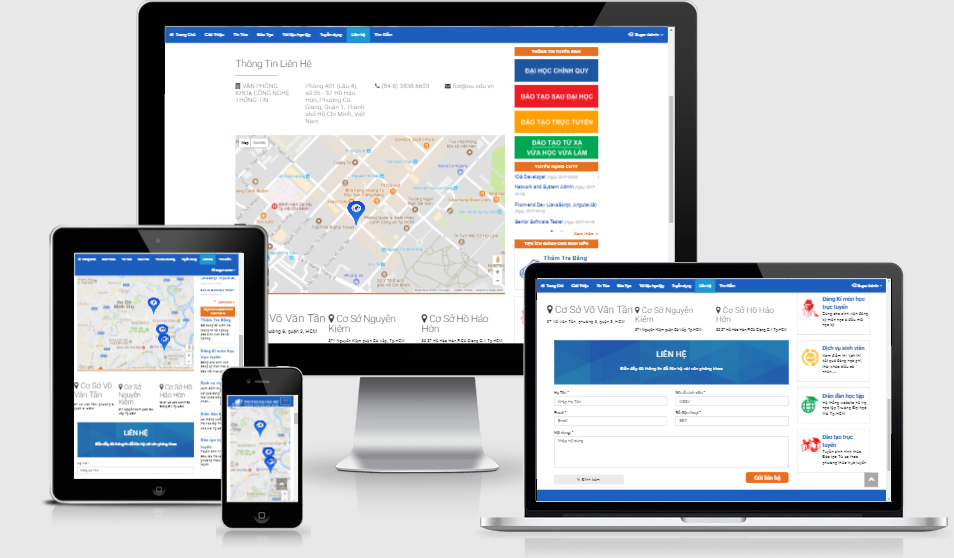
* + 1. Tuyển dụng

Hiển thị các thông tin tuyển dụng mới nhất, được cập nhật thường xuyên với logo công ty, tên công việc , ngày bắt đầu , ngày hết hạn…



Hình *:Trang tuyển dụng hiển thị trên nhiều loại màn hình thiết bị*

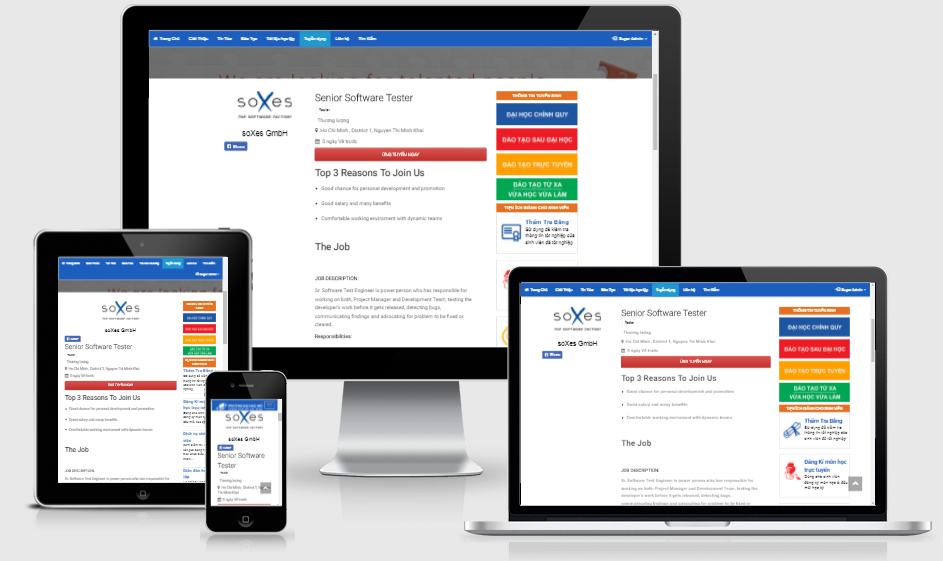
* + 1. Liên hệ



Hình 22*:Trang liên hệ*

Trang liên hệ , hiển thị bản đồ các cơ sở, có đường link mỗi khi nhấp vào bản đồ sẽ trỏ đến chi tiết cơ sở đó, ngoài ra phía dưới còn có form đăng nhập để lấy yêu cầu thông tin của người dùng để gửi đến văn phòng khoa có đính kèm file. người dùng có thể vào website gửi yêu cầu mà không cần phải đăng nhập

* + 1. Trang chi tiết tuyển dụng



Hình *:Trang chi tiết tin tuyển dụng*

Thể hiện chi tiết ,yêu cầu, cũng như chi tiết công việc của công ty đó, có kèm nút share. ngoài ra còn hiển thị các tin tuyển dụng liên quan, nút ứng tuyển sẽ đi đến trang web tuyển dụng hoặc công ty tuyển dụng.

