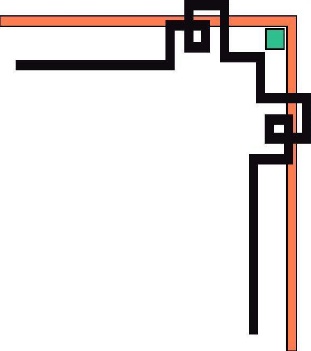
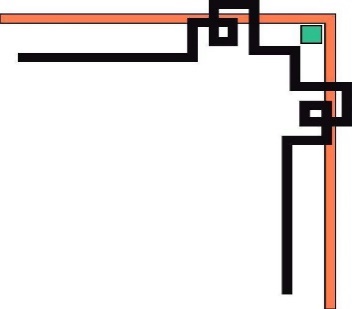
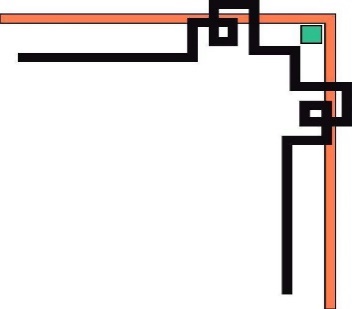
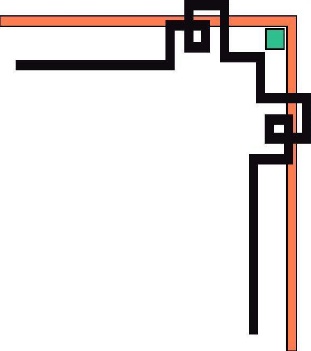
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC CẦN THƠ**

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

**LUẬN VĂN TỐT NGHIỆP NGÀNH HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI: HỆ THỐNG QUẢN LÝ NHÀ TRỌ DÀNH CHO SINH VIÊN TRÊN ĐỊA BÀN THÀNH PHỐ CẦN THƠ**



**NÀM**

**GIÁO VIÊN HƯỚNG DẪN: Th.S *Nguyễn Minh Khiêm***

**SINH VIÊN THỰC HIỆN: NGUYỄN MINH Ý**

**MSSV: B1411457**

**CHUYÊN NGÀNH: Hệ Thống Thông Tin**

**Khóa 40**

**MÃ LỚP: 1495A2**

**Học Kỳ II – Năm Học: 2017 – 2018**

**LỜI CÁM ƠN**

🙢☸🙠

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

🙢☸🙠

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

🙢☸🙠

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**TÓM TẮT**

**ABSTRACT**

**PHẦN GIỚI THIỆU**

1. **ĐẶT VẤN ĐỀ:**

**Cần Thơ** là thành phố lớn, hiện đại và phát triển nhất ở Đồng bằng Sông Cửu Long Là thành phố lớn thứ tư cả nước, đồng thời là một đô thị loại một, thành phố trưc thuộc Trung Ương. Không chỉ là là trung tâm kinh tế của vùng Đồng bằng Sông Cửu Long, Cần Thơ đồng thời là trung tâm giáo dục của vùng Đồng bằng Sông Cửu Long với hơn 25 trường đại học, cao đẳng, trung cấp chuyên nghiệp đóng tại địa bàn thành phố Cần Thơ. Trong đó lớn nhất là Trường đại học Cần Thơ cơ sở đào tạo đại học và sau đại học trọng điểm của Nhà nước ở Đồng bằng Sông Cửu Long, là trung tâm văn hóa - khoa học kỹ thuật  của vùng. Chính vì vậy số lượng sinh viên theo học tại các trường đại học ở Cần Thơ sẽ tăng lên theo mỗi năm. Đa số sinh viên đều học tập xa nhà, nhu cầu tất yếu của họ là tìm được nhà trọ để học tập và làm việc. Nhưng nếu tìm kiếm theo cách thông thường sẽ tốn rất nhiều thời gian cũng như công sức để tìm được một nhà trọ thích hợp để học tập và làm việc.

Chính vì sự bất tiện và khó khăn trên, em quyết định thực hiện đề tài “**Xây dựng Hệ thống quản lý nhà trọ dành cho  Sinh Viên trên địa bàn thành phố Cần Thơ”.** Trang WEB ra đời nhằm hỗ trợ sinh viên trong việc tìm kiếm nhà trọ được dễ dàng hơn, cũng như giúp các chủ nhà trọ có một kênh để có thể đăng tải, giới thiệu các thông tin nhà trọ của mình đến với sinh viên cũng như mọi người có nhu cầu tìm nhà trọ. Góp phần đem lại sự thuận tiện tối đa cho người cần tìm nhà trọ nói chung và sinh viên nói riêng.

1. **LỊCH SỬ ĐỀ TÀI:**

Tại Việt Nam nói chung và thành phố Cần Thơ nói riêng, từ những năm đầu của thế kỷ XX nhu cầu về nhà ở ngày càng nâng cao. Đặc biệt, đối với sinh viên, mỗi năm số lượng sinh viên đến để học tập tại thành phố Cần Thơ ngày càng cao. Để đáp ứng cho nhu cầu rất lớn về chỗ ở, nhiều nhà trọ đã được xây dựng lên, việc kinh doanh nhà trọ ngày càng phát triển sầm uất. Chính vì vậy, việc tìm kiếm một nhà trọ phù hợp với nhu cần của bản thân để làm việc và học tập trở thành một vấn đề cấp bách cần thực hiện ngay. Và phương pháp hiệu quả nhất giải quyết vấn đề này đó là xây dựng các Website cũng như các ứng dụng tìm kiếm nhà trọ trên smartphone.

Trên thực tế đã có rất nhiều Website hỗ trợ việc tìm kiếm nhà trọ trực tuyến, nhưng đa phần chỉ là đăng tải thông tin dạng chữ, ít có thông tin chi tiết về nhà trọ như tiện ích, vị trí, … cũng như vị trí của nhà trọ khá mơ hồ, đa phần chỉ là tìm kiếm nhà trọ nói chung chưa có sự hỗ trợ tối đa cho các bạn sinh viên tìm kiếm được một nhà trọ phù hợp với nhu cầu học tập của các bạn.

1. **MỤC TIÊU ĐỀ TÀI:**

Luận văn phải xây dựng ***“Hệ thống quản lý nhà trọ dành cho  Sinh Viên trên địa bàn thành phố Cần Thơ”*** hỗ trợ sinh viên cũng như người dùng với nhiều chức năng như:

* Trang người dùng: không những cung cấp đầy đủ các thông tin về nhà trọ, thông tin cho thuê, vị trí nhà trọ mà còn cho phép đặt giữ chỗ trực tuyến. Đặc biệt, người dùng có thể tìm kiếm nhà trọ theo nhiều tiêu chí khác nhau và đặc biệt hệ thống sẽ gợi ý những nhà trọ có khoảng cách gần với các trường đại học nhằm hỗ trợ tối đa cho các bạn sinh viên. Người dùng có thể bình luận nhận xét về nhà trọ giúp các bạn sinh viên có cái nhìn tổng quan hơn trước khi lựa chọn nơi làm việc và học tập.
* Trang Admin: Website cung cấp các chức năng cập nhật thông tin đầy đủ về nhà trọ, loại nhà trọ, các trường đại học, tài khoản người dùng, tiện ích nhà trọ, bài viết, bình luận. Quan trọng hơn, trang web cho phép định vị chính xác vị trí các trường đại học cũng như vị trí các nhà trọ hỗ trợ người dùng có một cái nhìn trực quan hơn về vị trí nhà trọ.
* Ngoài ra, website có cấu trưc, chuyên mục rõ rang, giao diện dễ nhìn, thân thiện với người dùng, tốc độ truy cập nhanh. Website có nội dung chính xác, rõ ràng, đáp ứng nhu cầu của người dùng về chức năng hệ thống và có hệ thống phân quyền truy cập cho các đối tượng người dùng, nhằm tăng cường khả năng bảo mật thông tin.

1. **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU:**
2. **Đối tượng nghiên cứu:**

Đối tượng bao gồm các Website, Forum, phần mềm, đề tài luận văn, đồ án, tiểu luận về lĩnh vực cho thuê và tìm kiếm nhà trọ. Ngoài ra còn có các nghiệp vụ về cho thuê nhà trọ, các tiêu chí tìm kiếm nhà trọ của sinh viên.

1. **Phạm vi nghiên cứu:**

* Về lý thuyết:
* Tìm hiểu, phân tích và thiết kế hệ thống dữ liệu qua CDM, DFD các cấp.
* Nghiên cứu cơ sở lý thuyết về ngôn ngữ lập trình PHP, hệ quản trị cơ sở dữ liệu MySQL, công cụ hỗ trợ thiết kế powerdesigner, phần mềm XAMPP, Laravel Framework, Google Map API
* Về người dùng và chức năng:
* *Chủ Nhà Trọ/ Người cần đăng tin cho thuê:*

Chủ nhà trọ/ Người cần đăng tin cho thuê đăng nhập vào hệ thống bằng tài khoản đã được cấp để thực hiện các chức năng sau:

* Đăng thông tin nhà trọ.
* Quản lý nhà trọ đã đăng kí trên hệ thống.
* Xem thông tin nhà trọ.
* Cập nhật thông tin nhà trọ.
* Cập nhật tiện ích nhà trọ.
* Quản lý thông tin tài khoản.
* Xóa nhà trọ.
* Quản lý đặt xem phòng.
* *Sinh viên/ Người cần tìm nhà trọ:*
* Tìm kiếm nhà trọ theo nhiều tiêu chí.
* Đăng tin tìm bạn ở ghép.
* Quản trị thông tin tài khoản.
* Đặt lịch hẹn xem phòng.
* Bình luận, đánh giá nhà trọ.
* Xem chỉ đường đến nhà trọ.
* Xem vị trí nhà trọ trên bản đồ.
* *Admin/ Quản trị viên:*
* Quản lý nhà trọ trên hệ thống.
* Cập nhật các điểm trường đại học trên địa bàn thành phố.
* Quản tri tài khoản.
* Quản trị bài đăng.
* Quản trị danh mục.
* Quản trị bình luận.
* Gửi Email tới người dùng.
* Về kỹ thuật:
* Đảm bảo tính chính xác, xử lý được nhiều nghiệp vụ thực tế, hiệu quả cao.
* Website đáp ứng các nhu cầu về tìm kiếm, định vị vị trí, dễ sử dụng, tốc độ xử lý nhanh, giao diện thân thiện, dễ sử dụng.
* Hỗ trợ tối đa người dùng trong việc tìm kiếm nhà trọ cũng như đăng thông tin cho thuê trên hệ thống.

