****

**Hướng dẫn cơ bản Magento**

-Ho Chi Minh City, 11/2015-

# Table of Contents

[Table of Contents 2](#_Toc437350955)

[List of Tables 2](#_Toc437350956)

[Hướng dẫn cơ bản Magento 3](#_Toc437350957)

[1. Magento là gì? 3](#_Toc437350958)

[2. Lịch sử phát triển và nền tảng kỹ thuật của Magento 3](#_Toc437350959)

[3. Cài đặt và sử dụng Magento. 3](#_Toc437350960)

[3.1 Cài đặt Web Server. 3](#_Toc437350961)

[3.2 Config server trước khi chạy Magento. 6](#_Toc437350962)

[3.3 Cài đặt Magento trên localhost. 8](#_Toc437350963)

[3.4 Một số thao tác cần thiết sau khi cài đặt thành công Magento. 15](#_Toc437350964)

[4. Import project Magento vào IDE 18](#_Toc437350965)

[4.1 Cài đặt debugger cho Eclipse. 18](#_Toc437350966)

[4.2 Tạo project mới và import source code Magento. 19](#_Toc437350967)

[5. Kiến trúc trong Magento. 23](#_Toc437350968)

[5.1 Cấu trúc thư mục của Magento. 24](#_Toc437350969)

[5.2 Cấu trúc module trong Magento. 27](#_Toc437350970)

[5.3 Blocks, Templates và Layouts trong Magento. 28](#_Toc437350971)

[6. Tạo một module mới trong Magento 31](#_Toc437350972)

[6.1 Khai báo 1 module mới cho Magento 31](#_Toc437350973)

[6.2 Tạo các thư mục cần thiết cho 1 module 32](#_Toc437350974)

[6.3 Cài đặt Router cho module. 33](#_Toc437350975)

[6.4 Tạo Controller cho module. 34](#_Toc437350976)

[6.5 Thiết lập Layout, Block và Template cho module. 36](#_Toc437350977)

[7. Một số tinh chỉnh giúp Magento hoạt động nhanh hơn ở phía front-end 39](#_Toc437350978)

[7.1 Điều chỉnh file .htaccess 39](#_Toc437350979)

[7.2 Điều chỉnh thiết lập trong Magento Admin 40](#_Toc437350980)

[7.3 Làm mới lại database 42](#_Toc437350981)

# List of Tables

[Table 1: Roles and Responsibility **Error! Bookmark not defined.**](#_Toc367122574)

# Hướng dẫn cơ bản Magento

## Magento là gì?

Magento là một mã nguồn mở được xây dựng bằng PHP giúp cho chúng ta có thể tạo ra các trang web thương mại điện tử. Magento được phát triển dựa trên Zend Framework.

Một số tính năng chính của Magento:

* Quản lý sản phẩm: với nhiều hình ảnh, tùy chọn nhận xét đánh giá sản phẩm, danh mục sản phẩm ưu thích, inventory.
* Quản lý category: dễ dàng tìm kiếm và chọn lọc sản phẩm theo category.
* Quản lý hàng tồn kho: quản lý sản phẩm còn lại trong kho, nhập, xuất …
* Tài khoản khách hàng: tình trạng tài khoản, lịch sử giao dịch, danh mục ưu thích, địa chỉ, giỏ hàng…
* Dịch vụ khách hàng: tăng cường các tính năng, hình thức liên hệ khách hàng; theo dõi toàn diện, dịch vụ email.
* Quản lý đơn hang.
* Thanh toán: nhiều phương thức thanh toán như thẻ tín dụng, PayPal, Authorize.net, Google Checkout, hỗ trợ các mô-đun thanh toán bên ngoài như CyberSource, ePay, eWAY và nhiều hơn thế.
* Công nghệ tìm kiếm: nhanh chóng, thân thiện, hỗ trợ tìm kiếm trên Google SiteMap.
* Hỗ trợ quốc tế: với đa ngôn ngữ và tiền tệ.
* Công cụ khuyến mãi và tiếp thị: phiếu giảm giá, khuyến mãi và nhiều tùy chọn.
* Phân tích và báo cáo: tích hợp với dịch vụ Google Analytics và cung cấp nhiều báo cáo.

## Lịch sử phát triển và nền tảng kỹ thuật của Magento

Magento chính thức bắt đầu phát triển vào đầu năm 2007. Tháng 6, 2011 eBay sở hữu 100% vốn đầu tư cho Magento. eBay tuyên bố rằng Magento sẽ tham gia X.Commerce sáng kiến ​​mới của eBay.

Magento được viết hoàn toàn bằng ngôn ngữ PHP, sử dụng hệ cơ sở dữ liệu MySQL. Cơ sở dữ liệu dựa trên mô hình EAV (entity – attribute – value).

## Cài đặt và sử dụng Magento.

Để có thể sử dụng Magento, chúng ta cần tải về và cài đặt nó, lưu ý là các bước cài đặt phải được tiến hành theo thứ tự sau:

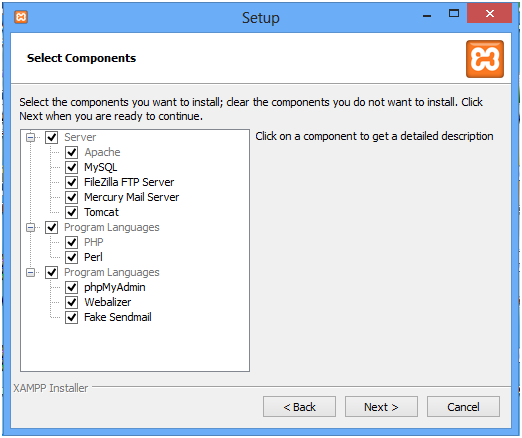
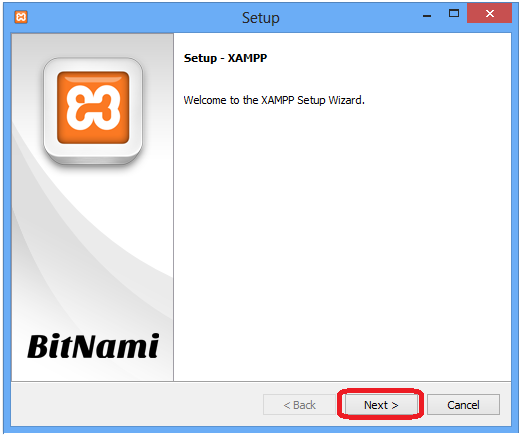
1. Cài đặt web server (xampp, wamp hoặc appserv).
2. Config server trước khi cài đặt Magento.
3. Cài đặt Magento trên localhost.
4. Một số thao tác cần thiết sau khi cài đặt thành công Magento.

### Cài đặt Web Server.

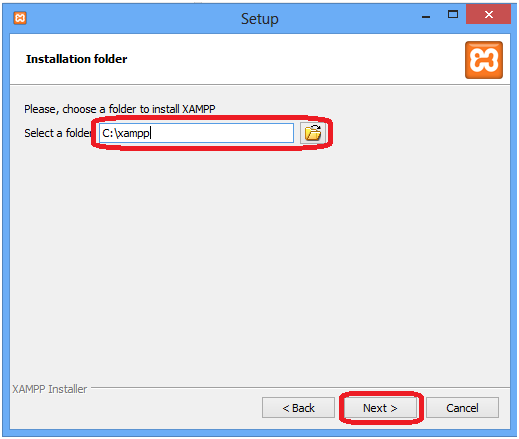
Các bạn có thể sử dụng bất kì webserver nào (xampp, wamp hoặc appserv), ở đây mình sẽ sử dụng XAMPP.

***Bước 1:*** Tải về và cài đặt phiên bản mới nhất của XAMPP tại đường dẫn <http://www.apachefriends.org/en/xampp.html>

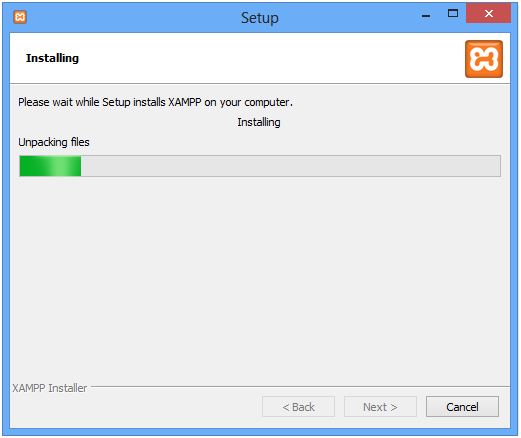
***Bước 2:*** Chạy file cài đặt, chọn các thành phần muốn cài.



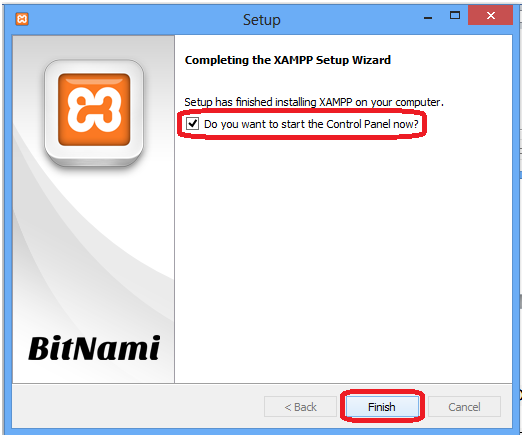
***Bước 3:*** Mặc định hệ thống sẽ cài đặt vào thư mục *xampp* ở ổ C, các bạn có thể thay đổi sang thư mục khác nếu muốn.



***Bước 4:*** Bấm Next để bắt đầu quá trình cài đặt.



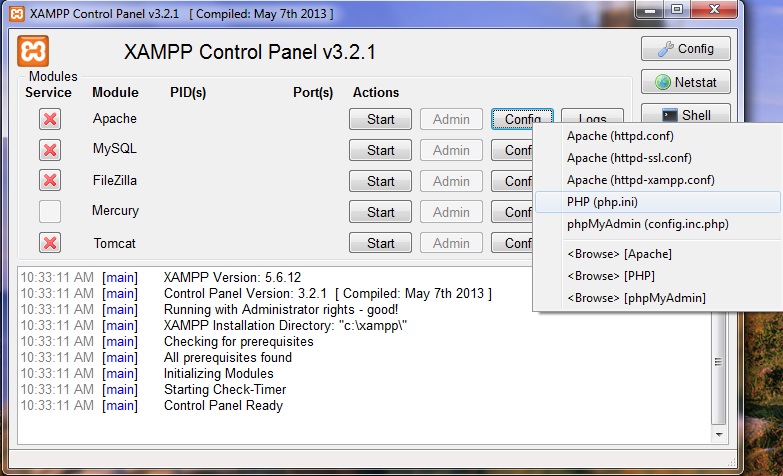
***Bước 5:*** Nhấn Finish để kết thúc quá trình cài đặt, các bạn có thể chạy Control Panel sau khi cài đặt xong nếu muốn.



### Config server trước khi chạy Magento.

* *Config file php.ini*

Mở Control Panel của XAMPP, click vào button “Config” của Apache, chọn *PHP*

*(php.ini).*

Cần sửa 1 số thông tin trong file php.ini như sau:

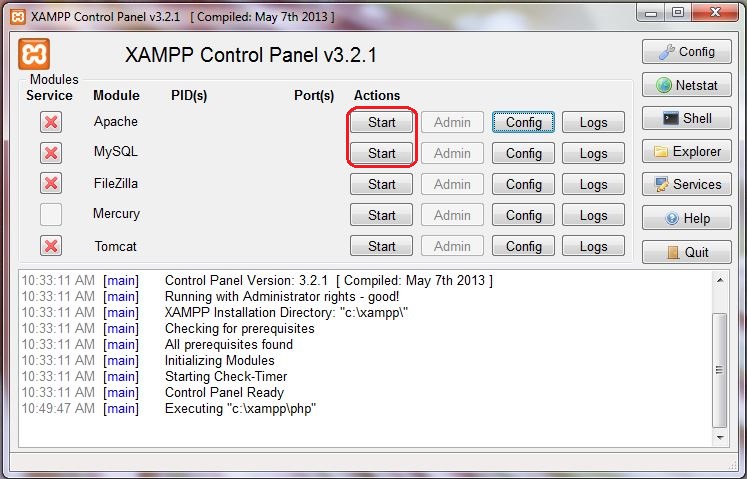
upload\_max\_filesize = 256M

post\_max\_size = 512M

memory\_limit = 1024M

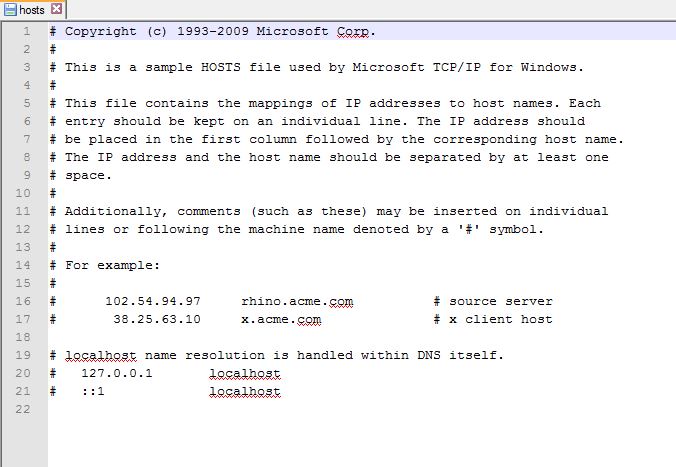
max\_input\_time = 3600

max\_execution\_time = 3600

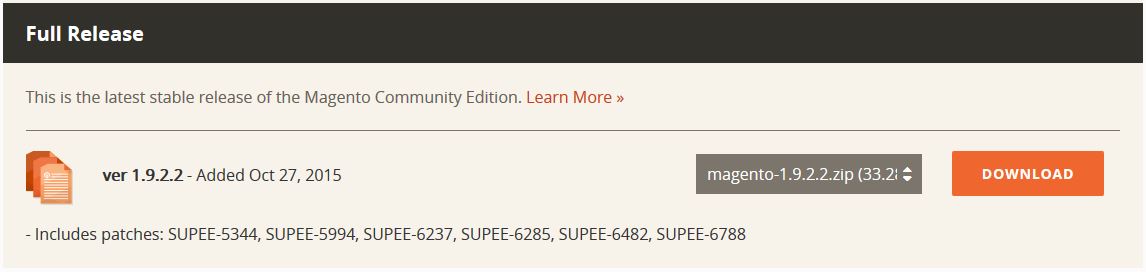
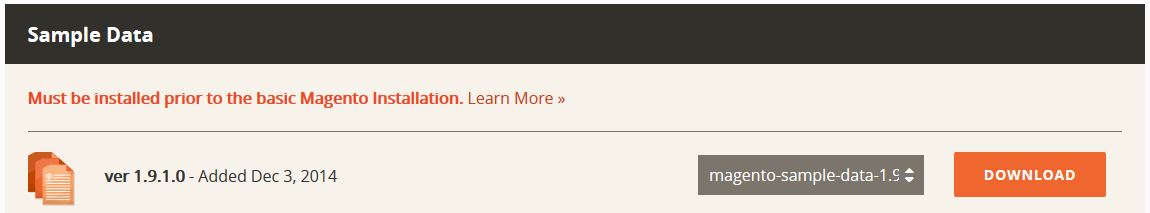
Sau khi config xong thì chúng ta có thể start Apache và MySQL.

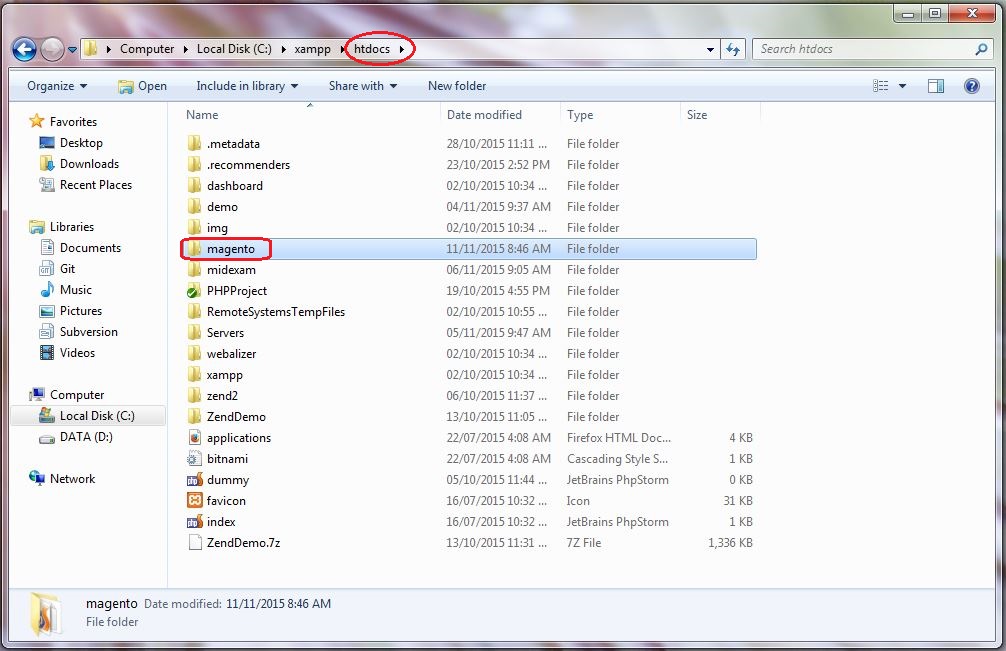
* *Config file host*

Mở file host ở thư mục *C:\Windows\System32\drivers\etc\hosts* bằng text editor, nếu chưa có thì thêm dòng này vào cuối file:

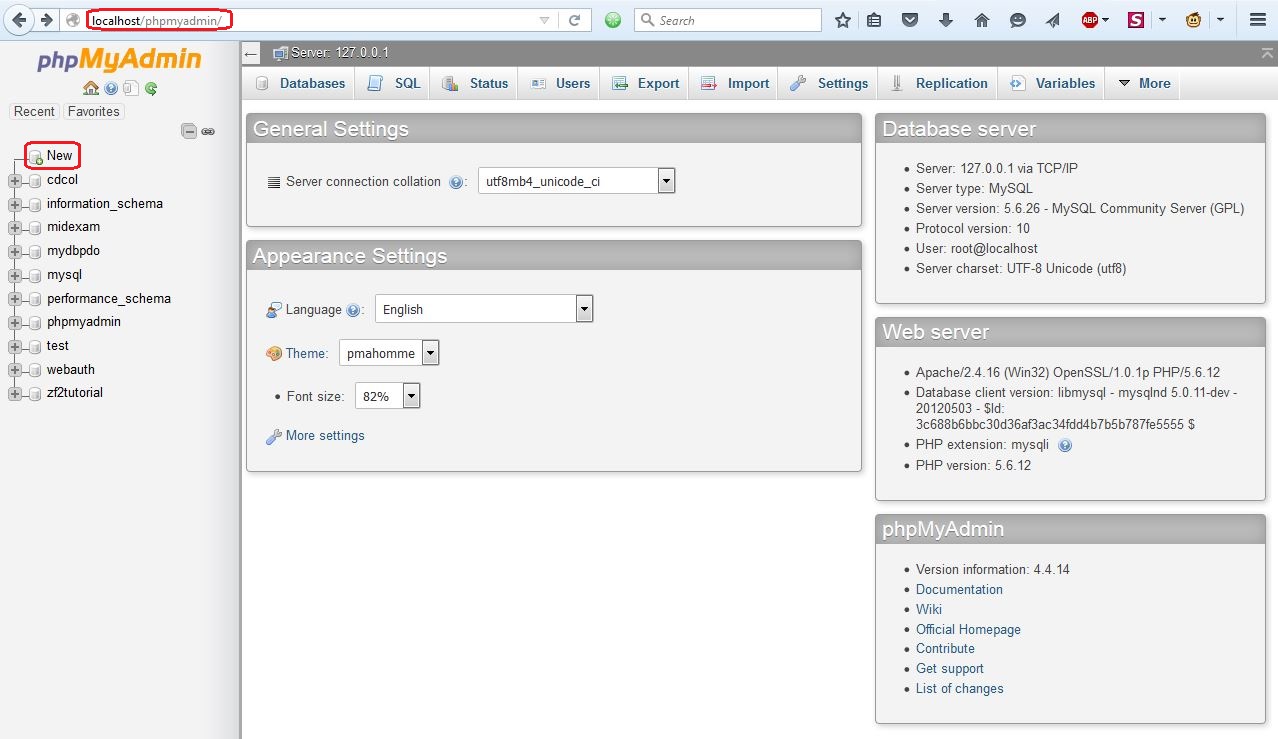
*# 127.0.0.1 localhost.com*

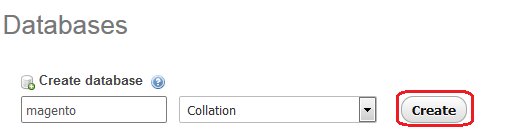
### Cài đặt Magento trên localhost.

***Bước 1:*** Tải phiên bản magento và sample data mới nhất định dạng zip từ trang chủ về. <http://www.magentocommerce.com/download>.

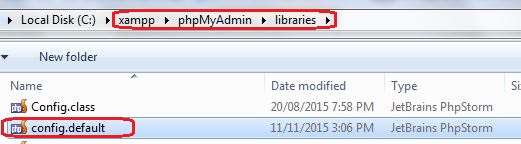
Sau khi download giải nén source code magento và copy vào thư mục chạy php của server. (www của appserv hoặc wamp, htdocs của xampp). Ở đây mình giải nén và copy vào thư mục tên là “*magento*”.

***Bước 2:*** Import sample data.

Vào phpadmin tạo một database mới (tên là “*magento*”).

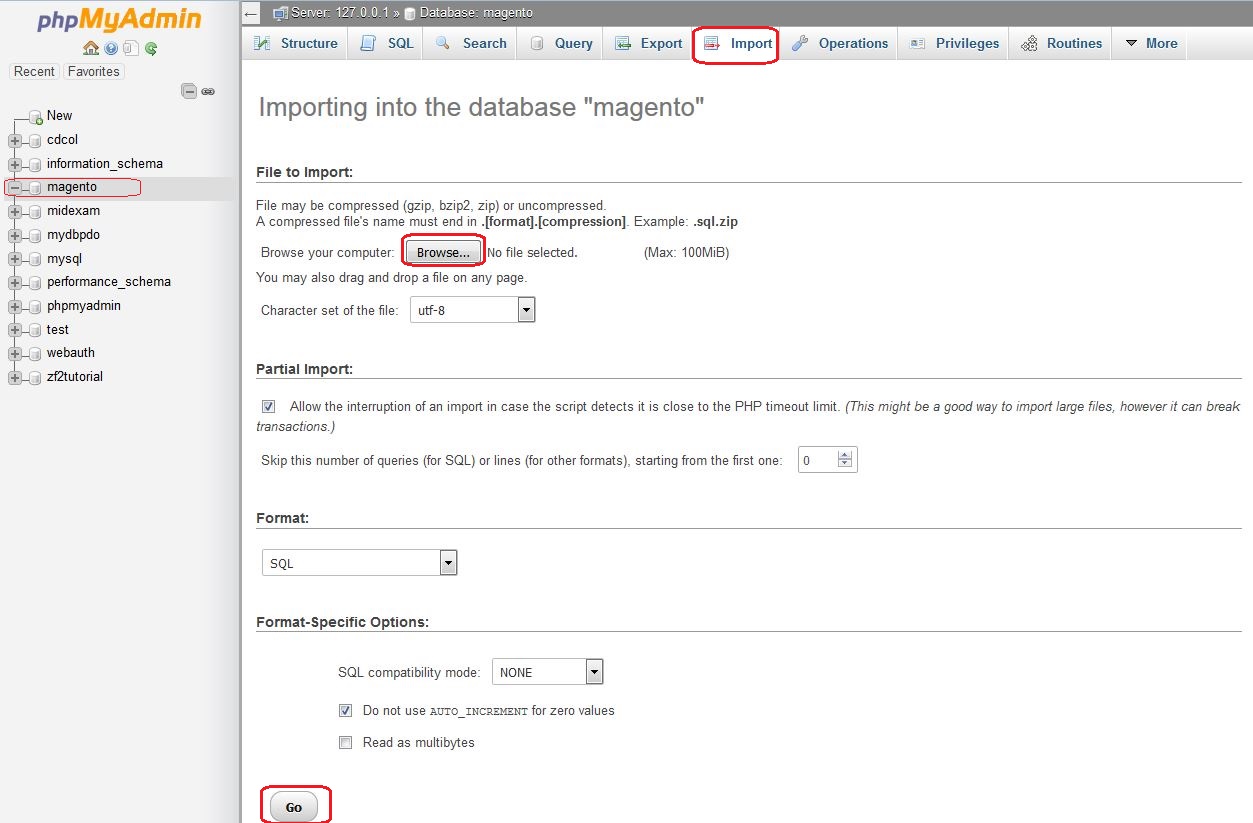
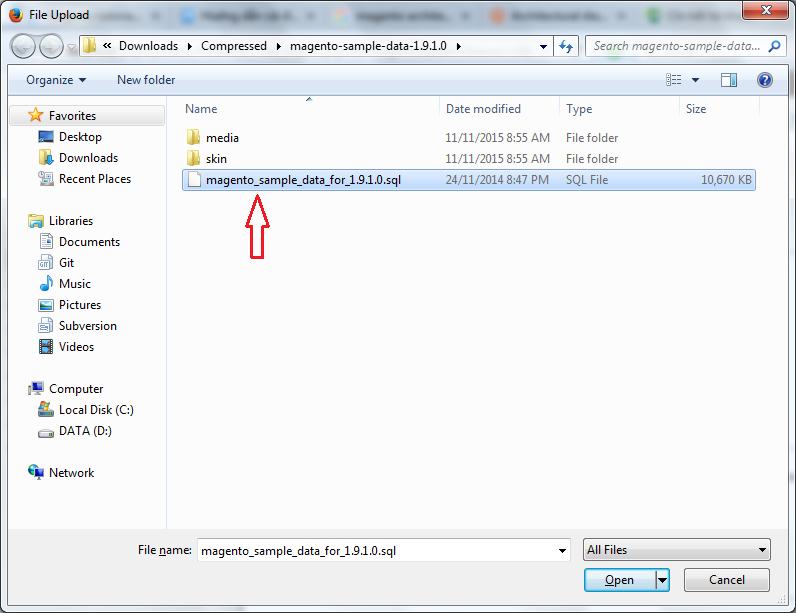


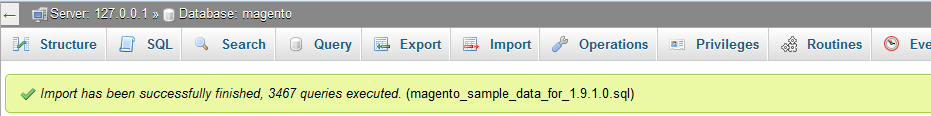
\* Lưu ý nhỏ: Vì file import khá lớn và tốn nhiều thời gian để import, chúng ta cần chỉnh sửa thời gian timeout khi thực hiện import, mặc định là 300 giây, mình chỉnh lại là 3600 giây tức là 1 tiếng, tránh tình trạng đang import thì quá thời gian timeout thì server sẽ ngừng import. Để chỉnh sửa, các bạn vào đường dẫn *\xampp\phpMyAdmin\libraries*, mở file *config.default.php* bằng text editor, tìm và thay đổi giá trị của dòng $cfg['ExecTimeLimit'] thành 3600. Sau đó restart server.