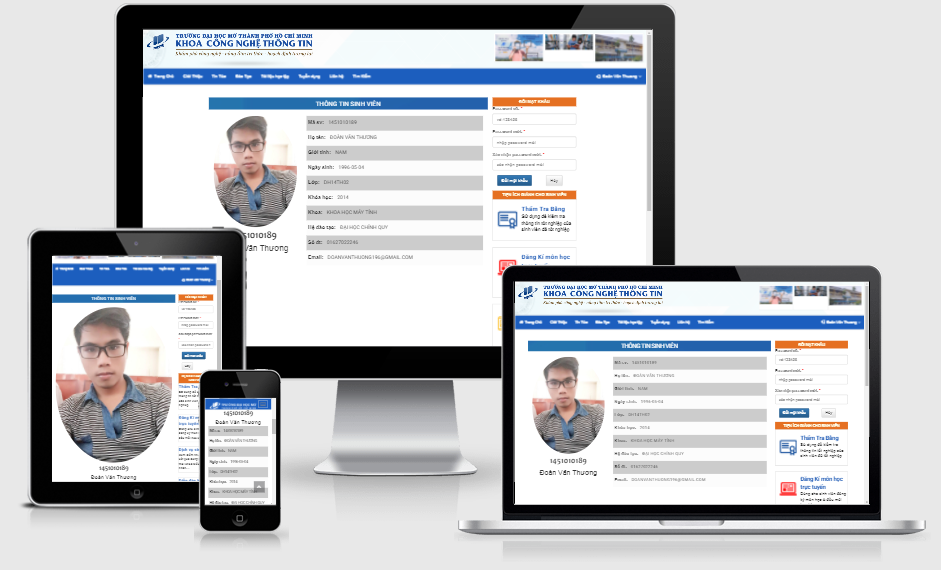
* + 1. Trang đăng nhập



Hình 24*:Trang đăng nhập*

Trang đăng nhập cho phép user đăng nhập, có kiểm soát lỗi. Nếu người đang nhập là admin thì sẽ đến trang quản trị ,ngược lại sẽ vào trang thông tin sinh viên

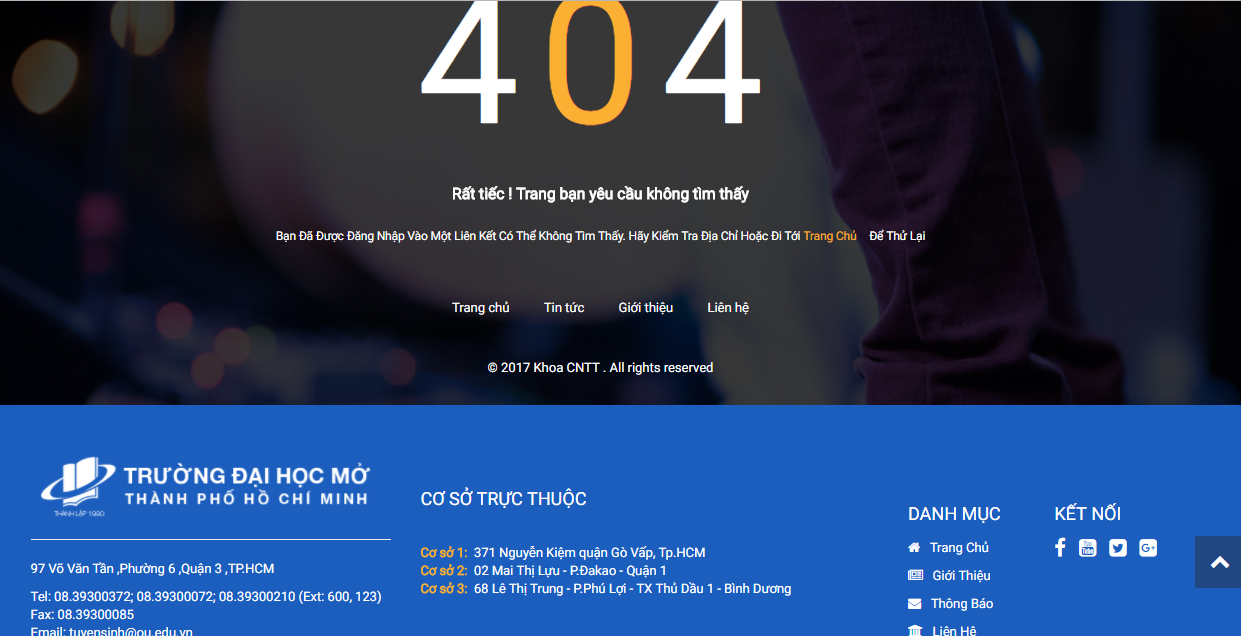
* + 1. Trang thông tin sinh viên



Hình 25*:Trang thông tin sinh viên*

Hiển thị thông tin về sinh viên như: tên, mã số, họ tên. lớp, điện thoại, emai…Ngoài ra còn có form đổi mật khẩu

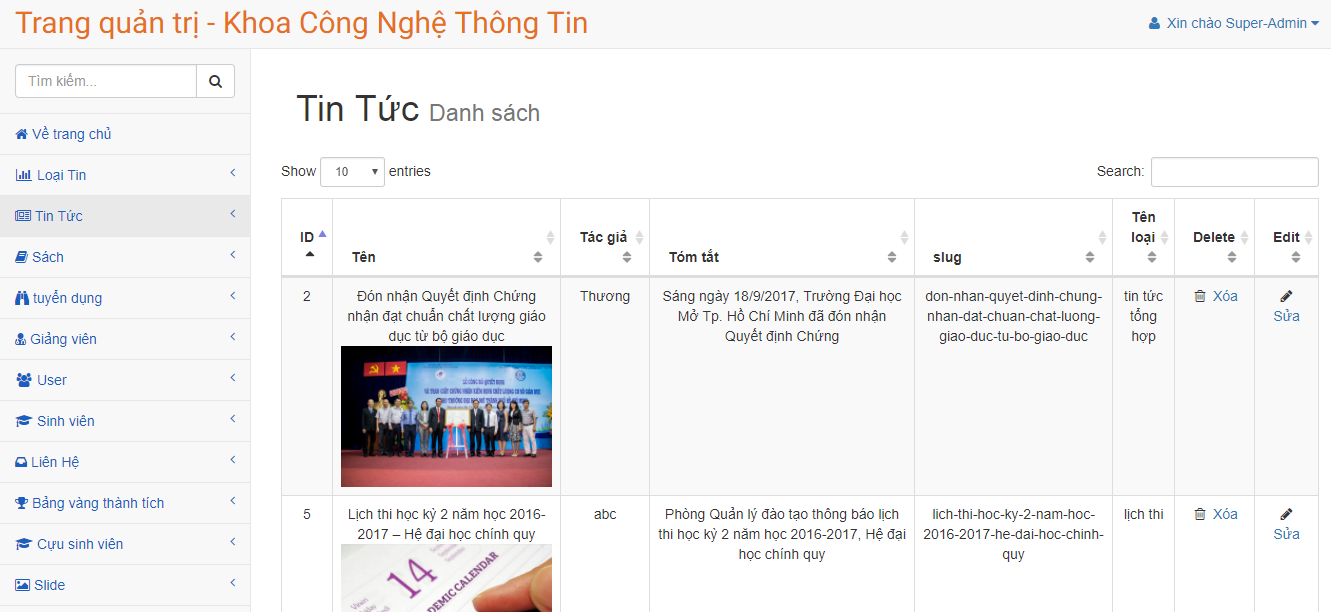
* + 1. Trang 404



Hình 26*:Trang lỗi 404*

Trang thông báo khi người yêu cầu một trang không tồn tại hoặc gõ sai đường dẫn url

