**I.Giới thiệu về PHP**

**1.PHP là gì?**

* PHP viết hồi qui của "PHP: Hypertext Preprocessor".
* PHP là ngôn ngữ lập trình kịch bản viết cho máy chủ mà được nhúng trong HTML. Nó được sử dụng để quản lý nội dụng động, Database, Session tracking, …
* Nó được tích hợp với một số Database thông dụng như MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.
* PHP thực thi rất tuyệt vời, đặc biệt khi được biên dịch như là một Apache Module trên Unix side. MySQL Server, khi được khởi động, thực thi các truy vấn phức tạp với các tập hợp kết quả khổng lồ trong thời gian Record-setting.
* PHP hỗ trợ một số lượng rộng rãi các giao thức lớn như POP3, IMAP, và LDAP. PHP4 bổ sung sự hỗ trợ cho Java và các cấu trúc đối tượng phân phối (COM và CORBA).

## **2.Các đặc điểm của php**

* PHP là một ngôn ngữ lập trình web rất được ưa chuộng, hiện là ngôn ngữ lập trình web phổ biến nhất. Nhờ vào một số đặc điểm sau:
* PHP dễ học và linh động
* Rất nhiều hàm hỗ trợ và nhiều phần mở rộng phong phú
* Đặc biệt mã nguồn mở, thường xuyên nâng cấp, chạy được trên nhiều máy chủ web, nhiều hệ điều hành (đa nền tảng)
* Cộng đồng sử dụng và hỗ trợ đông đảo
* Ngoài phần code chính (thường gọi là code thuần), các phần mở rộng cũng rất phong phú mà lại miễn phí như nhiều frame work, nhiều CMS
* Các mã nguồn chia sẻ trên mạng tìm được rất nhiều và dễ dàng
* Được tích hợp và sử dụng ổn định trong một mô hình LAMP = Linux+Apache+Mysql+Php, mã nguồn mở, chi phí thấp.
* Các hosting hỗ trợ nhiều

### **3.Các chức năng của php**

Trước hết và cũng là quan trọng nhất, nó đảm nhiệm vai trò của: **ngôn ngữ kịch bản phía máy chủ (Server Side script)** - máy chủ sẽ tiếp nhận request (yêu cầu) từ máy khách (client) - máy chủ web sẽ triệu gọi file mã nguồn tương ứng. Trong file mã nguồn này chứa các mã php để xử lý request - trình thông dịch sẽ dịch mã php sang mã HTML, CSS, XML,...trả ra cho máy chủ web, máy chủ web trả lại thông tin (reponse) cho máy khách. Dữ liệu nhận được từ máy khách là các đoạn mã dạng text như HTML, CSS,...mà không thể thấy mã php ( vì đã được thực thi thành dạng text ) - đảm bảo được tính bảo mật, đây cũng là chức năng cơ bản và quan trọng của một ngôn ngữ phía máy chủ. 

Ngoài ra PHP còn có thể:

Xử lý ảnh

Cho phép và xử lý file upload ( tải file lên server )

Thiếp lập và xử lý cookie, session

(\*) Thao tác tới cơ sở dữ liệu

Gửi email

(\*) Xử lý dữ liệu nhận được từ form nhập liệu

Thông qua php, một trang web tĩnh (static webpage), thường chỉ phần giao diện tại máy khách - tương tác tới máy chủ web , trở thành một trang web động ( nội dung thay đổi, tùy biến theo các sự kiện, yêu cầu) - hay được gọi là dynamic webpage. Học lập trình php vì thế cũng hay được gọi là: lập trình web động với php, lập trình ứng dụng web, ...

**II.MVC  
1.Mô hình MVC là gì?**

MVC là chữ viết tắt của **M**odel - **V**iew - **C**ontroller, đây là một mô hình kiến phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC, từ đó bạn có thể thấy sự quan trọng của nó như thế nào rồi đấy.

