VIỆN ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**VŨ ĐÌNH HẢI**

**TÌM HIỂU NỀN TẢNG ỨNG DỤNG WEB NGUỒN MỞ CODEIGNITER**

**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM VÀ ỨNG DỤNG**

**HÀ NỘI – NĂM 2017**

VIỆN ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**TÌM HIỂU NỀN TẢNG ỨNG DỤNG WEB NGUỒN MỞ CODEIGNITER**

**Chuyên ngành: Công nghệ thông tin**

***Giáo viên hướng dẫn* : ThS. Trần Tiến Dũng**

***Sinh viên thực hiện* : Vũ Đình Hải**

***Lớp* : 11B4**

**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM VÀ ỨNG DỤNG**

**HÀ NỘI – NĂM 2017**

MỤC LỤC

[LỜI NÓI ĐẦU](#_Toc499292769)

[TÓM TẮT CHUYÊN ĐỀ](#_Toc499292770)

[DANH MỤC CHỮ VIẾT TẮT](#_Toc499292771)

[Chương 1 TÌM HIỂU CODEIGNITER FRAMEWORK 1](#_Toc499292772)

[1.1. Ứng dụng website là gì? 1](#_Toc499292774)

[1.2. PHP 3](#_Toc499292775)

[1.3. Codeigniter Framework 8](#_Toc499292777)

[1.3.1. Codeigniter framework 8](#_Toc499292778)

[1.3.2. Cấu trúc thư mục trong Codeigniter 11](#_Toc499292779)

[1.3.3. Mô hình Model-View-Controller 14](#_Toc499292780)

[1.3.4. Controller 15](#_Toc499292781)

[1.3.5. Model 17](#_Toc499292782)

[1.3.6. View 19](#_Toc499292783)

[Chương 2 VẬN DỤNG CODEIGNITER CODE FROM ĐĂNG KÍ 22](#_Toc499292784)

[2.1. Thiết kế cơ sở dữ liệu 22](#_Toc499292786)

[2.2. Xây dựng chức năng 24](#_Toc499292787)

[2.2.1. Xây dựng form đăng kí 24](#_Toc499292788)

[2.2.2. Xây dựng bảng danh sách học sinh 25](#_Toc499292789)

[2.2.3. Xây dựng chức năng sửa 26](#_Toc499292790)

[2.3. Tổng kết 28](#_Toc499292791)

[2.3.1. Ưu điểm của codeigniter framework 28](#_Toc499292792)

[2.3.2. Nhược điểm của codeigniter framework 29](#_Toc499292793)

[TÀI LIỆU THAM KHẢO 31](#_Toc499292794)

LỜI NÓI ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và những ứng dụng của nó trong đời sống, máy tính điện tử không còn là một thứ phương tiện lạ lẫm đối với mọi người mà nó dần trở thành một công cụ làm việc và giải trí thông dụng và hữu ích của chúng ta, không chỉ ở công sở mà còn ngay cả trong gia đình.

Để người dùng tiếp cận một cách nhanh nhất với công nghệ thì website là phần không thể thiếu. Qua website người dùng có thể truy cập, tiếp cận với các thông tin, tin tức dù ở bất cứ nơi đâu. Vậy website từ đâu mà có? Khi nhắc đến làm một website mọi người sẽ nhắc đến các lập trình viên phát triển web, các ngôn ngữ để xây dựng website và không thể bỏ qua là các framework để xây dựng website. Framework có mặt hầu như ở mọi ngôn ngữ lập trình, có thể hiểu một cách đơn giản Framework chính là bộ khung của ứng dụng.

Để đi sâu và tìm hiểu framework trong chuyên đề này em sẽ tìm hiểu về một Framework của PHP đó là Codeigniter. Đây là một framework mạnh mẽ và rất phổ biến hiện nay cho các lập trình viên code web bằng ngôn ngữ PHP.

Do thời gian có hạn, kiến thực và kinh nghiệm bản thân còn hạn chế, chuyên đề còn nhiều thiếu sót, em mong nhận được ý kiến đóng góp quý báu từ quý thầy cô để chuyên đề được hoàn thiện hơn. Em xin cảm ơn!

Qua đây, em cũng xin gửi lời cảm ơn đến các thầy cô trong Khoa Công nghệ thông tin - Viện Đại học Mở Hà Nội. Các thầy cô đã theo sát em trong quá trình học tập, khảo sát hệ thống, giải đáp những thắc mắc trong quá trình làm chuyên đề vừa qua. Đặc biệt em xin gửi lời cảm ơn đến thầy ThS. Trần Tiến Dũng đã giúp đỡ em trong quá trình nghiên cứu framework và đưa ra những lời khuyên quý báu khi em thực hiện chuyên đề.

TÓM TẮT CHUYÊN ĐỀ

**Tên đề tài**: Tìm hiểu nền tảng ứng dụng web nguồn mở Codeigniter

**Sinh viên thực hiện**: Vũ Đình Hải. **Lớp**: 11B4.

**Giảng viên hướng dẫn**: ThS. Trần Tiến Dũng

**Tóm tắt chuyên đề**:

Nội dung của chuyên đề tốt nghiệp là tìm hiểu về Framework Codeginiter. Nghiên cứu cấu trúc tổ chức source code, các phiên bản, cách lập trình khi sử dụng framework. Vận dụng lý thuyết đã nghiên cứu, áp dụng thực hiện form đăng kí thành viên

DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên viết tắt | Tên đầy đủ | Dịch nghĩa tiếng Việt |
| 1 | MVC | Model-View-Controller |  |
| 2 | DRY | Don’t Repeat Yourself |  |
| 3 | WWW | World Wide Web |  |

# 

Chương 1  
TÌM HIỂU CODEIGNITER FRAMEWORK

1. 1. Ứng dụng website là gì?

Website còn gọi là trang web, trang mạng, là một tập hợp các trang web bao gồm văn bản, hình ảnh, video, flash …, thường chỉ nằm trong một tên miền(domain name) hoặc tên miền phụ (subdomain). Trang web được lưu trữ (web hosting) trên máy chủ web (server web) có thể truy cập thông qua Internet. Trang web đầu tiên mà người xem truy cập tên miền thường được gọi là trang chủ (home page), người xem có thể xem các trang khác thông qua các siêu liên kết (hyperlinks). Chỉ cẩn có Internet người dùng có thể xem, cập nhật, thay đổi thông tin ngay tức khắc, ở bất kì nơi nào, tiết kiệm được chi phí, thời gian với lượng thông tin không giới hạn số lượng hay phạm vi (toàn thế giới có thể truy cập).

Website có thể được phân loại dựa theo nhiều tiêu chí khác nhau: Phân loại theo giao diện người dùng, phân loại theo loại dữ liệu, phân loại theo đối tượng sở hữu.

Phân loại theo giao diện người dùng website được chia làm hai loại: Frontend và Backend. Frontend là định dạng trang web được trình bày trên màn hình máy tính, điện thoại… của người xem (máy khách) được xem bằng các phần mềm trình duyệt như Internet Exploer, Google Chrome, Firrefox, Safari.... Backend là phần lưu trữ trên máy chủ được người thiết kế web, lập trình viên web sử dụng và quản lý website.