* 1. **Giao diện trang quản trị**



Hình 27*:Trang quản trị*

Cho phép user có quyền admin đăng nhập, quản trị tất cả các thông tin trong trang quản trị trừ user. Chỉ có user có quyền super admin ( quyền tối thượng) mới vào được phần quản lý user.Tại đây ta có thể thêm, xóa , sắp xếp nhiều nội dung trên trang như :

* Thông tin sinh viên
* Tin tức
* Sách
* Giảng viên
* Slide ảnh trên các trang
* Xem người dùng gửi liên hệ
* …

*Hình 4.10: Giao diện trang thanh toán*

Giao diện quản trị

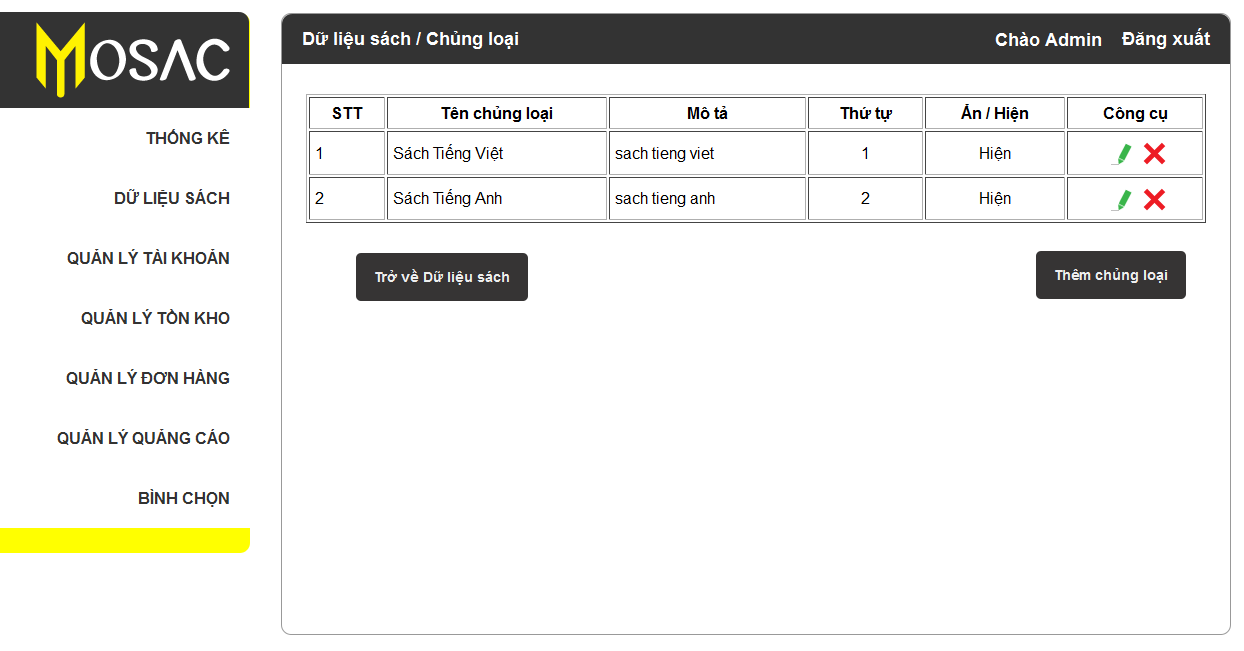