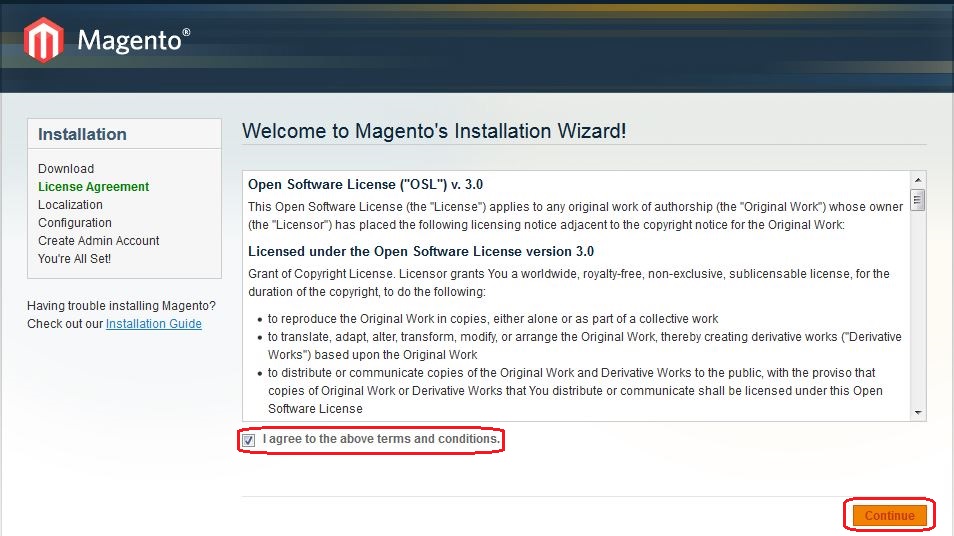


Sau khi tạo xong database thì chúng ta bắt đầu import sample data đã tải ở trên. Giải nén thư mục vừa tải về. Chọn tab Import, trỏ tới file sql trong thư mục vừa giải nén, bỏ tick phần Partial Import, chọn Go và đợi cho server import vào.



Kiểm tra đủ 359 bảng và có thông báo successfully sau khi import là đúng.

***Bước 3:*** Cài đặt magento.

Truy cập đường dẫn [*http://localhost/magento*](http://localhost/magento), màn hình cài đặt xuất hiện, nhấn Đồng ý điều khoản và Continue.

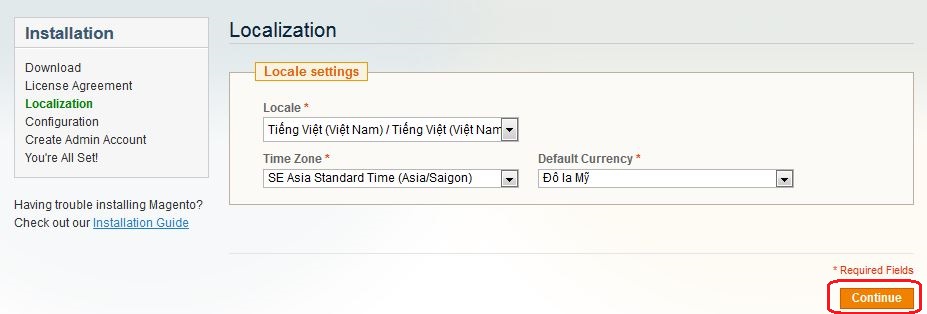
Thay đổi config cho phù hợp với website của mình:

+ Locale: ngôn ngữ

+ Timezone: thời gian

+ Default currency: tiền tệ mặc định

Sau đó click “Continue” để sang trang config database



Điền thông tin về database:

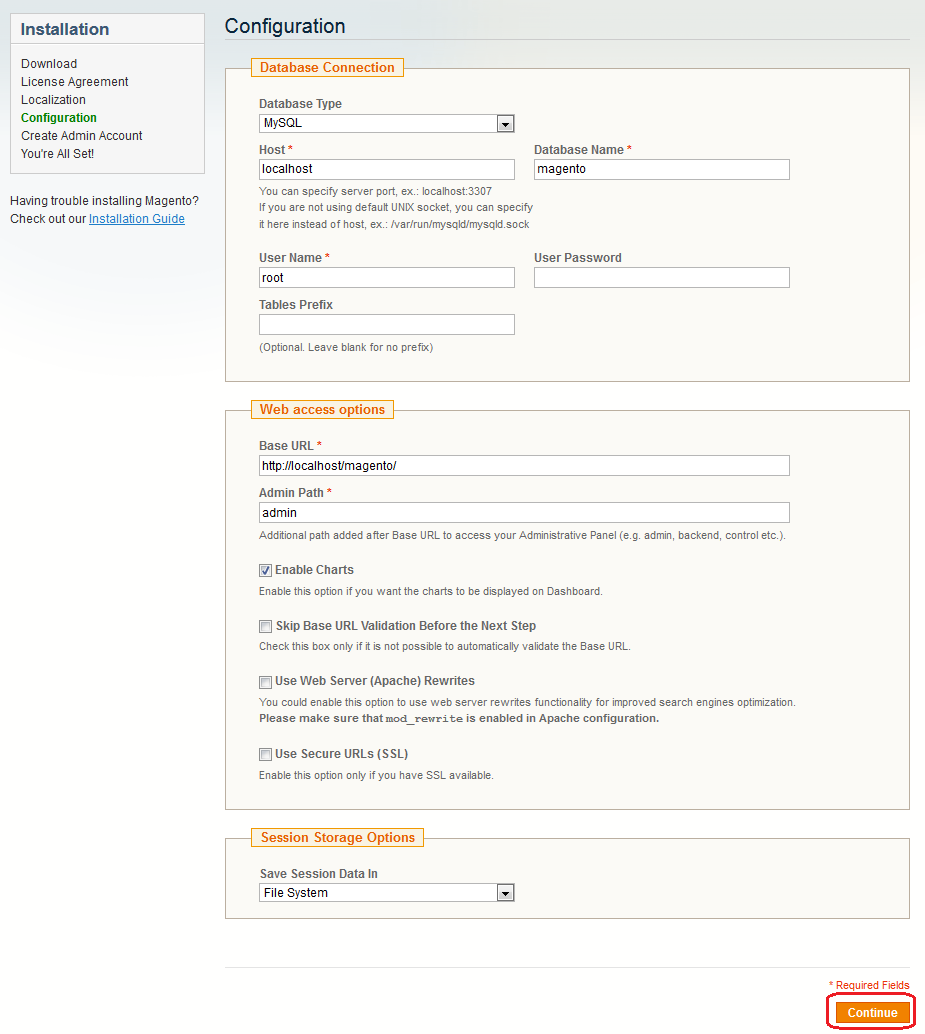
+ Host: host name (localhost).

+ User name: username của user quản lý database.

+ Password: password của user quản lý database.

+ Table prefix: prefix của table trong database.

Sau khi điền xong thông tin click “Continue” để bắt đầu quá trình cài đặt.



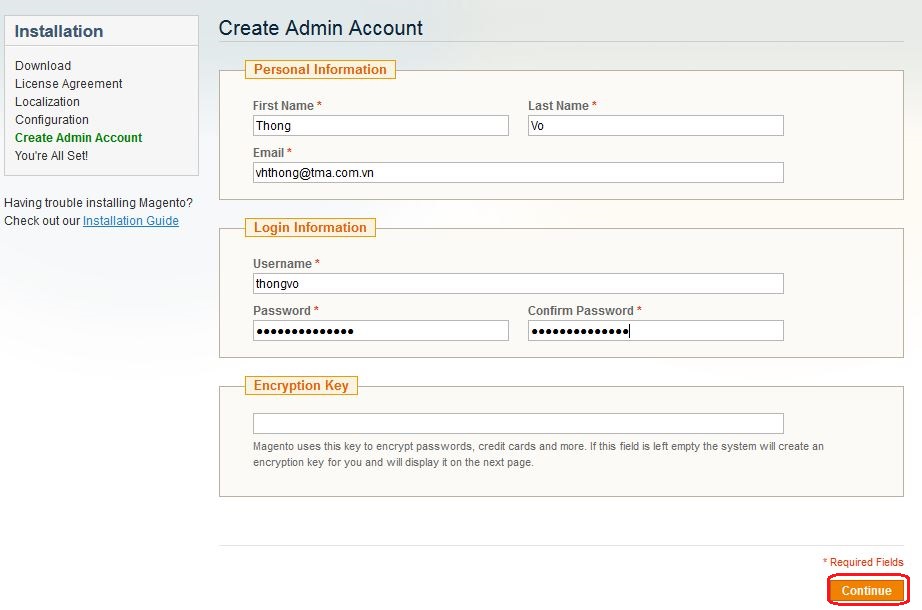
Trước khi kết thúc quá trình cài đặt, màn hình sẽ cho phép người dùng điền thông tin của admin:

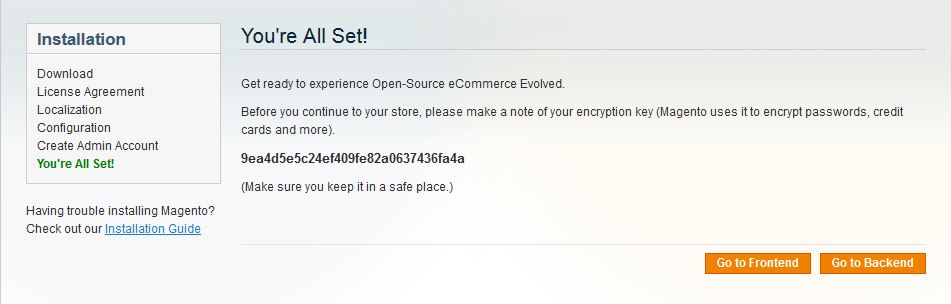
+ First Name.

+ Last Name.

+ Username.