1. **NỘI DUNG NGHIÊN CỨU:**

Nghiên cứu về nghiệp vụ cho thuê, tìm kiếm nhà trọ, cơ sở dữ liệu về quản lý một hệ thống quản lý nhà trọ sinh viên. Từ đó, đề tài được mô tả qua các sơ đồ như CDM, DFD các cấp.

Nghiên cứu về các ngôn ngữ lập trình như: PHP, Laravel Framework, CSS, HTML,… Các kỹ thuật lập trình như: Boostrap, Google Map Api,… Sau đó tiến hành lập trình toàn bộ hệ thống website và kiểm thử.

1. **NHỮNG ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI:**
2. **KẾ HOẠCH THỰC HIỆN:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **CÔNG VIỆC** | TUẦN | | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| 1 | Khảo sát hệ thống |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Phân tích yêu cầu |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Thiết kê mô hình |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Cài đặt chương trình |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Nhập liệu và kiểm thử |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Sửa lỗi chương trình |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Viết báo cáo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **BỐ CỤC QUYỂN LUẬN VĂN:**

Luận văn có 3 phần chính:

1. Nhận xét của giáo viên hướng dẫn.
2. Nhận xét của giáo viên phản biện.
3. Lời cám ơn.
4. Tóm tắt.
5. Abstract
6. Mục Lục
7. Phần giới thiệu
8. Phần nội dung
9. Phần kết luận

**PHẦN NỘI DUNG**

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL:**
2. **MySQL là gì?**

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở 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. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix,FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,…

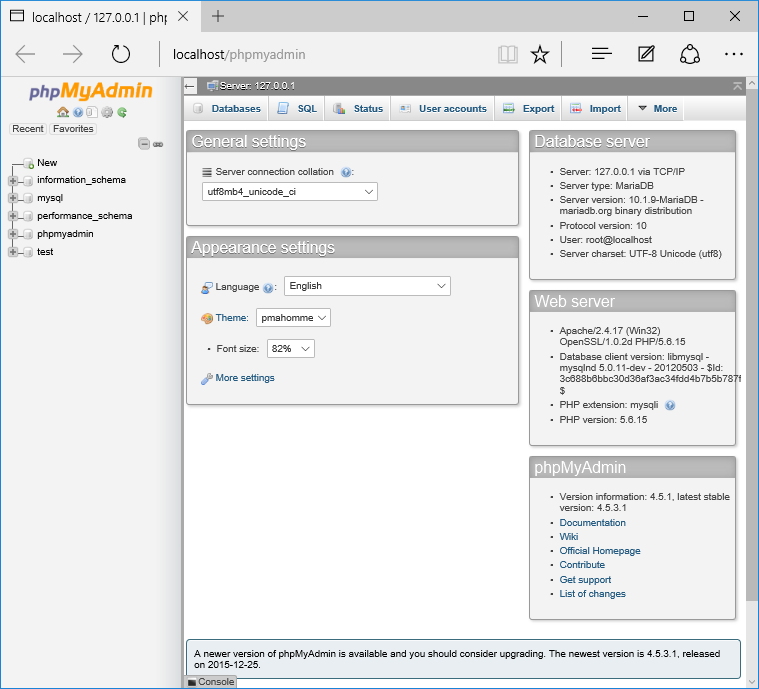
MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

1. **Tìm hiểu về phpMyAdmin:**

**phpMyAdmin** là phần mềm mã nguồn mở được viết bằng ngôn ngữ PHP nhằm giúp người dùng (thường là các nhà quản trị cơ sở dữ liệu hay database administrator) có thể quản lý cơ sở dữ liệu MySQL thông qua giao diện web thay vì sử dụng giao diện cửa sổ dòng lệnh (command line interface).

Sử dụng **phpMyadmin** người dùng có thể thực hiện được nhiều tác vụ khác nhau như khi sử dụng cửa sổ dòng lệnh. Các tác vụ này bao gồm việc tạo, cập nhật và xoá các cơ sở dữ liệu, các bảng, các trường, dữ liệu trên bảng, phân quyền và quản lý người dùng....



1. **NGÔN NGỮ LẬP TRÌNH PHP:**
2. **Tổng quan về PHP:**

**PHP** - viết tắt hồi quy của **"Hypertext Preprocessor"**, là một ngôn ngữ lập trình kịch bản được chạy ở phía server nhằm sinh ra mã html trên client. PHP đã trải qua rất nhiều phiên bản và được tối ưu hóa cho các ứng dụng web, với cách viết mã rõ rãng, tốc độ nhanh, dễ học nên PHP đã trở thành một ngôn ngữ [lập trình web](http://hoclaptrinhweb.org/hoc-thiet-ke-web/lap-trinh-web.html) rất phổ biến và được ưa chuộng.

**PHP** chạy trên môi trường Webserver và lưu trữ dữ liệu thông qua hệ quản trị cơ sở dữ liệu nên PHP thường đi kèm với Apache, MySQL và hệ điều hành Linux (LAMP).

* Apache là một phần mềm web server có nhiệm vụ tiếp nhận request từ trình duyệt người dùng sau đó chuyển giao cho PHP xử lý và gửi trả lại cho trình duyệt.
* MySQL cũng tương tự như các hệ quản trị cơ sở dữ liệu khác (Postgress, Oracle, SQL server...) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.
* Linux: Hệ điều hành mã nguồn mở được sử dụng rất rộng rãi cho các webserver. Thông thường các phiên bản được sử dụng nhiều nhất là RedHat Enterprise Linux, Ubuntu...

1. **Lịch sử phát triển của PHP:**

**PHP** mà chúng ta biết đến ngày nay là sự kế thừa của một sản phẩm có tên là PHP/FI. Được viết bởi Rasmus Lerdorf, bản PHP đầu tiên là một tập hợp đơn giản các Common Gateway Interface (CGI) – một chuẩn giao tiếp giữa client và server – nhị phân viết bằng ngôn ngữ C. Tháng 9/1995, Rasmus mở rộng PHP, thêm vào công cụ có tên là FI (Forms Interpreter), và một số chức năng như chúng ta biết ngày nay. Nó có các biến kiểu Perl và cú pháp cho phép nhúng mã HTML. Cú pháp này, tương tự Perl, nhưng hạn chế hơn, đơn giản và có phần không phù hợp. Trong thực tế, để nhúng mã lệnh PHP/FI vào một tập tin HTML, các lập trình viên phải sử dụng các comment HTML. Và mặc dù phương pháp này không được hoàn toàn đón nhận, FI vẫn tiếp tục phát triển và đem về lợi nhuận thương mại.

**PHP 3.0:** Được Andi Gutmans và Zeev Suraski tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. Lý do chính mà họ đã tạo ra phiên bản này là do họ nhận thấy PHP/FI 2.0 hết sức yếu kém trong việc phát triển các ứng dụng thương mại điện tử. PHP 3.0 như là phiên bản thế hệ kế tiếp của PHP/FI 2.0, và chấm dứt phát triển PHP/FI 2.0.PHP 3.0 cung cấp cho người dùng cuối một cơ sở hạ tầng chặt chẽ dùng cho nhiều cơ sở dữ liệu, giao thức và API khác nhau.Cho phép người dùng có thể mở rộng theo modul.  
=> Chính điều này làm cho PHP3 thành công so với PHP2. Lúc này họ chính thức đặt tên ngắn gọn là 'PHP' ( Hypertext Preprocessor ). PHP 3.0 đã chính thức được công bố vào tháng 6 năm 1998.

**PHP4:** Được công bố năm 2000 tốc độ xử lý được cải thiện rất nhiều, PHP 4.0 đem đến các tính năng chủ yếu khác gồm có sự hỗ trợ nhiều máy chủ Web hơn, hỗ trợ phiên làm việc HTTP, tạo bộ đệm thông tin đầu ra, nhiều cách xử lý thông tin người sử dụng nhập vào bảo mật hơn và cung cấp một vài các cấu trúc ngôn ngữ mới. Với PHP 4, số nhà phát triển dùng PHP đã lên đến hàng trăm nghìn và hàng triệu site đã công bố cài đặt PHP, chiếm khoảng 20% số tên miền trên mạng Internet. Nhóm phát triển PHP cũng đã lên tới con số hàng nghìn người và nhiều nghìn người khác tham gia vào các dự án có liên quan đến PHP như PEAR, PECL và tài liệu kĩ thuật cho PHP.