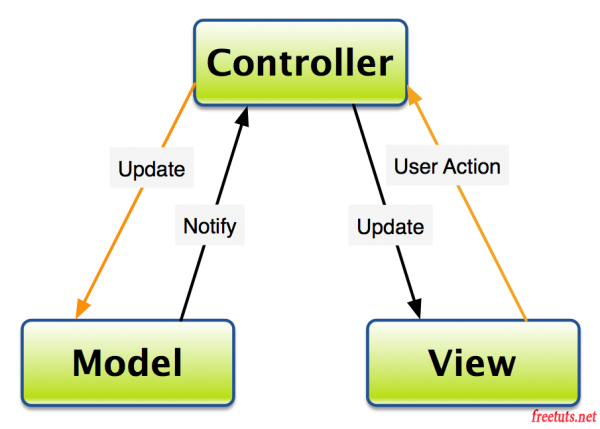
**Trong mô hình này thì**:

**Model**: có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua **View**

**View**: có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu nôm na đây người ta còn gọi là thành phần giao diện.

**Controller**: đóng vài trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client

Để rõ ràng hơn thì bạn xem hình dưới đây:



**2.Ưu điểm và nhược điểm mô hình MVC**

**Ưu điểm**:

Hệ thống phân ra từng phần nên dễ dáng phát triển

Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án

Vấn đề bảo trì cũng tương đối ok, dễ nâng cấp

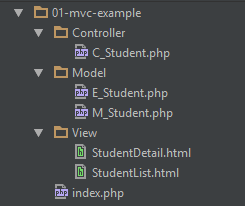
Dễ dàng debug trong quá trình xây dựng

**Nhược điểm:**

Hệ thống sẽ chạy chậm hơn PHP thuần, tuy nhiên nó ko phải là vấn đề :D

Xây dựng cầu kì và mất thời gian để xây dựng thư viện, cấu trúc

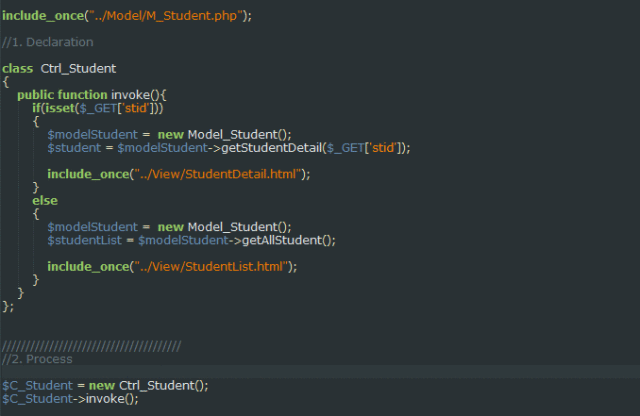
Trong ví dụ tới này,tổ chức source code minh hoạ như mô hình sau đây:

   
Tổ chức source code của ứng dung minh hoạ

**a. Tầng Controller**

– Đây là tầng điều phối của ứng dụng: nhận request từ client, phân tích request, gọi tầng Model để lấy dữ liệu và tầng View để tổ chức hiển thị dữ liệu.

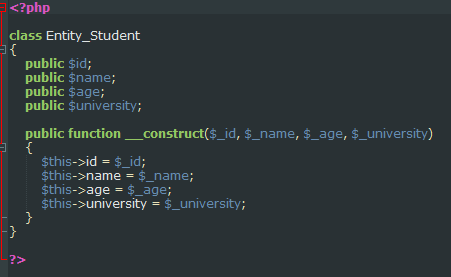
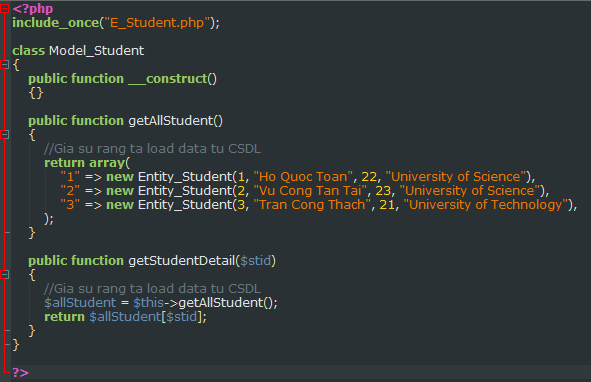
– Ở đây ta nói tới chức năng quản lí học sinh, nó sẽ được điều phối bởi controller C\_Student.php. Do trong chức năng có 2 chức năng con: Hiển thị danh sách học sinh, Hiển thị chi tiết học sinh; vậy nên nó sẽ phân biệt từng tính năng bằng cách kiểm tra tham số đầu vào là “stid”. Tuỳ trường hợp mà controller sẽ gọi view tương ứng.