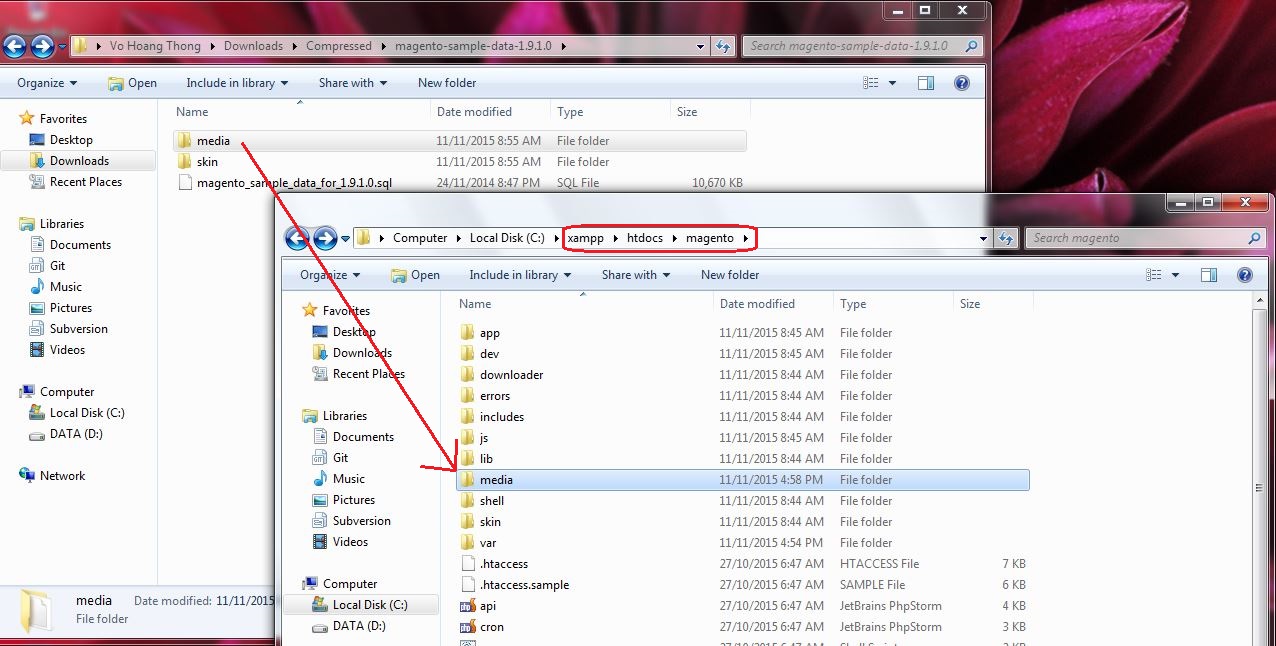
+ Password.



Như vậy chúng ta đã kết thúc quá trình cài đặt magento trên localhost. Chúng ta có thể truy cập vào trang admin khi click vào button “Go to Backend” và truy cập vào trang frontend khi click vào button “Go to Frontend”.

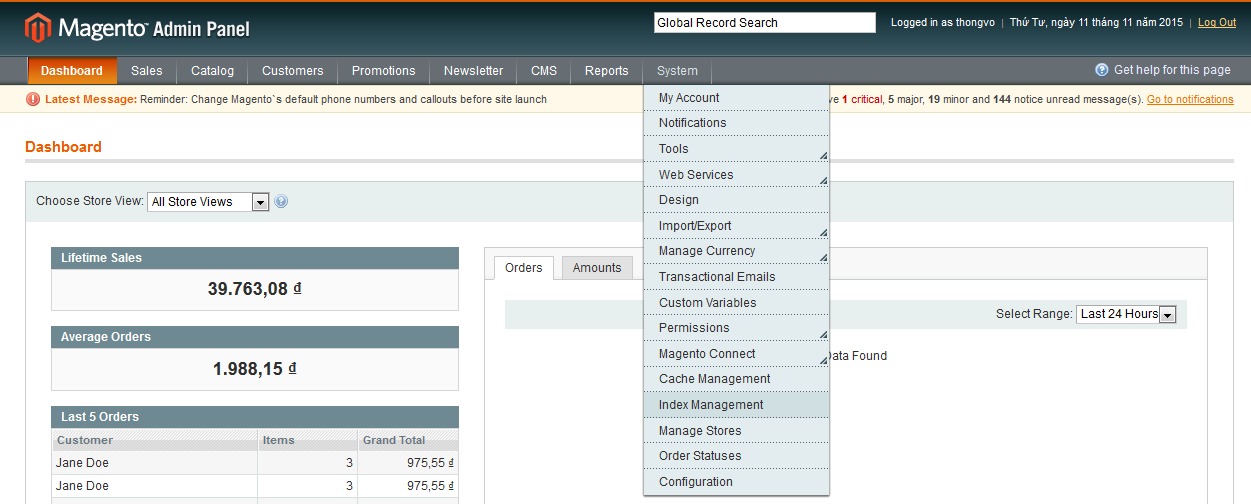
### Một số thao tác cần thiết sau khi cài đặt thành công Magento.

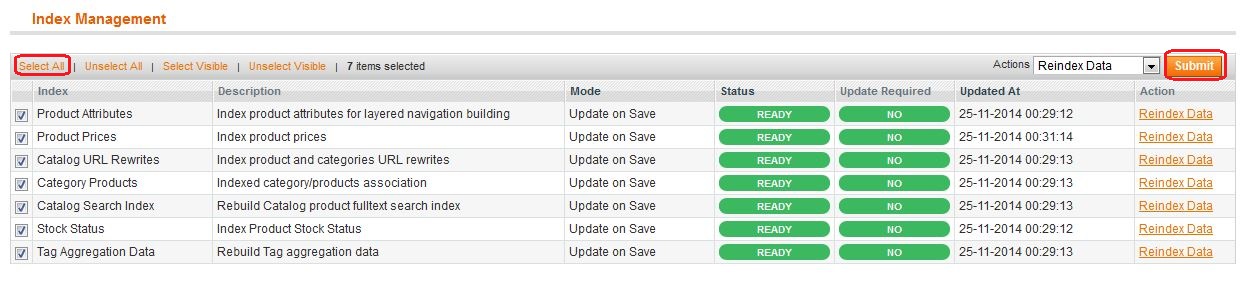
#### Hiển thị hình ảnh của sản phẩm trong sample data.

Copy thư mục *media* trong thư mục đã giải nén của sample data vào thư mục root của magento trong server.

#### Reindex data.

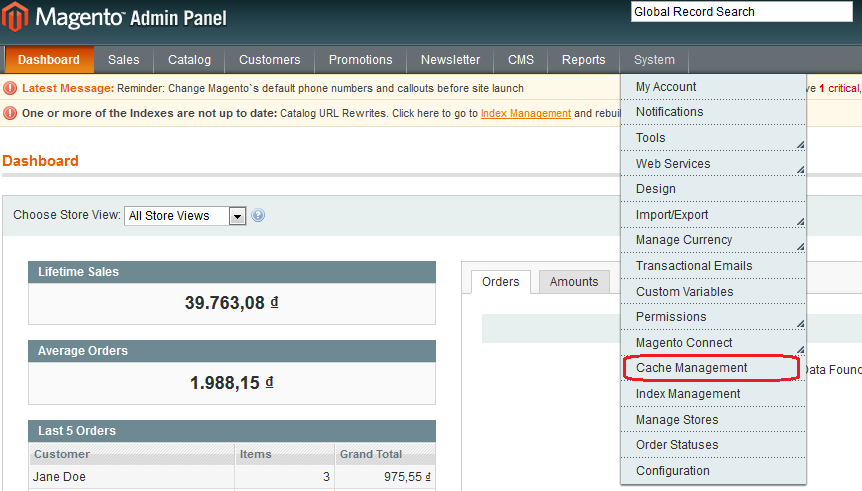
Magento sử dụng index nhằm tăng tốc độ truy xuất của trang web, chính vì vậy mỗi khi bảng chính thay đổi dữ liệu, chúng ta cần phải reindex lại cho các bảng indexer.

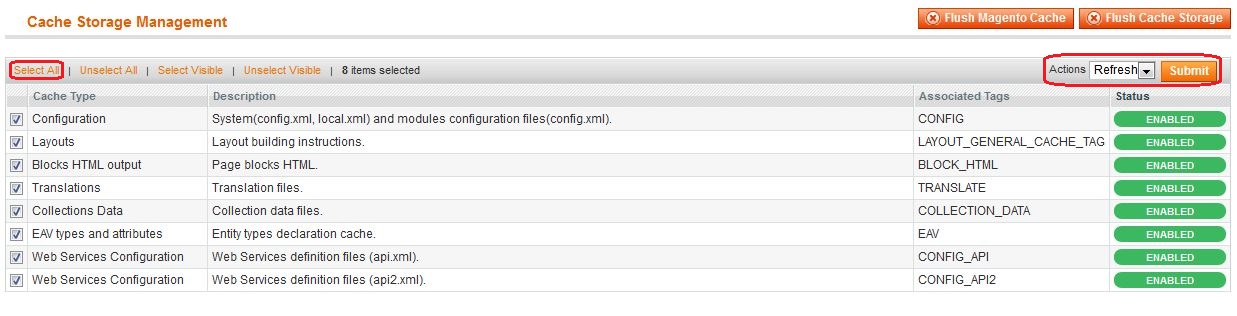
Sau khi điền user name và password của admin hệ thống sẽ redirect sang trang admin dasboard. Click vào menu ***System->Index Management.***

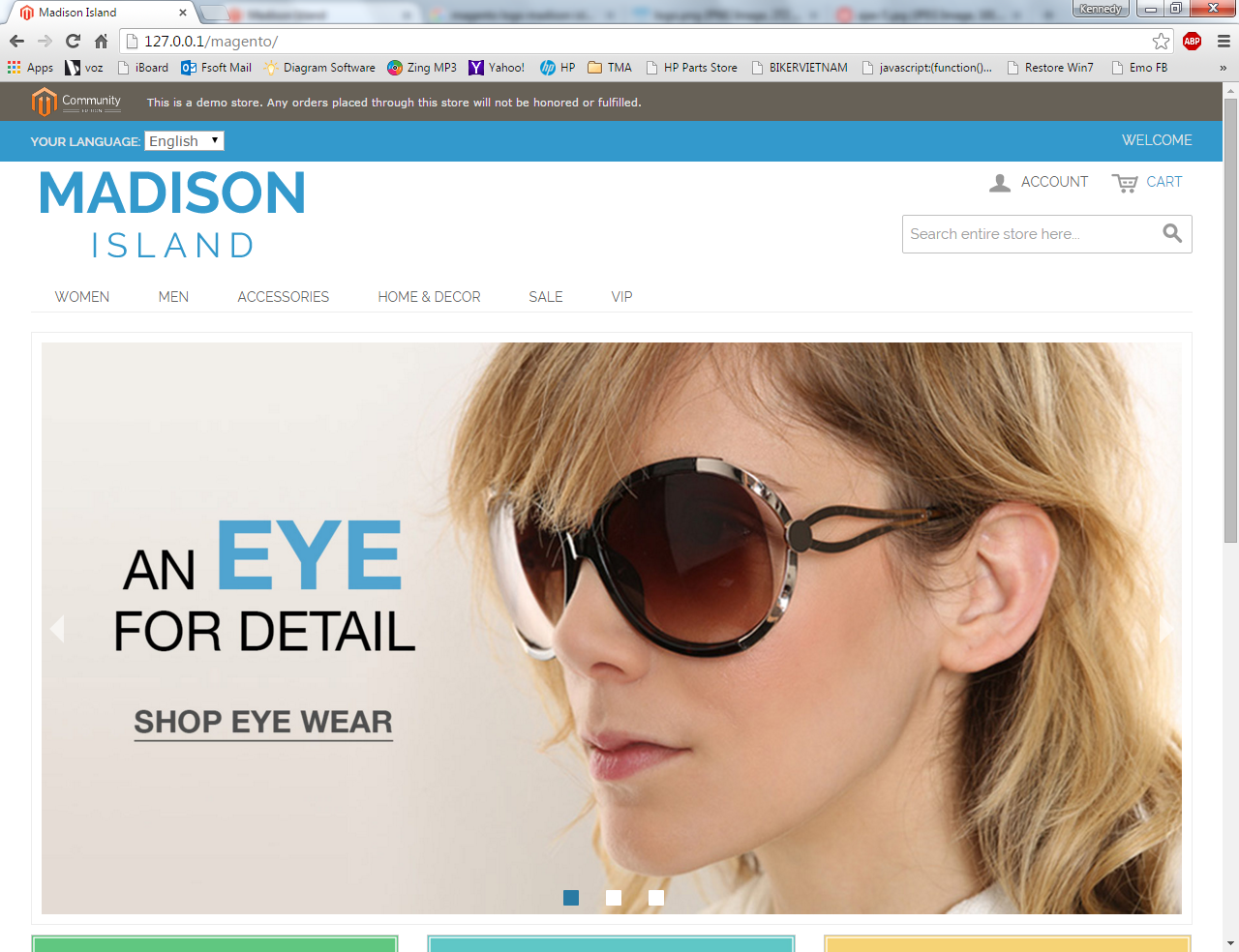
Click “Select All” sau đó chọn action là “Reindex data”. Click “Submit” để bắt đầu quá trình reindex dữ liệu.

#### Refresh cache.

Magento sử dụng cache để tăng tốc độ và giảm số câu query truy xuất cơ sở dữ liệu nên ta cần refresh nó mỗi khi có thay đổi 1 phần nào đó trong cache.

Sau khi đăng nhập vào admin, click vào menu ***System-> Cache Management***để tới trang quản lý cache trong Magento.