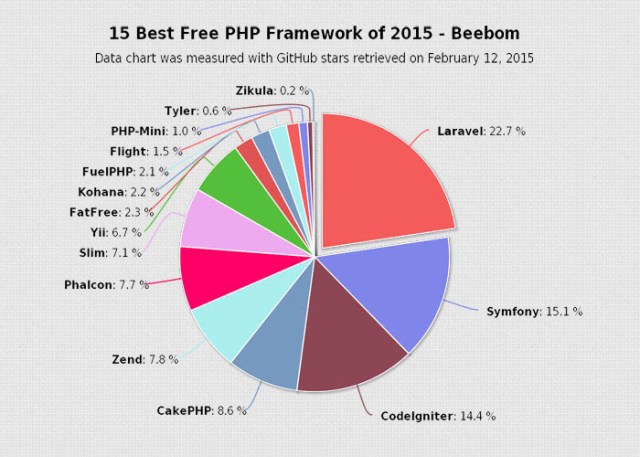
**PHP5:** Bản chính thức đã ra mắt ngày 13 tháng 7 năm 2004 sau một chuỗi khá dài các bản kiểm tra thử bao gồm Beta 4, RC 1, RC2, RC3. Mặc dù coi đây là phiên bản sản xuất đầu tiên nhưng PHP 5.0 vẫn còn một số lỗi trong đó đáng kể là lỗi xác thực HTTP.

**PHP6:** Hiện nay phiên bản tiếp theo của PHP đang được phát triển, PHP 6 bản sử dụng thử đã có thể được download tại địa chỉ snaps.php.net. Phiên bản PHP 6 được kỳ vọng sẽ lấp đầy những khiếm khuyết của PHP ở phiên bản hiện tại.

1. **KHÁI QUÁT VỀ PHP FRAMEWORK:**

Để phát triển một website hệu quả, thuận lợi và không tốn công xây dựng một hệ thống từ đầu thì các lập trình viên thường sử dụng PHP Framewwork. PHP Framewwork là một thư viện các hàm xây dựng sẵn, mục địch chính là hỗ trợ cho việc phát triển website bằng ngôn ngữ PHP. Khi sử dụng Framewwork, website sẽ có cấu trúc hệ thống rõ ràng và chặt chẽ, thời gian để phát triển website nhanh hơn và giúp hệ thống bảo mật hơn. Một PHP Framework thường được xây dựng trên kiến trúc MVC.

Ngoài ra, khi phát triển các website động dựa vào công nghệ PHP – MySQL, các công ty phát triển phần mềm đều sử dụng các PHP Framework như: Laravel, Zend, CakePHP, Symfony,…



1. **LARAVEL FRAMEWORK**
2. **Giới thiệu về Laravel Framwork:**

**Laravel** là một PHP framework  mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhắm vào mục tiêu hỗ trợ phát triển các ứng dụng web theo kiếm trúc model-view-controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng , một hệ thống đóng gói modular và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

Vào khoảng Tháng 3 năm 2015, các lập trình viên đã có một cuộc bình chọn PHP framework phổ biến nhất, Laravel đã giành vị trí quán quân cho PHP framework phổ biến nhất năm 2015, theo sau lần lượt là Symfony2, Nette, CodeIgniter, Yii2 vào một số khác. Trước đó, Tháng 8 2014, Laravel đã trở thành project PHP phổ biến nhất và được theo dõi nhiều nhất trên Github.

Laravel được phát hành theo giấy phép MIT, với source code được lưu trữ tại [Github](https://github.com/laravel/laravel).

1. **Lịch sử phát triển của Laravel:**

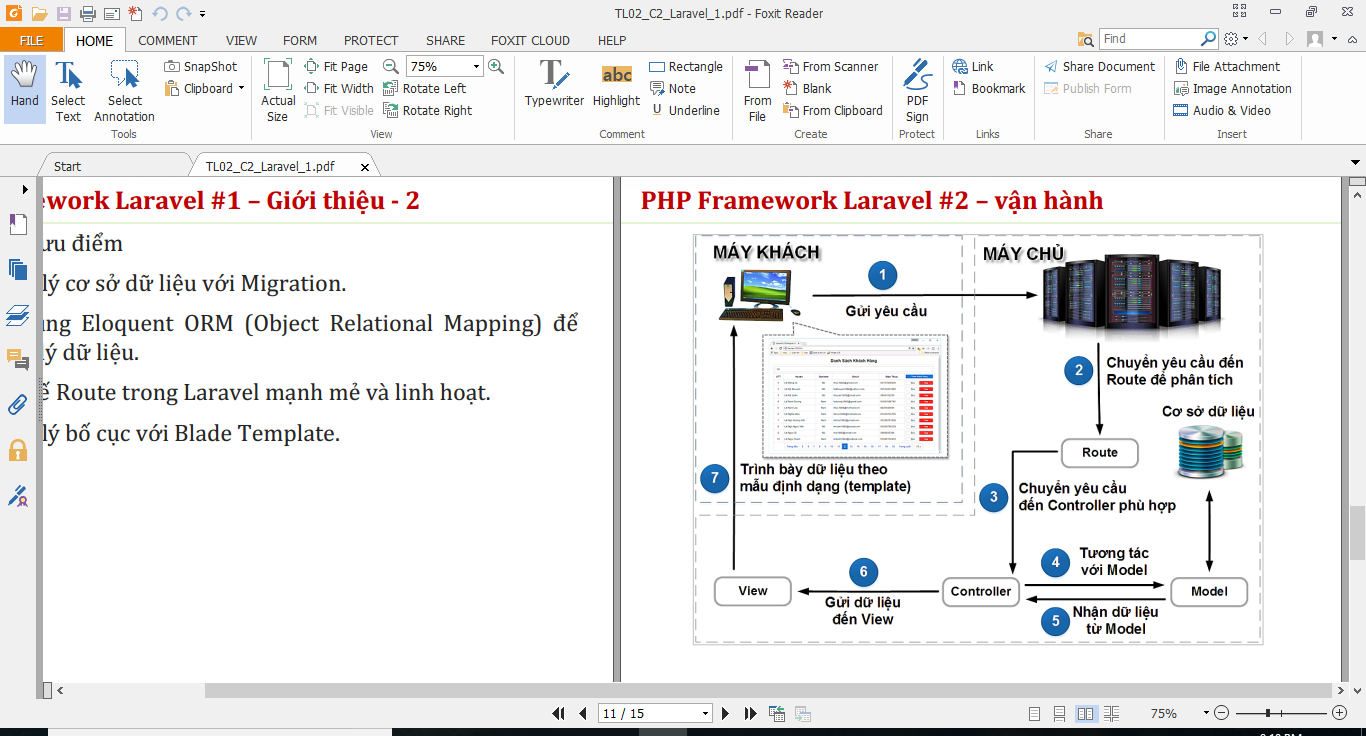
**Laravel** được Taylor Otwell tạo ra như một giải pháp thay thế cho CodeIgniter, cung cấp nhiều tính năng quan trọng hơn  như xác thực và phân quyền. Tôi cũng không chắc về điều này, nhưng có thể Taylor vốn là một .NET developer khi bắt đầu có nhu cầu làm việc với PHP khoảng vào những năm 2010-2011, đã chọn CodeIgniter khi đó đang là một ngôi sao mới nổi, thậm chí lấn át cả Symfony gạo cội. Và Taylor nhanh chóng nhận ra những điểm khiếm khuyết ở CodeIgniter, với tài năng và kiến thức xuất sắc về design-pattern của mình, Taylor quyết định tự mình tạo ra một framework sao cho thật đơn giản, dễ hiểu, hỗ trợ lập trình viên hiện thực ý tưởng một cách nhanh nhất bằng nhiều tính năng hỗ trợ như Eloquent ORM mạnh mẽ, xác thực đơn giản, phân trang hiệu quả, và hơn thế nữa.

