# LỜI MỞ ĐẦU

Ngành công nghệ thông tin là một ngành khoa học đang trên đà phát triển mạnh và ứng dụng rộng rãi trên nhiều lĩnh vực. Cùng với xu hướng phát triển của các phương tiện truyền thông như Báo, Radio… thì việc sử dụng Internet ngày càng phổ biến. Truy cập Internet, chúng ta có được một kho thông tin khổng lồ phục vụ mọi nhu cầu, mục đích của chúng ta chỉ bằng một cái nhấp chuột.

Nhận thức được nhu cầu trao đổi thông tin, sự kiện, với sự ra đời của hàng loạt website cho các mục đích thương mại, giải trí, tin tức…Để đáp ứng với việc cập nhật thông tin hàng ngày,… thì Website Cổng thông tin trường học tin tức ra đời là một nhu cầu tất yếu của mỗi trường học.

Do đó, chúng em đã vận dụng ngôn ngữ PHP, MySQL và công cụ Visual Studio Code để xây dựng ứng dụng Website Cổng thông tin trường học.

Hoàn thành xong để tài, chúng em vô cùng biết ơn thầy, cô **TRƯỜNG CAO ĐẲNG KINH TẾ - KỸ THUẬT CẦN THƠ** đã nhiệt tình giảng dạy, hướng dẫn và đặc biệt là thầy **VÕ HOÀNG TÚ**, người trực tiếp hướng dẫn nhiệt tình cho chúng em trong suốt quá trình nghiên cứu và thực hiện đề tài này.

Và chúng em cũng xin chân thành cảm ơn đến các anh chị tại **CÔNG TY TNHH PHÁT TRIỂN VÀ CHUYỂN GIAO PHẦN MỀM DTSOFT** đã tạo điều kiện cho em thực tập và cung cấp một số tài liệu liên quan để em hoàn thành bài báo cáo. Đồng thời, em xin gởi lời cảm ơn đến anh **NGUYỄN QUỐC THỐNG** đã tận tình hướng dẫn em trong suốt thời gian thực tập.

Chúng em xin chân thành cảm ơn!

*Cần Thơ, Ngày tháng năm 2019*

*Nhóm sinh viên thực hiện*

Lê Vũ Lâm  
Nguyễn Văn Quốc

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

🙢🕮🙠

|  |  |
| --- | --- |
|  | **Cần Thơ**, Ngày….Tháng….Năm 2019 |
|  | **GIẢNG VIÊN HƯỚNG DẪN** |

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

🙢🕮🙠

|  |  |
| --- | --- |
|  | **Cần Thơ**, Ngày….Tháng….Năm 2019 |
|  | **GIẢNG VIÊN HƯỚNG DẪN** |

# DANH MỤC HÌNH ẢNH

Hình 4.1: Đăng nhập hệ thống

Hình 4.2: Thêm người dùng

Hình 4.3: Sửa thông tin người dùng

Hình 4.4: Danh sách người dùng

Hình 4.5: Danh sách bài viết

Hình 4.6: Thêm bài viết

Hình 4.7: Danh sách và thêm loại tin

Hình 4.8: Danh sách trang

Hình 4.9: Thêm trang

Hình 4.10: Quản lý ảnh

Hình 4.11: Thêm và chỉnh sửa menu

Hình 4.12: Danh sách slide

Hình 4.13: Thêm slide

Hình 4.14: Sửa tiêu đề, Miêu tả, footer

Hình 4.15: Sửa Icon, Logo, Banner

Hình 5.1: Banner, Menu và Slide trang web

Hình 5.2: Giao diện tin tức theo loại tin và sidebar

Hình 5.3: Giao diện tin tức theo loại tin và sidebar

Hình 5.4: Danh sách bài viết theo loại tin

Hình 5.5: Chi tiết bài viết

Hình 5.6: Trang giới thiệu nhà trường

# MỤC LỤC

[**LỜI MỞ ĐẦU** 1](#_Toc13711399)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN** 2](#_Toc13711400)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN** 3](#_Toc13711401)

[**MỤC LỤC** 4](#_Toc13711402)

[1.1 Tổ chức hành chính, nhân sự 8](#_Toc13711403)

[1.2 Hoạt động chuyên ngành và môi trường làm việc 8](#_Toc13711404)

[1.2.1 Hoạt đông chuyên ngành 8](#_Toc13711405)

[1.2.2 Môi trường làm việc 9](#_Toc13711406)

[**CHƯƠNG 2: GIỚI THIỆU TỔNG QUAN** 10](#_Toc13711407)

[2.1 Khái niệm về Website 10](#_Toc13711408)

[2.2 Đặt vấn đề 10](#_Toc13711409)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU** 11](#_Toc13711410)

[3.1 Tổng quan 11](#_Toc13711411)

[3.2 Cơ sở lý thuyết 11](#_Toc13711412)

[3.2.1 Khái niệm dữ liệu và cơ sở dữ liệu 11](#_Toc13711413)

[3.2.2 Các mức biểu diển một cơ sở dữ liệu 12](#_Toc13711414)

[3.2.3 Mô hình phân rã chức năng (Bussiness Function Diagram\_BFD) 12](#_Toc13711415)

[3.2.3.1 Giới thiệu mô hình phân rã chức năng 12](#_Toc13711416)

[3.2.3.2 Các thành phần của mô hình phân rã chức năng 13](#_Toc13711417)

[3.2.3.3 Các bước xây dựng mô hình phân rã chức năng 13](#_Toc13711418)

[3.2.4 Mô hình luồng dữ liệu 13](#_Toc13711419)

[3.2.4.1 Giới thiệu mô hình luồng dữ liệu 13](#_Toc13711420)

[3.2.4.2 Các thành phần của mô hình luồng dữ liệu 14](#_Toc13711421)

[3.2.4.3 Các bước xây dựng mô hình luồng dữ liệu 15](#_Toc13711422)

[3.2.5 Mô hình thực thể kết hợp (Entity Relationshop Diagram\_ERD) 15](#_Toc13711423)

[3.2.5.1 Giới thiệu mô hình thực thể kết hợp 15](#_Toc13711424)

[3.2.5.2 Các thành phần của mô hình thực thể kết hợp 15](#_Toc13711425)

[3.2.5.3 Các quy tắt chuẩn hóa 17](#_Toc13711426)

[3.2.5.4 Các bước xây dựng mô hình thực thể kết hợp 17](#_Toc13711427)

[3.2.6 Mô hình quan hệ (Relational Data Model\_RDM) 17](#_Toc13711428)

[3.2.6.1 Giới thiệu mô hình quan hệ 17](#_Toc13711429)

[3.2.6.2 Các thành phần trong mô hình quan hệ 17](#_Toc13711430)

[3.2.6.3 Các quy tắt chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ 18](#_Toc13711431)

[3.3 Lý thuyết về công nghệ được sử dụng trong phần mềm 18](#_Toc13711432)

[3.3.1 Hệ quản trị cơ sở dữ liệu MySQL 18](#_Toc13711433)

[3.3.2 Laravel 18](#_Toc13711434)

[3.3.2.1 Giới thiệu PHP 18](#_Toc13711435)

[3.3.2.2 Giới thiệu laravel 19](#_Toc13711436)

[3.3.2.2 Ưu điểm của laravel 20](#_Toc13711437)

[3.3.2.3 Nhược điểm của Laravel 21](#_Toc13711447)

[3.3.2.4 Mô hình Model View Controller (MVC) 22](#_Toc13711453)

[3.3.2.5 Cấu trúc thư mục Laravel 23](#_Toc13711454)

[3.3.2.6 Route 24](#_Toc13711455)

[3.3.2.7 Controller 26](#_Toc13711458)

[3.3.2.8 View 27](#_Toc13711459)

[3.3.2.8 Blade template 27](#_Toc13711460)

[3.3.2.9 Database 29](#_Toc13711461)

[3.4 Phân tích yêu cầu 31](#_Toc13711462)

[3.4.1 Mô tả yêu cầu 31](#_Toc13711463)

[3.4.2 Các thông tin đầu vào 31](#_Toc13711464)

[3.4.3 Các thông tin đầu ra 31](#_Toc13711465)

[3.4.4 Các yêu cầu trang quản lý 31](#_Toc13711466)

[3.5 Xây dựng mô hình phân rã chức năng 31](#_Toc13711467)

[3.5.1 Quản lý Posts 31](#_Toc13711468)

[3.5.2 Quản lý Pages 32](#_Toc13711469)

[3.5.3 Quản lý Option 32](#_Toc13711470)

[3.5.4 Quản lý User 32](#_Toc13711471)

[3.5.5 Quản lý Category 32](#_Toc13711472)

[3.5.6 Quản lý Menu 33](#_Toc13711473)

[3.5.7 Quản lý Slides 33](#_Toc13711474)

[3.5.8 Thống kê 33](#_Toc13711475)

[3.5.9 Mô hình phân rã chức năng tổng hợp 34](#_Toc13711476)

[3.6 Xây dựng mô hình thực thể kết hợp 35](#_Toc13711477)

[3.7 Xây dựng mô hình quan hệ (RDM) 35](#_Toc13711478)

[3.8 Xây dựng các bảng 35](#_Toc13711479)

[**CHƯƠNG 4: CHỨC NĂNG VÀ GIẢI THUẬT** 38](#_Toc13711480)

[4.1 Các chức năng trang quản lý 38](#_Toc13711481)

[4.2 Giải thuật trang quản lý 39](#_Toc13711482)

[4.2.1 Đăng nhập hệ thống 39](#_Toc13711483)

[4.2.2 Thêm người dùng 40](#_Toc13711484)

[4.2.3 Thay đổi thông tin người dùng 40](#_Toc13711485)

[4.2.4 Danh sách người dùng 41](#_Toc13711486)

[4.2.6 Danh sách bài viết 41](#_Toc13711487)

[Hình 4.5: Danh sách bài viết 41](#_Toc13711488)

[4.2.7 Thêm bài viết 42](#_Toc13711489)

[4.2.8 Danh sách và thêm loại tin 42](#_Toc13711490)

[4.2.9 Danh sách trang 43](#_Toc13711491)

[4.2.10 Thêm trang 43](#_Toc13711492)

[4.2.11 File Manager (Quản lý ảnh) 44](#_Toc13711493)

[Hình 4.10: Quản lý ảnh 44](#_Toc13711495)

[4.2.12 Thêm và chỉnh sửa menu 44](#_Toc13711496)

[4.2.13 Danh sách slide 45](#_Toc13711497)

[4.2.14 Thêm slide 45](#_Toc13711498)

[4.2.15 Sửa Cài đặt 46](#_Toc13711499)

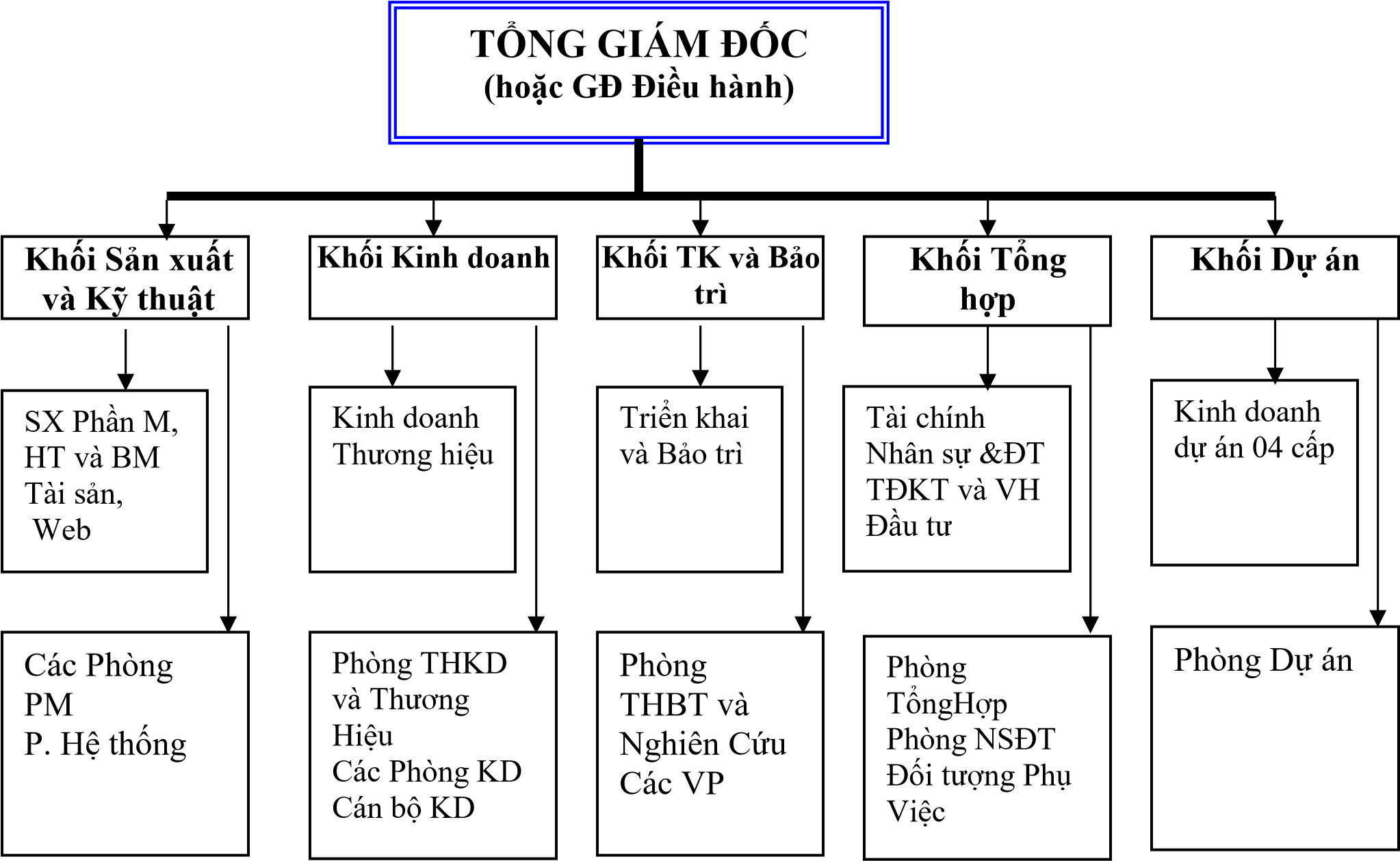
[**CHƯƠNG 5: GIAO DIỆN WEBSITE** 47](#_Toc13711501)

**CHƯƠNG 1: GIỚI THIỆU ĐƠN VỊ THỰC TẬP**

## 1.1 Tổ chức hành chính, nhân sự

Công ty TNHH Phát triển và Chuyển giao Phần mềm - DTSoft. Để thực hiện cam kết hiệu quả phần mềm 100% khi sử dụng cho khách hàng, Công ty đã tiến hành chia địa bàn hoạt động cho từng Văn phòng, tổng số có 32 Văn phòng. Mỗi văn phòng có từ 5 đến 7 cán bộ địa phương chịu trách nhiệm hỗ trợ và chăm sóc tốt các khách hàng trên địa bàn của mình. Sau đây là phân công hoạt động bảo trì và hỗ trợ của các Văn phòng DTSOFT tại 63 tỉnh thành trên toàn quốc.

-Cơ cấu tổ chức:



## 1.2 Hoạt động chuyên ngành và môi trường làm việc

### 1.2.1 Hoạt đông chuyên ngành

Công ty DTSoft hoạt động chuyên về thiết kế và chuyển giao các sản phẩm phần mềm, đặc biệt là các các phần mềm cho các cơ quan quản lý và các doanh nghiệp.

- Chức năng: Công ty DTSoft hoạt động chuyên về thiết kế và chuyển giao các phần mềm như: Kế toán, Nguồn lực, Giáo dục, Bệnh viện, Quan hệ khách hàng, Hành chính công, Văn phòng. Các sản phẩm đã chuyển giao cho nhiều tổ chức, đơn vị và đảm bảo hiệu quả.

- Nhiệm vụ: Cung cấp giải pháp và dịch vụ tổng thể trong công tác quản lý của các Cơ quan đơn vị cơ sở đến tổng hợp toàn bộ các báo cáo tại các đơn vị chủ quản.

Bằng kinh nghiệm thực tế trong công tác chuyển giao phần mềm của mình gần 20 năm qua, DTSoft đã xây dựng mạng lưới hỗ trợ tận nơi phủ kín cả nước mà các Công ty khác không thể có được, và rất vinh dự là đơn vị phần mềm duy nhất được Ban chỉ đạo Quốc gia về CNTT, Bộ Bưu chính Viễn thông, Bộ Khoa học và Công Nghệ, Hiệp hội Doanh nghiệp phần mềm Việt Nam Vinasa trao tặng GIẢI THƯỞNG SAO KHUÊ NĂM 2008 cho đơn vị có “ Dịch vụ hỗ trợ hậu mãi” tốt nhất.