1. Trang đăng nhập

*Hình 4.12: Giao diện trang chủ phần quản trị*

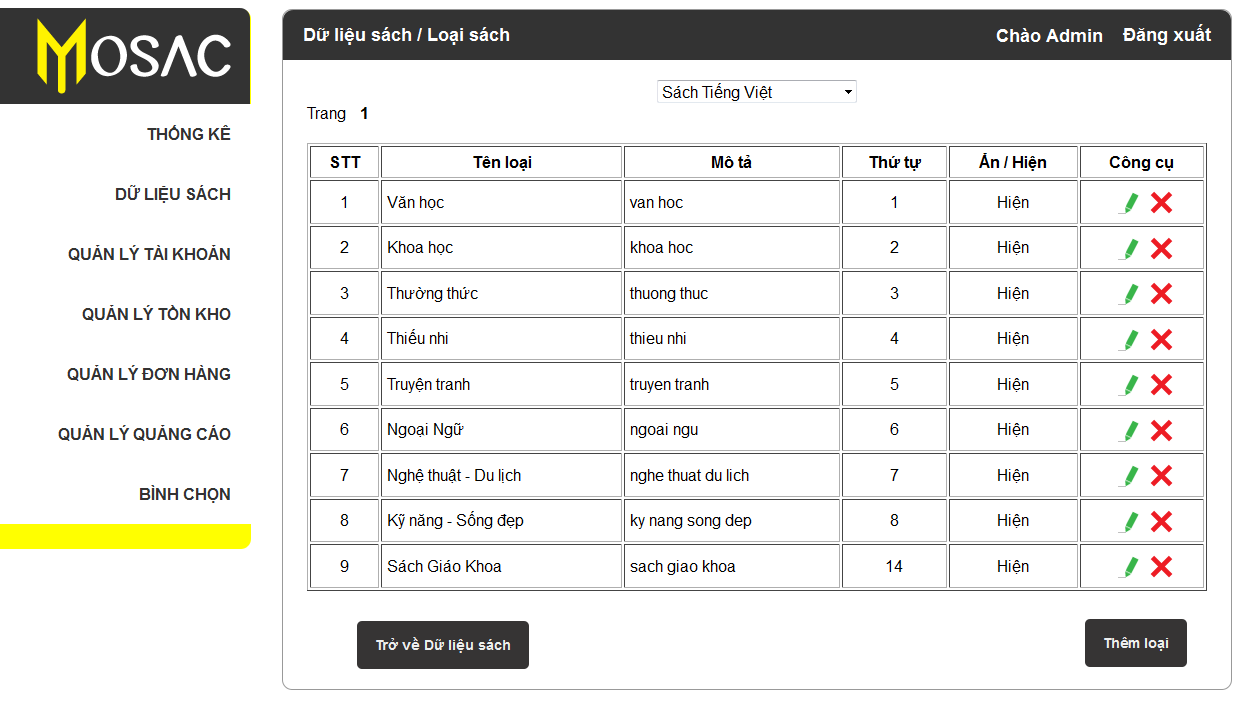
Trang quản lý dữ liệu sách

Phần quản lý dữ liệu sách trong website gồm có quản lý chủng loại sách, quản lý loại sách, quản lý các nhà xuất bản và quản lý các sách điện tử.

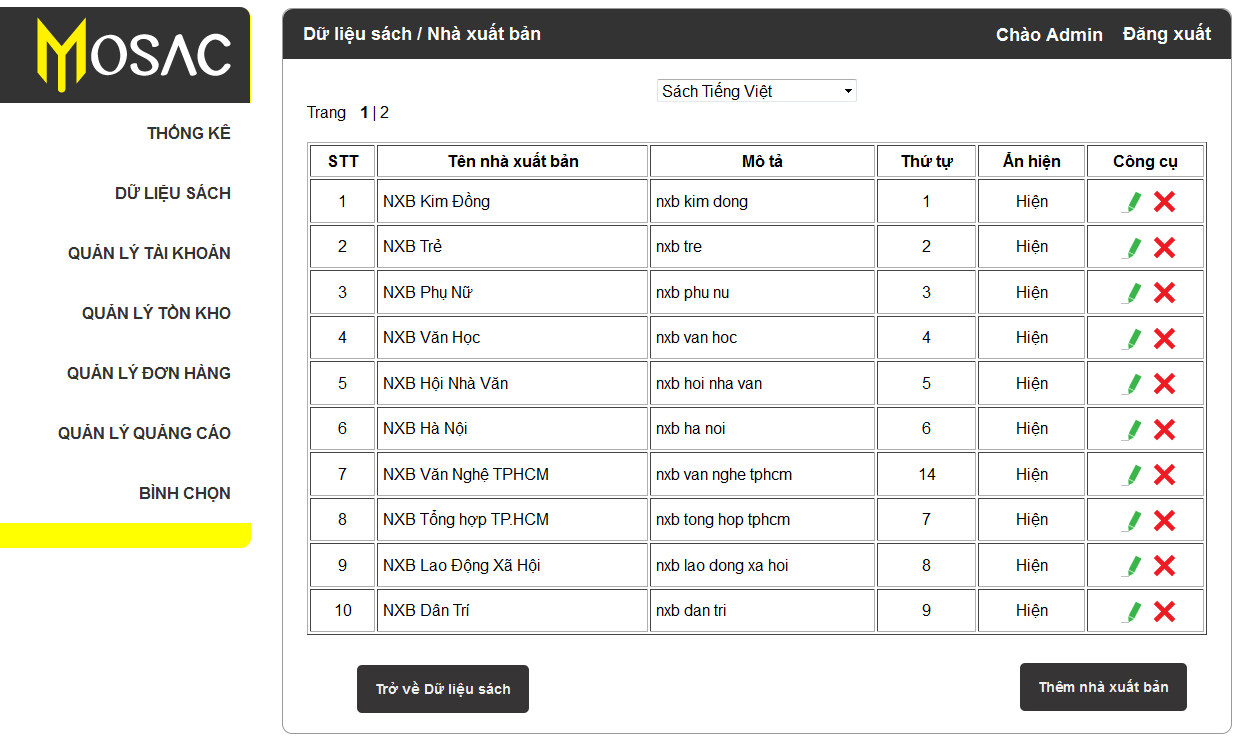
Trong mỗi trang quản lý dữ liệu sách đều có hiển thị thông tin chi tiết về từng dữ liệu mà người dùng chọn như chủng loại sách, loại sách… Ngoài ra, trong mỗi loại dữ liệu sách còn có các nút thêm, xóa, chỉnh sửa.