* **Bản Laravel beta đầu tiên** được phát hành vào ngày 9/6/2011, tiếp đó là **Laravel 1** phát hành trong cùng tháng. Laravel 1 bao gồm các tính năng như xác thực, bản địa hóa, model, view, session, định tuyến và các cơ cấu khác, nhưng vẫn còn thiếu controller, điều này làm nó chưa thật sự là một MVC framework đúng nghĩa.
* **Laravel 2** được phát hành vào tháng 9 năm 2011, mang đến nhiều cài tiến từ tác giả và cộng đồng. Tính năng đáng kể bao gồm hỗ trợ controller, điều này thực sự biến Laravel 2 thành một MVC framework hoàn chỉnh, hỗ trợ Inversion of Control ([IoC](https://en.wikipedia.org/wiki/Inversion_of_control)), hệ thống template Blade. Bên cạnh đó, có một nhược điểm là hỗ trợ cho các gói của nhà phát triển bên thứ 3 bị gỡ bỏ.
* **Laravel 3** được phát hành vào tháng 2 năm 2012, với một tấn tính năng mới bao gồm giao diện dòng lệnh (CLI) tên “Artisan”, hỗ trợ nhiều hơn cho hệ thống quản trị cơ sở dữ liệu, chức năng ánh xạ cơ sở dữ liệu Migration, hỗ trợ “bắt sự kiện” trong ứng dụng,  và hệ thống quản lý gói gọi là “Bundles”. Lượng người dùng và sự phổ biến tăng trưởng mạnh kể từ phiên bản Laravel 3.
* **Laravel 4**, tên mã “Illuminate”, được phát hành vào tháng 5 năm 2013. Lần này thực sự là sự lột xác của Laravel framework, di chuyển và tái cấu trúc các gói hỗ trợ vào một tập được phân phối thông qua Composer, một chương trình quản lý gói thư viện phụ thuộc độc lập của PHP. Bố trí mới như vậy giúp khả năng mở rộng của Laravel 4 tốt hơn nhiều so với các phiên bản trước. Ra mắt lịch phát hành chính thức mỗi sáu tháng một phiên bản nâng cấp nhỏ. các tính năng khác trong Laravel 4 bao gồm tạo và thêm dữ liệu mẫu (database seeding), hỗ trợ hàng đợi, các kiểu gởi mail, và hỗ trợ “xóa mềm”  (soft-delete: record bị lọc khỏi các truy vấn từ Eloquent mà không thực sự xóa hẳn khỏi DB).
* **Laravel 5** được phát hành trong tháng 2 năm 2015, như một kết quả thay đổi đáng kể cho việc kết thúc vòng đời nâng cấp Laravel lên 4.3. Bên cạnh một loạt tính năng mới và các cải tiến như hiện tại, Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ cho phát triển ứng dụng mới. Những tính năng mới của Laravel 5 bao gồm hộ trợ lập lịch định kỳ thực hiện nhiệm vụ thông qua một gói tên là “Scheduler”, một lớp trừu tượng gọi là “Flysystem” cho phép điều khiển việc lưu trữ từ xa đơn giản như lưu trữ trên máy local – dễ thấy nhất là mặc định hỗ trợ dịch vụ Amazone S3, cải tiến quản lý assets thông qua “Elixir”, cũng như đơn giản hóa quản lý xác thực với các dịch vụ bên ngoài bằng gói “Socialite”.
* **Laravel 5.1** phát hành vào tháng 6 năm 2015, là bản phát hành đầu tiên nhận được hỗ trợ dài hạng (LTS) với một kết hoạch fix bug lên tới 2 năm vào hỗ trợ vá lỗi bảo mật lên tới 3 năm. Các bản phát hành LTS của Laravel được lên kế hoạch theo mỗi 2 năm.

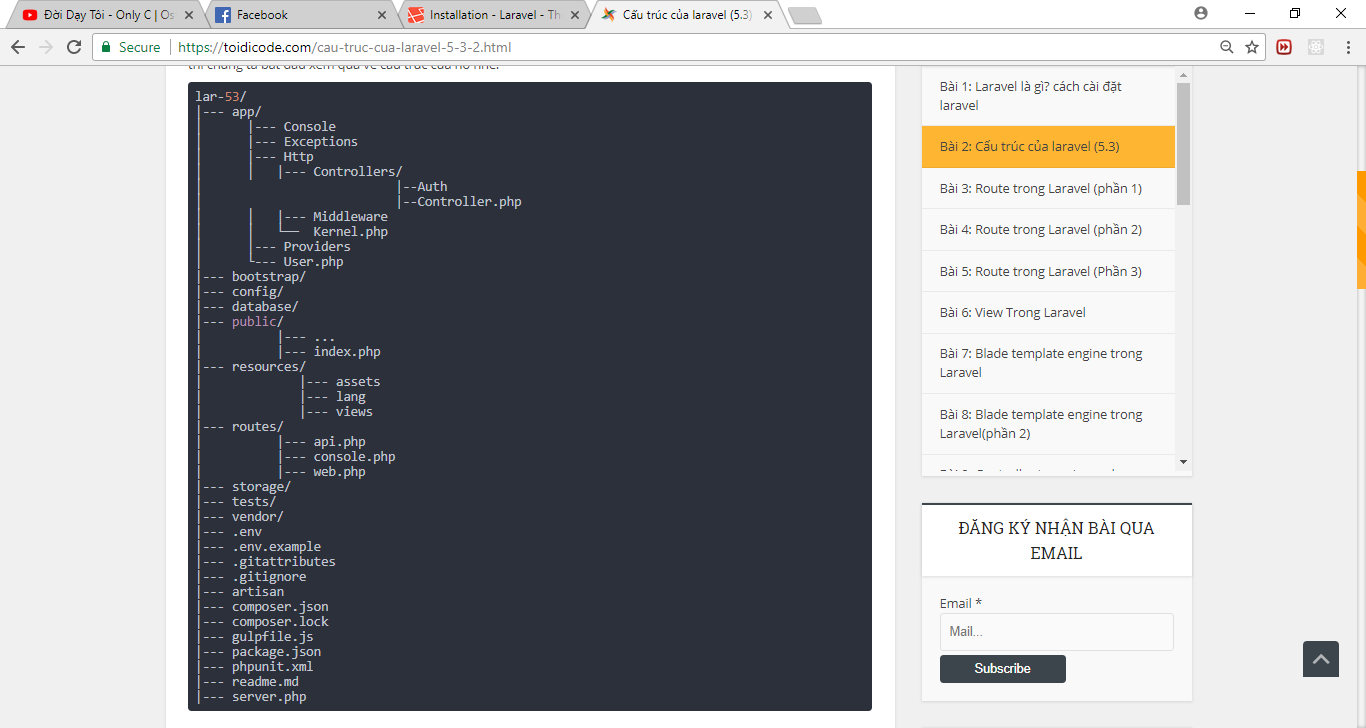
1. **Kiến trúc vận hành của Laravel:**

Quá trình vận hành của hệ thống dược xây dựng bởi Laravel được mô tả thông qua 7 bước:

* **Bước 1 :** người dùng ở máy khách( thành phần View) gửi yếu cầu đến máy chủ Web.
* **Bước 2:** các yêu cầu người dùng từ Wiev được gửi đến Route để phân tích.
* **Bước 3:** sau khi phân tích và xác định được Controller phù hợp , yêu cầu được gửi đến Controller để xử lý .
* **Bước 4:** thành phần Controller yêu cầu thành phần Model lấy dữ liệu theo yêu cầu.
* **Bước 5:** thành phần Controller nhận kết quả từ thành phần Model.
* **Bước 6:** thành phần Controller xử lý kết quả nhận được và chuyển đến View.
* **Bước 7:** thành phần View trình bày dữ liệu nhận được từ thành phần Controller theo các mẫu đã định dạng trước (template) và tiếp tục tương tác với người dùng.



1. **Cấu trúc lưu trữ của Laravel:**



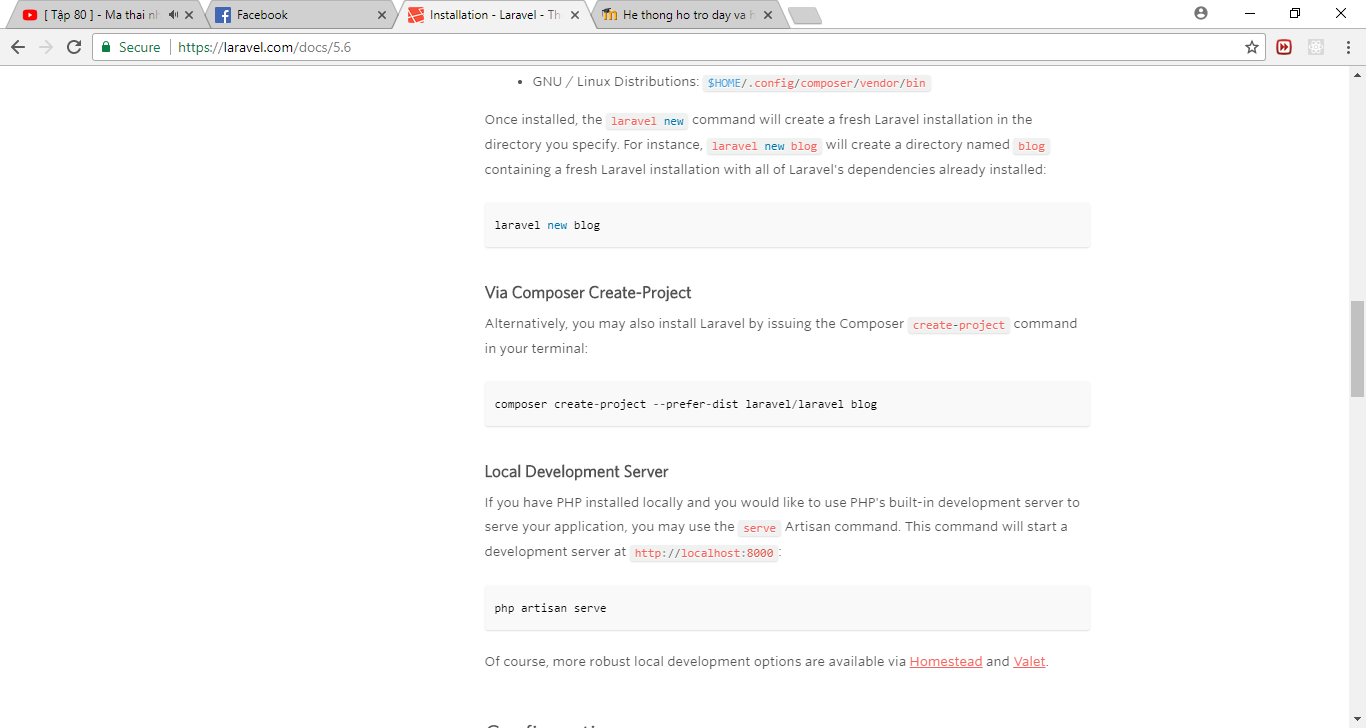
* **app:** Chứa các thư mục, các tập tin php, thư viện, models.
  + **Console:** Chứa các tập tin định nghĩa các câu lệnh trên artisan.
  + **Excerption:**  Chứa các tập tin quản lý, điều hướng lỗi.
  + **Http**
    - **Controllers** : Chứa các controllers của project.
    - **Middleware:**Chứa các tập tin lọc và ngăn chặn các requests.
    - **Kernel.php:**Cấu hình, định nghĩa Middleware.
  + **Providers:**Chứ các providers thực hiện việc **binding vào service container**(ở phần nâng cao mình sẽ nói sau).
  + **User.php:**Là model User mà Laravel tự địn sẵn cho chúng ta.
* **bootstrap:**Chứa tập tin điều hướng hệ thống.
* **config:** Chứa mọi tập tin cấu hình của Laravel.
* **database:** Chứa các thư mục tập tin vềdatabase.
  + **migrations:** Chứa các tập tin định nghĩa,khởi tạo và sửa bảng.
  + **seeds:** Chứa các tập tin định nghĩa dữ liệu insert(thêm) vào trong database.
  + **factories:** Chứa các tập tin định nghĩa các cột bảng dữ liệu để tạo ra các dữ liệu ảo.
* **public:** Chứa các tập tin css, js, image.
  + **index.php:**Đây là tệp tin root của Laraver
* **resources:** Chứa các views, ngôn ngữ(language) của project.
* **routes:** Chứa các tập tin định nghĩa các router, xử lý điều hướng router bao gồm: **web, api và console**(Mình sẽ nói thêm ở phần sau).
* **storage:** Chứa các tập tin hệ thống cache, session, ...
* **tests: ...**
* **vendor:** Chứa các thư viện của composer.
* **.env:** Là tập tin cấu hình chính của laravel như key app,database.
* **.env.example:**Tệp tin cấu hình mẫu của laraver.
* **composer.json:** tập tin của composer.
* **composer.lock:** tập tin của composer.
* **package.js:** Tập tin cấu hình của [nodejs](http://toidicode.com/nodejs-la-gi-26.html) (chứa các package cần dùng 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.