Phân loại theo dữ liệu được chia làm 2 loại là website động và website tĩnh. Website động là web có cơ sở dữ liệu, được cung cấp công cụ uản lý, có thể truy cập thường xuyên và quản lý các thành phần trên website. Website động khi xây dựng sẽ bao gồm hai phần: Một phần hiển thị trên trình duyệt mà chúng ta thường nhìn thấy khi truy cập.  Phần ngầm bên dưới dùng để điều khiển nộidung của trang web, phần nội dung ngầm này thường thì chỉ những người quảntrị website đó mới có thể có quyền truy cập vào. Web tĩnh ở đây được hiểu theo nghĩ là dữliệu không thay đổi thường xuyên, do lập trình bằng ngônngữ HTML theo từng trang. Với dạng web này người để thay đổi nội dung trên trang web người sở hữu phải truycập trực tiếp vào các mã lệnh để thay đổi thông tin. Không có cơ sở dữ liệu bên dưới hệ thống, không có công cụ để điều khiển nội dung gián tiếp. Dạng file của trang website tĩnh thường là html. Ưu điểm: Web tĩnh dành cho các web có nội dung đơn giản gọn nhẹ, không có sự can thiệp của người lập trình web, không phải xử lý những mã lệnh phức tạp vì vậy việc thiết kế website đặt để các đối tượng thoải mái và tự do sáng tạo của người thiết kế nên web tĩnh thường sở hữu được một hình thức hấp dẫn và bắt mắt. Khuyết điểm: vì không có hệ thống hỗ trợ thay đổi thông tin nên việc cập nhật thông tin web tốn nhiều chi phí và cần phải có người am hiểu kỹ thuật web thực hiện.

Phân loại theo đối tượng sở hữu, Có thể là côngviệc của một cá nhân, một doanh nghiệp hoặc các tổ chức, và thường dành riêng cho một số chủ đề cụ thể hoặc mục đích.  Bất kỳ trang web cóthể chứa một siêu liên kết vào bất kỳ trang web khác, do đó phân biệt các trang web cá nhân, như cảm nhận của người sử dụng. Tạm thời phân loại như sau:

**Thiết kế website** **cá nhân**: Các đối tượng như diễn viên, ca sĩ, ngừơi nổi tiếng, người thiết kế đồ hoạ, hoặc bất kỳ cá nhân nào thích giớ thiệu bản thân mình đều có thể tạo ra một website cho cá nhân mình.

**Thiết kế website** tin tức: Đây là một dạng website cung cấp thông tin chính trị, xã hội, kinh tế, khoa học, giáo dục, sức khoẻ,…thể loại này được phát triển trên nền tảng từ các thể loại báo giấy truyền thống.

**Thiết kế website** **Mạng xã hội** (blog): Là dạng web dành cho người sử dụng được quyền tạo cho mình một không gian riêng gồm nhiều trang độc lập, ở đây người dùng có thể đăng tải thông tin cá nhân sở thích, hình ảnh, viết nhật ký tại trang của mình… Các mạng xã hội nổi tiếng như: Twitter, Facebook, google+ …

**Thiết kế website cho doanh nghiệp**: thiết kế web với mục đích quảng bá công ty, giới thiệu các chức năng hoạtđộng, cập nhật những tin tức, sản phẩm mới của công ty nhằm dễ dàng tiếp cận đến khách hàng thông qua một kênh quảng bá mới là internet.

**Thiết kế website** **Diễn dàn**: Là dạng web tương tác với người dùng mà bất kỳ xem nào cũng có thể đăng ký tham gia là thành viên và được quyền tăng tải bài viết của mình và dĩ nhiên diễn đàn luôn có người kiểm soát thông tin người dùng đăng tải và có quyền can thiệp vào việc hiển thị thông tin đó hay không.

**Thiết kế website bán hàng**: là các dạng web cho phép bán hàng trực tuyến, việc thanh toán có nhiều hình thức như: tiền mặt, chuyển khoản, thanh toán bằng thẻ, hoặc thông qua cổng thanh toán của các dịch vụ hỗ trợ.

**Thiết kế website dành cho các tổ chức**, cơ quan nhà nước: Các bộ, sở, ban ngành, hiệp hội tổ chức…là đối tượng sở hữu website dạng này.

**Thiết kế website** **giải trí**: đăng tải phim ảnh, nhạc, game, ...

* 1. PHP

PHP (viết tắt hồi quy "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

PHP có một lịch sử phát triển lâu đời và được đông đảo lập trình viên đón nhận, ngày càng phát triển mạnh mẽ hơn. Nhiều người bảo chọn PHP vì lí do mã nguồn mở, miễn phí, dễ triển khai, chạy nhanh, những ai đã làm việc với PHP và theo đuổi nó vì có chung một lý do rất đơn giản, đó là làm việc với PHP rất thoải mái.

PHP có hướng đối tượng nhưng bạn hoàn toàn có thể “code rừng”, không cần triển khai bất cứ lớp nào để hoàn thành công việc. PHP có nhiều hàm build-in, chỉ cần gọi là chạy mà không cần phải nhớ namespace để include/import như các ngôn ngữ khác. Lưu trữ dữ liệu trên biến và mảng của PHP thì rất thoải mái; biến không cần khai báo và có thể xài bất kể đâu; mảng lưu theo kiểu một chiều, 2 chiều, đa chiều tùy ý, truy xuất ngẫu nhiên đơn giản với cặp “khóa/giá trị”… Nhưng ưu điểm “thoải mái” của PHP cũng chính là nhược điểm của nó. Với những khối code nhỏ, bạn hoàn toàn có thể code theo kiểu rất của riêng bạn; nhưng với các phần mềm lớn, bạn sẽ bị lạc trong chính khu rừng đó như: function name, variable name… Bạn sẽ không thể nào tiếp tục phát triển lớn hơn, thời gian bảo trì sẽ gấp nhiều nhiều lần thời gian phát triển… Cũng như nhiều ngôn ngữ lập trình khác, thấy được khó khăn này, người dùng PHP đã xây dựng những framework cho mình. Vậy nên Framework có mặt hầu như ở mọi ngôn ngữ lập trình, PHP cũng không nằm ngoại lệ.

Trước hết framework là một “tiêu chuẩn”.

Khi sử dụng framework nào, bạn phải tuân thủ những qui định của nó đặt ra: qui định về cách đặt tên biến, đặt tên hàm, tên lớp, … qui định về cách truyền dữ liệu, qui định về luồng xử lý dữ liệu, qui định về cách tổ chức code – file nào đặt ở đâu nhiệm vụ gì…. Tất cả những qui định đó nhằm giúp tạo ra một bản code sạch. Bạn có thể dễ dàng ghi nhớ vị trí – chức năng của các dòng code để chỉnh sửa, hay sử dụng. Nhờ tuân thủ qui định, mà khi làm việc nhóm, người này có thể đọc hiểu code của người kia, có thể sử dụng lại hàm được viết bởi một người khác.