### 1.2.2 Môi trường làm việc

- Công ty TNHH Phát triển và Chuyển giao Phần mềm - DTSoft có một môi trường làm việc năng động, hiện đại, với hệ thống cơ sở vật chất khang trang và đầy đủ thiết bị. Đội ngũ cán bộ tận tình, vững chuyên môn, nhiều kinh nghiệm trong công việc.

- Cán bộ trong công ty luôn nhiệt tình và sẵn sàng hỗ trợ trong quá trình học tập, nghiên cứu và thực hành thực tập.

# CHƯƠNG 2: GIỚI THIỆU TỔNG QUAN

## 2.1 Khái niệm về Website

Còn gọi là trang web (có thể nhầm lẫn với "*web page*") hoặc trang mạng, là một tập hợp trang web, thường chỉ nằm trong một [tên miền](https://vi.wikipedia.org/wiki/T%C3%AAn_mi%E1%BB%81n) hoặc [tên miền phụ](https://vi.wikipedia.org/wiki/T%C3%AAn_mi%E1%BB%81n_ph%E1%BB%A5) trên [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web) của [Internet](https://vi.wikipedia.org/wiki/Internet). Một trang web là tập tin [HTML](https://vi.wikipedia.org/wiki/HTML) hoặc [XHTML](https://vi.wikipedia.org/wiki/XHTML) có thể truy nhập dùng giao thức [HTTP](https://vi.wikipedia.org/wiki/Hypertext_Transfer_Protocol). Trang mạng có thể được xây dựng từ các tệp tin HTML (trang mạng tĩnh) hoặc vận hành bằng các [CMS](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_n%E1%BB%99i_dung) chạy trên máy chủ (trang mạng động).

Trang mạng có thể được xây dựng bằng nhiều ngôn ngữ lập trình khác nhau ([PHP](https://vi.wikipedia.org/wiki/PHP), [ASP.NET](https://vi.wikipedia.org/wiki/ASP.NET), [Java](https://vi.wikipedia.org/wiki/Java), [Ruby on Rails](https://vi.wikipedia.org/w/index.php?title=Ruby_on_Rails&action=edit&redlink=1), [Perl](https://vi.wikipedia.org/wiki/Perl),...).

Website được giao tiếp và hiển thị cho người dùng truy cập bằng các phần mềm được gọi là trình duyệt website. Một sô trình duyệt website nổi tiếng có thể kể đến như [Internet Explorer](https://vi.wikipedia.org/wiki/Internet_Explorer) được cài đặt mặc định vào mỗi máy tính cài hệ điều hành windows và được thay thế bởi Edge trên Windows 10 hay Chrome được phát triển bởi Google và Firefox được phát triển bởi Mozilla.

**2.2 Đặt vấn đề**

Từ những thập niên cuối thế kỷ XX đến nay, nhân loại đã phát minh và chứng kiến sự tiến triển thần kỳ của công nghệ thông tin. Những thành tựu của công nghệ thông tin đã góp phần rất quan trọng cho quá trình hình thành và phát triển nền kinh tế tri thức và xã hội thông tin mang tính chất toàn cầu. Công nghệ thông tin đóng vai trò quan trọng và không thể thiếu trong đời sống. Chính việc quản lý các công việc một cách hợp lý trên thiết bị máy tính bằng các chương trình quản lý, góp phần giúp cho công việc trở nên dễ dàng hơn.

Công nghệ thông tin càng phát triển thì con người càng có nhiều những phương pháp mới, công cụ mới để xử lý thông tin và nắm bắt được nhiều thông tin hơn. Công nghệ thông tin được ứng dụng trong mọi ngành nghề, mọi lĩnh vực của đời sống xã hội. Việc ứng dụng công nghệ thông tin trong quản lý, sản xuất, kinh doanh là một xu hướng tất yếu.

Kết quả của việc áp dụng công nghệ thông tin trong quản lý là việc hình thành các hệ thống thông tin quản lý nhằm phục vụ cho nhu cầu xử lý dữ liệu và cung cấp thông tin cho các chủ sở hữu hệ thống đó. Các cơ quan nhà nước áp dụng các ứng dụng tin học vào quản lý đang ngày một nhiều hơn, đa dạng hơn.

**Cổng thông tin trường học** là một website được viết dựa trên ngôn ngữ PHP (Framework Laravel) tương tác trên Hệ quản trị cơ sở dữ liệu MySQL giúp người dùng có thể xem tin tức, sự kiện, thông báo, thời khóa biểu,... của nhà trường cũng như giúp thông tin của nhà trường được đưa đến các thành viên trong nhà trường nhanh chóng, tiện lợi.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

**3.1 Tổng quan**

**Cơ sở dữ liệu** là một tập hợp các [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u_(m%C3%A1y_t%C3%ADnh)) có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

Mọi hệ thống đều phải sử dụng cơ sở dữ liệu của riêng mình và được lưu trữ trên một hệ quản trị cơ sở dữ liệu đó có thể là Access, SQL Server, MySQL, MongoDB,...

Việc phân tích và thiết kế cơ sở dữ liệu cho một hệ thống có thể tiến hành đồng thời với việc phân tích thiết kế hệ thống hoặc có thể tiến hành riêng. Vấn đề đặt ra là cần xây dựng một cơ sở dữ liệu giảm được tối đa sự dư thừa dữ liệu đồng thời phải dể khôi phục và bảo trì.

**3.2 Cơ sở lý thuyết**

**3.2.1 Khái niệm dữ liệu và cơ sở dữ liệu**

Dữ liệu là các thông tin liên quan đến đối tượng (ví dụ con người, vật, sự việc,...) được lưu trữ trên máy tính. Dữ liệu được mô tả dưới nhiều dạng khác nhau (như kí tự, hình ảnh, âm thanh,...).

Dữ liệu về đối tượng có thể khác nhau tùy thuộc vào mục đích quản lý. Ví dụ: cùng là dữ liệu về đối tượng nhân viên nhưng với mục đích quản lý khác nhau thì sẽ khác nhau, cụ thể như sau:

- Với mục đích quản lý lương, dữ liệu về đối tượng nhân viên sẽ bao gồm: Tên, Mã nhân viên, Lương, Phụ Cấp,...

- Với mục đích quản lý thông tin cá nhân, dữ liệu về đối tượng nhân viên sẽ bao gồm: Tên, Địa chỉ, Ngày sinh, Giới tính,...

Cơ sở dữ liệu là tập hợp dữ liệu được tổ chức một cách có cấu trúc liên quan với nhau và được lưu trữ trong máy tính. Cơ sở dữ liệu được thiết kế, xây dựng cho phép người dùng lưu trữ dữ liệu, truy xuất thông tin hoặc cập nhật dữ liệu.

Lợi ích của việc quản lý dữ liệu bằng cơ sở dữ liệu:

* Giúp dữ liệu lưu trữ một cách hiệu quả và có tổ chức.
* Giảm sự trùng lặp thông tin xuống mức thấp nhất và do đó đảm bảo được tính nhất quán và toàn vẹn dữ liệu.
* Đảm bảo dữ liệu có thể truy xuất theo nhiều cách khác nhau.
* Khả năng chia sẽ thông tin cho nhiều người và sử dụng cho nhiều ứng dụng khác nhau.

Những vấn đề cần phải giải quyết trong một cơ sở dữ liệu:

*- Tính chủ quyền của dữ liệu:* Do tính chia sẻ thông tin của cơ sở dữ liệu tiền tính chủ quyền của dữ liệu có thể bị là nhờ và làm thơ nhạt tinh thần trách nhiệm. Được thể hiện trên vấn đề an toàn dữ liệu, khả năng biểu diễn các mối liên hệ ngữ nghĩa của dữ liệu và tính chính xác của dữ liệu. Điều này có nghĩa là người khai thác cơ sở dữ liệu phải có nghĩa và cấp thất các thông tin thời nhất của cơ sở dữ liệu.

*- Tính bảo mật và quyền khai thác thông tin của người sử dụng:*Do có nhiều người được phép khai thác cơ sở dữ liệu một cách đồng thời nên cần phải có một cơ chê bao mật và phân quyền hạn khai thác cơ sở dữ liệu. Các hệ điều hành nhiều người sử dụng hay hệ điều hành mạng cục bộ (Window For WorkGroup, WinNT,...) đều có cung cấp cơ chế này

*- Tính tranh chấp dữ liệu:* Nhiều người được phép truy nhập vào cùng một tài nguyên dữ liệu của cơ sở dữ liệu với những mục đích khác nhau như: xem, thêm, sửa hoặc xóa dữ liệu. Cần phải có một cơ chế ưu tiên try nhập dữ liệu trong quá trình khai thác cạnh tranh. Cơ chế ưu tiên có thể được thực hiện bằng việc.

* Cấp quyền (hay mức độ) ưu tiên cho từng người khai thác người nào được cấp quyền hạn ưu tiên cao hơn thì được ưu tiên truy nhập dữ liệu trước.
* Theo biển có hoặc loại try nhập quyền đọc được ưu tiên trước quyền ghi dữ liệu.
* Dựa trên thời điểm truy nhập: ai có yêu cầu truy xuất trước thì có quyền truy nhập dữ liệu trước.

- Đảm bảo dữ liệu khi có sự cố: Việc quản lý dữ liệu tập trung có thể làm tăng khả năng mất mát hoặc sai lệch thông tin khi có sự cố như mất điện đột xuất, một phần đĩa lưu trữ cơ sở dữ liệu bị hư v.v... Một số hệ điều hành mạng có cung cấp dịch vụ Sao lưu ảnh đĩa, tự động kiểm tra và khắc phục lỗi khi có sự cố. Tuy nhiên, bên cạnh dịch vụ của hệ điều hành, để đảm bảo cơ sở dữ liệu luôn luôn ổn định, mộ cơ sở dữ liệu nhất thiết phải có một cơ chế khôi phục dữ liệu khi có sự cố bất ngờ xảy ra.

**3.2.2 Các mức biểu diển một cơ sở dữ liệu**

Đối với một cơ sở dữ liệu phức tạp, cần phải che giấu đi sự phức tạp đối với việc biểu diễn dữ liệu trong cơ sở dữ liệu đề nhằm đơn giản hóa sự tương tác giữa người sử dụng với hệ thống. Có 3 mức biểu diễn cơ sở dữ liệu:

-*Mức trong/ Mức vật lý (Physical level):* mô tả cách thức lưu trữ dữ liệu. Trả lời câu hỏi lưu dữ liệu gì và lưu trữ như thế nào? Người thao tác ở mức này là người quản trị cơ sở dữ liệu và những người sử dụng có chuyên môn.

-*Mức quan niệm/ Mức logic (Conception level):* trả lời cho câu hỏi cần phải lưu trữ bao nhiêu loại dữ liệu? Gồm những loại dữ liệu nào, mối quan hệ giữa chúng.

-*Mức ngoài/ Mức nhìn (View level):* là mức của người sử dụng. Mỗi người sử dụng có thể nhịn cơ sở dữ liệu theo một góc độ khác nhau. Tùy thuộc vào vai trò của người sử dụng mà có thể thấy tất cả hay một phần của cơ sở dữ liệu. Người sử dụng có thể không biết về cấu trúc tổ chức lưu trữ thông tin trong cơ sở dữ liệu, họ chỉ làm việc trên một phần của cơ sở dữ liệu.

**3.2.3 Mô hình phân rã chức năng (Bussiness Function Diagram\_BFD)**

**3.2.3.1 Giới thiệu mô hình phân rã chức năng**

Mô hình phân rã chức năng (BFD) là công cụ biểu diễn việc phân rã có thứ bậc các chức năng của hệ thống. Mỗi chức năng có thể gồm nhiều chức năng con, số mức chia ra phụ thuộc kích cỡ và độ phức tạp của hệ thống.

Mô hình BFD giúp xác định phạm vi của hệ thống cần phân tích, cho phép mô tả khái quát dần các chức năng của một tổ chức, phát hiện chức năng thiếu, trùng lặp. Rất gần gũi với sơ đồ tổ chức nhưng ta không đồng nhất nó với sơ đồ tổ chức.

**3.2.3.2 Các thành phần của mô hình phân rã chức năng**

*Chức năng*: là công việc mà tổ chức cần làm và được phân theo nhiều mức từ tổng hợp đến chi tiết. Tên chức năng phải là một mệnh đề động từ, gồm động từ và bổ từ. Được biểu diễn bằng hình chữ nhật.

Ví dụ:

|  |
| --- |
| In hóa đơn |

*Kết nối:* kết nối giữa các chức năng mang tính chất phân cấp và được ký hiệu bằng đoạn thẳng nối chức năng cho tới các chức năng con.

*Ví dụ:* Chức năng A phân rã thành chức năng B và C

**A**

**C**

**B**

**3.2.3.3 Các bước xây dựng mô hình phân rã chức năng**

- Bước 1: Xác định chức năng:

Một chức năng chính sẽ là 1 trong 3 loại sau:

* Cung cấp, sản xuất sản phẩm

Ví dụ: Phân phát hàng hóa.

* Cung cấp dịch vụ.

Ví dụ: Đặt hàng hóa

* Quản lý tài nguyên.

Ví dụ: Quản lý tiền lương

Sau khi xác định nó thuộc loại nào thì tiến hành đặt tên cho chức năng đó.

Tên chức năng phải đơn giản nhưng phải bao quát các chức năng con. Mỗi chức năng có một tên duy nhất, các chức năng không được trùng tên.

- Bước 2: Phân rã các chức năng:

Nguyên tắt phân rã:

* Sắp xếp các công việc trên một mức cùng một hàng đảm bảo cân đối.
* Hệ thống lớn không quá 6 mức, hệ thống nhỏ không quá 3 mức.
* Các chức năng con cùng một mẹ có kích thước, độ phức tạp gần như nhau.

**3.2.4 Mô hình luồng dữ liệu**

**3.2.4.1 Giới thiệu mô hình luồng dữ liệu**

Mô hình luồng dữ liệu là một công cụ mô tả có thứ tự các thông tin chuyển từ một chức năng này sang một chức năng khác.

Mục đích của biểu đồ luồng dữ liệu là giúp chúng ta thấy được đằng sau những cái gì thực tế xảy ra trong hệ thống, làm rõ những chức năng và thông tin nào cần thiết cho việc quản lý.

**3.2.4.2 Các thành phần của mô hình luồng dữ liệu**

Chức năng ( tiến trình): là một hoạt động có liên quan đến sự biến đổi hoặc tác động lên thông tin như tổ chức lại thông tin, bổ sung thông tin hoặc tạo ra thông tin, Một tiến trình luôn có thông tin đầu vào và thông tin đầu ra.

Ví dụ: Tiến trình Nhập mới thông tin học sinh.

*Luồng dữ liệu:* là luồng thông tin vào hoặc ra khỏi chức năng. Các luồng dữ liệu phải chỉ ra được thông tin logic chứ không phải tài liệu vật lý. Các luồng thông tin khác nhau phải có tên gọi khác nhau.

Ví dụ:

Biên nhận

Biên nhận hợp lệ

*Kho dữ liệu:* là nơi biểu diễn thông tin cần cất giữ, để một hoặc nhiều chức năng sử dụng chúng. Một tiến trình có thể trao đổi với kho dữ liệu như: đưa dữ liệu vào kho, lấy thông tin từ kho, vừa lấy thông tin ra xử lý vừa cập nhật lại kho.

Ví dụ: Đưa thông tin học sinh vào kho danh sách học sinh

|  |
| --- |
| Danh sách học sinh |

*Tác nhân ngoài:* là tên người hoặc một nhóm người nằm ngoài hệ thống nhưng có trao đổi trực tiếp với hệ thống.

Ví dụ:

Phòng giám đốc

Phòng kế toán

**3.2.4.3 Các bước xây dựng mô hình luồng dữ liệu**

*Phân cấp:* là sự phân cấp từ mức tổng quát đến mức chi tiết để nêu lên các chức năng của hệ thống. Tuỳ theo quy mô của hệ thống mà sự phân cấp này chi tiết đến mức nào, đa số được chia thành 3 mức.