1. **Những ưu điểm của Laravel so với các PHP Framework khác:**

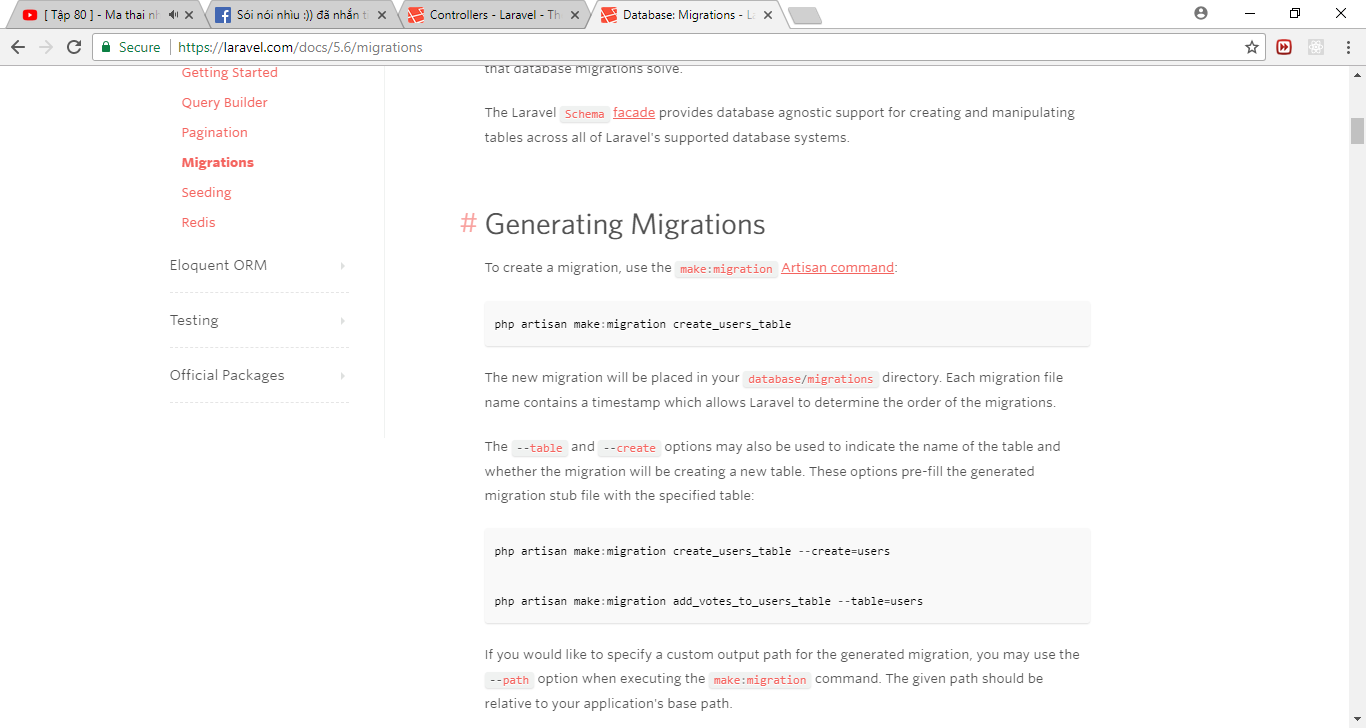
* Thừa hưởng những thế mạnh và khắc phục các nhược điểm của các Framework trước đó.
* Mức độ cạnh tranh cao vì được cập nhật liên tục nên các lỗi được khắc phục nhanh.
* Tài liệu hướng dẫn được thiết kế đầy đủ và rõ ràng
* Dễ dàng được hỗ trợ bởi cộng đồng vì hiện tại Laravel có số lượng người dùng nhiều nhất.
* Dễ dàng quản lý và tích hợp các thư viện vào dự án dựa vào Composer.
* Tự động nạp thư viện mã lệnh (autoload) theo không gian tên (namespace).
* Tính năng Migration giúp quản lý cơ sở dữ liệu hiệu quả hơn khi làm việc nhóm.
* Các lớp Eloquent giúp đơn giản hóa các thao tác xử lý cơ sở dữ liệu đối với quan hệ 1 – N và quan hệ N – N.
* Eloquent ORM (Object Relational Mapping): đây là một tính năng tuyệt vời, hỗ trợ việc quản lý và truy vết các thao tác truy vấn cơ sở dữ liệu. Tương thích với hầu hết các hệ quản trị cơ sở dữ liệu như MySQL, SQL Server, SQLite,…
* Cơ chế Route trong Laravel thật sự khác biệt và đầy mạnh mẽ so với các Framework khác.
* Việc quản lý bố cục rất đơn giản dựa vào Blade Template

1. **Các lệnh cơ bản của Laravel:**

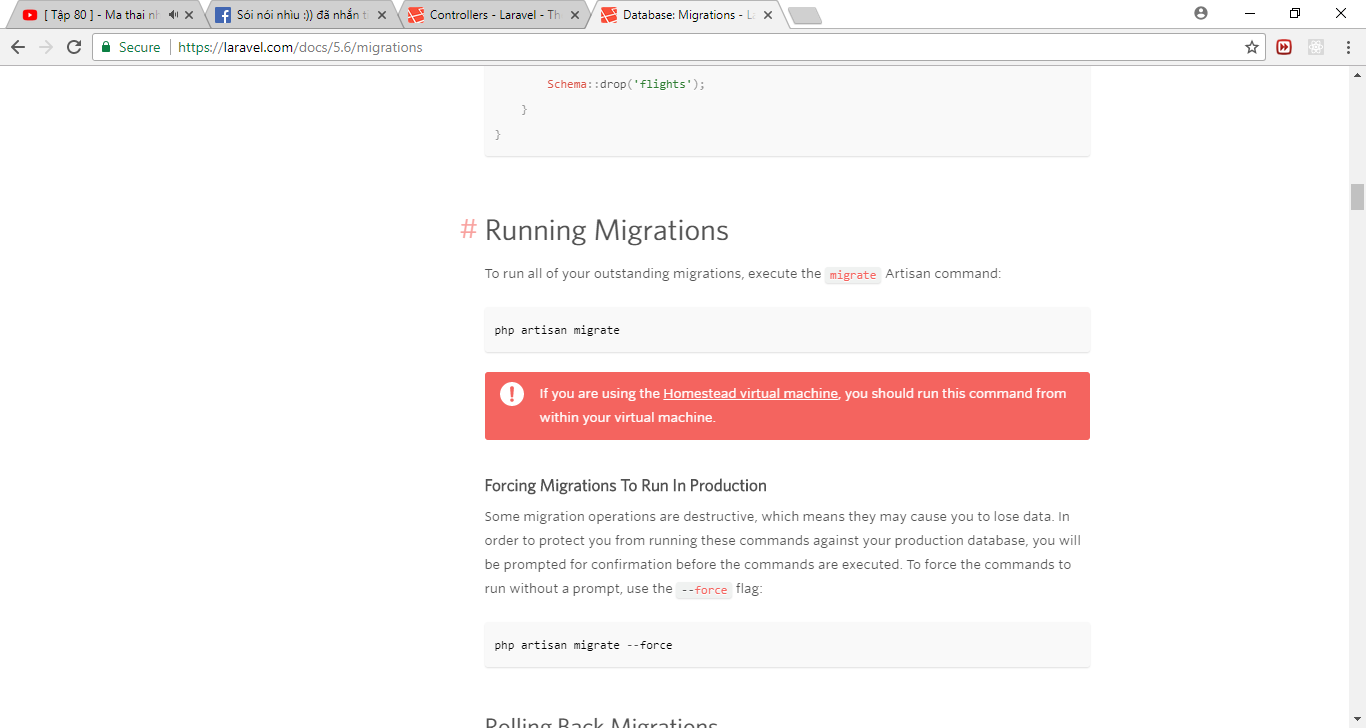
Tạo Project:



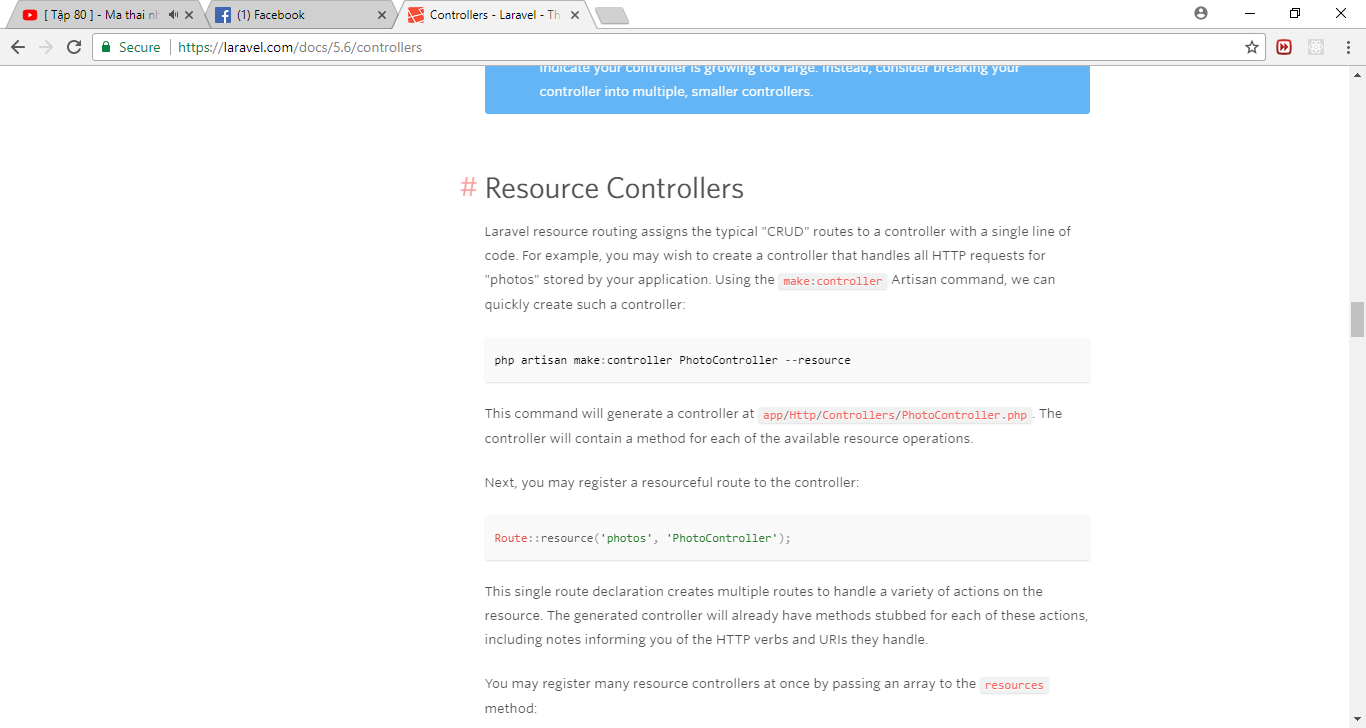
Tạo Migration:



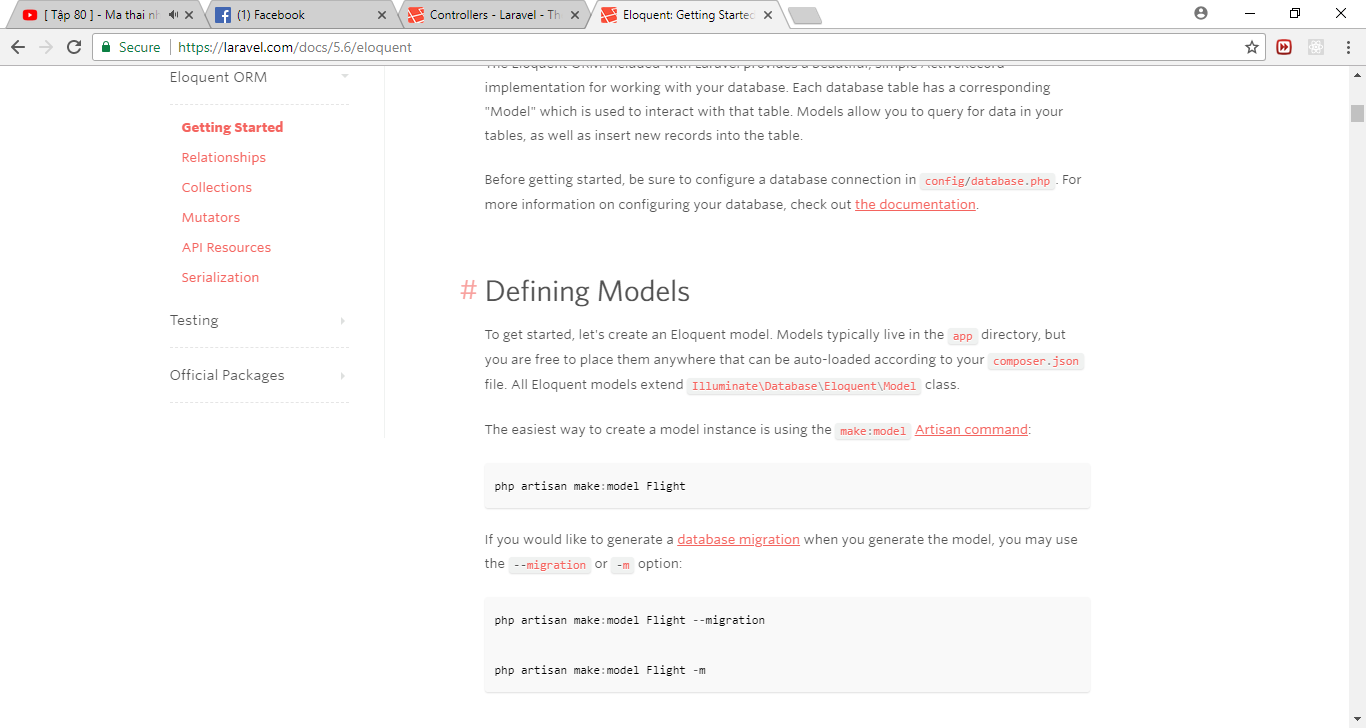
Migrate dữ liệu:



Tạo Controllers:



Tạo Model:



1. **GOOGLE MAP API:**
   * + 1. **Tổng quan về Google Map API:**

Google Map là một dịch vụ ứng dụng vào công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google, hỗ trợ nhiều dịch vụ khác của Google đặc biệt là dò đường và chỉ đường; hiển thị bản đồ đường sá, các tuyến đường tối ưu cho từng loại phương tiện, cách bắt xe và chuyển tuyến cho các loại phương tiện công cộng (xe bus, xe khách ...), và những địa điểm (kinh doanh, trường học, bệnh viện, cây ATM...) trong khu vực cũng như khắp nơi trên thế giới.

Map API là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của website A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các website cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google (di chuột, room, đánh dấu trên bản đồ…). Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ javascripts do vậy việc sử dụng API google rất dễ dàng.

Google Map API đã được nâng cấp lên phiên bản v3 không chỉ hỗ trợ cho các máy để bàn truyền thống mà cho cả các thiết bị di động; các ứng dụng nhanh hơn và nhiều hơn .

Các dịch vụ hoàn toàn miễn phí với việc xây dựng một ứng dụng nhỏ. Trả phí nếu đó là việc sử dụng cho mục đích kinh doanh, doanh nghiệp.

* + - 1. **Một số ứng dụng của Google Map API:**
* Đánh dấu các địa điểm trên bản đồ kèm theo thông tin cho địa điểm đó : khu vui chơi giải trí, nhà hàng khách sạn, cây ATM, bệnh viện, trường học,… bất cứ địa điểm nào bạn muốn.
* Chỉ dẫn đường đến các địa điểm cần tìm(đường tối ưu và nhiều option khác),chỉ dẫn đường giao thông công cộng, có thể là các địa điểm cung cấp như trên. Ở đây sử dụng các service google cung cấp.
* Khoanh vùng khu vực: các trung tâm kinh tế, khu đô thị, khu ô nhiễm…
* Tình trạng giao thông các khu vực… từ đó đưa ra các giải pháp…
  + - 1. **Google Map Event:** Sự kiện gồm 2 loại:
* **UI Event**: Lắng nghe sự kiện từ người dùng.
* **MVC State change**: Lắng nghe sự kiện từ sự thay đổi giá trị của các thuộc tính trên Map. Để lắng nghe sự kiện, ta sử dụng phương thức *addListener().* Phương thức này nhận vào 1 đối tượng, 1 kiểu sự kiện để lắng nghe và 1 phương thức xử lý khi sự kiện xảy ra.
  + 1. **UI event:**

Một số đối tượng trong Maps API được thiết kế để đáp ứng với sự kiện người sử dụng chẳng hạn như các sự kiện từ chuột hoặc bàn phím . Một đối tượng *google.maps.Marker* có thể lắng nghe người sử dụng các sự kiện sau đây: *click, dblclick, mouseup, mousedown, mouseover, mouseout*.