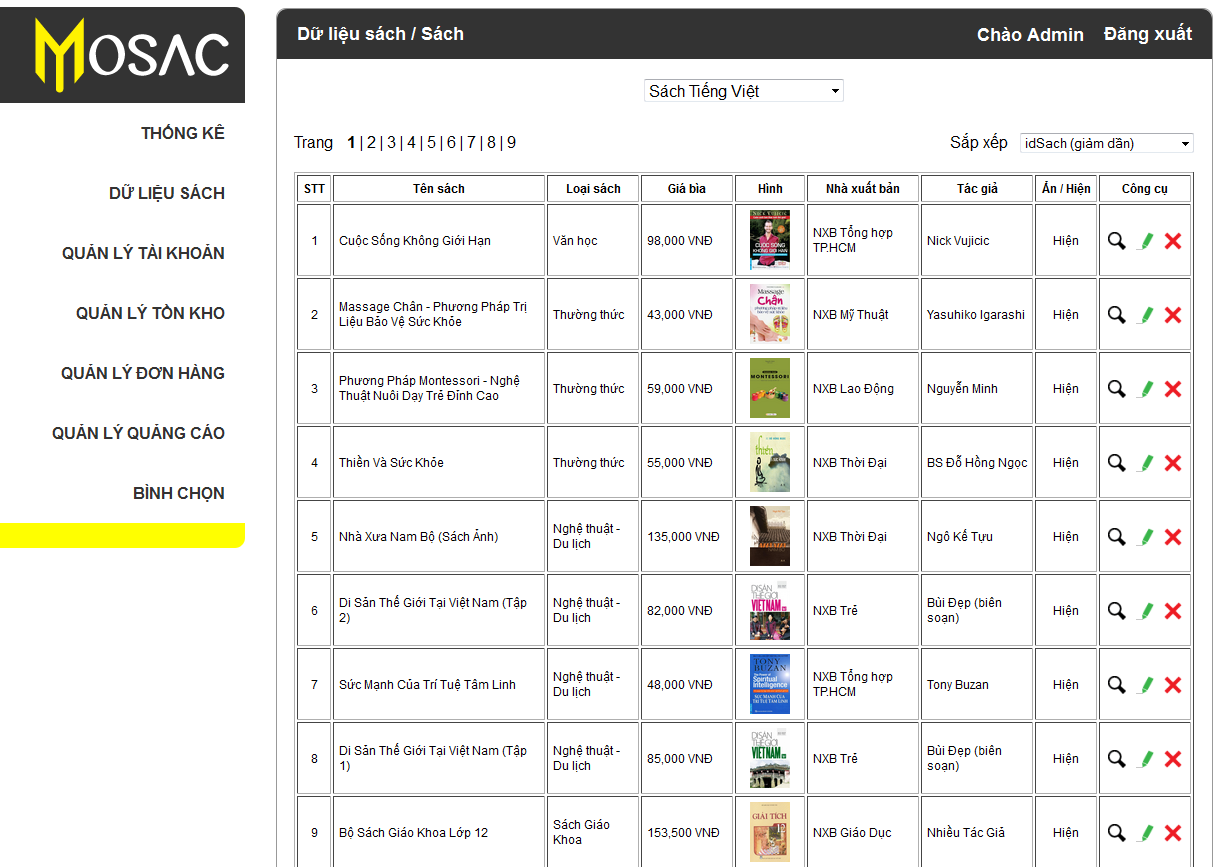
*Hình 4.13: Giao diện trang quản lý chủng loại sách*



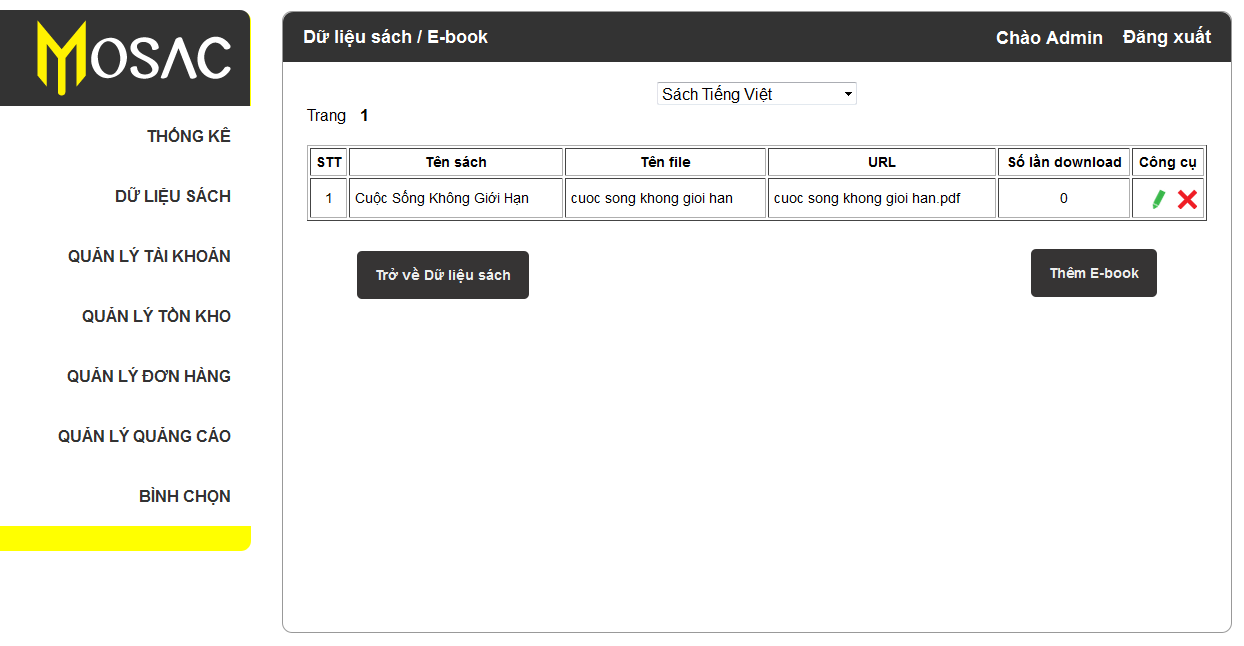
*Hình 4.14: Giao diện trang quản lý loại sách*