**Tại sao nên dùng một framework để phát triển dự án**. Bạn sẽ mất ít thời gian hơn để phát triển dự án hơn.Với framwork, bạn đã có một khung sườn làm việc. Bạn không mất thời gian phân tích kiến trúc ứng dụng để “sáng tạo” những thứ người khác đã làm rất tốt, viết lại những thứ quá đỗi nhàm chán.  
Bạn sẽ có một ứng dụng với mã nguồn sạch đẹp, dễ dàng phát triển, bảo trì.  
Team của bạn sẽ làm việc với nhau hiệu quả hơn, hiểu ý nhau hơn, tốc độ hơn.  
Bạn sẽ có cơ hội sử dụng lại nhiều đoạn code, những module, extension, plugin gắn vào ứng dụng của mình mà không tốn sức để phát triển cho các chức năng đó. Hiệu năng cũng như bảo mật sẽ được chăm sóc và cập nhật thường xuyên nhờ cộng đồng phát triển framework đó. Bạn sẽ có nhiều người giúp đỡ hơn. Những lập trình viên cùng sử dụng một framework sẽ tạo nên một cộng đồng. Không ai dám vỗ ngực xưng tui biết hết mọi thứ về framework đó, nhưng một đám đông 3000 người thì vô tư phán điều đó. Mà thường ai đã dùng cái gì rồi, họ cũng muốn mọi người xung quanh dùng giống họ, do đó họ rất nhiệt tình giúp đỡ, tư vẫn và giải đáp những vấn đề của bạn mà không hề đòi hỏi lợi ích nào. Vậy nên lựa chọn framework có cộng đồng lớn cũng là một tiêu chí hàng đầu. Ngoài ra khi sử dụng một framework ta còn nhận thấy những ưu điểm sau:

* Nâng cao tốc độ phát triển sản phẩm
* Cung cấp code được tổ chức tốt, có thể sử dụng lại và dễ dàng bảo trì
* Cho phép bạn phát triển theo thời gian khi các ứng dụng web chạy trên các framework này có khả năng mở rộng
* Giải phóng bạn khỏi những lo ngại về bảo mật ở mức thấp của một trang web
* Tuân theo mô hình MVC(Model-View-Controller) đảm bảo sự tách biệt phần trình diễn và phần logic nghiệp vụ
* Thúc đẩy phương thức phát triển web hiện đại như các công cụ lập trình hướng đối tượng

**Các Framework PHP phổ biến hiện nay**

**Laravel**: Mặc dù [Laravel](http://laravel.com/) là một framework PHP tương đối mới (nó được phát hành vào năm 2011), theo khảo sát trực tuyến gần đây của Sitepoint thì nó là framework phổ biến nhất mà các nhà phát triển hay sử dụng. Laravel có một [hệ sinh thái rất lớn](http://laravel.com/#ecosystem) với một [nền tảng giúp triển khai ứng dụng](https://forge.laravel.com/) của bạn một cách nhanh chóng, và trang web chính thức của nó cung cấp nhiều hướng dẫn bằng video gọi là [Laracasts.](https://laracasts.com/) Laravel có nhiều tính năng giúp cho việc phát triển ứng dụng nhanh chóng nhất có thể. Laravel sở hữu một light-weight templating engine có tên là "Blade", **cú pháp sáng sủa** hỗ trợ cho các tác vụ thường xuyên cần phải làm trong ứng dụng của bạn, chẳng hạn như authentication, sessions, queueing, caching và RESTful routing. Laravel cũng bao gồm một **môi trường phát triển cục bộ** gọi là [Homestead](http://laravel.com/docs/5.1/homestead) - đây thực chất là một Vagrant box được đóng gói.

## **Symfony**: [Symfony Components](http://symfony.com/components) là **các thư viện PHP có thể sử dụng lại giúp bạn hoàn thành nhiều tác vụ khác nhau**, chẳng hạn như việc tạo form, cấu hình đối tượng, routing, authentication, templating, và nhiều thứ khác. Bạn có thể cài đặt bất kỳ Components nào bằng trình quản lý dependency PHP [Composer.](https://getcomposer.org/) Trang web của Symfony có một phần [showcase](http://symfony.com/showcase/) rất đẹp mắt, nơi bạn có thể xem qua một số dự án đã hoàn thành với sự giúp đỡ của framework tiện dụng này

**CodeIgniter:**  [CodeIgniter](http://www.codeigniter.com/) là một framework PHP rất nhẹ và có tuổi đời gần 10 năm (được phát hành lần đầu tiên vào năm 2006). CodeIgniter có một quá trình cài đặt rất đơn giản và chỉ đòi hỏi một cấu hình tối thiểu, do đó nó có thể giúp bạn loại bỏ được rất nhiều rắc rối. Đây cũng là một sự lựa chọn lý tưởng nếu bạn muốn **tránh xung đột phiên bản PHP**, vì nó **hoạt động tốt trên hầu như tất cả các nền tảng shared và dedicated hosting**.

CodeIgniter không hoàn toàn dựa trên mô hình phát triển MVC. Việc sử dụng các lớp Controller là bắt buộc, nhưng Models và Views là tùy chọn, và bạn có thể sử dụng các tiêu chuẩn đặt tên và viết code theo cách của riêng mình, đó là bằng chứng cho thấy CodeIgniter cung cấp tự do tuyệt vời cho các nhà phát triển. Nếu bạn tải mã nguồn về, bạn sẽ thấy nó chỉ có dung lượng khoảng 2MB, vì đây là một lean framework, nhưng nó cho phép bạn bổ sung thêm các plugin của hãng thứ ba nếu bạn cần các chức năng phức tạp hơn.

**Yii 2:** Nếu bạn chọn framework [Yii](http://www.yiiframework.com/) thì trang web của bạn sẽ có [tốc độ thực thi rất nhanh](http://www.yiiframework.com/performance/)**, nhanh hơn so với các framework PHP khác**, bởi vì nó sử dụng **kỹ thuật lazy loading** một cách rộng rãi. Yii 2 hoàn toàn hướng đối tượng, và nó dựa trên khái niệm lập trình DRY(Don’t Repeat Yourself), do đó cung cấp cho bạn một code base sạch sẽ và hợp lý.

Yii 2 được tích hợp với jQuery đi kèm với một bộ tính năng AJAX-enabled, nó thực hiện một cơ chế để dễ dàng sử dụng skinning và theming, do đó đây có thể là một lựa chọn tuyệt vời cho những người đến từ một nền tảng frontend. Nó cũng có một bộ sinh code (class code generator) mạnh mẽ được gọi là [Gii](http://www.yiiframework.com/doc-2.0/guide-tool-gii.html) tạo điều kiện cho lập trình hướng đối tượng và tạo nguyên mẫu (prototyping) nhanh chóng, và cung cấp một giao diện dạng web cho phép bạn tương tác để sinh ra phần code mà bạn cần.

**Zend Framework**: [Zend](http://framework.zend.com/) là một framework PHP mạnh mẽ và ổn định được đóng gói với rất nhiều cấu hình tùy chọn vì thế nó được khuyến cáo **không nên sử dụng cho các dự án nhỏ** nhưng rất **tuyệt vời cho những dự án phức tạp**. Zend có các đối tác như IBM, Microsoft, Google và Adobe. Trong phiên bản lớn phát hành sắp tới, Zend Framework 3 sẽ được [tối ưu hóa cho PHP 7](http://framework.zend.com/blog/announcing-the-zend-framework-3-roadmap.html), nhưng vẫn sẽ hỗ trợ PHP 5.5 trở đi.