* + 1. **MVC State change:**

**Các sự kiện:**

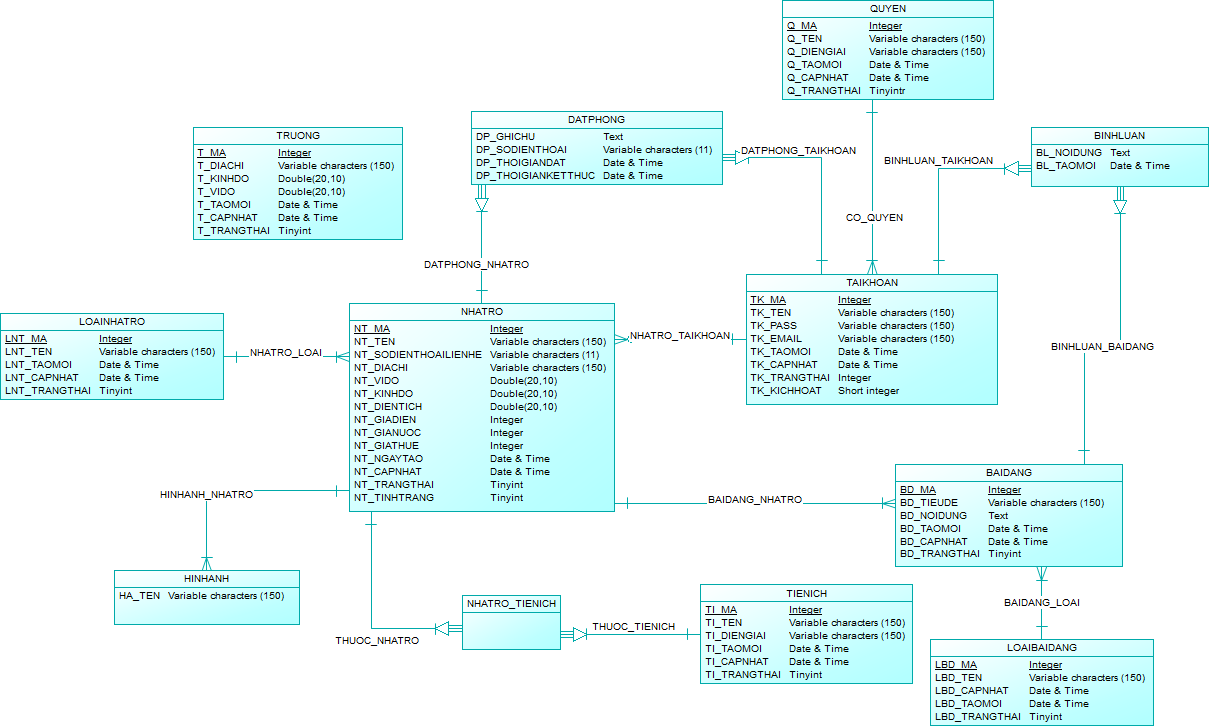
* *zoom\_changed*: sự kiện sẽ thực thi khi ta zoom size trên map
* *center\_changed*: sự kiện sẽ thực thi khi thuộc tính *center* của map thay đổi
* *title\_changed*: sự kiện sẽ thực thi khi thuộc tính title của map thay đổi
* *heading\_changed*: sự kiện sẽ thực thi khi thuộc tính heading của map thay đổi
* *dragstart*: sự kiện sẽ thực thi khi người dùng bắt đầu drag bản đồ
* *drag*: sự kiện sẽ thực thi khi người dùng drag bản đồ
* *dragend*: sự kiện sẽ thực thi khi người dùng kết thúc drag bản đồ
* *maptypeid\_changed*: sự kiện sẽ thực thi khi thuộc tính maptypeid của map thay đổi Event Listeners (lắng nghe sự kiện)
* Handling Events.
  + - 1. **Map Controls:**
* Control mặc định thiết lập cho Google Map gồm:
  + *Zoom* – hiển thị 1 slider hoặc button “+/-” để điều khiển độ zoom *(zoom level*) của bản đồ
  + *Pan* – công cụ để dịch chuyển bản đồ sang trái, phải, trên, dưới.
  + *MapType* – cho phép người dùng chuyển đổi giữa các loại bản đồ (*roadmap* và *satellite*)
  + *Street View* – hiển thị 1 icon *Pegman* (hình người) để có thể kéo bản đồ để enable Street View (xem chi tiết về đường phố).
* Một số controls khác:
  + *Scale* – hiển thị 1 thành phần chia tỷ lệ bản đồ
  + *Rotate* – hiển thị một biểu tượng nhỏ hình tròn để xoay bản đồ
  + *Overview Map* – hiển thị một *thumbnail overview map* cho phép xem bản đồ hiện tại trong một khu vực rộng hơn.
    - 1. **Google Maps Types:**
* *ROADMAP* (bản đồ truyền thống, default 2D map)
* *SATELLITE* (photographic map)
* *HYBRID* (photographic map + tên đường và tên thành phố)
* *TERRAIN* (bản đồ với hệ thống sông ngòi, núi....) Kiểu của map được xác định bởi thuộc tính *mapTypeId:*

1. **MÔ HÌNH MVC:**

**CHƯƠNG 2: NỘI DUNG VÀ KẾT QUẢ THỰC HIỆN**

1. **MÔ TẢ HỆ THỐNG:**
2. **Mô tả yêu cầu:**
3. **Các biểu mẫu liên quan:**
4. **THIẾT KẾ THÀNH PHẦN DỮ LIỆU:**
5. **Mô hình dữ liệu mức quan niệm (CDM):**

CDM mô tả toàn bộ cấu trúc và mối quan hệ giữa các thông tin cần quản lý trong hệ thống quản lý nhà trọ sinh viên tại thành phố Cần Thơ.



1. **Mô tả các thực thể:**
   1. **Thực thể: truong(t\_ma, t\_ten, t\_vido, t\_kinhdo, t\_taomoi, t\_capnhat, t\_trangthai)**

Diễn giải: Lưu thông tin của các trường đại học trên địa bàn như: tên, tọa độ,… Trong đó **t\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| t\_ma | Integer | X |  | X |  | Mã trường |
| t\_ten | Varchar |  | 150 | X |  | Tên trường |
| t\_vido | Double |  | 20,10 | X |  | Vĩ độ |
| t\_kinhdo | Double |  | 20,10 | X |  | Kinh độ |
| t\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| t\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| t\_trangthai | Tinyint | X |  |  | 2 | Trạng thái |

* 1. **Thực thể: taikhoan(tk\_ma, tk\_ten, tk\_email, tk\_matkhau, tk\_active, tk\_taomoi, tk\_capnhat)**

Diễn giải: Lưu thông tin của người dùng có tài khoản đăng nhập vào hệ thống. Trong đó **tk\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương. Email không được trùng và được dùng như tên đăng nhập để đăng nhập vào hệ thống.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| tk\_ma | Integer | X |  | X |  | Mã tài khoản |
| tk\_ten | Varchar |  | 150 |  |  | Tên người dùng |
| tk\_email | Varchar |  | 150 | X |  | Email người dùng |
| tk\_matkhau | Varchar |  | 150 |  |  | Mật khẩu người dùng |
| tk\_kichhoat | Tinyint | X |  |  | 0 | Trạng thái kích hoạt |
| tk\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| tk\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |

* 1. **Thực thể: tien\_ich(ti\_ma, ti\_ten, ti\_diengiai, ti\_taomoi, ti\_capnhat, ti\_trangthai)**

Diễn giải: Lưu thông tin của các tiện ích đi kèm nhà trọ như tên tiện ích, diễn giải tiện ích,… Trong đó, **ti\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| ti\_ma | Integer | X |  | X |  | Mã tiện ích |
| ti\_ten | Varchar |  | 150 | X |  | Tên tiện ích |
| ti\_diengiai | Varchar |  | 150 |  |  | Diễn giải tiện ích |
| ti\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| ti\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| ti\_trangthai | Tinyint | X |  |  | 2 | Trạng thái tiện ích |

* 1. **Thực thể: quyen(q\_ma, q\_ten, q\_diengiai, q\_taomoi, q\_capnhat, q\_trangthai)**

Diễn giải: Lưu thông tin của các quyền của người dùng khi sử dụng hệ thống. Trong đó, **q\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| q\_ma | Integer | X |  | X |  | Mã quyền |
| q\_ten | Varchar |  | 150 | X |  | Tên quyền |
| q\_diengiai | Varchar |  | 150 |  |  | Diễn giải quyền |
| q\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| q\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| q\_trangthai | Tinyint | X |  |  | 2 | Trạng thái quyền |

* 1. **Thực thể: nhatro(nt\_ma , nt\_ten, nt\_sodienthoailienhe, nt\_diachi, nt\_kinhdo, nt\_vido, nt\_giadien, nt\_gianuoc, nt\_giathue, nt\_tinhtrang, nt\_taomoi, nt\_capnhat, nt\_trangthai)**

Diễn giải: Lưu thông tin của các nhà trọ khi người dùng đăng kí trên hệ thống. Trong đó, **nt\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| nt\_ma | Integer | X |  | X |  | Mã nhà trọ |
| nt\_ten | Varchar |  | 150 |  |  | Tên nhà trọ |
| nt\_sodienthoailienhe | Varchar |  | 11 |  |  | Số điện thoại liên hệ |
| nt\_diachi | Varchar |  | 150 | X |  | Địa chỉ nhà trọ |
| nt\_kinhdo | Double |  | 20,10 | X |  | Kinh độ nhà trọ |
| nt\_vido | Double |  | 20,10 | X |  | Vĩ độ nhà trọ |
| nt\_dientich | Double | X | 20,10 |  |  | Diện tích nhà trọ |
| nt\_giadien | Varchar | X |  |  |  | Giá điện |
| nt\_gianuoc | Integer | X |  |  |  | Giá nước |
| nt\_giathue | Integer | X |  |  |  | Giá cho thuê |
| nt\_tinhtrang | Tinyint | X |  |  |  | Tình trạng nhà trọ (còn/hết phòng) |
| nt\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| nt\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| nt\_trangthai | Tinyint | X |  |  | 2 | Trạng thái nhà trọ |