*Hình 4.15: Giao diện trang quản lý nhà xuất bản*



*Hình 4.16: Giao diện trang quản lý thông tin sách*



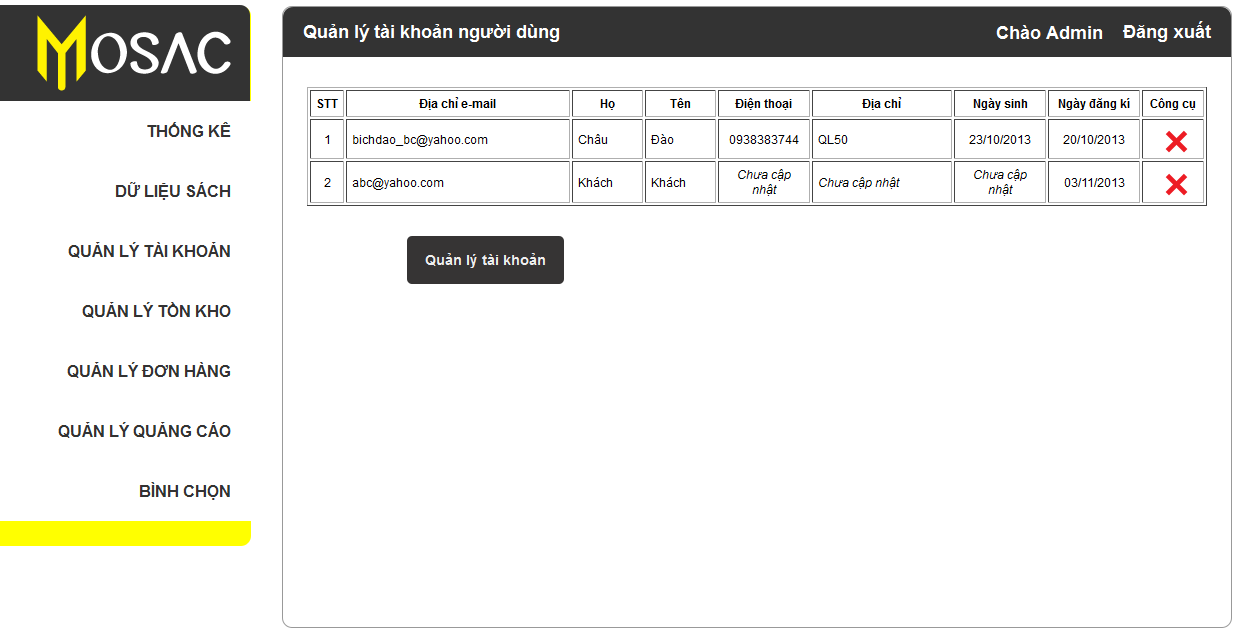
*Hình 4.17: Giao diện trang quản lý sách điện tử*

Trang quản lý tài khoản

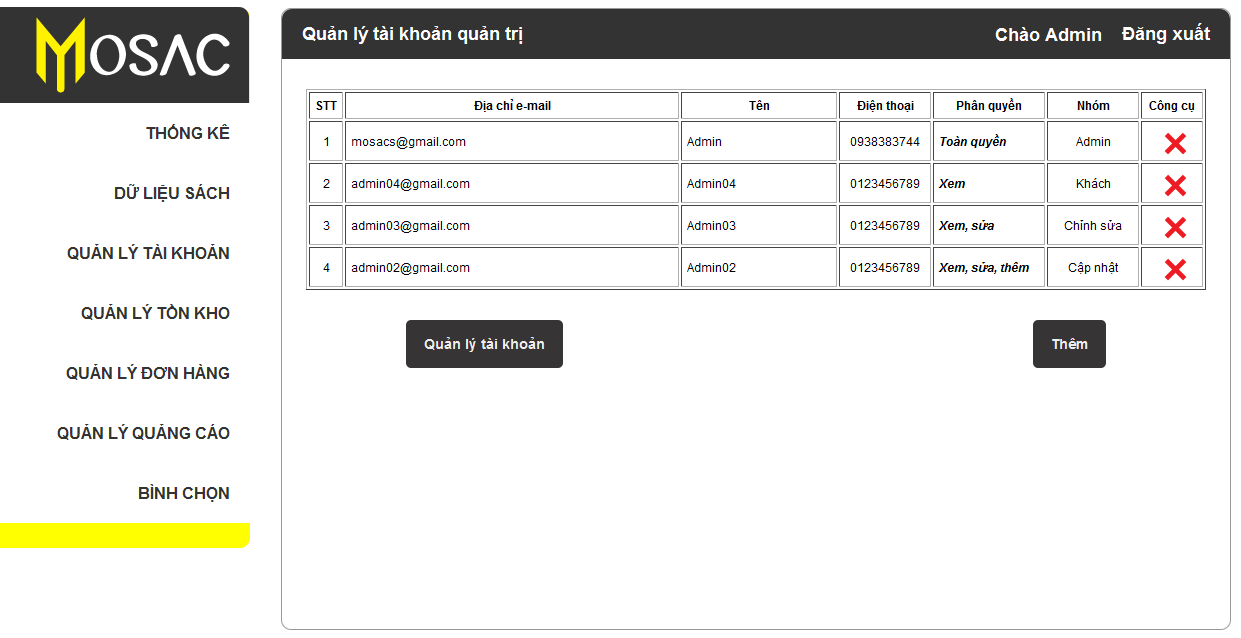
Người quản trị có thể quản lý cả tài khoản của người dùng lẫn tài khoản của người quản trị tại trang quản lý tài khoản.

Đối với tài khoản người dùng, người quản trị chỉ có quyền xóa tài khoản đó chứ không được thêm và sửa. Vì đây là thông tin của người dủng khi đăng ký tài khoản nên nếu muốn sửa thông tin tài khoản thì người dùng sẽ đăng nhập và thực hiện cập nhật thông tin tài khoản. Bên cạnh đó, nếu người quản trị muốn tạo một tài khoản người dùng, người quản trị có thể vào giao diện website dành cho khách hàng để tạo tài khoản mới.

Đối với các tài khoản quản trị website, người quản trị chính có thể tạo tài khoản mới và cấp quyền cho các tài khoản đó. Để đảm bảo mỗi người quản trị viên chỉ có thể thực hiện những việc trong giới hạn cho phép, người quản trị chính cần phân quyền cho các tài khoản quản trị viên một cách chặt chẽ nhất.