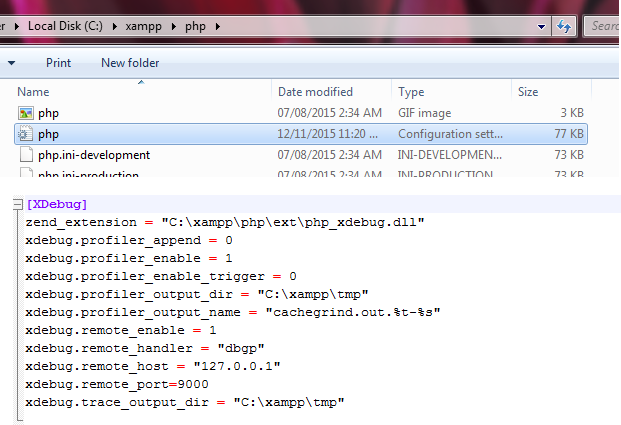
Click “Select all” sau đó chọn action là “Refresh”. Click “Submit” để bắt đầu quá trình refresh tất cả các loại cache trong Magento.

Như vậy các bạn đã hoàn thành phần cài đặt Magento trên localhost, để kiểm tra chúng ta vào trình duyệt và gõ đường dẫn [*http://localhost/magento*](http://localhost/magento)*,* nếu ra được màn hình như bên dưới tức là đã thành công.

## Import project Magento vào IDE

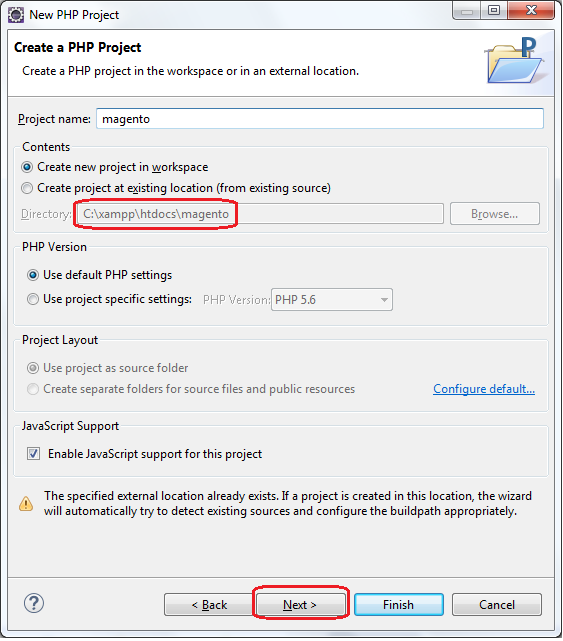
Để có thể cấu hình, phát triển các module và chức năng trên trang web, chúng ta cần import source code của Magento vào IDE để tiện cho việc đó. Mình sẽ sử dụng IDE là Eclipse, cần cài thêm plugin PHP IDT để hỗ trợ cho việc code PHP, ai chưa có thì có thể tải ở đây *http://projects.eclipse.org/projects/tools.pdt*

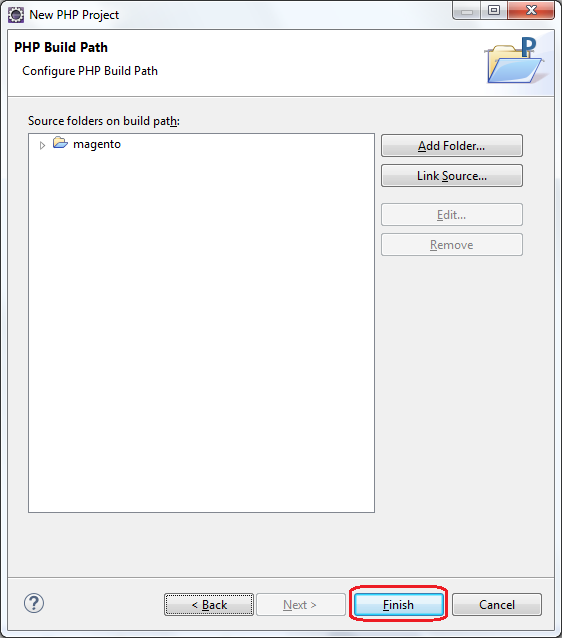
### Cài đặt debugger cho Eclipse.

Mở file *php.ini* trong XAMPP lên và chỉnh sửa [XDebug] như trong hình

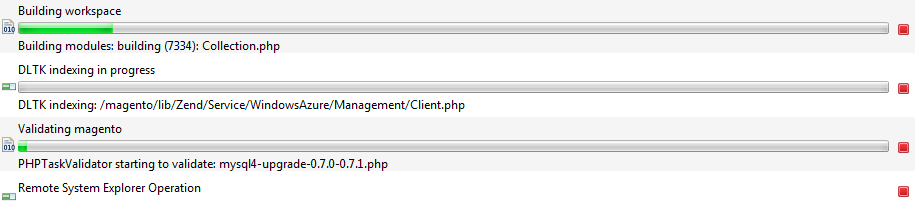
### Tạo project mới và import source code Magento.

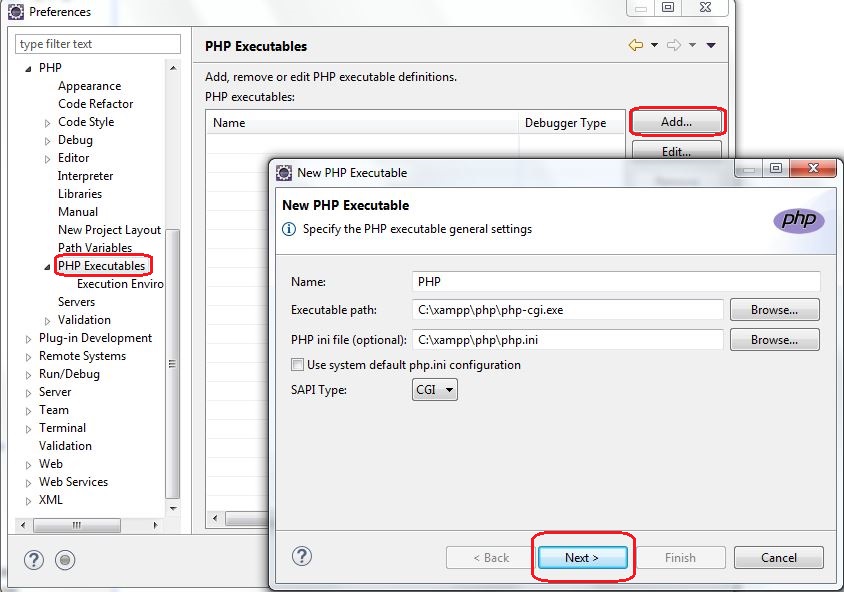
Tạo 1 PHP Project trong Eclipse, chú ý là vị trí của project phải nằm trong thư mục *htdocs* của XAMPP. Vì mình để workspace của mình nằm trong thư mục đó sẵn rồi nên không phải chỉnh gì thêm, bạn nào không thiết lập như mình thì nhớ lưu ý vị trí project. Đặt tên đúng với thư mục chứa source code của Magento mà bạn đã tải về.

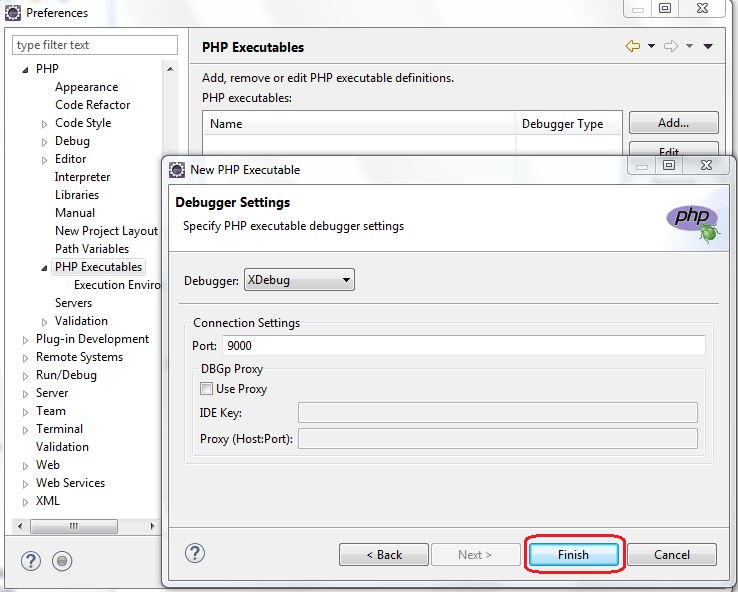


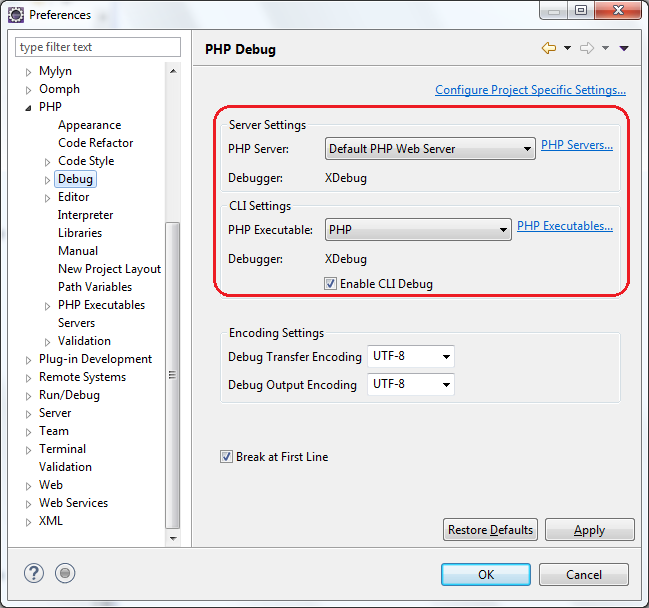


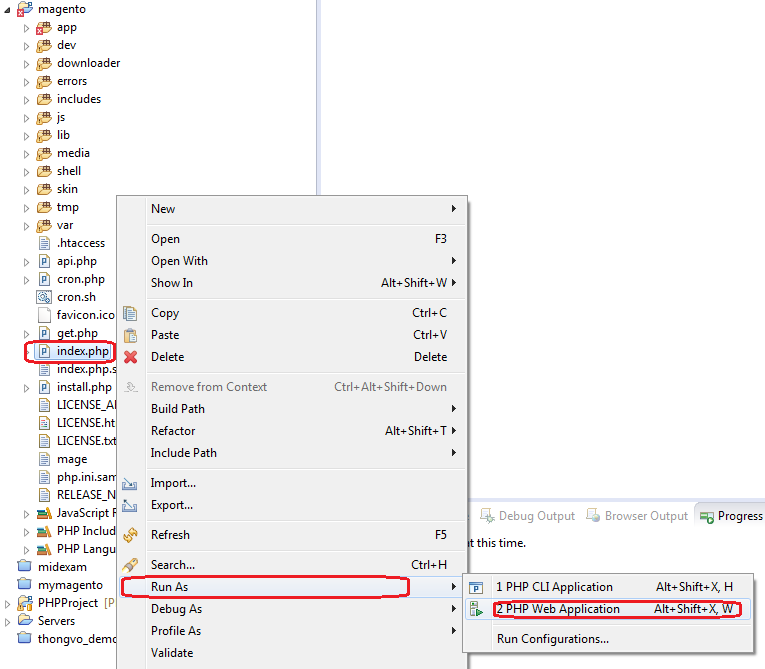
Lúc này sau khi tạo xong project, Eclipse sẽ chạy khá nhiều task để giúp hỗ trợ chúng ta gợi ý code khi lập trình, nên đợi cho các task này chạy xong rồi tiếp tục.



Sau khi import xong chúng ta sẽ cấu hình PHP Excutable để dùng với PHP Engine của XAMPP. Chúng ta sẽ vào Preferences của Eclipse để chỉnh.



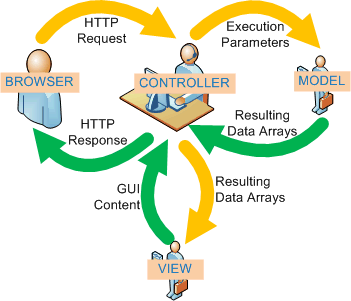


Vậy là bây giờ các bạn đã có thể chạy được project bằng cách click phải vào *index.php* và chọn *Run As PHP Web Application.*

## Kiến trúc trong Magento.

Như đã nói ở phần 1 của bài, Magento được xây dựng dựa trên nền tảng của Zend Framework với kiến trúc MVC, đây là 1 framework rất mạnh và được sử dụng khá phổ biến bởi tính dễ quản lý và mở rộng.

Mô hình MVC (Model-View-Controller) giúp lập trình viên tách riêng từng lớp xử lý của ứng dụng, mỗi lớp đảm nhiệm từng vị trí khác nhau, từ đó có thể dễ dàng xây dựng, bảo trì ứng dụng một cách dễ dàng nhất.



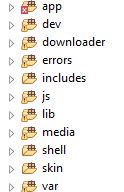
+ *Model:* Là các lớp cung cấp dữ liệu, dịch vụ liên quan đến dữ liệu và business logic. Các lớp này làm việc trực tiếp với dữ liệu và cung cấp dữ liệu cho các thành phần khác. Trong Magento, các lớp này được chứa trong thư mục Model.