* Bước 1: Xây dựng DFD mức khung cảnh (mức 0)
* Bước 2: Xây dựng DFD mức đỉnh (mức 1)
* Bước 3: Xây dựng DFD mức dưới đỉnh (mức 2)

**3.2.5 Mô hình thực thể kết hợp (Entity Relationshop Diagram\_ERD)**

**3.2.5.1 Giới thiệu mô hình thực thể kết hợp**

Mô hình thực thể kết hợp (ERD) hay còn là mô hình thực thể liên kết hoặc mô hình mức quan niệm cho dữ liệu (MCD Conceptual Data Model) là mô hình mô tả thế giới thực một cách dễ hiểu và rõ ràng, gần với suy nghĩ của con người.

Mô hình ERD dùng để thiết kế CSDL ở mức quan niệm.

Mô hình ERP có những tính chất sau:

* *Tính diễn đạt:* mô tả khối lượng lớn các khái niệm sao cho nó có thể biểu diễn toàn diện thế giới thực.
* *Tính đơn giản:* giúp người sử dụng và người thiết kế hiểu nhau.
* *Tỉnh hình thức:* các khái niệm thể hiện đồng nhất, chính xác.
* *Tinh dễ đọc:* mỗi khái niệm được biểu diễn bằng một ký hiệu đồ họa rõ ràng.

**3.2.5.2 Các thành phần của mô hình thực thể kết hợp**

**3.2.5.2.1 Thực thể**

Thực thể là một nhóm các đối tượng có trong thế giới mà ta đang khảo sát. Nó tồn tại một cách độc lập, các đối tượng được nhóm lại dựa vào một số đặc tính chung nào đó.

Ví dụ: tại một trường cao đẳng ta có các thực thể như khoa, lớp, sinh viên, giáo viên,...

**3.2.5.2.2 Liên kết**

Một liên kết giữa hai thực thể là một kết hợp về ngữ nghĩa giữa các thực thể đó.

Ví dụ: Khoa Quản lý nhiều lớp, lớp gồm nhiều sinh viên.

**3.2.5.2.3 Thuộc tính**

Thuộc tỉnh là dữ liệu mà người ta nhận biết trên một thực thể hoặc một liên kết. Tên của thuộc tỉnh này được ghi bên dưới của thực thể hoặc liên kết.

* *Thuộc tính của thực thể:* phụ thuộc vào thực thể đó.
* *Thuộc tính của liên kết:* phụ thuộc vào các thực thể tham gia vào liên kết.

Ví dụ:

* Thực thể SINHVIEN có các thuộc tính: mã sinh viên, họ tên sinh viên,
* giới tính, địa chỉ,...
* Liên kết Học có thuộc tính Điểm Trung Bình, và là liên kết giữa thực thể SINHVIEN và thực thể MONHOC.

**3.2.5.2.4 Thể hiện của thực thể và thể hiện của liên kết**

*- Thể hiện của một thực thể:* là một đối tượng cụ thể trong tập hợp đối tượng.

*- Thể hiện của một liên kết:* là một liên kết được tách rời ra riêng rẽ, được tạo thành từ một và chỉ một thể hiện của từng thực thể tham gia vào liên kết đó.

**3.2.5.2.5 Bản số**

Bản số là sự thể hiện của một thực thể, thể hiện cho một liên kết, bản số có 2 loại.

- *Bản số tối thiểu:* là số lần tối thiểu mà một thể hiện của thực thể tham gia vào các thể hiện của một liên kết. Bản số tối thiểu có 2 giá trị: 0 và 1.

- *Bản số tối đa:* là số lần tối đa là một thể hiện của thực thể tham gia vào các thể hiện của một liên kết. Bản số tối đa có 2 giá trị: 1 và n.

Ví dụ: Một sinh viên thuộc tối thiểu một lớp, thuộc tối đa nhiều lớp. Bản số tối thiểu là 1. Một lớp gồm tối thiểu một sinh viên, gồm tối đa nhiều sinh viên. Bản số tối thiểu là 1, bản số tối đa là n.

**3.2.5.2.6 Khóa**

- Khóa của thực thể là một thuộc tính đặc biệt của thực thể sao cho với mỗi giá trị của thuộc tỉnh này tương ứng với một và chỉ một thể hiện của thực thể. Các khóa này thường được ghi đầu tiên trong danh sách các thuộc tính của thực thể và phải gạch dưới.

- Khóa của liên kết: thường được tạo thành bởi tất cả các khóa của các thực thể tạo nên liên kết đó.

**3.2.5.2.7 Chiều của liên kết**

Chiều của liên kết là số lượng đối tượng tham gia vào lien kết đó. Có các loại liên kết sau:

* Liên kết 2 chiều.
* Liên kết 3 chiều.
* Liên kết lớn hơn 3 chiều.
* Nhiều liên kết giữa cùng thực thể.
* Liên kết tự thân.

**3.2.5.2.8 Tổng quát hóa và chuyên biệt hóa**

Chuyên biệt hỏa là sự phân hoạch một thực thể thành một tập các thực thể. Các thực thể chuyên biệt (thực thể con) thừa hưởng tất cả các thuộc tính của thực thể tiền bối (thực thể cha).

Tổng quát hóa là gộp các thực thể thành một thực thể bao hàm tất cả các thể hiện của các thực thể con.

**3.2.5.2.9 Phụ thuộc hàm**

*Phụ thuộc hàm giữa các thuộc tính:* là tồn tại một phụ thuộc hàm giữa thuộc tỉnh A và thuộc tính B, nếu với mỗi giá trị của thuộc tính A đều xác định duy nhất một thuộc tính B.

*Phụ thuộc hàm sơ cấp giữa các thuộc tính:* tồn tại một phụ thuộc hàm sơ cấp giữa hai thuộc tính A và B, nếu tồn tại một phụ thuộc hàm giữa A và B và không tồn tại 1 thuộc tính con nào của A để xác định được B.

*Phụ thuộc hàm sơ cấp trực tiếp giữa các thuộc tính:* tồn tại một phụ thuộc hàm sơ cấp trực tiếp giữa hai thuộc tỉnh A và B, nếu có:

- Một phụ thuộc hàm sơ cấp giữa A và B.

- Không tồn tại một thuộc tính C nào, sao cho C phụ thuộc hàm A rối C cũng phụ thuộc hàm B.

Phụ thuộc hàm giữa các thực thể là tồn tại một phụ thuộc hàm giữa hai thực thế X và Y, nếu mỗi thể hiện của thực thể X xác định duy nhất một thể hiện của thực thể Y.

**3.2.5.3 Các quy tắt chuẩn hóa**

Mục đích của việc chuẩn hóa là nhằm để loại bỏ sự lặp đi lặp lại của các thuộc tính trong các thực thể của các quan hệ, nhằm tránh sự dư thừa dữ liệu, giải quyết sự nhập nhằng giữa các mối quan hệ của các thực thể.

- Quy tắc 1: đối với bất kỳ một thể hiện nào của thực thể, mọi thuộc tỉnh đều phải có ý nghĩa và phải có giá trị duy nhất.

- Quy tắc 2: mọi thuộc tính của một thực thể đều phụ thuộc vào khoá của thực thể đó bằng một phụ thuộc hàm sơ cấp trực tiếp.

- Quy tắc 3: mọi thuộc tính của liên kết phải phụ thuộc vào khóa của liên kết đó bằng một phụ thuộc hàm sơ cấp trực tiếp.

- Lỗi Fan Traps: lỗi này thường xuất hiện khi có 2 phụ thuộc hàm cũng chỉ vào một thực thể đích, mà hai thực thể nguồn có mối liên hệ.

**3.2.5.4 Các bước xây dựng mô hình thực thể kết hợp**

- Liệt kê chính xác và lựa chọn các thông tin cơ sở.

- Xác định các thực thể, thuộc tính và định danh của nó.

- Xác định các mối quan hệ và các thuộc tính của nó.

- Vẽ biểu đồ mô hình thực thể kết hợp.

- Xác định bản số, chuẩn hóa và thu gọn biểu đồ.

**3.2.6 Mô hình quan hệ (Relational Data Model\_RDM)**

**3.2.6.1 Giới thiệu mô hình quan hệ**

Mô hình cơ sở dữ liệu quan hệ (ROM) hay còn là mô hình quan hệ hoặc mô hình mức luận lý cho dữ liệu MLD (Data Model) là mô hình được E.F.Codd phát triển vào đầu những năm 1970. Mô hình được thiết lập trên cơ sở lý thuyết tập hợp nên nó rất dễ hiểu và được sử dụng rộng rãi trong việc tổ chức dữ liệu cho các hệ thống.

**3.2.6.2 Các thành phần trong mô hình quan hệ**

Quan hệ các thông tin lưu trữ trong cơ sở dữ liệu được tổ chức thành bảng (Table) 2 chiều gọi là quan hệ.

Lược đồ quan hệ: bao gồm tên quan hệ và tập các thuộc thỉnh của quan hệ.

Lược đồ cơ sở dữ liệu: gồm một tập hợp các lược đồ quan hệ.

**3.2.6.3 Các quy tắt chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ**

Mô hình thực thể kết hợp đã cung cấp một cách trình bày độc lập với kỹ thuật về các thông tin của hệ thống. Nó quá mềm dẻo về mặt ngôn ngữ do đó hệ quản trị cơ sở dữ liệu khó có thể hiểu được. Vì vậy, cần thiết phải có một giai đoạn chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ.

- Quy tắc 1: mỗi thực thể trong mô hình ERD sẽ được chuyển thành một quan hệ trong mô hình quan hệ. Mỗi thuộc tính trong mô hình ERD sẽ được chuyển thành một thuộc tỉnh trong mô hình quan hệ. Khóa của thực thể trở thành khóa chính của quan

hệ.

- Quy tắc 2: mỗi liên kết phụ thuộc hàm sẽ biến mất trong mô hình quan hệ. Thực thể đích sẽ chuyển thành quan hệ đích, thực thể nguồn sẽ trở thành quan hệ nguồn trong mô hình quan hệ. Quan hệ nguồn này sẽ gồm tất cả các thuộc tính của thực thể nguồn và chứa thêm khóa của thực thể đích. Khóa này được xem như là khóa ngoài của quan hệ nguồn. Nếu liên kết phụ thuộc hàm có thuộc tính, thì ở mô hình quan hệ các thuộc tính này cũng được thêm vào quan hệ nguồn.

- Quy tắc 3: một liên kết không là phụ thuộc hàm của ERD sẽ trở thành một quan hệ trong mô hình quan hệ. Quan hệ này gồm có khóa của các thực thể liên kết với nó và có các thuộc tính của liên kết. Trong đó, khóa chính của quan hệ chính là khóa của các thực thể tham gia vào liên kết đó.

- Quy tắc 4: một liên kết tự thân nếu là liên kết có phụ thuộc hàm thì áp dụng quy tắc 2. Nếu là liên kết không có phụ thuộc hàm thì áp dụng quy tắc 3.

**3.3 Lý thuyết về công nghệ được sử dụng trong phần mềm**

**3.3.1 Hệ quản trị cơ sở dữ liệu MySQL**

MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/w/index.php?title=Novell_NetWare&action=edit&redlink=1), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,..

### 3.3.2 Laravel

**3.3.2.1 Giới thiệu PHP**

**PHP** là viết tắt **Hypertext Preprocessor** một ngôn ngữ lập trình kịch bản mã nguồn mở chạy ở phía server. **PHP** đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến và được ưa chuộng nhất thế giới nhờ tốc độ nhanh, nhỏ gọn, cú pháp dễ học...

**PHP** được tạo ra năm 1994 bởi Rasmus Lerdorf, trải qua quá trình phát triển từ PHP/FI, PHP 3, PHP 4, PHP 5 với sự đóng góp của cộng động phiển bản mới nhất PHP 7 được kỳ vọng sẽ lấp đầy những khiếm khuyết của PHP ở phiên bản hiện tại..

#### 3.3.2.2 Giới thiệu laravel

[**Laravel**](https://laravel.com/)  là một Framework Open Source PHP miễn phí được khởi tạo bởi Taylor Otwell vào năm 2011 nhằm mục đích tạo lựa chon thay thế cho CodeIgniter, Laravel cung cấp một bộ khung chuẩn dựa trên mẫu kiến trúc **design parttent MVC**và hàng loạt các quy tắc tạo code, các mẫu thiết kế, chính điều này giúp Laravel hỗ trợ đắc lực cho các lập trình viên trong quá trình triển khai các ứng dụng web trên nền tảng mã nguồn mở Php một cách nhanh chóng. Tổ chức code của Laravel rất linh động, mềm dẻo và không kém phần chặt chẽ, chính điều này làm cho các khối code được lập trình viên viết luôn luôn đảm bảo đúng chuẩn, sạch sẽ và dễ nâng cấp sau này.

Vào tháng 6/2011 Laravel được phát hành phiên bản beta nhằm tạo nên một lựa chọn mới thay thế cho Codeiginter. Phiên bản 1.0 được phát hành vào cùng tháng với phiên bản beta này, Laravel cung cấp hàng tá các tính năng mà Codeiginter đang thiếu như: xác thực người dùng, đa ngôn ngữ, MVC. Tuy nhiên tại phiên bản 1.0 Laravel vẫn chưa đưa vào controllers điều này khiến nó chưa thực sự chuẩn MVC

**Laravel 2.0** được phát hành vào tháng 9/2011 với việc chuẩn hóa mô hình MVC, controllers được đưa vào, hỗ trợ thiết kế **IOC**, và giới thiệu một tính năng mới là template engine Blade. Một thiếu sót và nhược điểm của phiên bản này là loại bỏ các thư viện bên ngoài, điều này khiến Laravel 2.0 chưa thực sự hấp dẫn lắm so với CodeIginer.

**Laravel 3.0** được ra mắt vào tháng 2/2012 với một loạt tính năng hấp dẫn như việc thêm vào môi trường code dòng lệnh CLI với tên artisan. Bổ sung tính năng kết nối với các hệ quản trị cơ sở dữ liệu khác.Tạo,xóa, sửa tổ chức dữ liệu bằng các phiên bản trong hệ thống migrate. Tại phiên bản này tính năng hấp dẫn Events (tạo sự kiện) cũng là một điểm mạnh khiến Laravel phát triển được một cộng đồng người dùng.

**Laravel 4.0** với tên mã **lluminate** được phát hành vào tháng 5/2013 đây là một cải tiến vượt bậc, loại bỏ toàn bộ những khuyết điểm của các phiên bản trước. Được viết lại hoàn toàn theo chuẩn MVC chặt chẻ Laravel 4.0 cung cấp 1 dàn khung cơ bản mạnh mẽ cho các ứng dụng được viết trên nó. Có thể nói đây là phiên bản mới hoàn toàn, tất cả các thành phần tổ chức thư viện đều được đóng gói và hỗ trợ qua composer,, chính điều này khiến cho áp dụng các thư viện bên ngoài dễ dàng hơn bao giờ khác, nó tách biệt rõ ràng từng phần thư viện. Hỗ trợ hàng đợi, nâng cao tính năng Migrate, hỗ trợ nhiều phương thức gửi mail…

Laravel 5.0 phát hành tháng 2/2015 đã khiến nó vươn lên vượt bậc, và lọt vào top các framework được lựa chon ưu tiên khi phát triển các ứng dụng doanh nghiệp bằng PHP. Các tính năng mới trong bản phát hành Laravel 5 bao gồm hỗ trợ lập kế hoạch các tác vụ được thực hiện định kỳ thông qua gói có tên *Scheduler* , một lớp trừu tượng có tên *Flysystem* cho phép lưu trữ từ xa theo cách tương tự như các [hệ thống tệp](https://en.wikipedia.org/wiki/File_system) cục bộ , cải thiện việc xử lý assets thông qua *Elixir* . xác thực được xử lý bên ngoài đơn giản thông qua gói *Socialite* tùy chọn. Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ mới cho các ứng dụng đã phát triển. Vào tháng 3 năm 2015, một khảo sát SitePoint đã liệt kê Laravel là khung công tác PHP phổ biến nhất.

#### 3.3.2.2 Ưu điểm của laravel

**3.3.2.2.1 Sử dụng các tính năng mới nhất của PHP**

Một trong những tính năng làm cho PHP Lararel framework trở nên nổi bật hơn tất cả đó là nó sử dụng tất cả các tính năng mới của PHP. Các framework khác không tận dụng được điểu này. Sử dụng Laravel 5 bạn sẽ sở hữu một số tính năng mới nhất mà PHP cung cấp, đặc biệt là trong Namespaces, Interfaces, Overloading, Anonymous functions và Shorter array syntax. Đã có rất nhiều bộ CMS nổi bật được viết trên nền Laravel framework ra đời, từ những bộ CMS đơn giản nhất đến những bộ "siêu mạnh", từ mã nguồn mở đến thương mại.