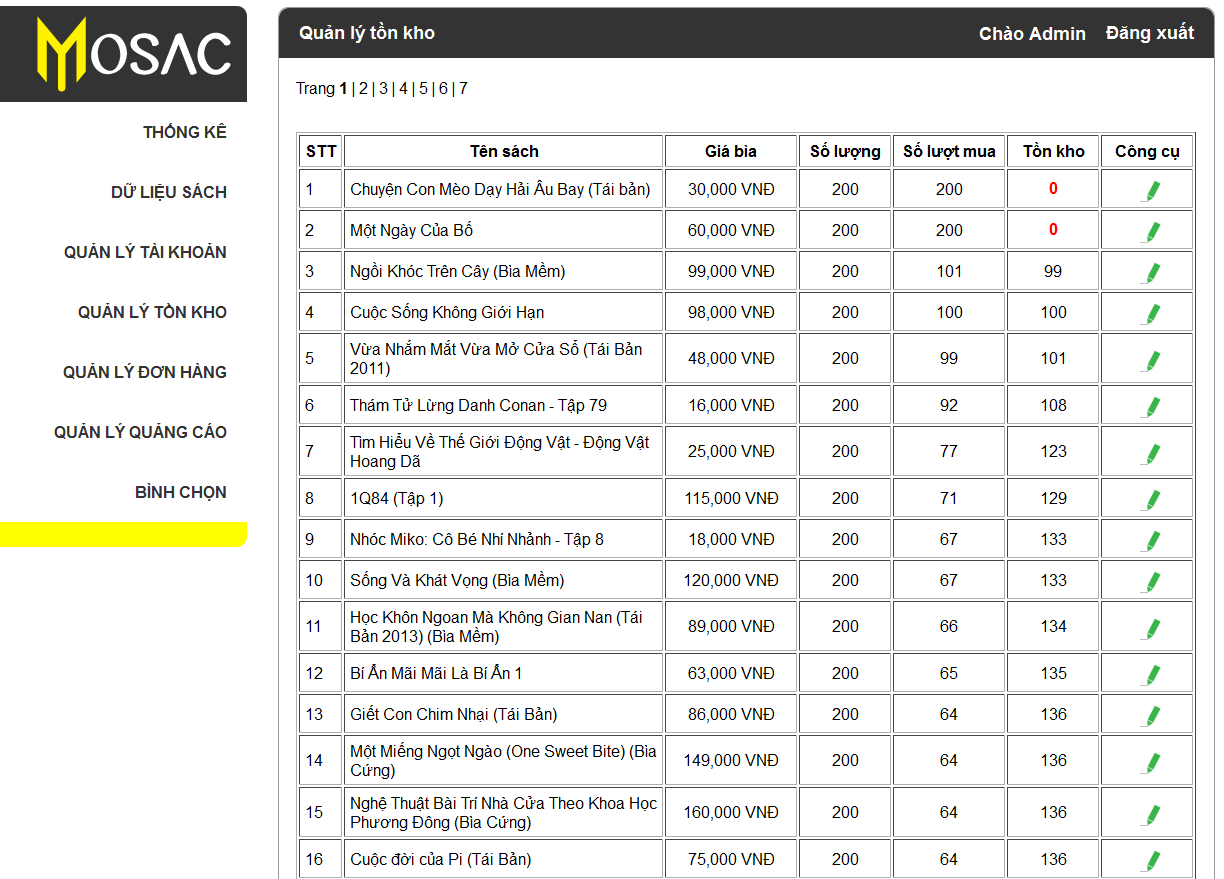
*Hình 4.18: Trang quản lý tài khoản người dùng*



*Hình 4.19: Giao diện trang quản lý tài khoản quản trị*

Trang quản lý tồn kho

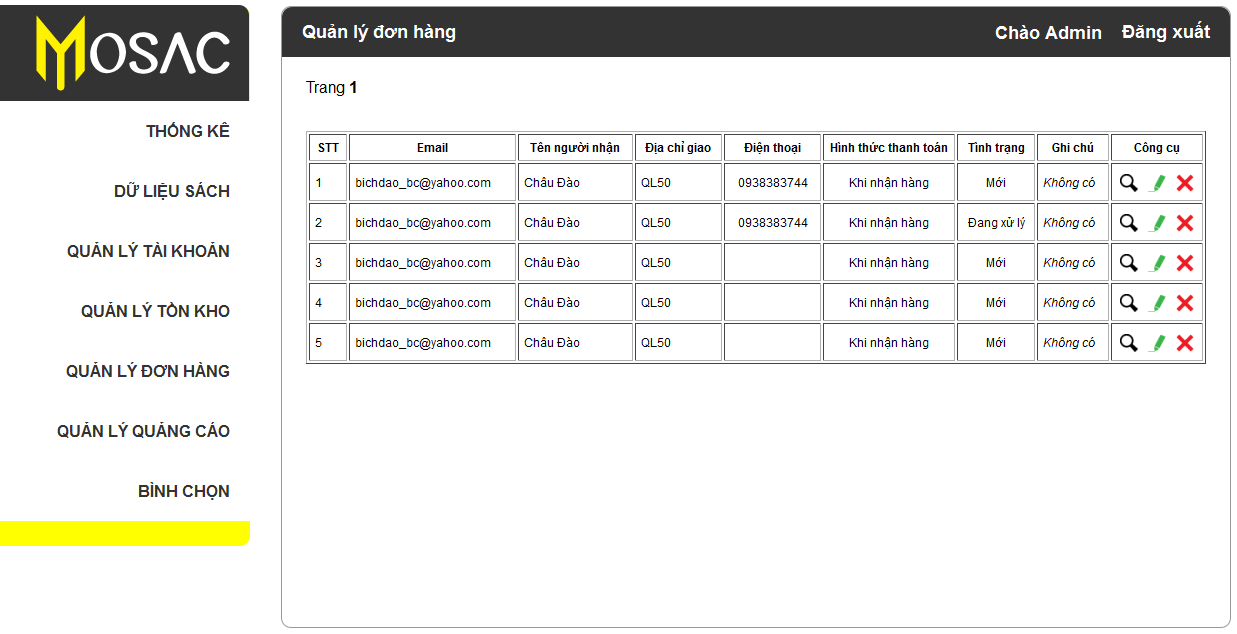
Đây là trang quản lý số lượng của từng loại sách. Trang này sẽ cung cấp thông tin để người quản trị nắm rõ và nhập thêm các quyển sách còn lại với số lượng ít.