+ *Controller:* Là các lớp điều khiển luồng ứng dụng, nó tiếp nhận đầu vào là các yêu cầu của người dùng thông qua HTTP request, chuyển tiếp các yêu cầu đó đến các lớp phụ trách trực tiếp xử lý yêu cầu. Từ đường link, router sẽ tìm đến Controller điều khiển đường link đó. Trong Magento, các lớp này được đặt trong thư mục controllers.

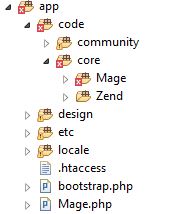
+ *View:* Là các lớp định nghĩa cách thức trình bày dữ liệu (không cập nhật dữ liệu). Các lớp này được chứa trong thư mục Block, template của module.

### Cấu trúc thư mục của Magento.

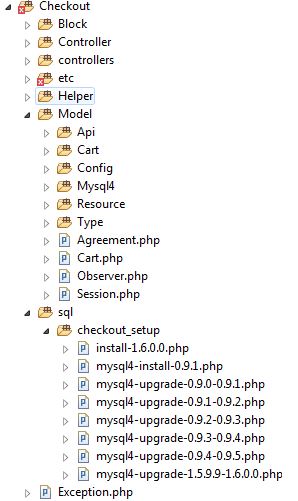
Để phát triển được ứng dụng trên Magento thì chúng ta phải hiểu rõ từng thự mục của nó.



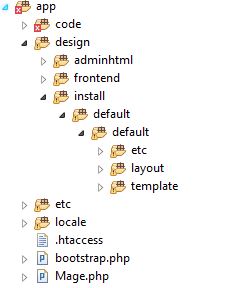
* *app* - Chứa tất cả code(modules), template, file cấu hình, ngôn ngữ. Đồng thời cũng chứa các theme, cài đặt mặc định của hệ thống.
* *downloader* - Dùng để cài đặt và nâng cấp magento mà không cần sử dụng SSH
* *includes* – Chứa file config.php
* *js* - Chứa các mã JavaScript.
* *lib* - Chứa các thư viện của các nhà phát triển.
* *media* - Chứa các file tải lên hệ thống như ảnh sản phẩm, tài liệu...
* *skin* - Chứa các file css, image, javascrip dùng để thiết kê các theme, packages, templates.
* *var* - Nơi chứa các tệp. file của bộ nhớ cache, session, dữ liệu import, export.

**

* */ app* – Nơi chứa ứng dụng chính của chúng ta.
* */ app / etc* – Chứa những config global cho hệ thống.
* */ app / code* - Chứa những modules ( models, controllers, helpers, blocks… ).
* */ app / code / core* - Những module do team Magento phát triển.
* */ app / code / community* - Những module do cộng đồng phát triển.
* */ app / code / core / Mage* – Default namespace của Magento.
* */ app / locale* – File dùng cho việc phát triển ứng dụng đa ngôn ngữ.
* */ app / locale /* {locale (en\_US)} - Zend\_Translate file dùng cho việc chuyển đổi ngôn ngữ.
* */ app / design* – Chứa những package cho việc design (layouts, templates, translations).



* */ app / code / core / Mage /* {Module} – Thư mục root của Module
* */ app / code / core / Mage /* {Module} / etc – Chứa các file cấu hình cho module.
* */ app / code / core / Mage /* {Module} / controllers – Chứa các Controller cho module.
* */ app / code / core / Mage /* {Module} / Block – Chứa các hàm business logic.
* / app / code / core / Mage */* {Module} / Model – Chứa các model thao tác với DB.
* / app / code / core / Mage */* {Module} / Helper – Chứa các helper class.
* */ app / code / core / Mage /* {Module} */ Model / Mysql4* – Chứa các resource models cung cấp bởi module.
* */ app / code / core / Mage /* {Module} */ sql –* Chứa các file cài đặt và nâng cấp sql khi có thay đổi version cho module.
* */ app / code / core / Mage /* {Module} */ sql / {resource} /* - Chứa các file nâng cấp resource model cụ thể.
* */ app / code / core / Mage /* {Module} */ sql /* {resource} */* {type}-{action}-{versions}.(sql|php) – Các file update resource model.



* */ app / design / frontend* – Chứa các thiết kế cho bên frontend
* */ app / design / adminhtml* – Chứa HTML cho trang quản lý của admin.
* */ app / design /* {area} */* {package} */* {theme} – Custom file cho theme.
* */ app / design /* {area} */* {package} */* {theme} */ layout* – Các file xml định nghĩa các block cho mỗi trường hợp khác nhau trong flow của website.
* */ app / design /* {area} */* {package} */* {theme} */ template* – Chứa các file templates phtml.

### Cấu trúc module trong Magento.

Magento tổ chức Code của nó vào các module riêng biệt. Trong một mô hình PHP [**Model-View-Controller (MVC)**](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) điển hình, tất cả Controller được đặt trong một folder, và tất cả Models trong một thư mục khác, v.v. Trong Magento, những file được nhóm lại dựa vào chức năng của chúng và được đặt trong cùng folder với nhau, cả Controller và Model, được gọi là Module.



Một module trong magento bao gồm có các phần chính sau:

+ *Block*: Là nơi dùng để load dữ liệu, điều chỉnh dữ liệu từ database trước khi cho hiển thị dữ liệu ra template.

+ *controllers*: Nhận yêu cầu từ phía người dùng từ http sau đó sẽ chuyển yêu cầu tới các lớp xử lý.

+ *etc*: Bao gồm các file xml dùng để config cho module. Tùy theo mỗi module mà có những file xml khác nhau:

- *etc/config.xml*: Dùng để khai báo model, helper, block…

- *etc/system.xml*: Config tạo ra 1 số field, hiển thị trên menu bên trái khi click vào system\config.

- *etc/adminhtml.xml*: Dùng để config hiển thị trên menu chính của phần quản trị.

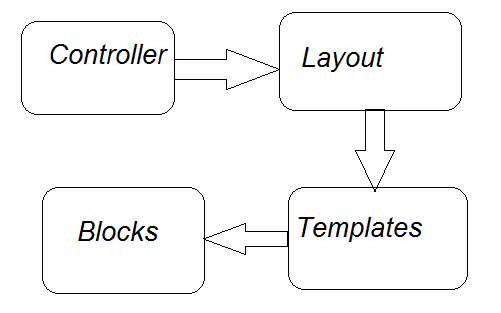
+ *Helper*: Trong này dùng để viết các function được sử dụng ở nhiều nơi khác nhau trong hệ thống. Cách gọi 1 helper: Mage::helper(‘tenmodule/tenhelper’)->helperTenfunction();

+ *Model*: Dùng để viết các câu lệnh truy vấn trực tiếp với cơ sở dữ liệu.

+ *sql*: Dùng để tạo bảng, cập nhật bảng dữ liệu, tương tác thay đổi dữ liệu…

### Blocks, Templates và Layouts trong Magento.

Trong Magento thì tầng View bao gồm có Block và Template. Block là những đối tượng PHP còn Template là những file kết hợp cả mã html và php (có đuôi là phtml). Bên trong file template có sử dụng $this để reference đến đối tượng block của nó.



#### Blocks.

Là nơi dùng để load dữ liệu, điều chỉnh dữ liệu từ database trước khi cho hiển thị dữ liệu ra template.

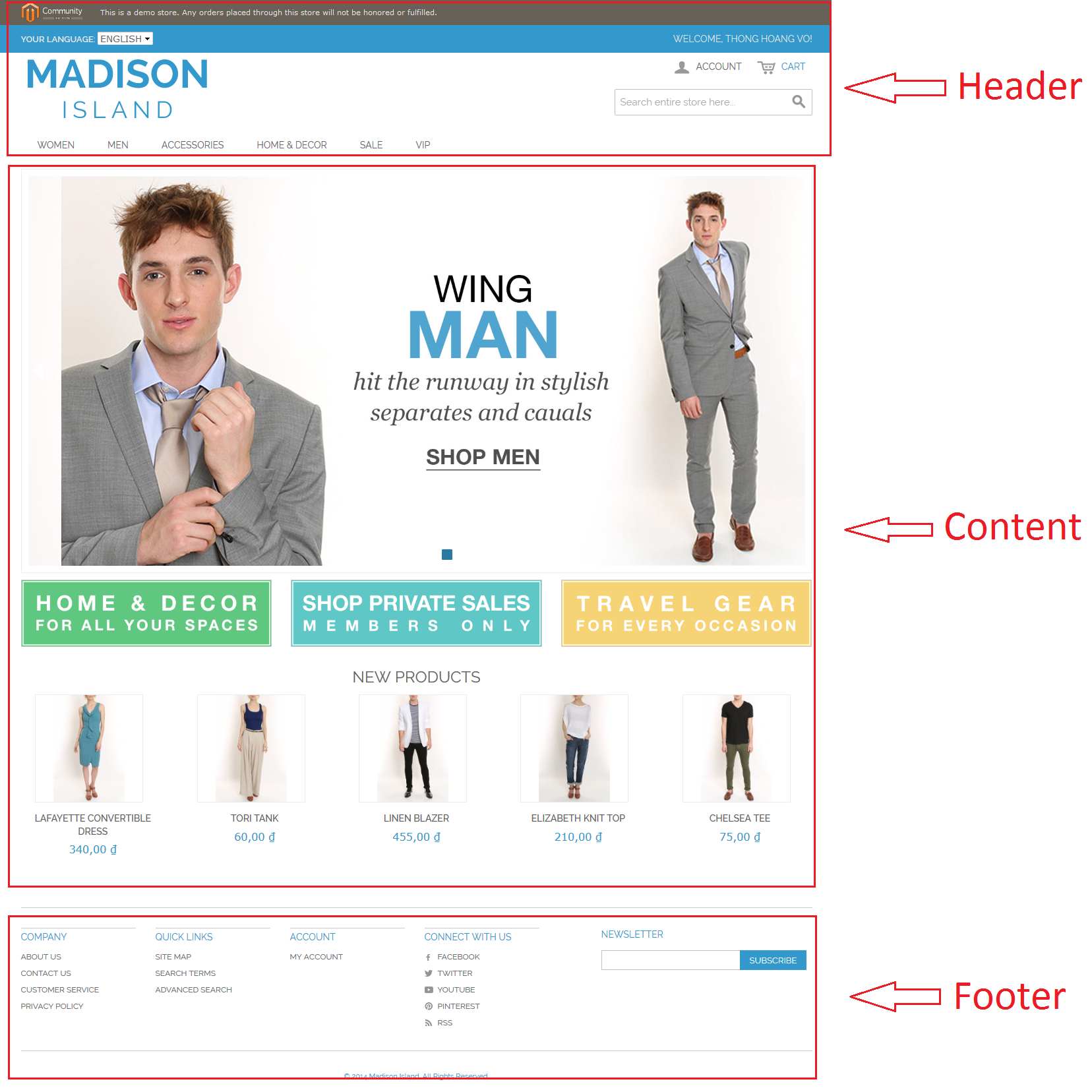
*Đặc điểm của Block:*

* Block chỉ được tải bởi layout hoặc controller.
* Block có thể và nên được cached.
* Block chỉ nên quan tâm đến phần hiển thị.