* 1. **Thực thể: loainhatro(lnt\_ma, lnt\_ten, lnt\_taomoi, lnt\_capnhat, lnt\_trangthai)**

Diễn giải: Lưu thông tin của các loại nhà trọ trên hệ thống. Trong đó, **lnt\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| lnt\_ma | Integer | X |  | X |  | Mã loại nhà trọ |
| lnt\_ten | Varchar |  | 150 | X |  | Tên loại nhà trọ |
| lnt\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| lnt\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| lnt\_trangthai | Tinyint | X |  |  | 2 | Trạng thái loại nhà trọ |

* 1. **Thực thể: loaibaidang(lbd\_ma, lbd\_ten, lbd\_taomoi, lbd\_capnhat, lbd\_trangthai)**

Diễn giải: Lưu thông tin của các loại bài đăng trên hệ thống. Trong đó, **lbd\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| lbd\_ma | Integer | X |  | X |  | Mã loại bài đăng |
| lbd\_ten | Varchar |  | 150 | X |  | Tên loại bài đăng |
| lbd\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| lbd\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| lbd\_trangthai | Tinyint | X |  |  | 2 | Trạng thái loại bài đăng |

* 1. **Thực thể: baidang(bd\_ma, bd\_tieude, bd\_noidung, bd\_taomoi, bd\_capnhat, bd\_trangthai)**

Diễn giải: Lưu thông tin của các bài đăng tin cho thuê trên hệ thống. Trong đó, **bd\_ma** làm khóa chính không được trùng nhau, không được NULL và bắt buộc dương.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| bd\_ma | Integer | X |  | X |  | Mã bài đăng |
| bd\_tieude | Varchar |  | 150 | X |  | Tiêu đề bài đăng |
| bd\_noidung | Text |  |  |  |  | Nội dung bài đăng |
| bd\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày tạo mới |
| bd\_capnhat | Date&time |  |  |  | Current\_timestamp | Ngày cập nhật |
| bd\_trangthai | Tinyint | X |  |  | 2 | Trạng thái duyệt bài đăng |

* 1. **Thực thể: binhluan(bl\_noidung, bl\_taomoi)**

Diễn giải: Thể hiện thông tin bình luận của người dùng về nhà trọ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| bl\_noidung | Text |  |  |  |  | Nội dung bình luận |
| bl\_taomoi | Date&time |  |  |  | Current\_timestamp | Ngày đăng bình luận |

* 1. **Thực thể: datphong(dp\_ghichu, dp\_sodienthoai, dp\_thoigiandatphong, dp\_thoiganketthuc)**

Diễn giải: Lưu thông tin đặt phòng của người dùng trên hệ thống. Thời gian kết thúc được tính sau 6h từ lúc đặt phòng.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| dp\_ghichu | Text |  |  |  |  | Ghi chú đặt phòng |
| dp\_sodienthoai | Varchar |  | 11 |  |  | Số điện thoại người đặt |
| dp\_thoigiandatphong | Date&time |  |  |  | Current\_timestamp | Thời gian đặt phòng |
| dp\_thoigianketthuc | Date&time |  |  |  | Current\_timestamp | Thời gian kết thúc (sau 6h kể từ lúc đặt) |

* 1. **Thực thể: hinhanh\_nhatro(ha\_ten)**

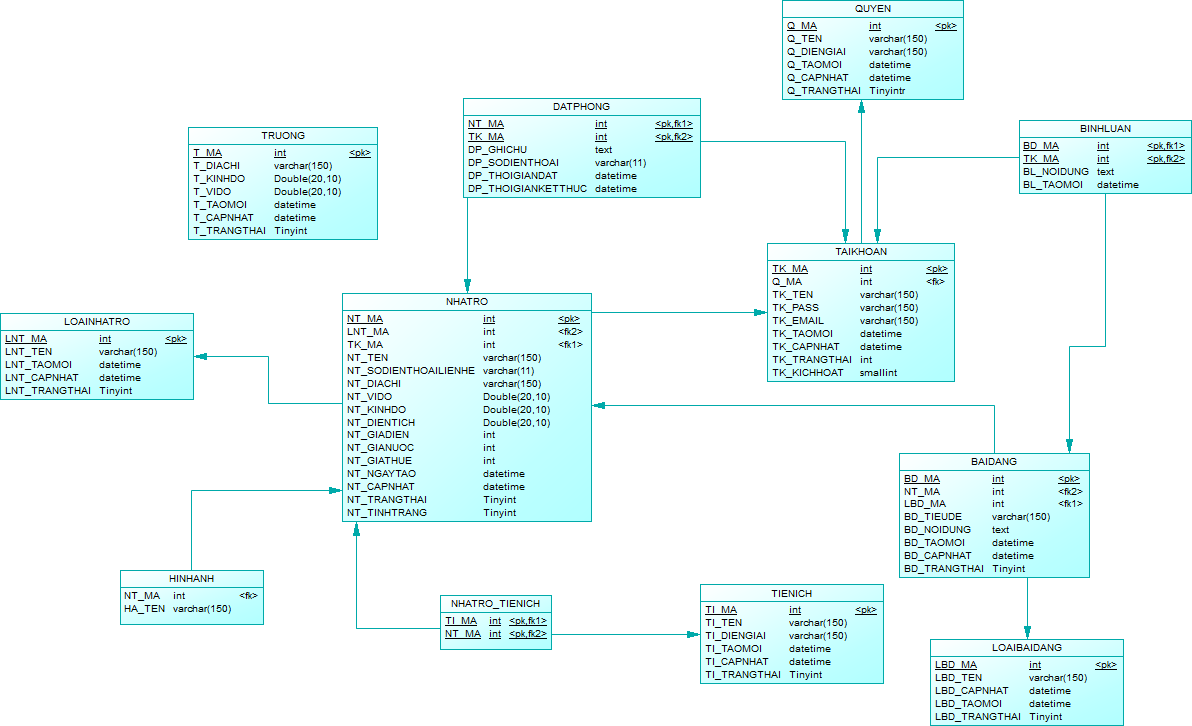
Diễn giải: Lưu tên hình ảnh của nhà trọ được người dùng upload lên hệ thống.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu Dữ Liệu** | **Số Dương** | **Độ Dài Tối Đa** | **Duy Nhất** | **Giá Trị Mặc Định** | **Diễn Giải** |
| ha\_ten | Varchar |  | 150 | X |  | Tên hình ảnh nhà trọ |

* 1. **Bảng nhatro\_tienich()**

Diễn giải: Thể hiện các tiện ích có trong nhà trọ.

1. **Mô hình dữ liệu mức luận lý:**
2. **Mô hình dữ liệu mức vật lý:**

****

1. **Các ràng buộc tham chiếu:**

LOAINHATRO(LNT\_MA) 🡺 NHATRO(LNT\_MA)

TAIKHOAN(TK\_MA) 🡺 NHATRO(TK\_MA)

NHATRO (NT\_MA) 🡺 HINHANH(NT\_MA)

NHATRO\_TIENICH(NT\_MA) 🡺 NHATRO(NT\_MA)

NHATRO\_TIENICH(TI\_MA) 🡺 TIENICH(TI\_MA)

LOAIBAIDANG(LBD\_MA) 🡺 BAIDANG(LBD\_MA)

BINHLUAN(BD\_MA) 🡺 BAIDANG(BD\_MA)

BINHLUAN(TK\_MA) 🡺 TAIKHOAN(TK\_MA)

QUYEN(Q\_MA) 🡺 TAIKHOAN(Q\_MA)

DATPHONG(NT\_MA) 🡺 NHATRO(NT\_MA)

DATPHONG(TK\_MA) 🡺 TAIKHOAN(TK\_MA)

HINHANH\_NHATRO(NT\_MA) 🡺 NHATRO(NT\_MA)

1. **Lưu đồ dòng dữ liệu (DFD):**
   1. **DFD cấp 0:**
   2. **DFD cấp 1:**
2. **Sơ đồ chức năng:**
   1. **Sơ đồ chức năng người dùng chưa có tài khoản:**

Người dùng chưa có tài khoản

Tìm nhà trọ

Đăng kí tài khoản

Xem thông tin nhà trọ

Xem chỉ đường đến nhà trọ

* 1. **Sơ đồ chức năng người dùng đã có tài khoản:**

Người dùng đã có tài khoản

Tìm nhà trọ

Quản lý tài khoản

Xem thông tin nhà trọ

Bình luận nhà trọ

Xem danh sách nhà trọ

Xem danh dách bài đăng bài đăng

Đổi mật khẩu

Đăng nhập

Xem chỉ đường đến nhà trọ

Hẹn xem nhà trọ

Cập nhật thông tin nhà trọ

Cập nhật bài đăng

Xem danh sách hẹn xem nhà trọ

* 1. **Sơ đồ chức năng Admin:**

1. **THIẾT KẾ THÀNH PHẦN XỬ LÝ:**
2. **THIẾT KẾ GIAO DIỆN:**
3. **THẢO LUẬN:**

**CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

1. **KẾT LUẬN:**
2. **HƯỚNG PHÁT TRIỂN:**