Phiên bản hiện tại, Zend Framework 2 cũng có nhiều tính năng thú vị như các công cụ mã hóa code, một editor có thể kéo & thả dễ dàng sử dụng với sự hỗ trợ cho các công nghệ front-end (HTML, CSS, JavaScript), debugging trực tuyến nhanh, các công cụ PHP Unit testing, và một Database Wizard giúp kết nối cơ sở dữ liệu. Zend Framework được tạo ra với các phương pháp Agile tạo điều kiện để cung cấp các ứng dụng chất lượng cao cho khách hàng doanh nghiệp.

* 1. Codeigniter Framework
     1. Codeigniter framework là gì?

CodeIgniter là một nền tảng ứng dụng web nguồn mở được viết bằng ngôn ngữ PHP bởi Rick Ellis (CEO của EllisLab, Inc). Phiên bản đầu tiên được phát hành ngày 28.02.2006, phiên bản hiện tại: 1.7.2 (phát hành ngày 11.09.2009). Ý tưởng xây dựng CodeIgniter được dựa trên Ruby on Rails, một nền tảng ứng dụng web được viết bằng ngôn ngữ Ruby. Hiện tại, CodeIgniter đang được phát triển bởi ExpressionEngine Development Team thuộc EllisLab, Inc. Cấu trúc thư mục trong Codeigniter

**Những điểm nổi bật**

Được thiết kế theo mô hình Model-View-Controller: Mô hình MVC giúp tách thành phần hiển thị giao diện (presentation) và xử lý (business logic) của một phần mềm thành những thành phần độc lập, từ đó giúp cho việc thiết kế, xử lý và bảo trì mã nguồn dễ dàng, đồng thời tăng khả năng mở rộng của phần mềm. CodeIgniter vận dụng mô hình này trong thiết kế, giúp tách biệt các tập tin giao diện với các tập tin xử lý dữ liệu, nâng cao khả năng quản lý và dễ bảo trì.

Nhỏ gọn: Gói cài đặt chỉ 404KB (không bao gồm phần User Guide). So với các PHP framework khác như CakePHP (1.3MB), Symfony (5.08MB) hay Zend Framework (5.66MB)…kích thước của CodeIgniter giúp giảm thiểu đáng kể không gian lưu trữ.

Tốc độ nhanh: CodeIgniter được đánh giá là PHP framework có tốc độ nhanh nhất hiện nay. Bằng cơ chế lưu nội dung vào bộ đệm (cache), kiểm tra bộ đệm trước khi tiến hành thực hiện yêu cầu, CodeIgniter giảm số lần truy cập và xử lý dữ liệu, từ đó tối ưu hóa tốc độ tải trang.

Miễn phí: CodeIgniter được phát hành dưới giấy phép Apache/BSD mở rộng, cho phép người dùng tự do thay đổi, phát triển và phân phối mã nguồn.

Hỗ trợ Search Engine Optimization: Cấu trúc URL của CodeIgniter rất thân thiện với các robot tìm kiếm.

Hệ thống thư viện phong phú: CodeIgniter cung cấp các thư viện phục vụ cho những tác vụ thường gặp nhất trong lập trình web, chẳng hạn như truy cập cơ sở dữ liệu, gửi email, kiểm tra dữ liệu, quản lý session, xử lý ảnh…đến những chức năng nâng cao như XML-RPC, mã hóa, bảo mật…

Bảo mật hệ thống: Cơ chế kiểm tra dữ liệu chặt chẽ, ngăn ngừa XSS và SQL Injection của CodeIgniter giúp giảm thiểu các nguy cơ bảo mật cho hệ thống.

**Những điểm hạn chế**

Chưa hỗ trợ Object-Relational Mapping: Object Relational Mapping (ORM) là một kỹ thuật lập trình, trong đó các bảng của cơ sở dữ liệu được ánh xạ thành các đối tượng trong chương trình. Kỹ thuật này giúp cho việc thực hiện các thao tác trong cơ sở dữ liệu (Create Read Update Delate – CRUD) dễ dàng, mã nguồn ngắn gọn hơn. Hiện tại, CodeIgniter vẫn chưa hỗ trợ ORM.

Chưa hỗ trợ AJAX: AJAX (Asynchronous JavaScript and XML) đã trở thành một phần không thể thiếu trong bất kỳ ứng dụng Web 2.0 nào. AJAX giúp nâng cao tính tương tác giữa người dùng và hệ thống, giúp cho người dùng có cảm giác như đang sử dụng ứng dụng desktop vì các thao tác đều diễn ra “tức thời”. Hiện tại, CodeIgniter vẫn chưa có thư viện dựng sẵn nào để hỗ trợ xây dựng ứng dụng AJAX. Lập trình viên phải sử dụng các thư viện bên ngoài, như jQuery, Script.aculo.us, Prototype hay Mootools…

Chưa hỗ trợ một số module thông dụng: So sánh với framework khác, CodeIgniter không có các module thực thi một số tác vụ thường gặp trong quá trình xây dựng ứng dụng web như Chứng thực người dùng (User Authorization), Trình phân tích RSS (RSS Parser) hay Trình xử lý PDF…

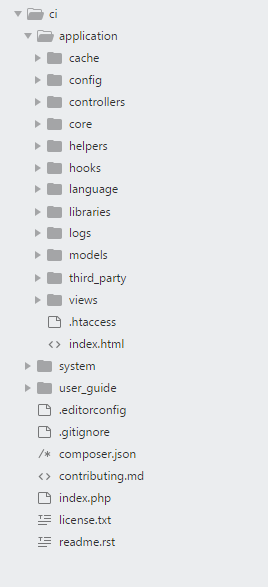
Chưa hỗ trợ Event-Driven Programming: Event-Driven Programming (EDP) là một nguyên lý lập trình, trong đó các luồng xử lý của hệ thống sẽ dựa vào các sự kiện, chẳng hạn như click chuột, gõ bàn phím…Đây không phải là một khuyết điểm to lớn của CodeIgniter vì hiện tại, chỉ có một số ít framework hỗ trợ EDP, bao gồm Prado, QPHP và Yii.

* + 1. Cấu trúc thư mục trong Codeigniter

Việc **download và cài đặt Codeigniter**rất đơn giản. Trước tiên vào trang chủ của codeigniter để download source. Sau khi download xong sẽ có được 1 file nén .rar, bạn vào folder **WWW**của Web Server tạo một folder mới tên là Codeigniter sau đó copy file vừa download vào trong folder mới tạo này và giải nén tại đó.

Sau khi giải nén các bạn ra trình duyệt gõ localhost/codeigniter, nếu có kết quả xuất hiện thì đã cài đặt thành công, ngược lại là cài đặt bị lỗi.

*Hình 1.1. Màn hìnhwebsite khi cài đặt thành công*



*Hình 1.2. Cấu trúc thư mục trong Codeigniter*

Tập tin **index.php** được xem như controller đầu vào, tiếp nhận các yêu cầu từ phía client và chuyển các yêu cầu này cho hệ thống xử lý.