**3.3.2.2.2 Tài liệu tuyệt vời**

Tài liệu của Laravel rất thân thiện với nhà phát triển. Tất cả các phiên bản của Laravel được phát hành cùng với các tài liệu phù hợp, bạn sẽ tìm thấy những giải thích chi tiết về coding style, methods và classes.

**3.3.2.2.3 Tích hợp với dịch vụ mail**

Các ứng dụng web bắt buộc phải thông báo cho người dùng về các deal và offer mới. Đăng ký người dùng mới cũng rất quan trọng (nghĩa là thông báo cho người dùng mới khi anh ấy/ cô ấy đăng ký). Framework được trang bị API sạch trên thư viện SwiftMailer. Và có các driver cho SMTP, Mailgun, SparkPost, Mandrill, PHP's "mail" function, Amazon SES, và "sendmail" giúp bạn gửi thư qua các dịch vụ dựa trên đám mây hoặc dịch vụ local.

**3.3.2.2.4 Hỗ trợ cache backend phổ biến**

Laravel framework hỗ trợ các cache backend như Memcached và Redis out-of-the-box. Bạn cũng có thể tùy chỉnh nhiều cấu hình cache.

**3.3.2.2.5 Công cụ tích hợp cho dòng lệnh - Artisan**

Các nhà phát triển thường thấy tẻ nhạt khi thực hiện các nhiệm vụ lập trình lặp đi lặp lại vì chúng rất tốn thời gian. Công cụ dòng lệnh có tên Artisan giúp họ tạo khung code và quản lý hệ thống cơ sở dữ liệu hiệu quả. Artisan giúp tạo các tệp MVC cơ bản và quản lý tài sản, bao gồm cả các cấu hình tương ứng của chúng.

**3.3.2.2.6 Gói và nguồn lực sẵn có**

Bạn sẽ được hưởng lợi ích từ npm package và bower package bằng cách kết hợp framework với Gulp và elixir, giúp trong việc sửa đổi tài nguyên. Việc tích hợp này sẽ giải quyết các phụ thuộc và là nguồn gói đáng tin cậy nhất trong thế giới PHP.

**3.3.2.2.7 Giảm chu kỳ phát triển sản phẩm**

Bạn có thể giảm đáng kể chu kỳ phát triển sản phẩm vì framwork giúp tích hợp nhanh hơn, bên cạnh đó là cộng đồng hỗ trợ rất lớn - Laracasts.

**3.3.2.2.8 Container IoC**

Còn được gọi là Inversion of Control, đây là một phương thức để tạo các đối tượng mới mà không phải khởi động bất kỳ thư viện bên ngoài nào. Nói cách khác, bạn có thể truy cập các đối tượng này từ bất cứ nơi nào bạn đang mã hóa, bạn sẽ không còn gặp khó khăn với các cấu trúc nguyên khối không linh hoạt.

**3.3.2.2.9 Reverse Routing**

Đây là một tính năng rất hữu ích trong framwork, bạn có thể tạo liên kết đến các route được đặt tên. Vì vậy, khi tạo các liên kết, bạn chỉ cần sử dụng tên của bộ định tuyến, framework sẽ tự động chèn URL chính xác. Nếu bạn cần thay đổi các route trong tương lai, những thay đổi sẽ tự động được thực hiện ở mọi nơi.

**3.3.2.2.10 Eloquent ORM**

Đây là một trong những tính năng tốt nhất của Laravel, cung cấp một triển khai ActiveRecord đơn giản và hiệu quả, làm việc với cơ sở dữ liệu. Các mô hình bạn tạo trong MVC sẽ có một bảng tương ứng trong cơ sở dữ liệu. ORM có các mối quan hệ, nếu bạn thao tác một bảng trong cơ sở dữ liệu, bạn cũng sẽ quản lý các dữ liệu liên quan. Các mối quan hệ đó là: one-to-one, one-to-many, many-to-many, has-many-through, polymorphic, và many-to-many polymorphic.

#### 3.3.2.3 Nhược điểm của Laravel

**3.3.2.3.1 Không hỗ trợ tính năng thanh toán**

Sẽ không đáng lo ngại nếu bạn không tự mình quản lý các khoản thanh toán, bởi vì bạn sẽ phải tuân thủ các quy tắc tuân thủ PCI. Trì hoãn các dịch vụ như Stripe và Paypal sẽ giải quyết vấn đề đó. Bạn cũng có thể thử bất kỳ trang web thương mại trực tuyến nào và xây dựng ứng dụng của mình trong kho template có sẵn, hoặc sử dụng các thư viện của Framework cho phép bạn tích hợp các phương thức thanh toán. Tuy nhiên, hầu hết các nhà giao dịch điện tử nên được nhúng bộ xử lý thanh toán của bên thứ ba nhằm thuận tiện hơn.

**3.3.2.3.2 Thiếu sự liên tục giữa các phiên bản**

Không có chuyển đổi liền mạch từ Laravel 4 sang 5. Nếu cố cập nhật code, bạn có thể sẽ phá vỡ ứng dụng.

**3.3.2.3.3 Chất lượng**

Một số thành phần trong framework không được thiết kế tốt. Ví dụ, dependency injection đôi khi trở nên phức tạp không cần thiết. Các tài liệu khá nặng. Bạn phải học hỏi nhiều trước khi bắt đầu xây dựng các ứng dụng.

Do đó, đây không phải là một lựa chọn tốt cho các nhà phát triển nghiệp dư. Tuy nhiên, framework vẫn đang được cải thiện rất nhiều. Phiên bản 5 đã tốt hơn nhiều với số lượng sai sót cũng ít hơn.

**3.3.2.3.4 Một số nâng cấp có thể có vấn đề**

Đây không chỉ là vấn đề của Laravel mà là của các PHP framework. Vì vậy, các nhà phát triển nên có biện pháp phòng ngừa trước khi nâng cấp mobile application/website.

**3.3.2.3.5 Thường không cung cấp sự phong phú cho ứng dụng di động**

Việc tải lại toàn trang có thể hơi nặng trong các mobile app khi so sánh với các website. Trong những trường hợp như vậy, các nhà phát triển web có xu hướng chỉ sử dụng framework như backend JSON API.

#### 3.3.2.4 Mô hình Model View Controller (MVC)

MVC - viết tắt của Model View Controller. Đây là một mô hình, cho phép chúng ta tách biệt các thành phần xử lý trong hệ thống. Để từ đó chúng ta có thể giải quyết các công việc một các nhanh gọn và dễ dàng hơn.

* Controller : có chức năng điều khiển, sắp xếp và xử lý các yêu cầu của người dùng.
* Model : sẽ đảm nhiệm các công việc trao đổi dữ liệu với database.
* View : là thành phần giao diện, hiển thị dữ liệu cho người dùng.

Khi người dùng gửi một yêu cầu lên hệ thống, hệ thống sẽ gửi về cho phía Controller xử lý các yêu cầu của người dùng. Trong quá trình làm việc, Controller sẽ phải thông qua lớp Model để làm việc với CSDL. Sau khi xử lý xong công việc, Controller sẽ đưa sang Views để hiển thị cho người dùng.

#### 3.3.2.5 Cấu trúc thư mục Laravel

**Project-laravel/**

**├── 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 chứa tất cả các thư mục, các tập tin php, các class php, thư viện, models để xây dựng project của bạn.
* **Console** thư mục chứa các tập tin định nghĩa từ lệnh thực thi trên Artisan.
* **Excerption** thư mục chứa các tập tin quản lý, điều hướng lỗi
* **Http**
* **Controller** là thư mục chứa các tập tin controllers
* **Middleware** là thư mục chứa các tập tin lọc và ngăn chặn các requests.
* **Kernel.php**là tập tin cấu hình, định nghĩa Middleware hoặc nhóm Middleware.
* **bootstrap** thư mục chứa tập tin điều hướng khởi động hệ thống, thường thì chúng ta sẽ không làm gì đến nó.
* **config** chứa mọi tập tin cấu hình của Laravel.
* **database** chứa các thư mục tập tin về CSDL
* **migrations** chứa các tập tin định nghĩa khởi tạo và sử bảng.
* **seeds** chứa các tập tin định nghĩa dữ liệu insert 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.
* **resources** chứa các tập tin giao diện (js, css, less, sass, coffeescript,...), views, ngôn ngữ.
* **storage** chứa các tập tin hệ thống như upload, cache, session, cookie, log...
* **routes** là thư mục chứa các tập tin định nghĩa các router, xử lý router hoặc điều hướng router (tức là URL, laravel không tự đặt url theo kiểu example.com/controller/action/value mà chúng ta phải tự định nghĩa chúng) bao gồm 3 loại là 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 key app, tên app, url app, email,  env mode, CSDL hay bật tắt debug.
* **composer.json**, **composer.lock** tập tin của composer.
* **package.js** tập tin cấu hình của nodejs chứa các package cần thiết cho projects.
* **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.

#### 3.3.2.6 Route

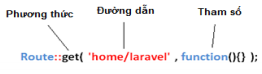
**Route** trong laravel có nhiệm vụ định hướng request của laravel.

Các Route trong laravel định nghĩa trong file **routes/web.php**. Danh sách các loại Route cơ bản và thường sửa dụng:

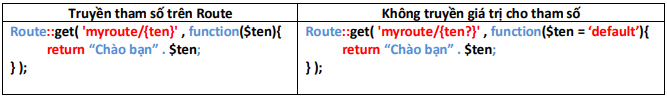
* Route::get Phương thức Get
* Route::post Phương thức Post
* Route::match Kết hợp nhiều phương thức
* Route::any Nhận tất cả phương thức
* Route::filter Tạo ra bộ lọc nhằm mục đích nào đó
* Route::group Gom các Route lại thành một nhóm
* Route::controller gọi đến Controller tương ứng
* Route::resource Sử dụng với Resource Controller

### Ngoài các method POST và GET phổ biến trong laravel còn có thêm PUT, PATCH, DELETE.

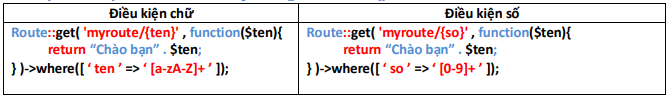
### Cấu trúc Route



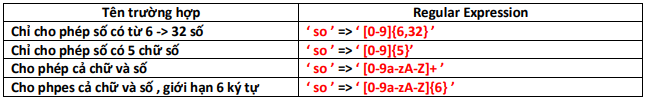
* Truyền tham số trên Route:



* Đặt điều kiện cho tham số với phương thức where:



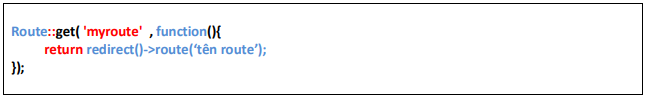
* Các trường hợp khác:



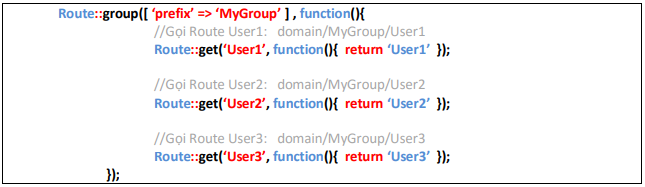
* **Định danh Route**



* Gọi Route bằng tên:



* **Route group**



#### 3.3.2.7 Controller

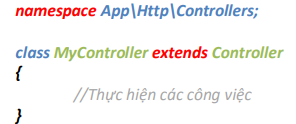
* **Tạo controller**

Tất cả controller trong laravel được chứa trong thư mục **app\http\controller**.

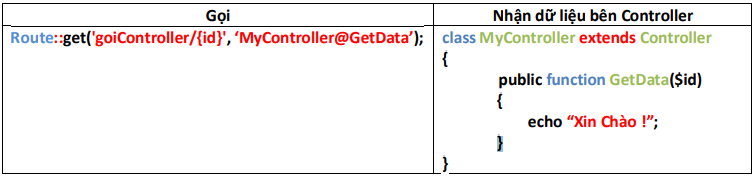
Tạo controller thông qua composer:

**php artisan make:controller {Tên controller}**

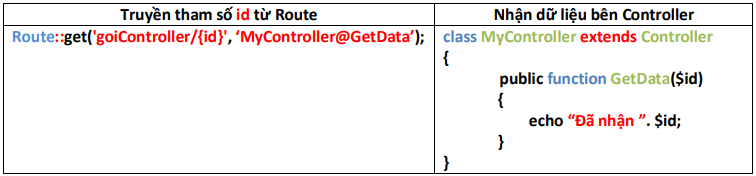
* **Cấu trúc Controller**



* **Gọi Controller**



* **Nhận dữ liệu từ Route**

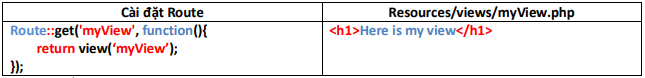


#### 3.3.2.8 View

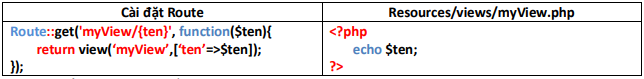
* **View là gì ?**

Là các file có đuôi **.php**, chứa mã nguồn html, hiển thị dữ liệu cho người dùng xem và view được lưu ở thư mục **resources/views.**

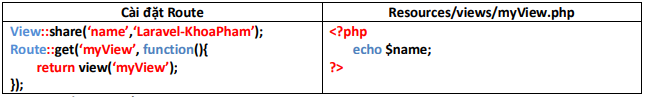
* **Gọi View**



* **Truyền tham số sang View**



* **Dùng chung dữ liệu trên Views**

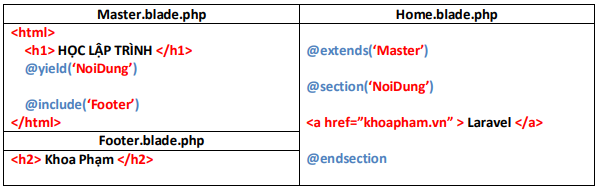


#### 3.3.2.8 Blade template

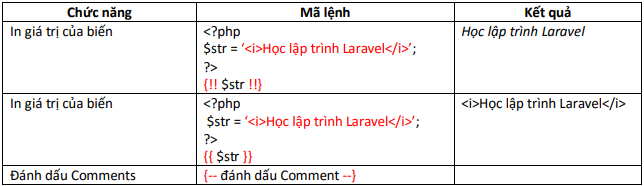
Muốn sử dụng **Blade Template** thì các tên file phải có chứa **.blade** đằng trước **.php**.

Vd: **MyFile.blade.php**

* **Sử dụng Blade Template**
* Kế thừa



* Hiển thị dữ liệu



* Các câu lệnh điều kiện



#### 3.3.2.9 Database

* **Kết nối với CSDL trong laravel**

Mở file .env cấu hình lại:

DB\_HOST = localhost

DB\_DATABASE = Tên CSDL

DB\_USERNAME = Tên người dùng

DB\_PASSWORD = Mật khẩu

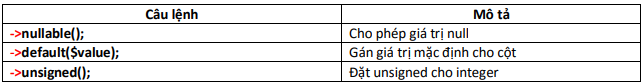
* **Schema**
* Tạo bảng



* Mở rộng



* Điều kiện



* Sửa bảng



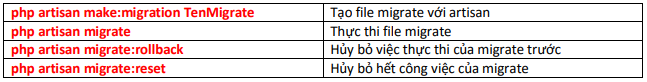
* Xóa bảng



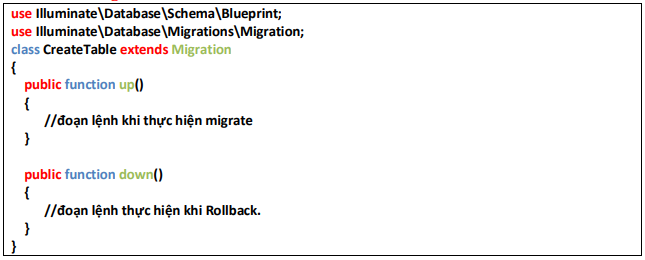
* **Migrate**

**Migrate** dùng để tạo lên cấu trúc các bảng trong cơ sở dữ liệu. Ta có thể sử dụng migrate để tạo ra các bảng cũng như back up, restore lại theo ý muốn. Các file migrate sẽ được lưu tại **database/migrations**.