Có 2 loại block trong Magento:

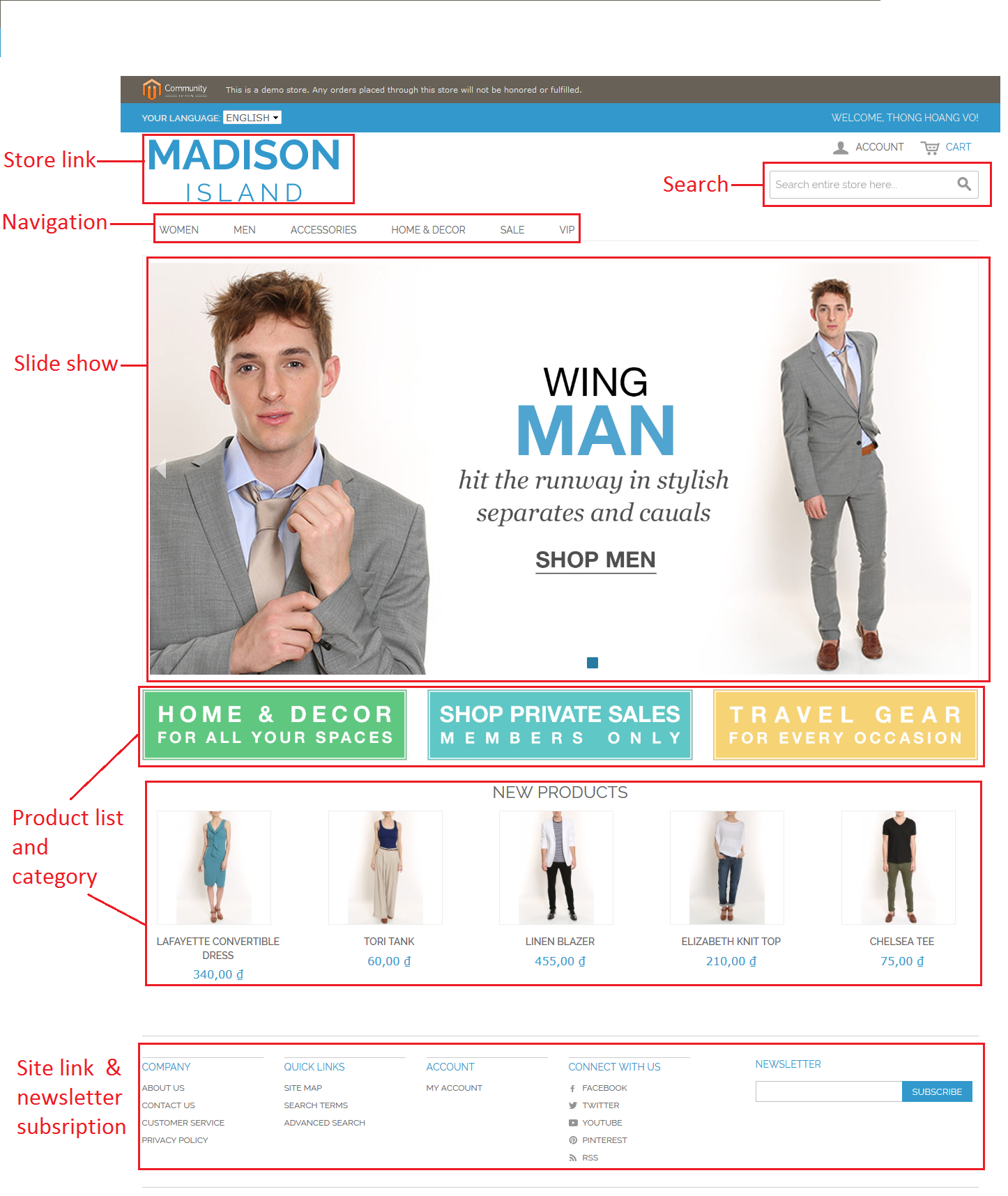
1. *Structural block*

* Là những block được tạo ra để định dạng vị trí cho những block khác trong một page. Ví dụ như trang chủ Magento đang sử dụng các structural block sau:
* Header
* Content
* Footer



1. *Content block:*

* Đây là những block nội dung, chúng được hiển thị bên trong những structural block. Mỗi content block thường hiển thị một khối nội dung nào đó thông qua những file template và chèn vào những block cha là structural block. Ví dụ trong các structural block phía trên của Magento sẽ chứa các content block như sau



Khi nhận được 1 request từ người dùng để hiển thị 1 page:

– Magento sẽ load những vùng cấu trúc.

– Mỗi vùng cấu trúc có những content block. Magento sẽ tập hợp những block này lại và gán cho các structure block tương ứng theo layout để xử lý đầu ra.

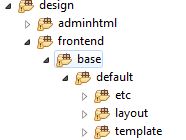
– Cuối cùng hệ thống sẽ trả lại kết quả sau cùng cho trình duyệt phần nội dung đã được định dạng vị trí.

#### Templates.

Templates là những file html được nhúng code php vào nên có phần đuôi là .phtml, được đặt trong thư mục design. Template có tác dụng lấy dữ liệu từ block và hiển thị ra cho người dùng.

Thư mục chứa các file template của magento, bao gồm:

* Layout: chứa các file .xml định ra các vị trí, block trên trang web.
* Template: bao gồm các file phtml chứa thông tin hiển thị ra web.



#### Layout.

Layout là cách bố trí các nội dung trên các trang magento được xác định theo các tập tin xml, và được chứa trong thư mục *app/design/frontend/base/default/layout*. Trong những tập tin XML, chúng ta sẽ thấy một số đoạn XML được viết theo cách parent, chidren. Các đoạn xml được sử dụng để xác định vị trí hiển thị trên các trang.

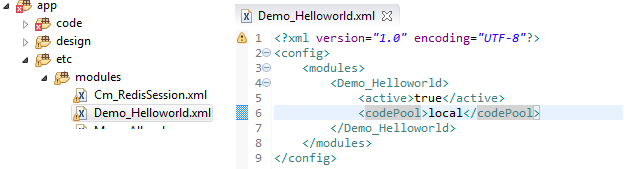
## Tạo một module mới trong Magento

Chúng ta đã biết Magento được tổ chức theo từng module, nên bây giờ mình sẽ hướng dẫn các bạn tạo 1 module mới có tên là Helloworld với chức năng là hiện dòng chữ “Hello World” ra màn hình.

### Khai báo 1 module mới cho Magento

Mỗi module cần 1 tập tin xml cấu hình chúng để Magento biết cách truy cập đến module đó.

Trước khi bắt đầu tạo 1 module mới, chúng ta cần lưu ý rằng các code của chúng ta sẽ được đặt trong thư mục *app/code/local* để tránh ảnh hưởng tới các code có sẵn của Magento.

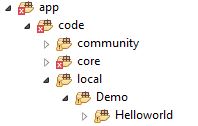
Tạo 1 tập tin xml với tên là Demo\_Helloworld.xml trong thư mục *app/etc/modules* với nội dung như sau.

Thẻ <active> tức là module này đang active hay không, thẻ <codePool> cho biết module này nằm ở thư mục nào. Lưu ý tên module sẽ tương ứng với tên đường dẫn tới module đó.

### Tạo các thư mục cần thiết cho 1 module

Như đã giới thiệu qua ở phía trên, chúng ta cần tạo các thư mục cần thiết của 1 module để chứa các code của nó.

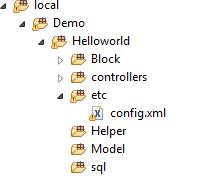
Tạo đường dẫn *app/code/local/<Namespace>/<Modulename>* với namespace có thể là tên công ty hoặc tên tổ chức của bạn, Modulename chính là tên module Helloworld của chúng ta. Ở đây mình sẽ tạo 1 namespace là Demo.



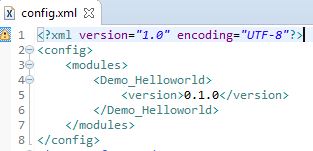
Tiếp theo tạo các thư mục cần thiết cho 1 module:

* Block.
* Controllers.
* Model.
* etc.
* Helpers.
* sql.

Tạo 1 file cấu hình của riêng module trong thư mục */etc*, đặt tên là *config.xml.*



Thiết lập cấu hình cho module của chúng ta.



Kiểm tra xem Magento đã nhận diện module mới của chúng ta chưa bằng cách vào trang quản lý của admin, chọn *System->Configuration->Advanced*, ở mục *Disable modules output* tìm tên module của mình mới tạo, nếu đang active tức là Magento đã nhận module mới. Nếu chưa thấy thì clear cache của Magento để refresh lại.

### Cài đặt Router cho module.

Tiếp theo chúng ta sẽ thiết lập router cho module này, để Magento biết là module này sẽ được kích hoạt khi có 1 request dưới dạng [*http://localhost/magento/helloworld*](http://localhost/magento/helloworld)*.*

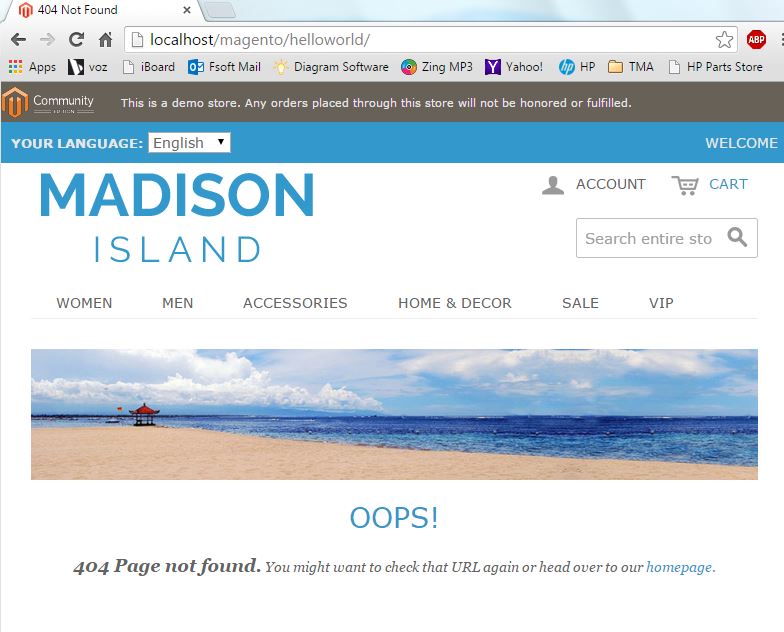
Thêm dòng cấu hình router vào file config của module.



* Router name: Tên của router, không được trùng với các module khác.
* FrontName: Tên của URL để kích hoạt router, nên trùng với tên của router.

Như vậy với mỗi request URL chứa frontName là helloworld sẽ được module Helloworld xử lý.

Thử chạy trang helloworld bằng đường dẫn trên ta sẽ được lỗi 404 vì chưa tạo Controller.



### Tạo Controller cho module.

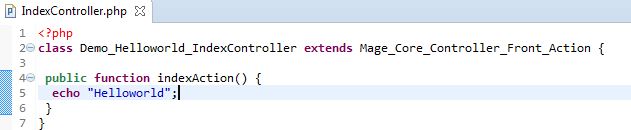
Controller có nhiệm vụ điều phối và xử lý tác vụ tương ứng với từng request đến từ user. Chúng ta tiến hành tạo một *IndexController* cho module Helloworld.



Thử refresh lại trang, ta sẽ nhận được lỗi *Controller file was loaded but class does not exist*.



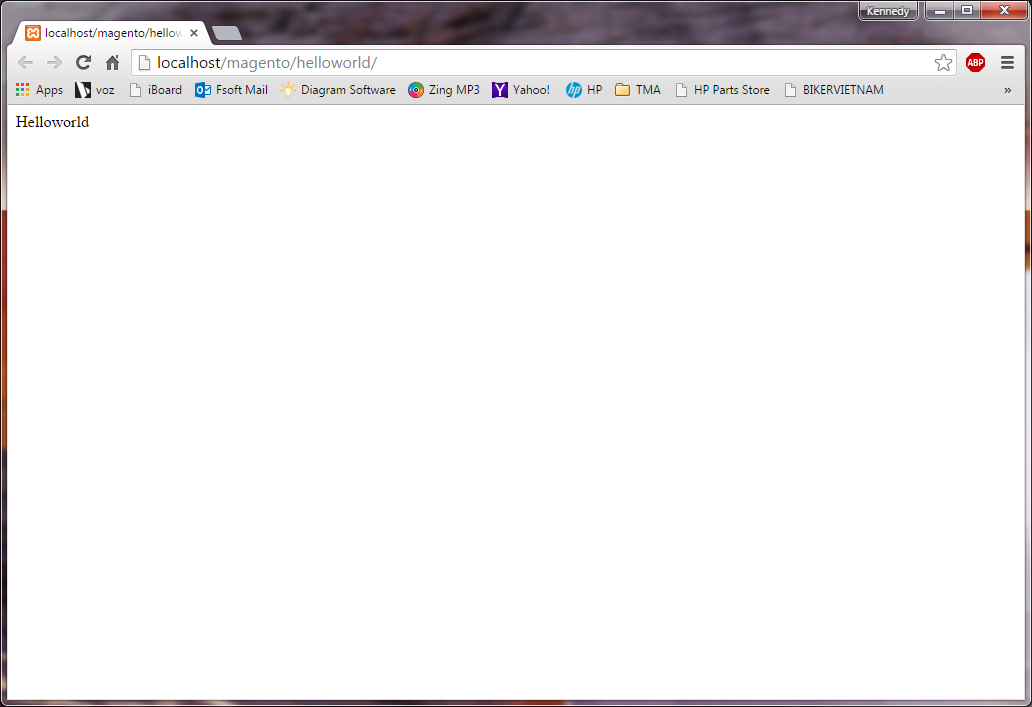
Bây giờ chúng ta tiến hành tạo class cho Controller, tên class cần đúng với tên ta đã khai báo trong file config (*Demo\_Helloworld\_[controllerName]Controller)* và thêm 1 action cho nó.



Cách thức Magento gọi module sẽ giống với Zend Framework, đường dẫn sẽ có dạng [*http://yoursite.com/magento/index.php/frontName/controllerName/actionName*](http://yoursite.com/magento/index.php/frontName/controllerName/actionName):

* Yoursite: Tên miền website của bạn, ở đây mình chạy local nên nó sẽ là localhost.
* frontName: Chính là tên router mà bạn đã khai báo ở file config.xml của module.
* controllerName: Tên Controller khai báo với router tương ứng.
* actionName: Tên action được định nghĩa trong controller.

Bây giờ refresh lại trang ta sẽ được kết quả như mong đợi.