Thư mục **system** chứa phần lõi của CodeIgniter, bao gồm các thư viện xây dựng sẵn, các tập tin ngôn ngữ, ghi chú hệ thống... . Trong đó các thư mục sau khá quan trọng:

Thư mục**application**: dành cho lập trình viên, các tập tin được lập trình cho ứng dụng sẽ được lưu trong thư mục này.

Thư mục **cache**: bộ đệm của hệ thống, chứa các trang đã xử lý trước đó.

Thư mục **helpers**: chứa các hàm hỗ trợ cho lập trình viên khi viết ứng dụng.

Thư mục**libraries**: chứa thư viện dựng sẵn của CodeIgniter.

Đối với lập trình viên, các tập tin lập trình sẽ được lưu vào thưmục **system/application**. Trong đó:

- Thư mục **config**: chứa các tập tin cấu hình hệ thống.

- Thư mục **controllers**: chứa các lớp controllers.

- Thư mục **errors**: chứa các tập tin lỗi.

- Thư mục **helpers**: chứa các hàm tiện ích do người dùng định nghĩa.

- Thư mục **hooks**: chứa các tập tin để mở rộng mã nguồn CodeIgniter.

- Thư mục **language**: chứa các tập tin ngôn ngữ.

- Thư mục **libraries**: chứa các thư viện cho người dùng định nghĩa.

- Thư mục **models**: chứa các lớp model.

- Thư mục **views**: chứa các lớp view.

**Các phiên bản hiện tại của Codeigniter**

Hiện tại trên trang chủ của Codeigniter framework đang có 3 phiên bản gồm: Codeigniter 2.x, Codeigniter 3.x và Codeigniter 4.x .

**Codeigniter 2.x** CodeIgniter 2.2.6 là phiên bản tiêu chuẩn hiện tại.

Phiên bản 2.x ban đầu được phát hành tháng 1 năm 2011, bản cập nhật chính tiếp theo(2.1.0) vào mua thu năm 2011, Phiên bản (2.2.0) xuất hiện vào tháng 7 năm 2014, và phiên bản hiện tại (2.2.6) xuất hiện tháng 10 năm 2015.

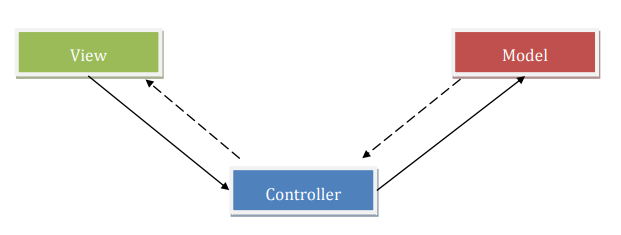
CodeIgniter 2 đã hết hạn sử dụng và sẽ không được cập nhật hay hỗ trợ tính đến ngày 31 tháng 10 năm 2015. Không có kế hoạch cập nhật và hỗ trợ cho phiên bản này nữa. Trang chủ của Codeigniter khuyến khích mọi người chuyển sang dùng CodeIgniter 3.x!

**Codeigniter 3.x** với CodeIgniter 3.1.là phiên bản tiêu chuẩn hiện tại. Đã có một số cải tiến kể từ phiên bản 2.x, đáng chú ý là với cơ sở dữ liệu, xử lý phiên và mã hóa. Sự phát triển của phiên bản này đang được tiến hành.

**Codeigniter 4.x** CodeIgniter 4 là phiên bản sắp tới của framework. Phiên bản này vẫn đang được tiến hành phát triển, với giai đoạn 1 hoàn thành vào tháng 6 năm 2016. Nó vẫn “không” sẵn sàng cho việc sử dụng để xây dựng và phát triển - phiên bản hiện tại được coi là "pre-alpha1".

* + 1. Mô hình Model-View-Controller

Model-View-Control (MVC) là một kiến trúc phần mềm, hiện đang được xem là một mẫu thiết kế trong công nghệ phần mềm. Mô hình MVC tách biệt phần xử lý dữ liệu ra khỏi phần giao diện, cho phép phát triển, kiểm tra và bảo trì các thành phần một cách độc lập.



*Minh họa mô hình MVC*

Theo đó:

- Model thể hiện các cấu trúc dữ liệu. Các lớp thuộc thành phần Model thường thực hiện các tác vụ như truy vấn, thêm, xóa, cập nhật dữ liệu. Khi dữ liệu trong Model thay đổi, thành phần View sẽ được cập nhật lại. - View là thành phần thể hiện dữ liệu trong Model thành các giao diện tương tác với người sử dụng. Một Model có thể có nhiều View tùy thuộc vào các mục đích khác nhau. - Controller đóng vai trò trung gian giữa Model và View. Thông tin người dùng từ View được gửi cho Controller xử lý, sau đó Controller tương tác với Model để lấy dữ liệu được yêu cầu, sau cùng Controller trả dữ liệu này về cho View. Mô hình MVC thường được sử dụng trong các ứng dụng web, vì thành phần View (mã HTML/XHTML) được sinh ra từ các ngôn ngữ lập trình web. Thành phần Controller sẽ nhận các dữ liệu GET/POST, xử lý những dữ liệu này, sau đó chuyển sang Model xử lý.

* + 1. Controller

Controller là những lớp đóng vai trò trung gian giữa view và model. Controller nhận các yêu cầu từ phía người dùng, kiểm tra chúng trước khi chuyển qua cho model. Sau khi model xử lý yêu cầu và trả dữ liệu về, controller chuyển sang view để hiển thị dữ liệu cho người dùng. Trong CodeIgniter, các lớp controller được gán vào segment thứ nhất trong URI. 

Một lớp controller chuẩn trong CodeIgniter có mẫu:   
class Example extends Controller {   
/\*\*   
\* Hàm tạo   
\*/   
function \_\_contruct()   
{   
parrent::\_\_contruct();   
}