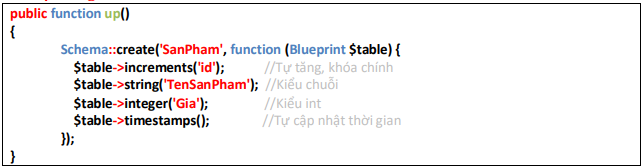
* Tạo sử dụng Migrate thông qua composer



* Cấu trúc migration



* Tạo bảng với Schema



## 3.4 Phân tích yêu cầu

### 3.4.1 Mô tả yêu cầu

Website cổng thông tin trường học bao gồm quản lý tin tức, quản lý thông báo, quản lý các phòng, khoa, thông tin tuyển sinh, các trung tâm tin học, anh văn, các biểu mẫu cần thiết của nhà trường. Ngoài ra còn quản lý quyền đăng nhập sửa đổi các thông tin của nhà trường.

### 3.4.2 Các thông tin đầu vào

* Khi muốn đăng bài viết lên trang tin tức người dùng cần đăng nhập vào hệ thống bằng tài khoản user do admin cung cấp, chọn loại tin , nhập tên bài viết, nội dung bài, chọn ảnh bài viết, trạng thái bài viết.
* Có nhiều bộ phận làm việc cần quản lý trang thông tin của các bộ phận.
* Có nhiều loại tin cần quản lý tin tức của mỗi loại tin.
* Có nhiều người dùng cần quản lý thông tin người dùng, quyền.

### 3.4.3 Các thông tin đầu ra

* Giới thiệu về trường học (CTEC)
* Hiển thị tin tức, thông báo
* Hiển thị các thông tin tuyển dụng liên kết với nhà trường
* Hiển thị thời khóa biểu
* Hiển thị lịch thi, kết quả thi học kì
* Tìm kiếm tin tức

### 3.4.4 Các yêu cầu trang quản lý

* Cập nhật, tìm kiếm thống kê tin tức
* Cập nhật, tìm kiếm thống kê người dùng
* Cập nhật, tìm kiếm thống kê các thông báo
* Phân quyền người dùng

### 3.4.5 Quyền truy cập hệ thống

**- Quản trị**: (chỉ duy nhất một tài khoản có quyền quảng trị)

+ Toàn quyền trên hệ thống

+ Tạo ra các tài khoản thành viên

+ Tạo ra các loại tin, trang

+ Quản lý thông tin trang web, menu, logo, banner, slide,….

**- Thành viên:**

+ Được quyền đăng bài viết

+ Sửa, xóa bài viết của mình

### 3.5 Xây dựng mô hình phân rã chức năng

### 3.5.1 Quản lý Posts

**3.5.2 Quản lý Pages**

**3.5.3 Quản lý Option**

**3.5.4 Quản lý Users**

**3.5.5 Quản lý Categories**

**3.5.6 Quản lý Menus**

**3.5.7 Quản lý Slides**

**3.5.8 Thống kê**

### 3.5.9 Mô hình phân rã chức năng tổng hợp

Admin Cổng thông tin CTEC

Quản lý Posts

Thêm mới

Sửa

Xóa

Tìm kiếm

Quản lý Pages

Thêm mới

Sửa

Xóa

Tìm kiếm

Quản lý Menu

Thêm mới

Sửa

Xóa

Quản lý Category

Thêm mới

Sửa

Xóa

Quản lý User

Đăng nhập

Đăng kí

Đổi mật khẩu

Cập nhật thông tin

Quản lý Slides

Thêm mới

Sửa

Xóa

Quản lý Options

Tên website

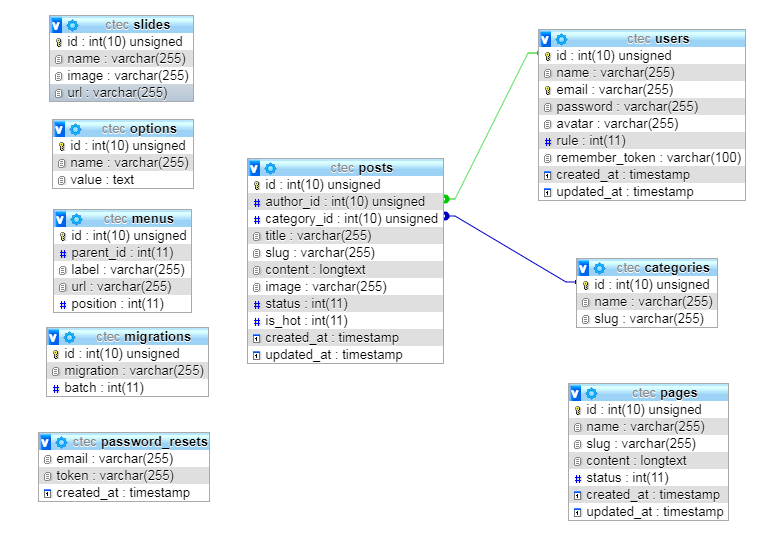
Miêu tả website

Icon website

Logo website

Banner website

## 3.6 Xây dựng mô hình thực thể kết hợp



## 3.7 Xây dựng mô hình quan hệ (RDM)

Từ mô hình thực thể kết hợp ta được mô hình quan hệ như sau:

* **Users (id**, name, email, password, avatar, rule**)**
* **Posts (id**, *author\_id*, *category\_id*, title, slug, content, image, status, is\_hot **)**
* **Categories (id**, name, slug**)**
* **Pages (id**, name, slug, content, status**)**
* **Menus (id**, name, slug, parent\_id**)**
* **Options (id**, name, value**)**
* **Slides (id**, name, image, link**)**

## 3.8 Xây dựng các bảng

* Bảng **Users** (Người dùng):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Name | Varchar(255) |  | Tên |
| 3 | Email | Varchar(255) |  | Email |
| 4 | Password | Varchar(255) |  | Mật khẩu |
| 5 | Avatar | Varchar(255) |  | Ảnh đại diện |
| 6 | Rule | Int(10) |  | Phân quyền |

* Bảng **Categories** (Loại tin):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Name | Varchar(255) |  | Tên |
| 3 | slug | Varchar(255) |  | Tên không dấu |

* Bảng **Posts** (Bài viết):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Author\_id | Int(10) | Khóa phụ | Id\_User |
| 3 | Category\_id | Int(10) | Khóa phụ | Id\_Category |
| 4 | Title | Varchar(255) |  | Tiêu đề |
| 5 | Slug | Varchar(255) |  | Tên không dấu |
| 6 | Content | Longtext |  | Nội dung |
| 7 | Image | Varchar(255) |  | Ảnh |
| 8 | Status | Int(11) |  | Trạng thái |
| 9 | Is\_hot | Int(11) |  | Nổi bật |

* Bảng **Pages** (Trang):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Name | Varchar(255) |  | Tên |
| 3 | Slug | Varchar(255) |  | Tên không dấu |
| 4 | Content | Varchar(255) |  | Nội dung |
| 5 | Status | Int(11) |  | Trạng thái |

* Bảng **Menus** ():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Lable | Varchar(255) |  | Tên |
| 3 | URL | Varchar(255) |  | Đường dẫn |
| 4 | Position | Int(11) |  | Vị trí |

* Bảng **Slides** ():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Name | Varchar(255) |  | Tên |
| 3 | Image | Varchar(255) |  | Ảnh slide |
| 4 | URL | Int(11) |  | Đường dẫn |

* Bảng **Options** (Cài đặt):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | Id | Int(10) | Khóa chính |  |
| 2 | Name | Varchar(255) |  | Tên |
| 3 | Value | Text |  | Tên không dấu |

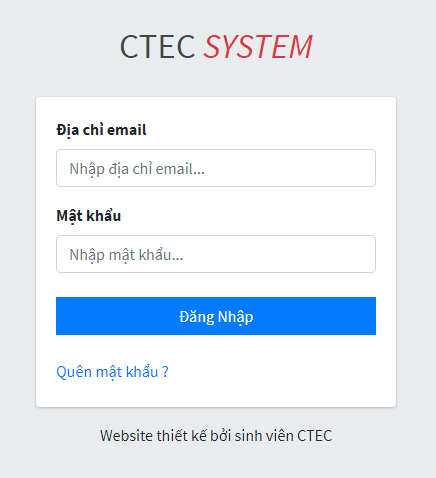
# CHƯƠNG 4: CHỨC NĂNG VÀ GIẢI THUẬT

## 4.1 Các chức năng trang quản lý

* Quản lý hệ thống
  + Đăng nhập hệ thống
  + Đổi mật khẩu
* Quản lý Bài viết
* Cập nhật (Thêm, Sửa, Xóa) Bài viết
* Tìm kiếm bài viết theo tên, người viết, ngày viết
* Quản lý Trang
* Cập nhật (Thêm, Sửa, Xóa) Trang
* Quản lý người dùng
* Cập nhật (Thêm, Sửa, Xóa) Thông tin người dùng
* Tìm kiếm thông tin người dùng
* Quản lý Loại tin
* Cập nhật (Thêm, Sửa, Xóa) Loại tin
* Quản lý Menu
* Cập nhật (Thêm, Sửa, Xóa) Menu
* Quản lý slides
* Cập nhật (Thêm, Sửa, Xóa) Slide
* Quản lý Cài đặt
* Sửa title Website
* Sửa description Website
* Sửa footer Website
* Sửa icon Website
* Sửa logo Website
* Sửa banner Website

## 4.2 Giải thuật trang quản lý

### 4.2.1 Đăng nhập hệ thống

****

Hình 4.1: Đăng nhập hệ thống

Giải thuật:

**Nhập email đăng nhập**

**Nhập mật khấu**

scscc

**Email = Email CSDL**

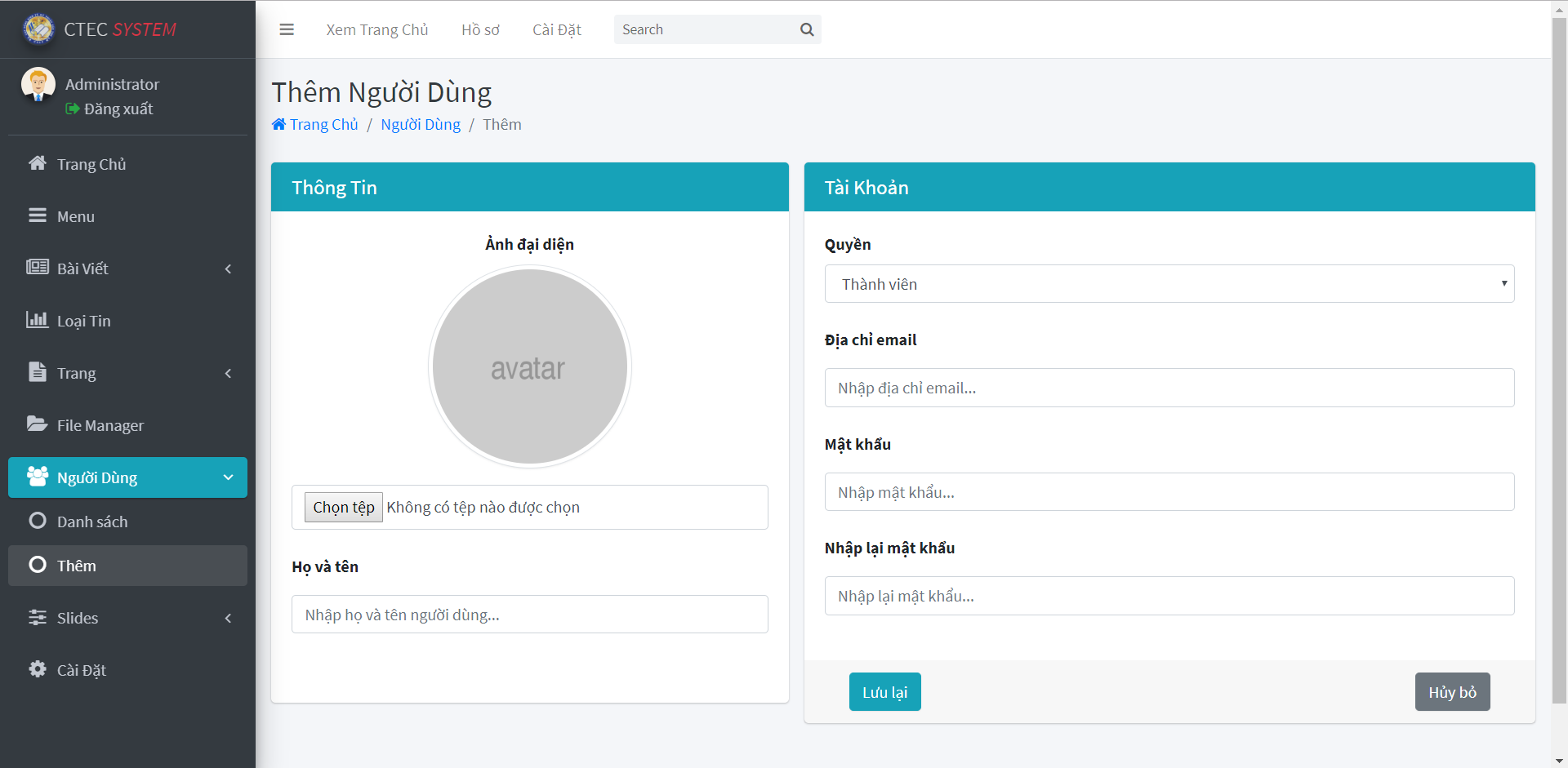
**Mật khẩu = Mật khẩu CSDL**

**Đăng nhập vào hệ thống**

Sai

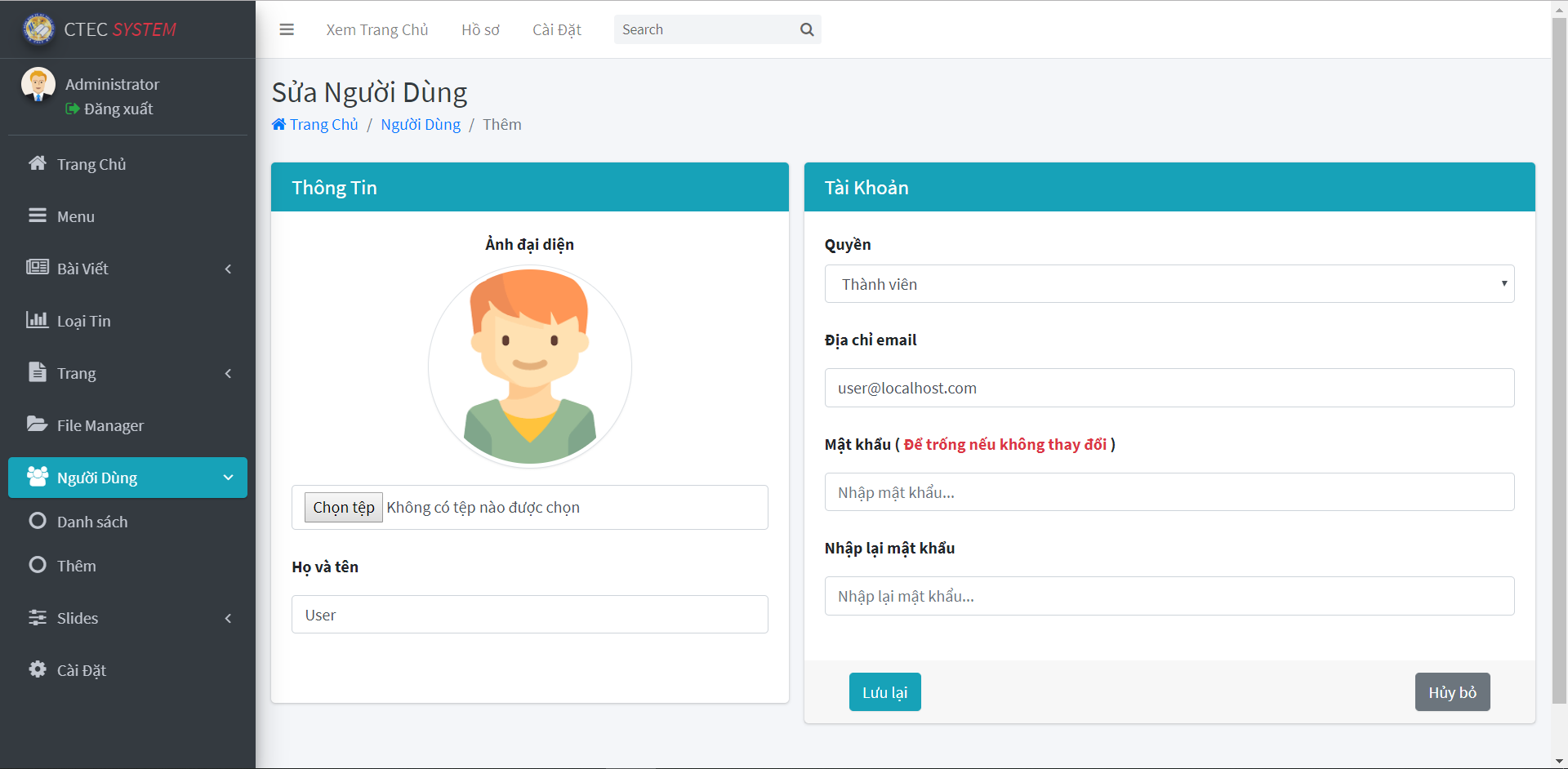
Đúng

### 4.2.2 Thêm người dùng



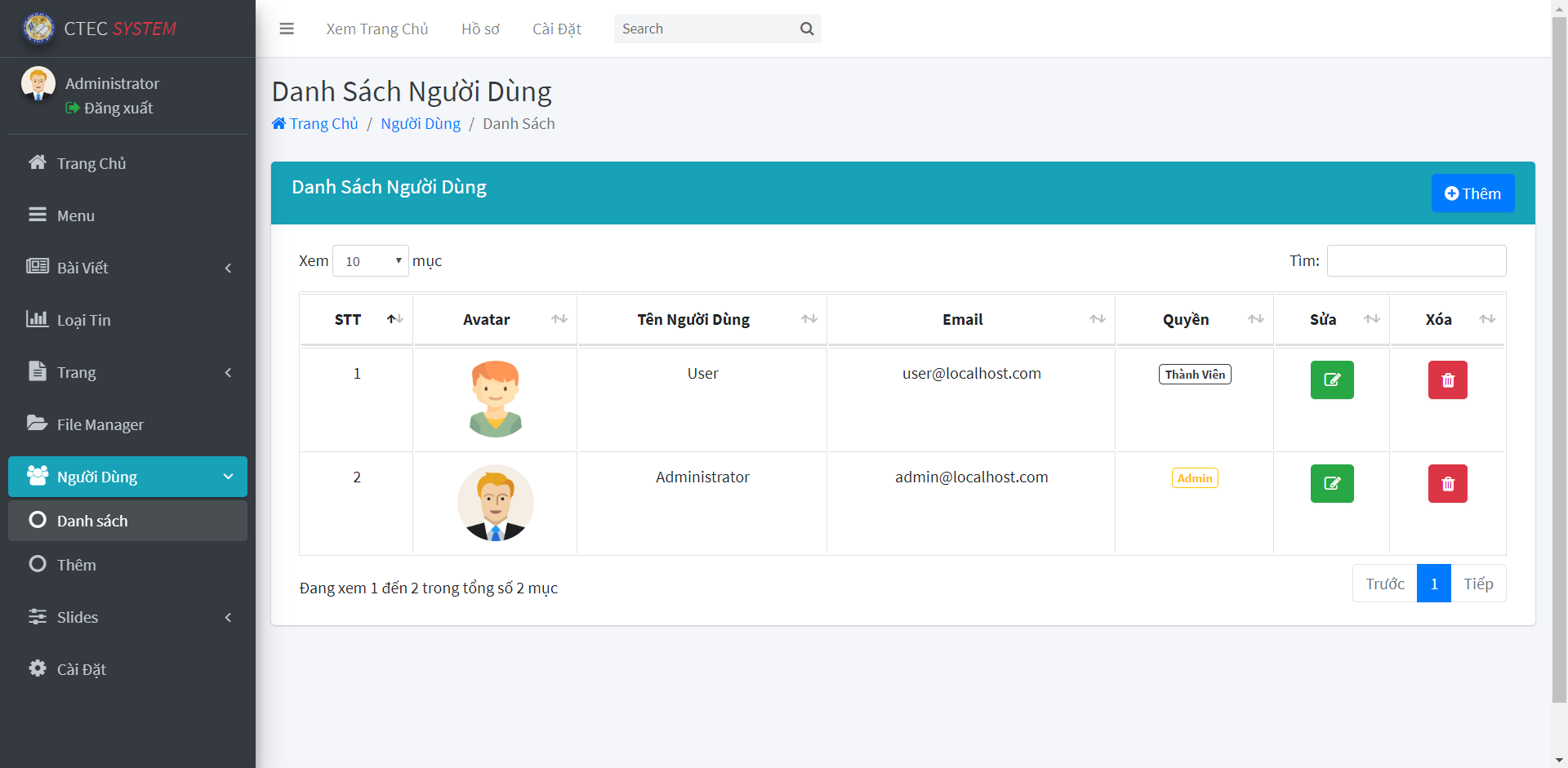
Hình 4.2: Thêm người dùng

### 4.2.3 Thay đổi thông tin người dùng

****

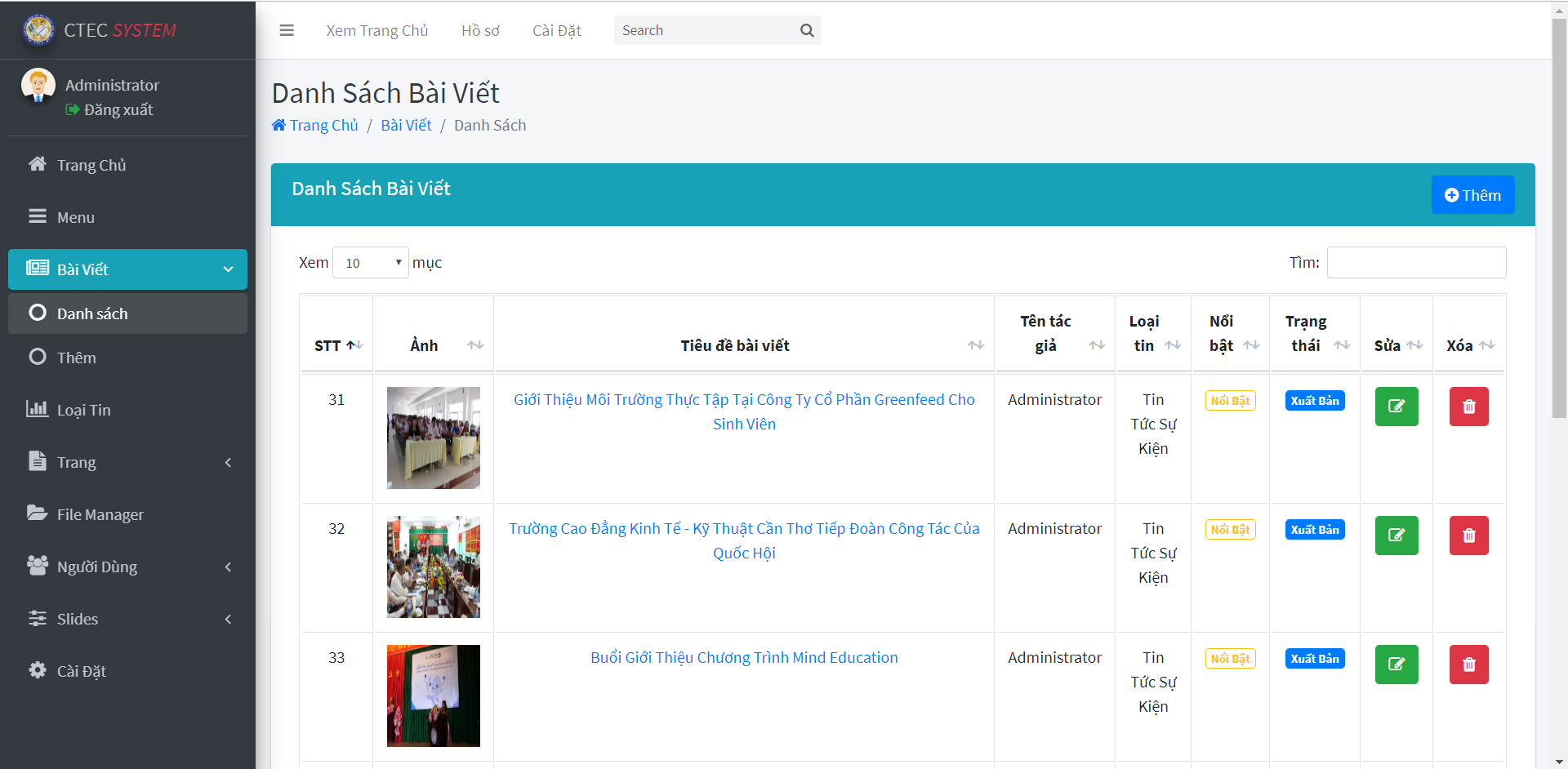
Hình 4.3: Sửa thông tin người dùng

### 4.2.4 Danh sách người dùng

****

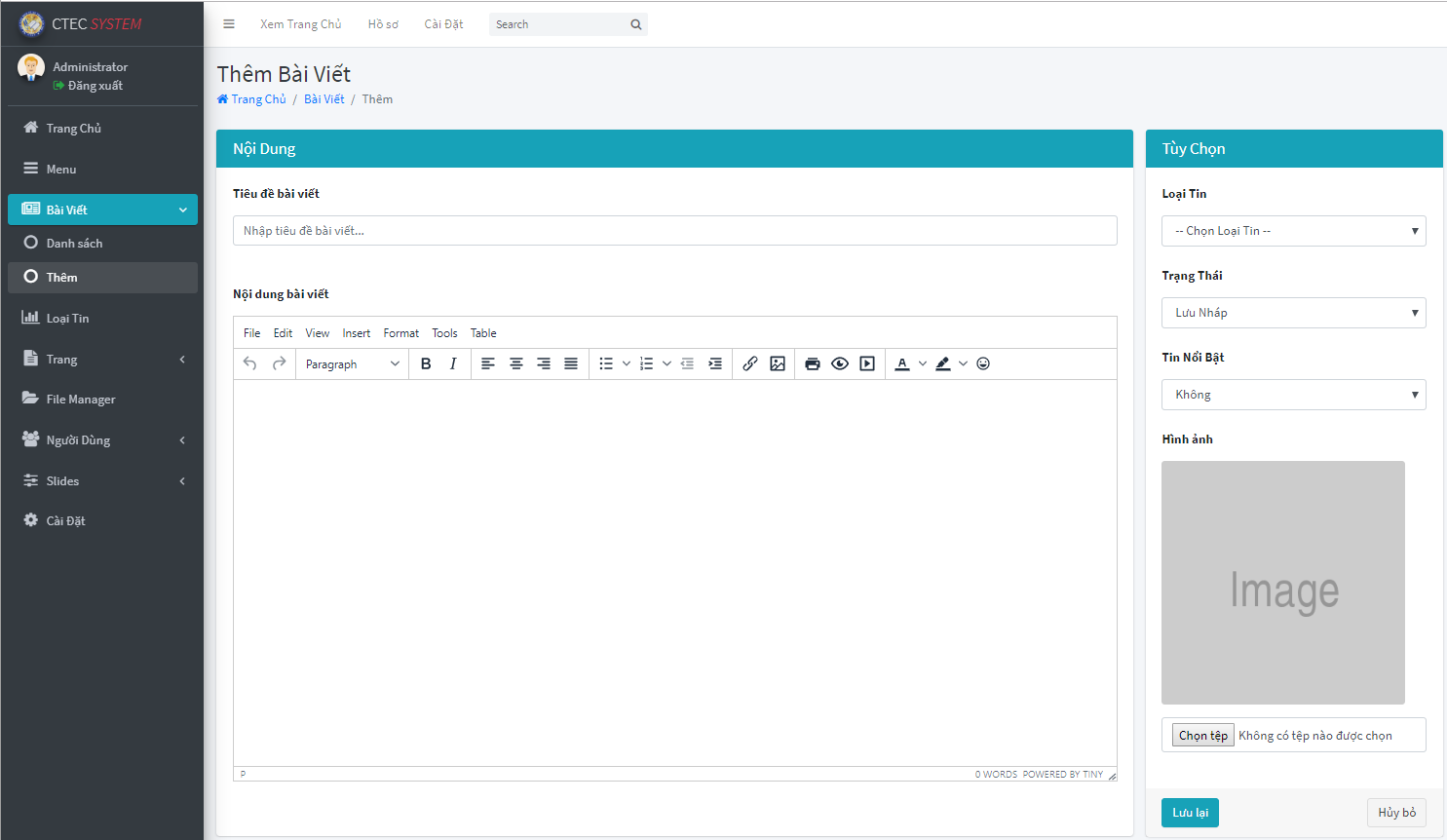
Hình 4.4: Danh sách người dùng

### 4.2.6 Danh sách bài viết

****

### Hình 4.5: Danh sách bài viết

### 4.2.7 Thêm bài viết

****

Hình 4.6: Thêm bài viết

### 4.2.8 Danh sách và thêm loại tin

****

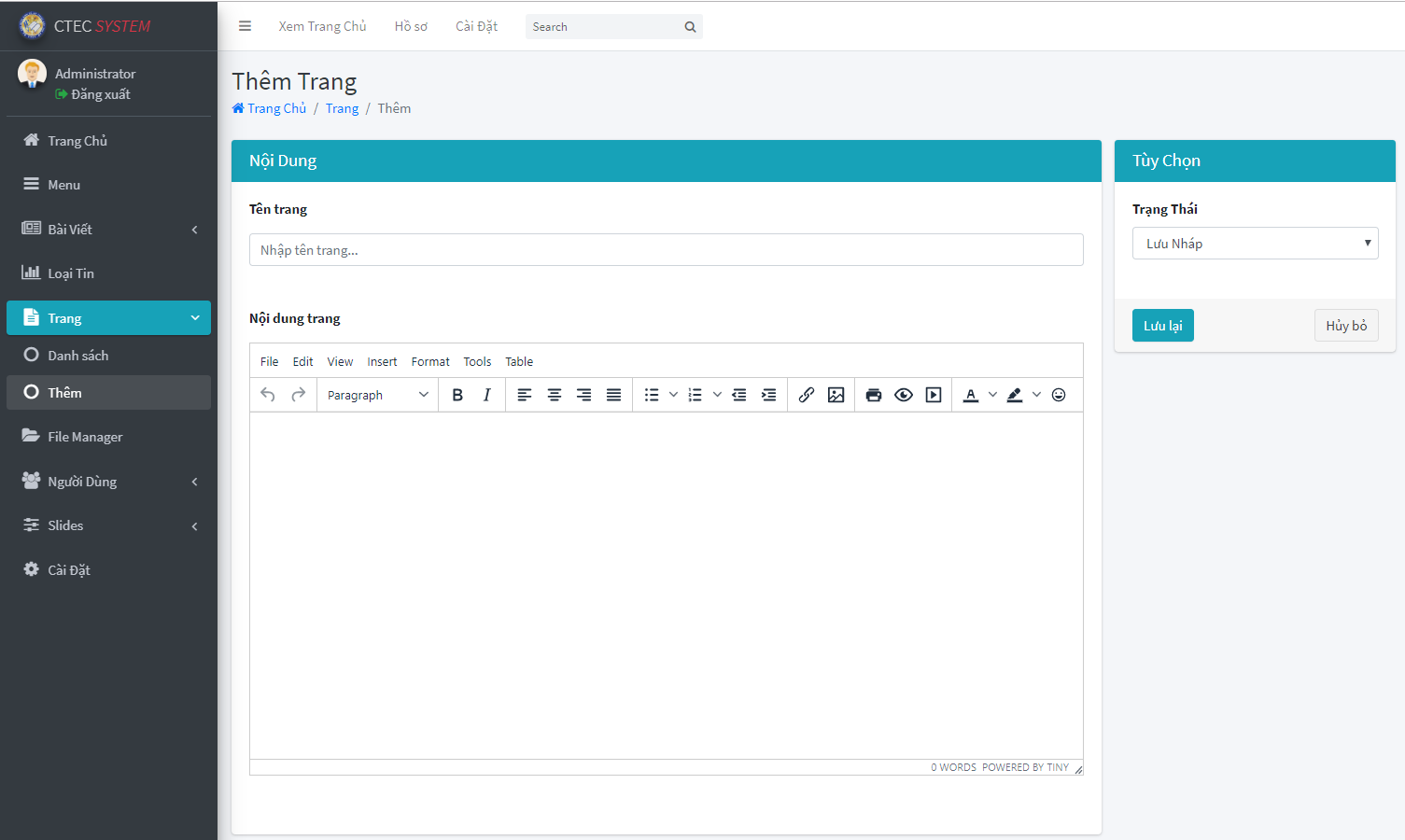
Hình 4.7: Danh sách và thêm loại tin

### 4.2.9 Danh sách trang

****

Hình 4.8: Danh sách trang

### 4.2.10 Thêm trang

****

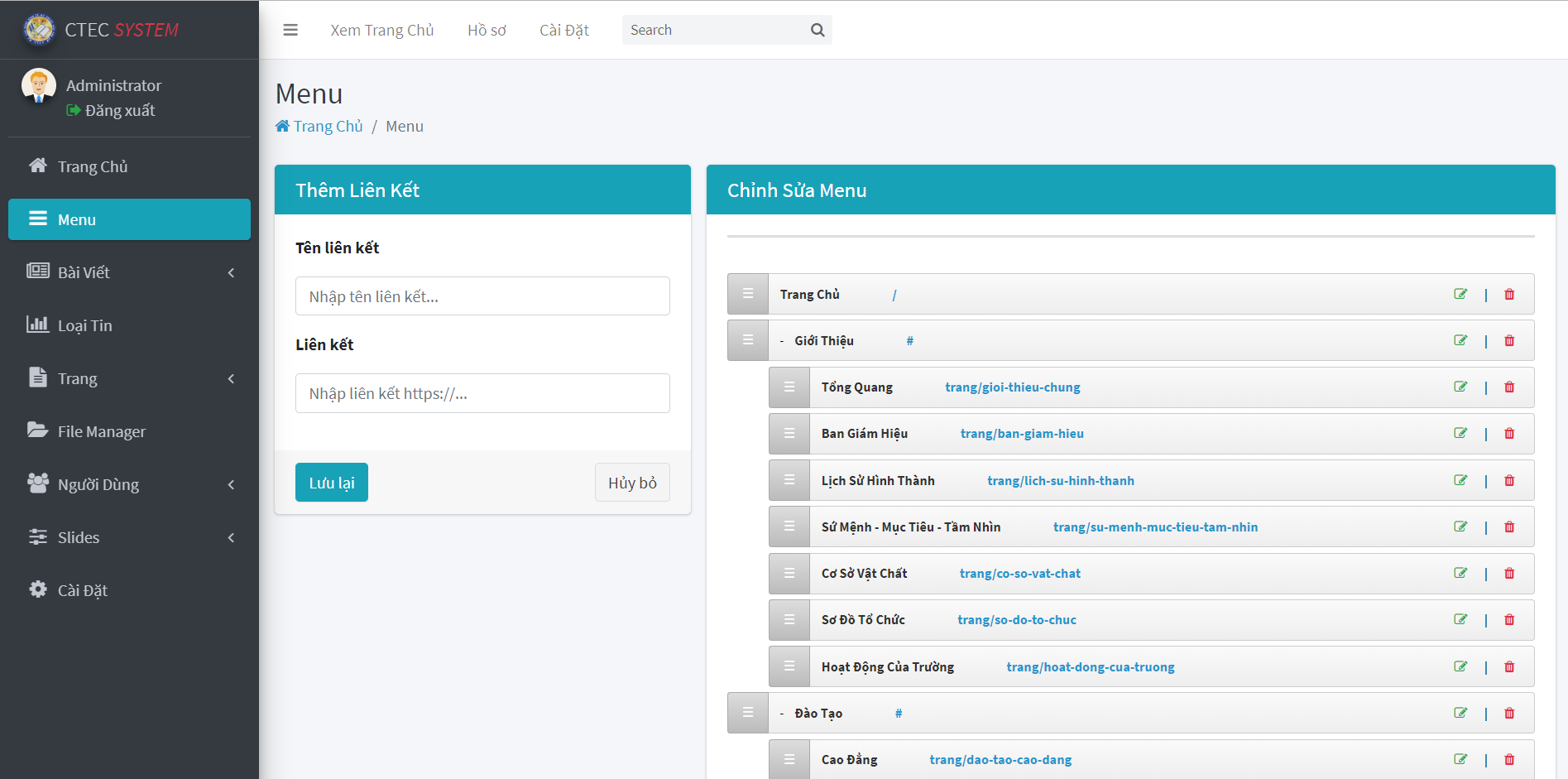
Hình 4.9: Thêm trang

### 4.2.11 File Manager (Quản lý ảnh)

### 

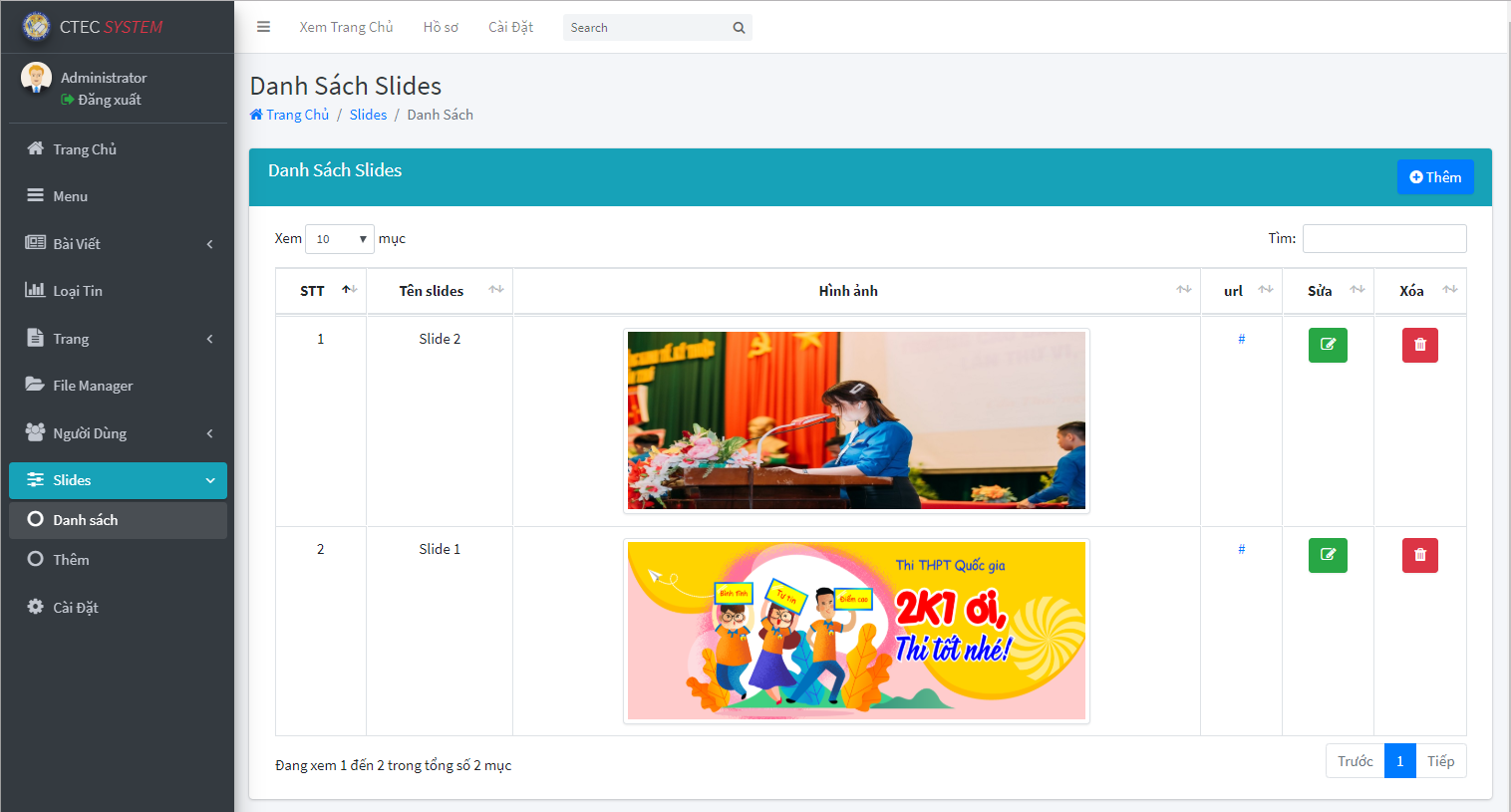
### Hình 4.10: Quản lý ảnh

### 4.2.12 Thêm và chỉnh sửa menu

****

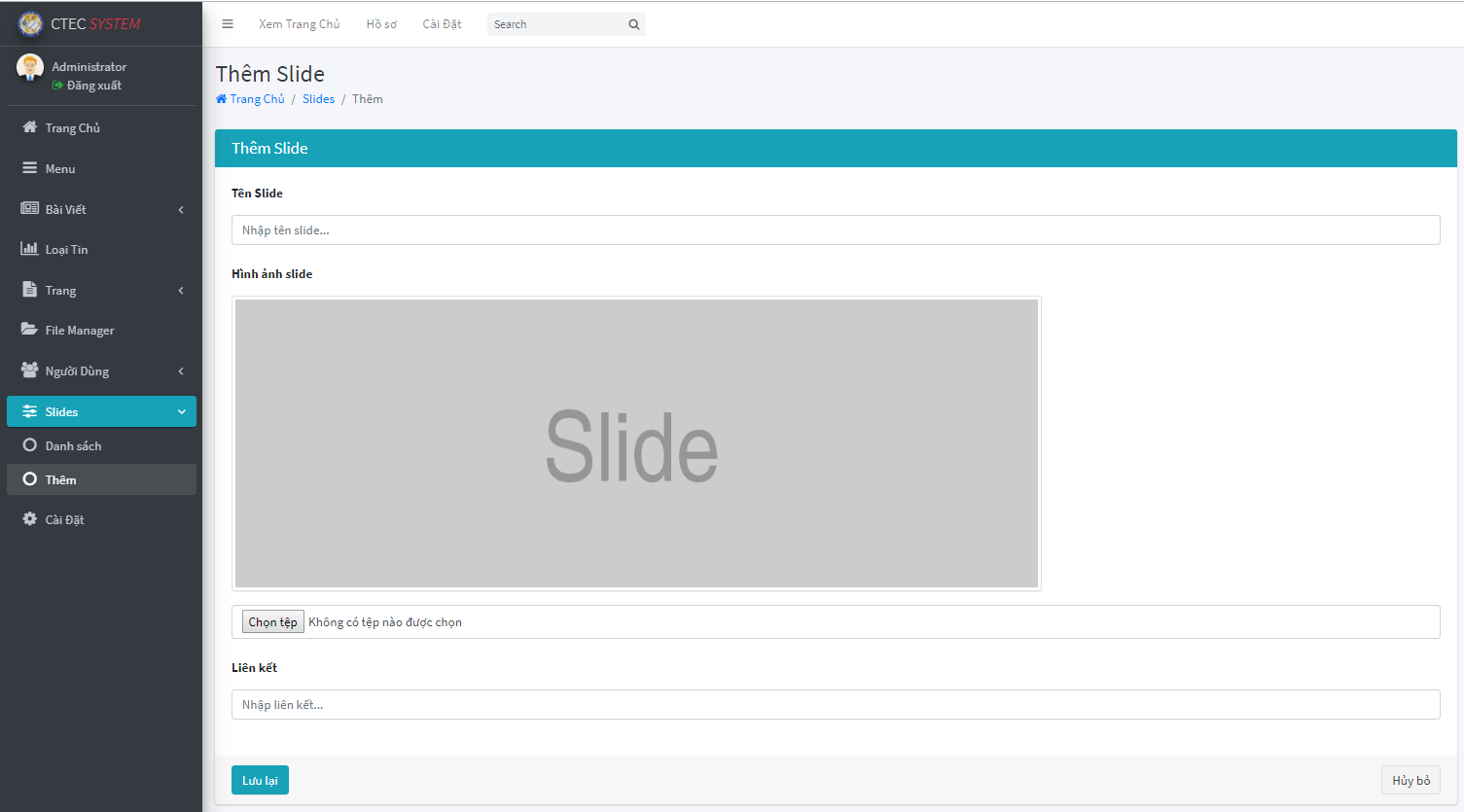