*Hình 4.20: Giao diện trang quản lý sách tồn kho*

Trang quản lý đơn hàng

Trang quản lý đơn hàng sẽ hiển thị các thông tin đơn hàng để người quản trị hay bộ phận đóng gói chuẩn bị các quyển sách trong đơn hàng. Ngoài ra, người quản trị có thể xem chi tiết từng đơn hàng để gọi điện thoại xác nhận với khách hàng về đơn hàng đã đặt mua, sau đó chuyển đơn hàng cho bộ phận đóng gói và vận chuyển.



*Hình 4.21: Giao diện trang quản lý đơn hàng*

CHƯƠNG V: TỔNG KẾT

Những hình thức kinh doanh thương mại điện tử đã trở nên khá phổ biến, được triển khai trong hầu hết các doanh nghiệp và đem lại doanh thu đáng kể. Cùng với lượng người sử dụng Internet và thẻ tín dụng tăng nhanh, số lượng khách hàng mua sắm qua mạng cũng theo đó tăng lên nhanh chóng. Đó là nguồn khách hàng lớn nhất của các doanh nghiệp triển khai thương mại điện tử.

Website thương mại điện tử lả công cụ hỗ trợ tốt nhất cho các doanh nghiệp muốn quảng bá thông tin về các sản phẩm của mình cho khách hàng. Từ website thương mại điện tử, doanh nghiệp có thể khảo sát nhu cầu thị trường, nghiên cứu thói quen mua sắm của khách hàng, bán hàng trực tuyến, cung cấp các dịch vụ hỗ trợ khách hàng… Hiện nay, hàng hóa và các dịch vụ được cung cấp trên website thương mại điện tử vô cùng phong phú và đa dạng. Các website này hoàn toàn có thể đáp ứng mọi nhu cầu mua sắm của người dùng.

Với thương mại điện tử, doanh nghiệp đã tận dụng triệt để các lợi ích từ Internet để giảm thiểu các chi phí phát sinh khi thực hiện mua bán theo kiểu truyển thống. Việc triển khai thương mại điện tử giúp các doanh nghiệp lôi kéo và thu hút sự quan tâm của khách hàng thông qua khả năng đáp ứng được các nhu cầu của khách hàng. Nhờ vào Internet, doanh nghiệp có thể cung cấp các thông tin, bảng báo giá, các chương trình khuyến mãi đến nhiều đối tượng khách hàng một cách nhanh chóng, hiệu quả.

Sự phát triển nhanh chóng của thương mại điện tử đã tạo ra thị trường rộng lớn cho các doanh nghiệp trung gian. Trong khi các doanh nghiệp trực tiếp cung cấp các sản phẩm do mình sản xuất đến người tiêu dùng thì các doanh nghiệp trung gian sẽ thu thập thông tin về nhu cầu của người dùng và nhập các mặt hàng của nhiều nhà cung cấp khác nhau để đáp ứng hầu hết các nhu cầu của khách hàng. Do đó, thông qua các doanh nghiệp trung gian, người tiêu dùng có thể lựa chọn nhiều loại sản phẩm của nhiều nhà cung cấp khác nhau.

Bên cạnh rất nhiều lợi ích đem lại cho doanh nghiệp và người tiêu dùng, thương mại điện tử vẫn còn tồn tại một số hạn chế nhất định. Các doanh nghiệp cần có kế hoạch thích hợp để tối thiểu hóa ảnh hưởng từ các hạn chế của thương mại điện tử đến hoạt động kinh doanh của mình. Vấn đề được doanh nghiệp và người tiêu dùng quan tâm nhiều nhất là vấn đề bảo mật trong thương mại điện tử. Khi thực hiện giao dịch thông qua mạng Internet, cả người dùng và doanh nghiệp đều đối mặt với nguy cơ về an toàn bảo mật thông tin vì thông tin truyền qua mạng Internet có thể bị lấy cắp, trong đó có thể có thông tin cá nhân của khách hàng và doanh nghiệp. Những kẻ xấu có thể lợi dụng những lổ hỗng về bảo mật trên website của doanh nghiệp để lấy cắp cơ sở dữ liệu chứa toàn bộ dữ liệu của website và các thông tin của khách hàng.

Như vậy, thương mại điện tử tạo ra rất nhiều lợi ích đối với cả doanh nghiệp và người tiêu dùng. Để áp dụng thương mại điện tử vào các hình thức kinh doanh, doanh nghiệp cần phải có chiến lược cụ thể và có thể triển khai theo từng giai đoạn. Tuy nhiên, thương mại điện tử vẫn còn tồn tại những hạn chế mà doanh nghiệp và khách hàng cần lưu ý. Doanh nghiệp cần phải dự tính những rủi ro của thương mại điện tử để hoàn thiện chiến lược kinh doanh. Người tiêu dùng cũng cần cảnh giác khi thực hiện các giao dịch qua Internet, kiểm tra kỹ các doanh nghiệp có uy tín và đảm bảo an toàn.

TÀI LIỆU THAM KHẢO

1. TS. Nguyễn Hoài Anh, ThS. Ao Thu Hoài, Thương mại điện tử, NXB Thông tin và Truyền thông – 2011.
2. TS. Bùi Văn Danh, Thương mại điện tử, NXB Phương Đông – 2011.
3. PHP Manual, <http://php.net>.
4. W3Schools, http://www.w3schools.com.