Theo mặc định, cấu trúc URL của CodeIgniter được thiết kế dựa vào các segment thay cho kiểu query truyền thống. Cách tiếp cận này giúp URL trở nên ngắn gọn, có ý nghĩa, dễ ghi nhớ và thân thiện với các bộ máy tìm kiếm. Một URL trong CodeIgniter có dạng:   
domain.com/index.php/controller/method/param/… Trong đó:   
• Segment controller là tên của lớp controller được gọi.   
• Segment method là tên của phương thức trong lớp controller ở trên.   
• Segment param là các đối số của phương thức đó   
Như vậy, URI domain.com/index.php/product/view/1 sẽ mang ý nghĩa: Gọi đến phương thức view() thuộc lớp Product với đối số truyền vào bằng 1. Định tuyến URI (URI Routing)   
Trong thực tế, URI domain.com/index.php/product/view/1 lại không thân thiện với các bộ máy tìm kiếm bằng URI domain.com/index.php/product/view/appple-ipad-1.   
CodeIgniter cho phép lập trình viên có thể thay đổi cơ chế xử lý URL mặc định thông qua chức năng Định hướng URI bằng cách thiết lập các quy luật trong tập tin application/config/routes.php. Tập tin này chứa mảng $route, với khóa là URI mới và giá trị là URI cũ. Các quy luật được đọc theo thứ tự từ trên xuống, quy luật trước có độ ưu tiên cao hơn quy luật sau. CodeIgniter cho phép sử dụng các ký tự đại diện (wildcard) để thay thế. CodeIgniter đã quy định sẵn hai ký tự đại diện:  
• num: tương ứng với các giá trị số.   
• any: tương ứng với bất kỳ giá trị nào.   
Ví dụ:   
$route ['product/view/(:num)'] = "product/view/$1";   
Ý nghĩa: Gọi đến phương thức view() trong lớp Product với tham số truyền vào là giá trị kiểu số. $route ['product/: any'] = "product/find\_product";   
Ý nghĩa: Tất cả các yêu cầu bắt đầu bằng product sẽ gọi đến hàm find\_product() của lớp Product.   
Ta cũng có thể sử dụng biểu thức chính quy (Regular Expression) để thiết lập các quy luật. Chẳng hạn: $route ['product/( [a-z]+)/(\d+)'] = "$1/id\_$2";   
Thêm hậu tố vào URL Các bộ máy tìm kiếm có xu hướng thân thiện hơn với các tập tin HTML. Bằng cách thay đổi giá trị biến $config ['url\_suffix'] trong tập tin system/application/config/config.php, ta có thể thêm hậu tố HTML hay bất cứ hậu tố nào khác vào cho URL. Ví dụ, với $config ['url\_suffix'] = '.html', URL của ta sẽ như sau:   
  
domain.com/index.php/controller/method/param.html   
Loại bỏ chuỗi index.php trong URL   
Theo mặc định, chuỗi index.php được thêm vào URL. Để loại bỏ chuỗi này, ta có thể sử   
dụng một tập tin .htaccess có nội dung như sau:   
Options +FollowSymLinks All -Indexes   
RewriteEngine On   
RewriteCond $1 !^(index\.php|resources|robots\.txt)   
RewriteCond %{REQUEST\_FILENAME} !-f   
RewriteCond %{REQUEST\_FILENAME} !-d   
RewriteRule ^(.\*)$ index.php/$1 [L,QSA]

* + 1. Model

Model là những lớp được xây dựng nhằm thực hiện việc trao đổi thông tin với cơ sở dữ liệu. Một lớp Model có thể thực hiện các tác vụ truy vấn, thêm, xóa, cập nhật dữ liệu. Trong CodeIgniter, việc khai báo các lớp model dành cho một thực thể đó là không cần thiết, vì trong controller của thực thể đó, ta có thể gọi đến thư viện database để thực hiện các thao tác với cơ sở dữ liệu. Tuy nhiên, để tiện cho việc quản lý, xây dựng lớp model cho một thực thể được khuyến khích. Một lớp model chuẩn trong CodeIgniter có cấu trúc như sau:

class Example\_model extends Model {   
/\*\*   
\* Hàm tạo   
\*/   
function \_\_contruct()   
{   
parrent::\_\_contruct();   
}   
\* Mô tả hàm   
\* @param kiểu dữ liệu $param mô tả biến   
\*/   
function exampleFunction($param)   
{   
// Do something here   
}  
}   
Khi khai báo một lớp model, ta cần tuân theo một số quy tắc:   
• Tên lớp được viết hoa chữ đầu tiên, phần còn lại viết thường. Ví dụ: User\_model, Blog\_model, Article\_model…   
• Tên tập tin được đặt như tên lớp, và được đặt trong thư mục application/models/. Các thư mục có thể được lưu lồng vào nhau để thuận tiện cho việc quản lý. Ví dụ: application/models/user\_model.php,   
application/models/front\_end/blog\_model.php,   
application/models/back\_end/article\_model.php…   
• Bắt buộc phải kế thừa từ lớp Model. Trong hàm tạo của lớp con, phải gọi đến hàm tạo của lớp cha. Để sử dụng model trong controller, ta sử dụng đoạn mã sau:   
$this->load->model('model\_name');   
Trong đó, model\_name là tên của lớp model. Nếu lớp model được lưu trong một thư mục, ta cần ghi rõ đường dẫn đến thư mục đó. Ví dụ lớp Blog\_model được lưu trong thư mục application/models/front\_end/blog\_model.php. Để sử dụng lớp này, ta sử dụng đoạn mã sau:   
$this->load->model('front\_end/blog\_model');   
Sau khi load lớp model, ta có thể sử dụng lớp này trong chương trình bằng cách gọi $this->model\_name->method(); Để gán một tên khác cho lớp model khi sử dụng trong chương trình, ta có thể truyền vào tham số thứ hai như sau:   
$this->load->model('model\_name', 'ModelName');   
Khi đó, để sử dụng các phương thức của lớp model, ta gọi:   
$this->ModelName->method();

* + 1. View

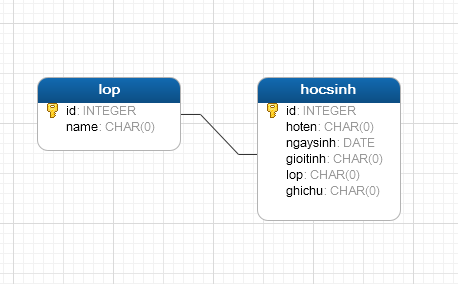
View là những tập tin HTML được xây dựng nhằm thể hiện dữ liệu trong model thành các giao diện tương tác với người dùng. View có thể là một trang web hoàn chỉnh, hay chỉ là một phần của trang web (header, footer, sidebar…). Nội dung của tập tin view, ngoài mã HTML còn có thể chứa mã PHP. View không bao giờ được gọi trực tiếp mà phải thông qua controller.   
Để sử dụng view trong controller, ta sử dụng đoạn mã sau:   
$this->load->view ('view\_name', $data);   
Trong đó, view\_name là tên của view, $data chứa các dữ liệu sẽ được hiển thị trong view.   
Cũng giống như model, ta có thể lưu view trong các thư mục để tiện cho việc quản lý.  
  
Khi sử dụng, chỉ cần khai báo đường dẫn tương đối đến thư mục chứa view. Ví dụ tập tin header.tpl.php được lưu trong thư mục application/views/front\_end. Để hiển thị view này, ta sử dụng đoạn mã sau:   
$this->load->view ('front\_end/header.tpl');   
CodeIgniter cho phép sử dụng nhiều view trong cùng một phương thức của controller.   
Dữ liệu trong các view sẽ được kết hợp lại với nhau. Ví dụ:   
class Example extends Controller{   
function showExample()   
{   
// Loading multiple views   
$this->load->view('header.tpl');   
$this->load->view('content.tpl');   
$this->load->view('sidebar.tpl');   
$this->load->view('footer.tpl');   
}   
}   
Để hiển thị dữ liệu trong model ra view, ta có thể truyền một mảng hoặc đối tượng làm   
đối số thứ hai khi load view. Ví dụ:   
$data = array(   
'name' => 'Maxwell Smart',   
'email' => 'maxwell.smart@control.org',   
'age' => '33'   
);   
$this->load->view('employee\_detail', $data);   
Hoặc   
$objEmployee = new Employee();   
$this->load->view('employee\_detail', $objEmployee);