Cài đặt controller của ứng dung

**b. Tầng Model và các lớp thực thể (Entity class)**

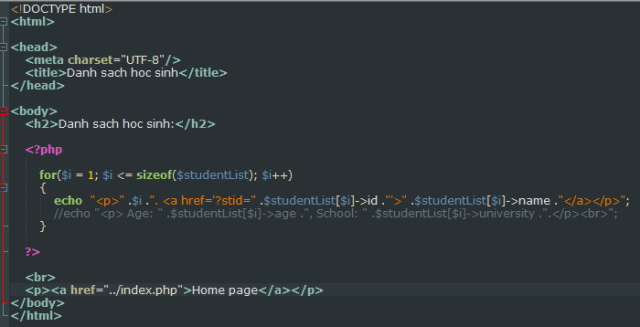
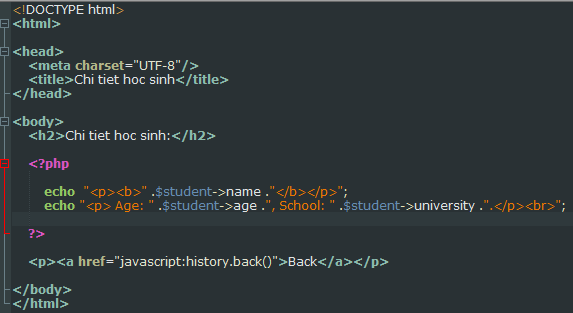
– Nhiệm vụ chính của tầng này là việc thao tác với cơ sở dữ liệu để lưu trữ cũng như truy vấn dữ liệu. Các thao tác với CSDL phải được đặt hết ở trong tầng này, nó sẽ chứa các thao tác cơ bản để truy vấn CSDL như: đọc, thêm, xoá, sửa.

– Ngoài ra thì tằng này còn chứ các lớp thực thể (Entity class), nó là các lớp để mô phỏng những đối tượng trong thực tế. Trong ví dụ này, tầng Model bao gồm 2 class: Model\_Student và Entity\_Student. Lớp thực thể Entity\_Student chính là 1 thực thể mô phỏng về 1 student trong thực tế (tên, tuổi, …), lớp Model\_Student sẽ sử dụng lớp thực thể này để cấu trúc dữ liệu sẽ lấy từ Database. Lớp Model\_Student đóng vai trò giao tiếp với CSDL.

  
Entity class, mô tả các bảng trong CSDLLớp Model phụ trách phần lien kết dữ lieu Database

**c. Tầng View**

– Lớp này dùng để định dạng hiển thị dữ liệu lên cho người dùng, do ta có 2 tính năng trong phần “quản lí học sinh” là: “danh sách học sinh” và “chi tiết học sinh”, nên ta sẽ có 2 view khác nhau tương ứng. Controller sẽ truyền các biến dữ liệu vào các View để chúng có thể dùng các dữ liệu này để hiển thị lên. Việc này được thực hiện bằng cách nhúng mã PHP vào trong mã HTML.

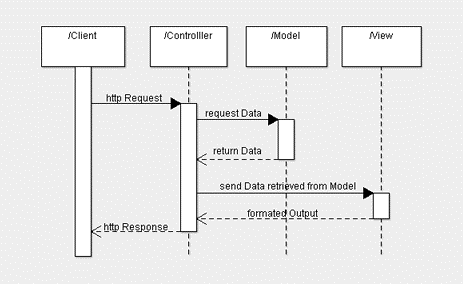
 View của “Danh sách học sinh”Fig View của “Chi tiết học sinh”

Nhận xét View:

Việc nhúng mã PHP vào HTML khiến mã nguồn trở nên phức tạp và khó kiểm soát, việc này dẫn tới sự ra đời của các “template engine” sau này, các “template engine” tách biệt mã PHP ra khỏi mã HTML nên sẽ khiến code trở nên rõ ràng hơn rất nhiều. Chúng ta sẽ tìm hiểu sau.

Đối với các ứng dụng thuần AJAX, phần View có thể được triển khai bởi javascript, kéo theo việc sử dụng cấu trúc JSON.

– Mô hình tuần tự sau sẽ giúp bạn hình dung luồng hoạt động của một http request trong mô hình MVC:

  
Sơ đồ tuần tự mô tả luồng chạy của 1 xử lí trong mô hình MVC

**III.Giới thiệu về Laravel Framework**

**1. Laravel Framework?**

[Laravel](http://laravel.com/)là 1 open source, là một framework dùng để xây dựng web application, được thiết kế dựa trên mô hình MVC (Model, Controller, View), toàn bộ source code được đặt trên github. Theo kết qủa khảo sát của các Developer vào tháng 12 năm 2013, thì Laravel Framework đứng top 1 một trong những framework phổ biến nhất, tiếp sau là [Phalcon](http://en.wikipedia.org/wiki/Phalcon_%28framework%29), [Symfony2](http://en.wikipedia.org/wiki/Symfony2), [CodeIgniter](http://en.wikipedia.org/wiki/CodeIgniter) và các framework khác. Tháng 8 năm 2014, Laravel Framework được xem như là một dự án PHP phổ biến nhất trên Github