### Thiết lập Layout, Block và Template cho module.

Lúc này ta đã có module và action để gọi tuy nhiên nó hiển thị chưa được đẹp cho lắm, nên bây giờ chúng ta sẽ tạo layout, block và template cho module mới để trông đẹp mắt hơn và tăng khả năng tùy chỉnh về phía front-end.

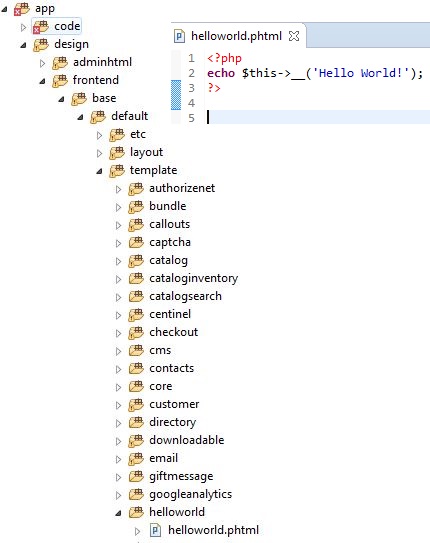
#### Cấu hình config.xml cho layout và block.

Thêm các dòng cấu hình vào file config của module để cấu hình cho layout và block.



#### Tạo file template.

Tiến hành tạo file template cho module, đặt tên là *helloworld.phtml*, nằm trong thư mục *app/design/frontent/base/default/template/helloworld*, template này sẽ in ra màn hình dòng chữ “Hello World”.



#### Tạo file layout.

Ta tạo layout để map giữa block và template, đặt tên là *helloworld.xml* và nằm trong thư mục app/design/frontend/base/default/layout.

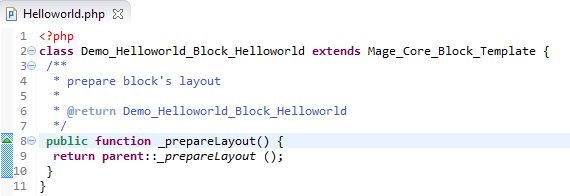
* Router name phải đúng với bên file config.
* Type của block: blockname/blockclass với block name là *helloworld* đã định nghĩa bên file config, block class sẽ được định nghĩa trong 1 file php nằm ở đường dẫn */Block/Helloworld.php* của module. Chú ý là kí tự đầu tiên của block class sẽ được viết hoa lên để mapping với đường dẫn tới block class. Như vậy *helloworld/helloworld* sẽ được map thành block name là *helloworld* và block class là *Helloword.*
* Chú ý format của tag, đã được giải thích trong hình.

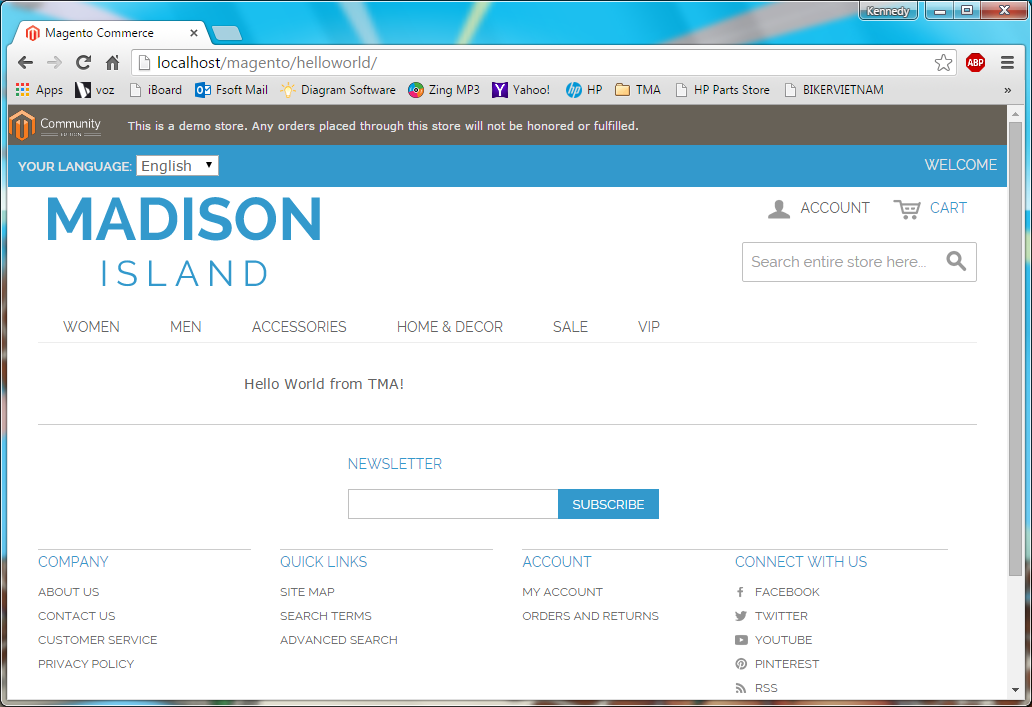
#### Tạo file block.

Sau khi đã thiết lập đầy đủ cấu hình cho template và block thì chúng ta tiến hành tạo file block để điều chỉnh giao diện của trang web trước khi đẩy ra cho người dùng.

Như đã đề cập ở trên thì file block của chúng ta sẽ có tên là *Helloworld.php* và nằm ở đường dẫn */Block*, tên của class bắt buộc phải là *<Package>\_<Module name>\_<block class>.*





Tất cả đã xong, chúng ta refresh lại trang để tận hưởng thành quả.

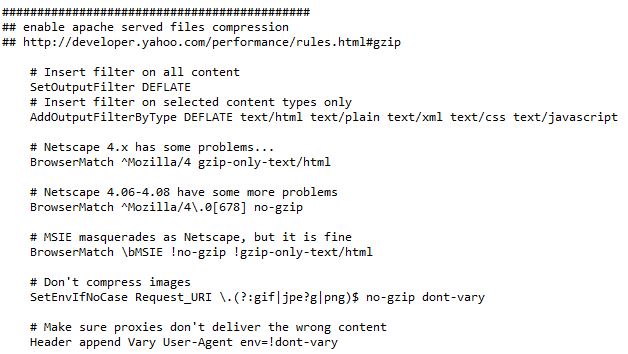
## Một số tinh chỉnh giúp Magento hoạt động nhanh hơn ở phía front-end

Trong thiết lập mặc định của Magento thì tốc độ khá là chậm, chúng ta nên thiết lập lại 1 vài thứ để tối ưu trang web của chúng ta.

### Điều chỉnh file .htaccess

#### Kích hoạt nén dữ liệu output

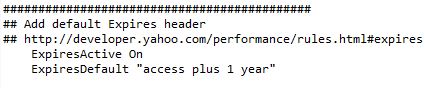
Chức năng này nén dữ liệu đầu ra như text, css ,javascript trước khi gửi tới browser. Điều này giúp cho lượng dữ liệu tải về browser nhỏ hơn từ đó giúp trang load nhanh hơn.

Mở file *.htaccess* lên và uncomment các đoạn tương ứng sau.

#### Kích hoạt Expires Headers

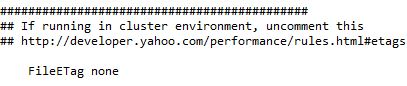
Expires Headers được browser dùng để cache lại các component trên page, giúp tốc độ tải trang nhanh hơn. Bằng cách sử dụng expires header thì chúng ta có thể tăng số lượng component được browser cache lại, từ đó tăng khả năng tái sử dụng trên page thay vì tải lại.

Tìm và chỉnh sửa file *.htaccess* như hình.



#### Tắt Etags.

Etag là cách mà browser xác thực các component qua những lần truy cập vào trang web. Nó có thể làm chậm hệ thống nếu như triển khai không đúng nên tốt nhất là tắt nó đi bằng cách chỉnh file như sau.



### Điều chỉnh thiết lập trong Magento Admin

#### Gộp CSS và JavaScript

Điều chỉnh này giúp cho các file CSS và JS được gộp lại từ đó giúp giảm thiểu dung lượng của request, giúp trang load nhanh hơn.

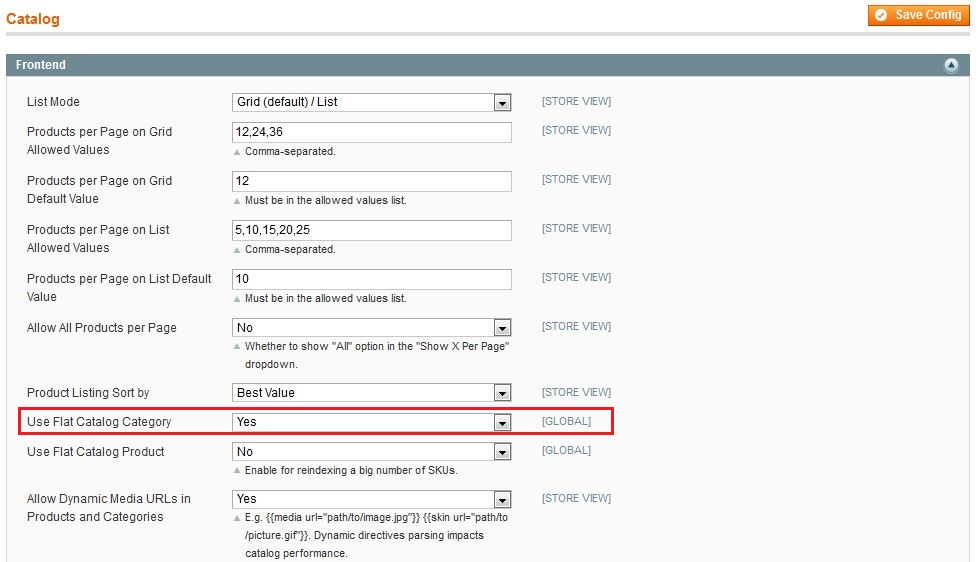
Thiết lập trong Magento Admin theo các bước sau:

1. Magento admin -> System Configuration -> Developer -> ở mục “Javascript Settings”, đổi trường “Merge Javascript Files” thành YES.
2. Ở mục “CSS Settings”, đổi trường “Merge CSS Files” thành YES
3. Xóa cache.



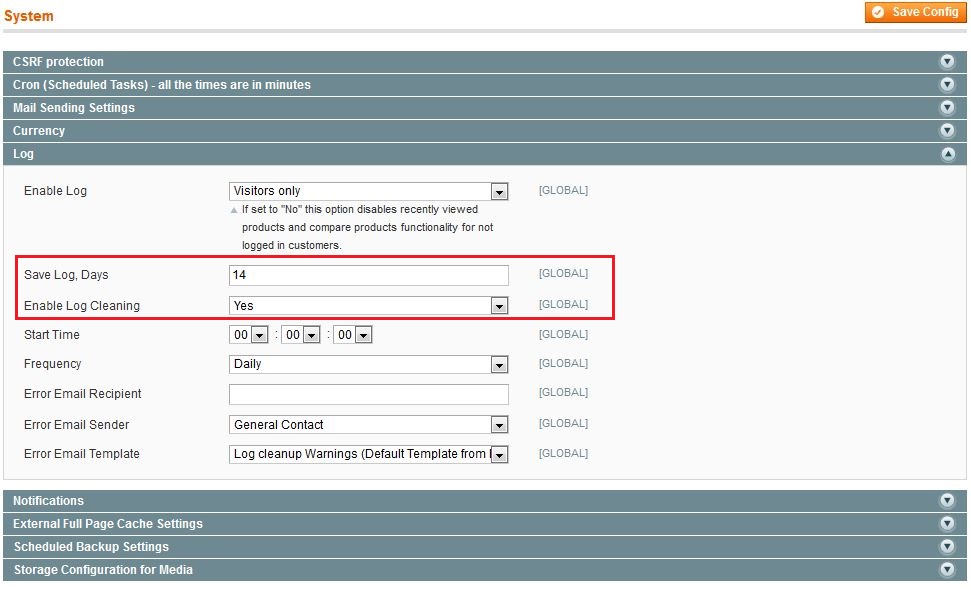
#### Kích hoạt Flat Catalog

Magento dùng cơ chế dữ liệu dạng EAV nên 1 record có thể được lưu ở rất nhiều bảng. Kích hoạt Flat Catalog giúp gộp tất cả lại thành 1 bảng từ đó truy xuất vào database nhanh hơn. Thiết đặt như sau:

1. Vào System -> Configuration -> Catalog.
2. Ở mục “Frontend”, đổi mục “Use Flat Catalog Category” thành YES.
3. Xóa cache.

#### Xóa file log.

1. System > Configuration.
2. Dưới mục Advanced, chọn System.
3. Ở mục “Log Cleaning”, đổi “Enable Log Cleaning” thành “yes” và cấu hình Save Log thành 14 ngày:



### Làm mới lại database

Login vào database của Magento và thực hiện xóa hết tất cả các dữ liệu (truncate) có trong các bảng sau:

* log\_customer
* log\_quote
* log\_summary\_type
* log\_url
* log\_url\_info
* log\_visitor
* log\_visitor\_info
* log\_visitor\_online