Khi đó, các khóa của mảng hoặc các thuộc tính của đối tượng sẽ được chuyển thành các   
biến để sử dụng trong tập tin view. Dưới đây là nội dung mẫu của tập tin   
employee\_detail.php:   
<p>Họ tên: <?php echo $name ?></p>   
<p>Ngày sinh: <?php echo $email ?></p>   
<p>Age: <?php echo $age ?></p>   
Ta cũng có thể sử dụng các cấu trúc điều khiển (if, else, switch…case…), lặp (for,   
while, do…while…)…bên trong view để hiển thị dữ liệu.  
CodeIgniter còn cho phép lập trình viên có thể lấy về nội dung của view thay vì xuất   
trực tiếp trình duyệt, bằng cách truyền true làm đối số thứ ba khi tải view.   
$string = $this->load->view('myfile', '', true);

Chương 2  
VẬN DỤNG CODEIGNITER CODE FROM ĐĂNG KÍ

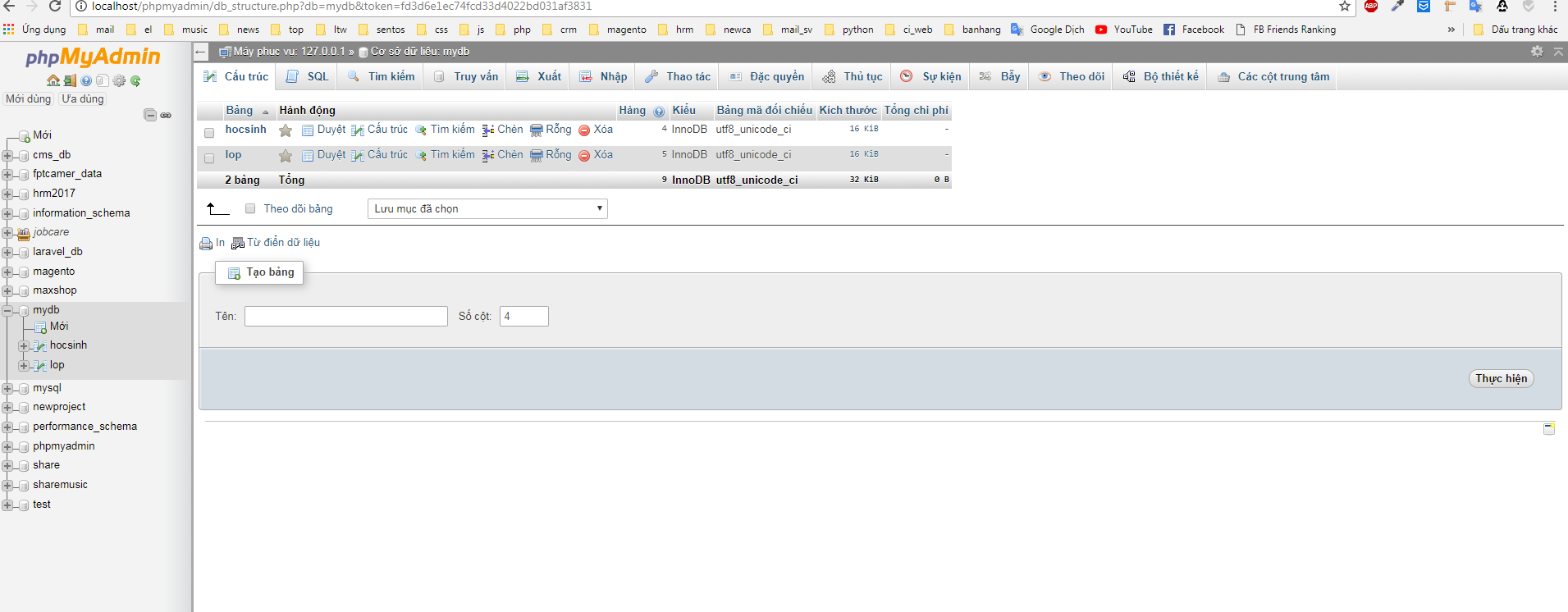
1. 1. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu gồm 2 bàng Học sinh và Lớp. Bảng học sinh để lưu thông tin đăng kí mới, bang lớp để lấy dữ liệu ra select box



*Hình 2.1 Cở sở dữ liệu*

Cài đặt xampp và sử dụng phpadmin để xây dựng database



*Hình 2.2 Giao diện phpadmin*

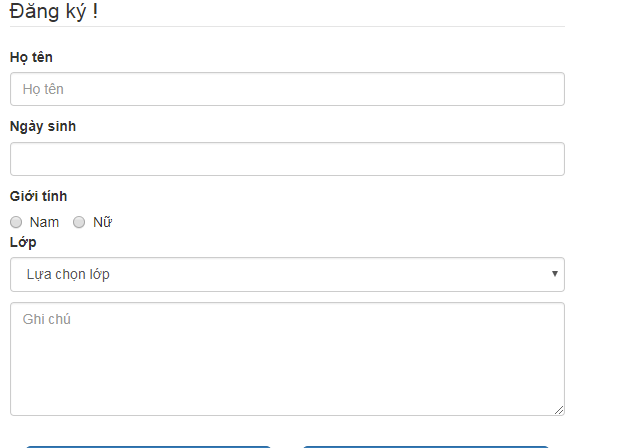
Tạo bảo lớp

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu |
| 1 | Id | Interger |
| 2 | Name | Char |

Tạo bảo học sinh

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu |
| 1 | Id | Interger |
| 2 | Hoten | Char |
| 3 | Ngaysinh | Date |
| 4 | Gioitinh | Char |
| 5 | Lop | Interger |
| 6 | Ghi chu | Char |

* 1. Xây dựng chức năng
     1. Xây dựng form đăng kí

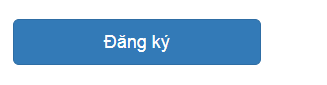


*Hình 2.3 Giao diện form đăng kí*

From đăng kí sẽ gồm các trường:

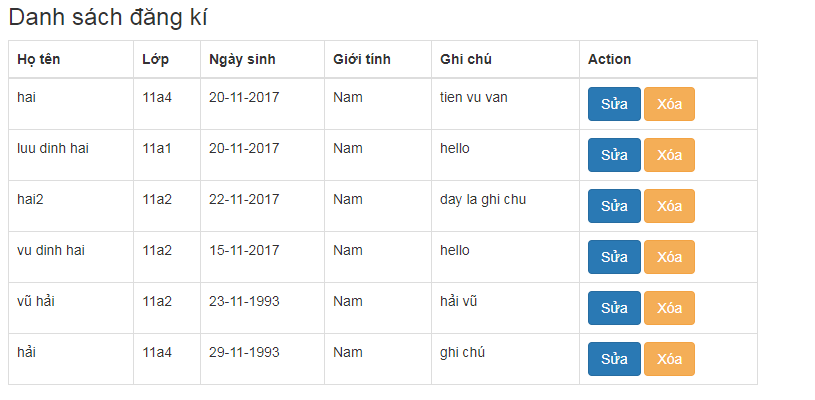
* Họ tên (Cho phép nhập text và không được trống)
* Ngày sinh (Định dạng ngày / tháng / năm): Sử dụng thư viện datepicker của boostrap.
* Giới tính: Sử dụng checkbox
* Lớp: sử dụng selectbox , lấy danh sách lớp trong bảng lớp thông qua model và truyền qua controller
* Ghi chú: sử dụng textarea