Hình 4.11: Thêm và chỉnh sửa menu

### 4.2.13 Danh sách slide

****

Hình 4.12: Danh sách slide

### 4.2.14 Thêm slide

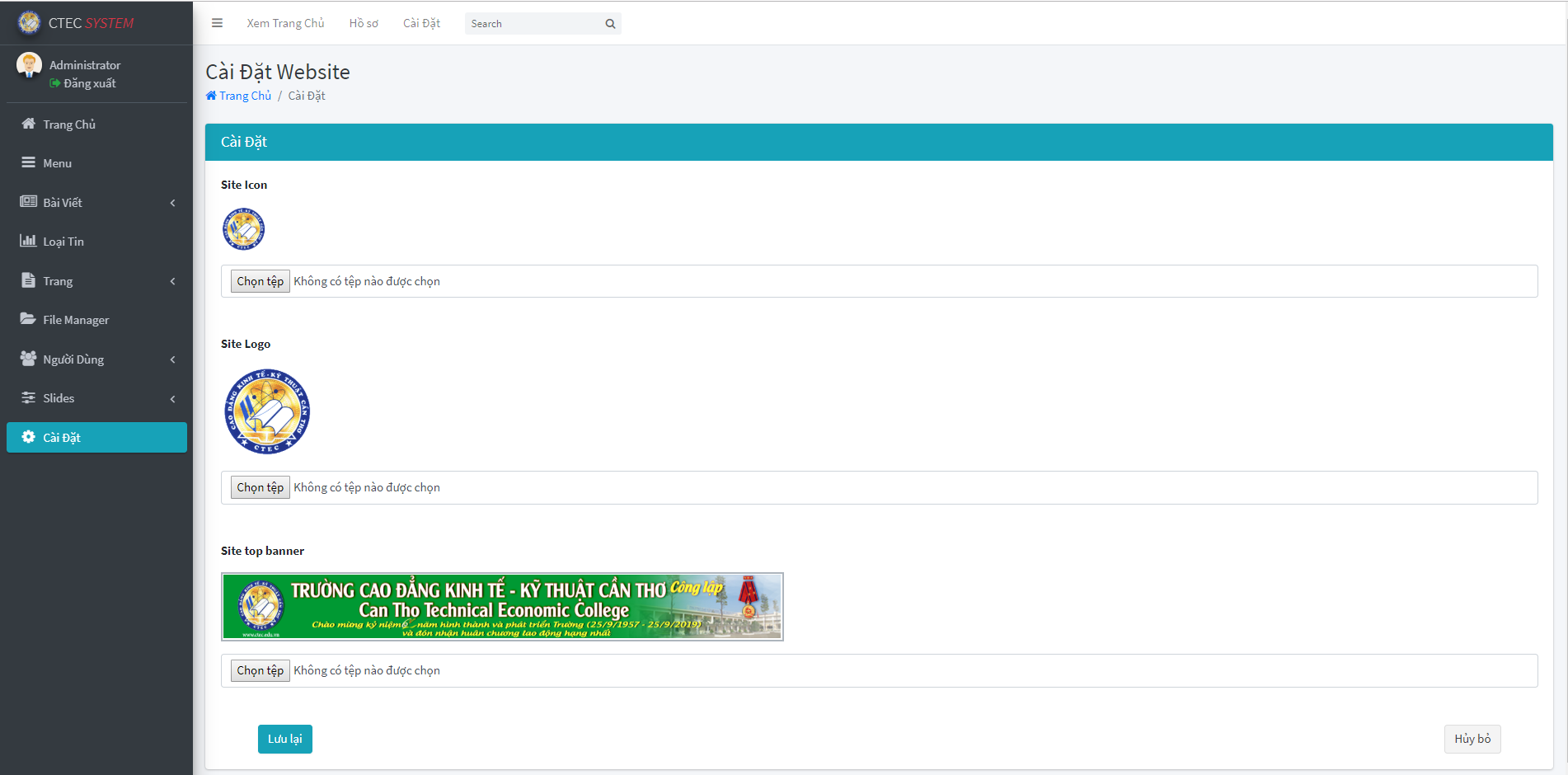
****

Hình 4.13: Thêm slide

### 4.2.15 Sửa Cài đặt

### 

Hình 4.14: Sửa tiêu đề, Miêu tả, footer



Hình 4.15: Sửa Icon, Logo, Banner

# CHƯƠNG 5: GIAO DIỆN WEBSITE

A screenshot of a computer

Description automatically generated

Hình 5.1: Banner, Menu và Slide trang web

A screenshot of a social media post

Description automatically generated

Hình 5.2: Giao diện tin tức theo loại tin và sidebar

A screenshot of a social media post

Description automatically generated

Hình 5.3: Giao diện tin tức theo loại tin và sidebar

A screenshot of a computer

Description automatically generated

Hình 5.4: Danh sách bài viết theo loại tin

A screenshot of a social media post

Description automatically generated

Hình 5.5: Chi tiết bài viết

A screenshot of a social media post

Description automatically generated

Hình 5.6: Trang giới thiệu nhà trường

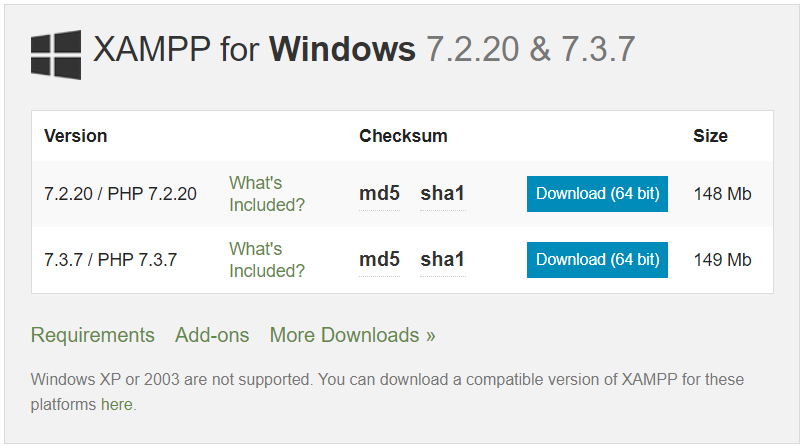
# PHỤ LỤC

* 1. **Cài đặt Xampp**

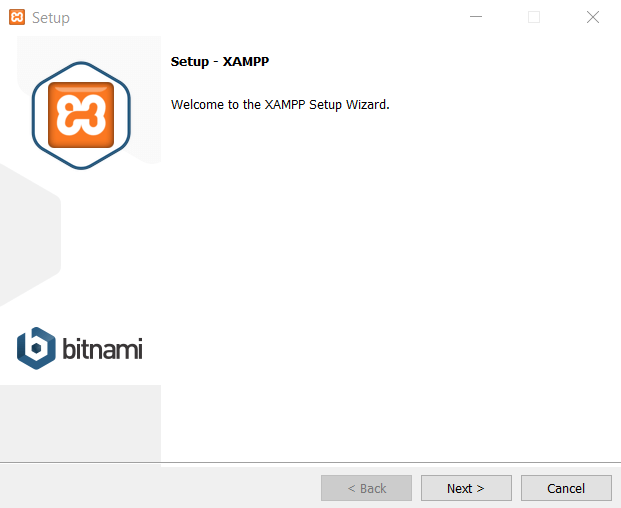
Để tải XAMPP, đầu tiên bạn truy cập vào địa chỉ:

|  |
| --- |
| <https://www.apachefriends.org/download.html> |

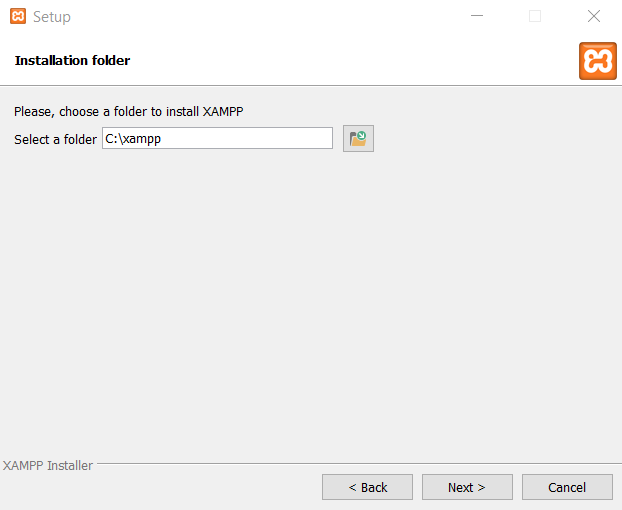
Chọn phiên bản XAMPP phù hợp với hệ điều hành của máy tính bạn đang sử dụng, và bạn nên chọn phiên bản PHP 7.1 trở lên. Lưu ý là XAMPP chỉ có phiên bản cho hệ điều hành 32bit nhưng 64bit vẫn hoạt động bình thường.

****

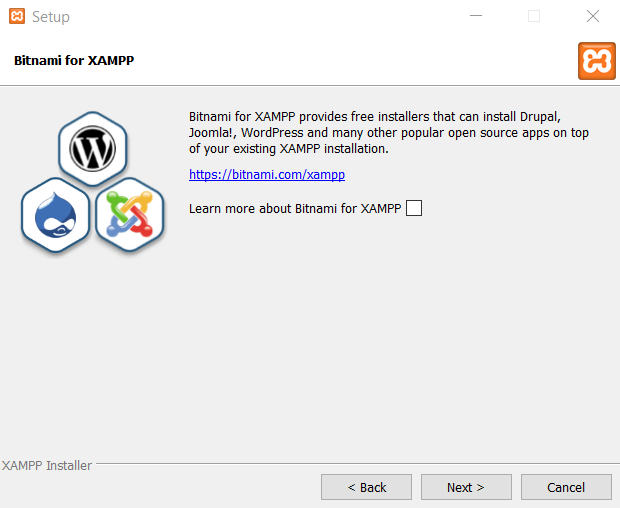
Sau khi tải file cài đặt về xong, hãy chạy nó, sau đó chọn **Next**.



Ở phần chọn đường dẫn, bạn hãy chọn đường dẫn cần lưu cài đặt của XAMPP. Lưu ý rằng đường dẫn này bạn phải nhớ vì khi cài đặt web lên localhost, bạn phải truy cập vào thư mục này. Bạn nên để mặc định là **c:\xampp**. Tiếp tục ấn Next.



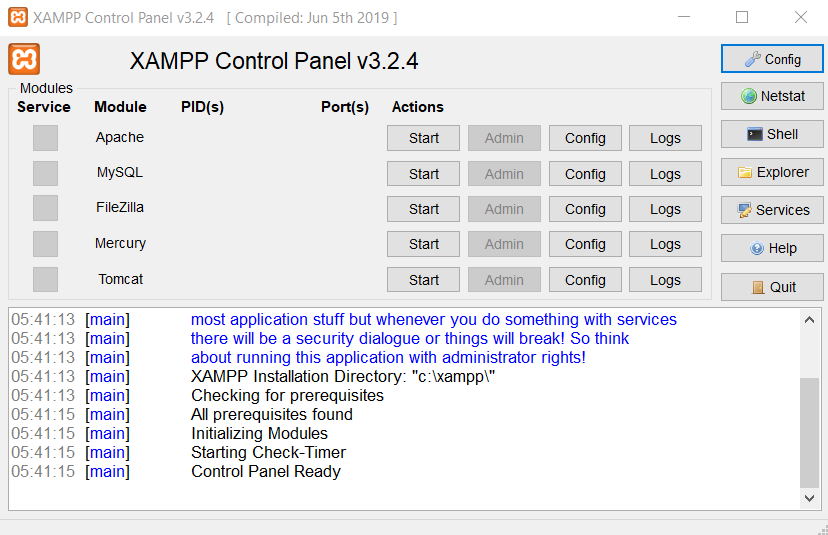
Ở trang kế tiếp, bạn bỏ chọn phần “Learn more about Bitnami for XAMPP“. Và ấn Next 2 lần nữa để bắt đầu quá trình cài đặt XAMPP.



Sau khi cài xong, ấn nút **Finish** để kết thúc cài đặt và mở bảng điều khiển của XAMPP. Tuy nhiên, hãy khởi động lại máy sau khi cài đặt xong để tránh tình trạng không khởi động được localhost.

#### Khởi động Localhost

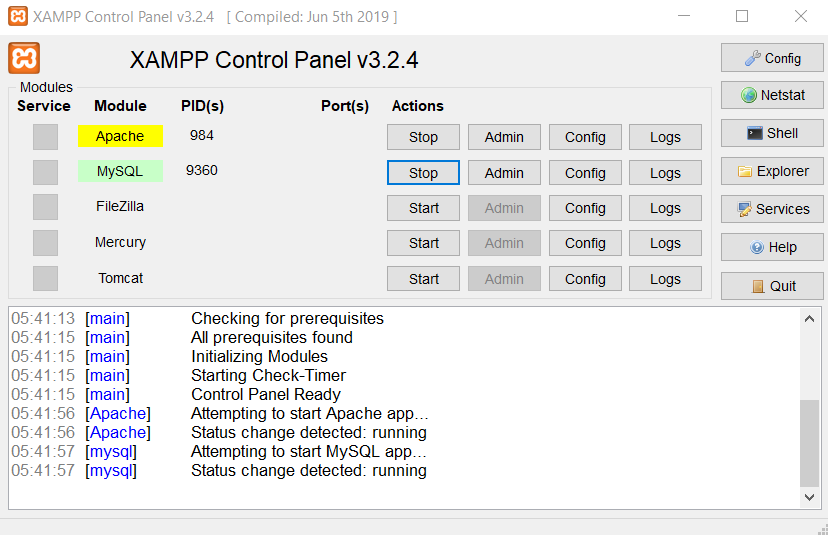
Bây giờ bạn hãy vào thư mục **c:\xampp** và mở file**xampp-panel.exe** lên để bật bảng điều khiển của XAMPP.



*Bảng điều khiển của XAMPP*

Bạn để ý sẽ thấy hai ứng dụng Apache và MySQL có nút **Start**, đó là dấu hiệu bảo 2 ứng dụng này chưa được khởi động, hãy ấn vào nút Start của từng ứng dụng để khởi động Webserver Apache và MySQL Server lên thì mới chạy được localhost.

Nếu cả hai ứng dụng chuyển sang màu xanh như hình dưới là đã khởi động thành công.



Với các bước như trên bạn đã cài thành công Xampp.

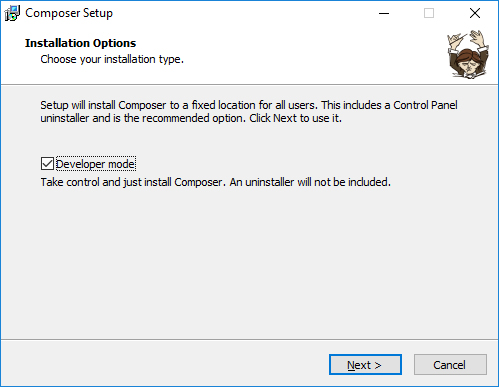
1. **Cài đặt Composer**

* Download trực tiếp từ trang chính: <https://getcomposer.org/download/>

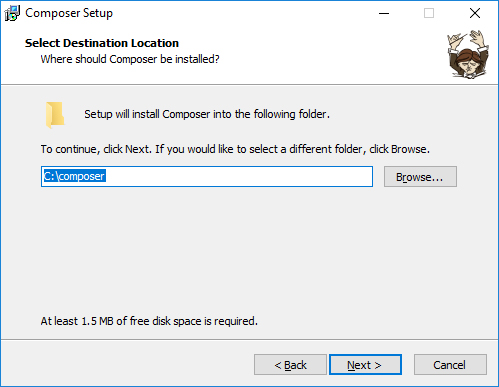
Do trong phạm vi bài hướng dẫn, composer sẽ được sử dụng cùng với Xampp, nên các bạn lựa chọn download file về cài đặt nhe.

Trường hợp các bạn không sử dụng chung với Xampp (ví dụ tự cài đặt từ VPS, Vagrant, ...), thì có thể chọn cách cài đặt bằng Command-line.

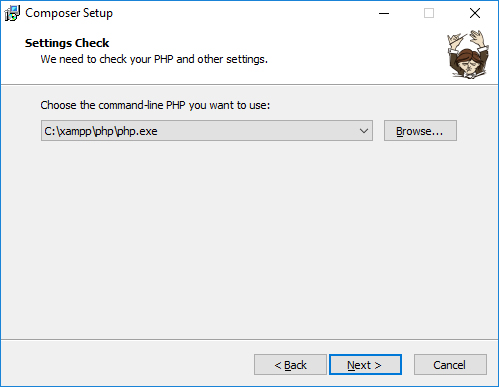
* Double click vào file vừa download, tiến hành cài đặt theo hình bên dưới.
* Click "Next" để tiến hành cài đặt.
* Lưu ý: Trong quá trình cài đặt, nếu có xuất hiện lỗi nào đó thì có thể bỏ qua bằng cách click "Next" để tiếp tục cài.



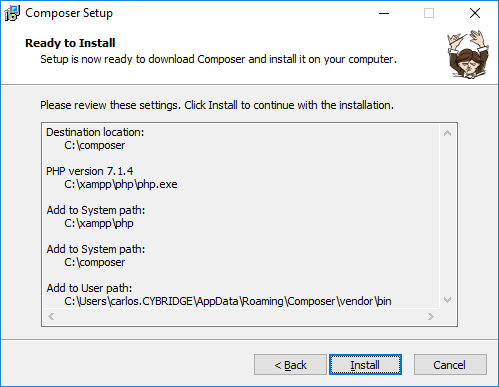
* Giữ mặt định để chọn folder cài đặt và tiếp tục click "Next".



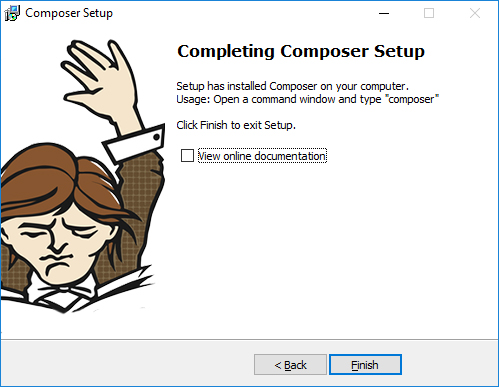
* Chọn khu vực chứa file chạy PHP và tiếp tục Click "Next".



* Xem lại lựa chọn lần cuối trước khi Click "Install" để tiến hành cài đặt.



* Tới đây là kết thúc quá trình cài đặt, Click "Finish" để kết thúc.



1. **Cài đặt Laravel**

### 3.1 Yêu cầu cấu hình

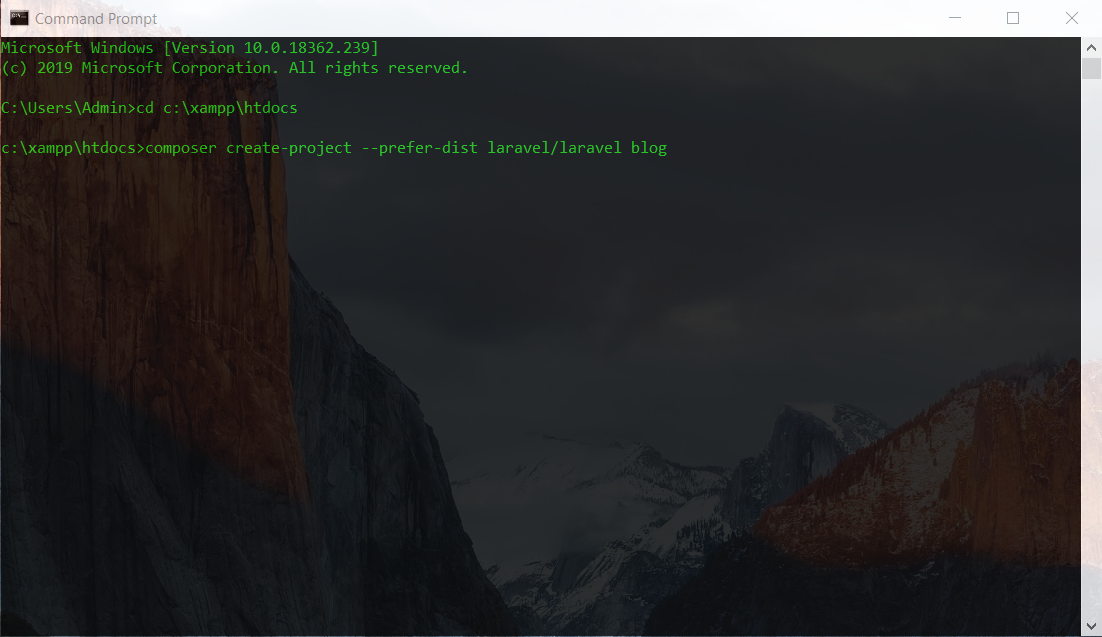
Để có thể cài Laravel, yêu cầu bắt buộc như sau:

* PHP >= 7.1
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension

### 3.2 Hướng dẫn cài đặt Laravel

Cài đặt **Laravel** thông qua **Composer:**

Chúng ta sẽ di chuyển thẳng vào thư mục htdocs của XAMPP, tại đây các bạn mở cửa sổ lệnh (như trên) và gõ dòng lệnh sau:



Trong đó blog chính là tên thư mục **laravel project** của bạn. Vậy là chúng ta đã cài đặt xong **Laravel**rồi đấy.