Laravel hiện đang là framework tốt nhất, đứng TOP 1 trong 3 năm liền trên bảng xếp hạng các Framework tốt nhất.   
**2. Sơ lược các tính năng cơ bản của Laravel Framework**

**Bundles** : Ở laravel phiên bản 3.x, cung cấp một hệ thống đóng gói các module, với rất nhiều tính năng đi kèm.

**Composer** : Ở laravel phiên bản 4.x, được sử dụng như một công cụ quản lý với tính năng như thêm các gói cài đặt, các chức năng PHP phụ trợ cho Laravel có trong kho Packagist.

**Eloquent ORM** (object relation mapping) : ánh xạ các đối tượng và quan hệ cơ sở dữ liệu, cung cấp các phương thức nội bộ để thực thi đồng thời cũng bổ sung các tính năng hạn chế về mối quan hệ giữa các đối tượng cơ sở dữ liệu. Eloquent ORM trình bày các bảng trong cơ sở dữ liệu dưới dạng các lớp, cung cấp thêm lựa chọn truy cập cơ sở dữ liệu trực tiếp mới mẻ hơn, chuyên nghiệp hơn.

**Application logic** : Là một phần của phát triển ứng dụng, được sử dụng bởi bộ điều khiển controllers.

**Routes** : Định nghĩa mối quan hệ giữa các đường dẫn (url), các liên kết (link) . Khi một liên kết được tạo ra bằng cách sử dụng tên của routes, thì một định danh liên kết thống nhất sẽ được tạo ra bởi laravel.

**Restful Controller** : cung cấp các tùy chọn để tách các logic phía sau các request HTTP POST, GET

**Class auto loading** : cung cấp việc tải tự động các class trong PHP, mà không cần include các class vào. Tùy thuộc vào yêu cầu các class cần thiết sẽ được nạp vào, hạn chế các class không cần thiết.

**View** : chưa các mã html, hiển thị dữ liệu được chỉ định bởi controller

**Migrations** : cung cấp một hệ thống kiểm soát các phiên bản lược đồ cơ sở dữ liệu (database cheme), làm cho web ứng dụng có khả năng tương tác phù hợp những thay đổi logic, các đoạn mã code của ứng dụng và những thay đổi cần thiết trong việc bố trí cơ sở dữ liệu, triển khai nới lỏng và cập nhậtcác ứng dụng.

**Unit Testing** : đóng một vai trò quan trọng trong Laravel, Unit testting chứa rất nhiều các hệ thống unit testing, giúp phát hiện và ngăn chặn lỗi trong khuôn khổ nhất định. Unit Testing có thể đượcchạy thông qua tiện ích command-line.

**Automatic pagination** : Tính năng tự động phân trang được tích hợp vào Laravel giúp đơn giản hóa các nhiệm vụ thực hiện phân trang so với các phương pháp thông thường.

**3.Tại sao chọn Laravel?**

* Được thừa hưởng những ưu điểm và thế mạnh của các framework khác.
* Có số lượng người sử dụng nhiều nhất
* Document rõ dàng, dễ học
* Autoload theo namespace.
* Sử dụng mô hình ORM rất đơn giản khi thao tác với DB
* Các lệnh tương tác với cơ sở dữ liệu cực kỳ ngắn gọn và thân thiện.
* Việc quản lý layout thật sự giản đơn với Balade Templating .
* Dễ dàng tích hợp các thư viện khác vào dự án, và được quản lý với Composer
* Phần route rất mạnh!!!

**IV.Sandbox   
1.Sandbox là gì?**

·         Sandbox là một môi trường thử nghiệm khép kín, được xây dựng để hoạt động tương tự môi trường sử dụng ngoài thực tế. Tại đây, quí vị có thể bắt đầu xây dựng và theo dõi quá trình hoạt động của ứng dụng mà không ảnh hưởng tới bất kì tài khoản PayPal thực tế.

·         Môi trường này sẽ hỗ trợ cho người phát triển phần mềm tiếp cận, làm thử và testing ứng dụng mà không làm ảnh hưởng đến bất kỳ đối tượng nào trong thực tế. Để từ đó, họ đánh giá, so sánh và quyết định lựa chọn dịch vụ mà họ nên tích hợp khi triển khai thực ứng dụng thực sự