Sau khi nhập các dữ liệu thành công bấm nút “đăng kí” để thêm mới bản ghi học sinh vào bảng học sinh



Dữ liệu sẽ được gửi qua controller và model để lưu 1 bản ghi mới vào bảng học sinh

* + 1. Xây dựng bảng danh sách học sinh



*Hình 2.4 Danh sách form học sinh*

Sử dụng controller trỏ vào hàm “get\_all” trong model để lấy ra danh sách học sinh

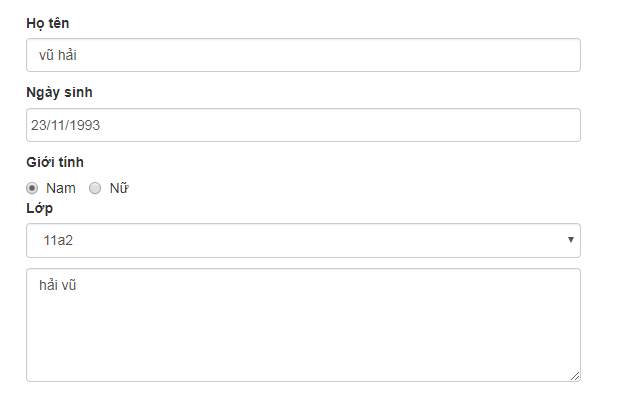
Sau khi có dữ liệu sẽ gửi qua view để đổ dữ liệu ra table

Sử dụng foreach để đổ dữ liệu vào bảng. Các dữ liệu hiển thị bao gồm :

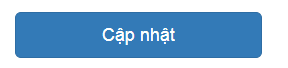
* Họ tên
* Lớp
* Ngày sinh
* Giới tính
* Ghi chú
* Cột thể hiện các hành động sửa, xóa
  + 1. Xây dựng chức năng sửa

Trong danh sách học sịnh chọn button “sửa” để sửa dữ liệu

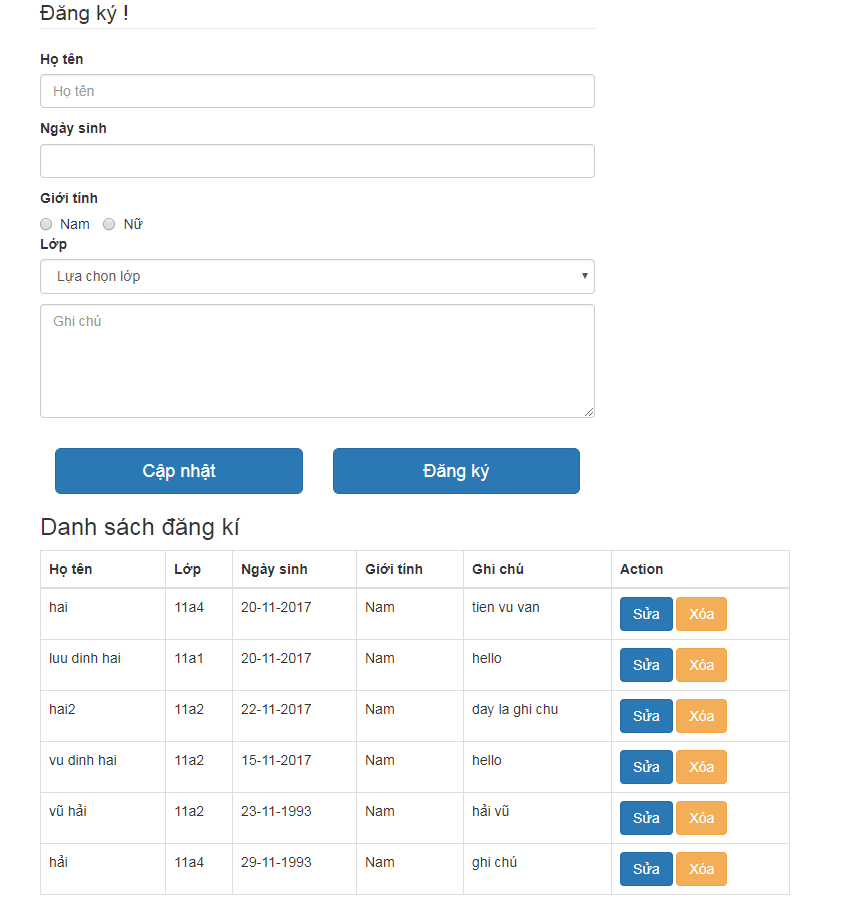


Sau khi bấm sửa, dữ liệu của học sinh đó sẽ được load lại lên form đăng kí

Có thể tủy chỉnh dữ liệu trong form sau đó bấm button “cập nhật” để thay đổi dữ liệu



Dữ liệu sẽ được cập nhật vào database sau đó trang sẽ tự động refresh để hiển thị lại danh sách vừa cập nhật



* 1. Tổng kết
     1. Ưu điểm của codeigniter framework

Sau quá trình thực hành với Codeigniter framework và PHP em nhận thấy đây là framework rất tốt cho những người mới tìm hiểu về framework PHP như em. Cụ thể CI framework khá nhẹ, cài đặt dễ dàng, cấu trúc thư mục rõ ràng, dễ hiểu. Khi sử dụng codeigniter rất thuận tiện với các modules có sẵn, các thư viện hỗ trợ nhiều đa dạng giúp người tiện lợi trong việc xử lý. CI yêu cầu lập trình viên làm hầu hết mọi việc phù hợp với các lập trình viên mới rèn luyện các kỹ năng lập trình.

* + 1. Nhược điểm của codeigniter framework

Mặc dù có rất nhiều ưu điểm, nhưng sau quá trình thực hành với codeigniter framework em cũng nhận thấy một số điểm yếu của CI framework. Đó là CI chưa hỗ trợ mapping database, nên việc kết nối và thao tác với database mất nhiều thời gian và công sức. CI cũng chưa hỗ trợ một số modules thông dụng như đăng nhập, đăng kí.

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1]. Thái Thanh Tùng, Slide bài giảng môn lập trình web – Khoa CNTT – Viện Đại học Mở Hà Nội,2015

[2]. Nguyễn Thị Tâm – Trương Tiến Tùng – Trịnh Thị Xuân, *Cơ sở dữ liệu*, Khoa Công Nghệ Thông Tin – Viện Đại học Mở Hà Nội, 2012.

[3]. Nguyễn Văn Vỵ, Giáo trình phân tích và thiết kế tin học hệ thống, NXB ĐHQG, 2007.

**Tiếng Anh**

[1]. Kevin Yank, *Build Your Own Database Driven Web Site Using PHP & MySQL*, Sitepoint, 2011.

[2]. Joel Murach và Ray Harris, Lập trình nâng cao PHP và MySQL – **Nhà xuất bản Murach, Trường Đại học FPT**