

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----oOo-----



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2012

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----

XÂY DỰNG TRANG WEB BÁN HÀNG ĐIỆN TỬ BẰNG OPENCART

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công Nghệ Thông Tin

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----

XÂY DỰNG TRANG WEB BÁN HÀNG ĐIỆN TỬ BẰNG OPENCART

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công Nghệ Thông Tin

Sinh viên thực hiện: Nguyễn Bá Tú.

Giáo viên hướng dẫn: Thạc sĩ Nguyễn Trịnh Đông.

Mã số sinh viên: 121239

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Nguyễn Bá Tú.

Mã số sinh viên: 121239.

Lớp: CT1201.

Ngành: Công nghệ thông tin.

Tên đề tài: **Xây dựng trang Web bán hàng điện tử bằng Opencart**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

b. Các yêu cầu cần giải quyết

2. Các số liệu cần thiết để thiết kế, tính toán.

3. Địa điểm thực tập

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....

.....

.....

.....

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....

.....

Đề tài tốt nghiệp được giao ngày tháng năm 2012

Yêu cầu phải hoàn thành trước ngày tháng năm 2012

Đã nhận nhiệm vụ: Đ.T.T.N

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Cán bộ hướng dẫn Đ.T.T.N

Hải phòng, ngày tháng năm 2012

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHÂN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....
.....
.....
.....

3. Cho điểm của cán bộ hướng dẫn:
(Điểm ghi bằng số và chữ)

.....
.....
.....

Ngày tháng năm 2012
Cán bộ hướng dẫn chính
(Ký, ghi rõ họ tên)

**PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI
TỐT NGHIỆP**

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế,...)

2. Cho điểm của cán bộ phản biện:

(Điểm ghi bằng số và chữ)

.....
.....
.....

Ngày tháng năm 2012

Cán bộ chấm phản biện

(Ký, ghi rõ họ tên)

MỤC LỤC

MỤC LỤC.....	1
DANH SÁCH CÁC HÌNH	4
BẢNG DANH SÁCH VIẾT TẮT	6
LỜI CẢM ƠN	9
MỞ ĐẦU.....	10
CHƯƠNG 1: CÔNG NGHỆ WEB	11
1.1. Giới thiệu Web.....	11
1.1.1. Phân loại.....	11
1.1.2. Các mô hình mạng.....	12
1.1.3. Mô hình vật lý	12
1.2. Các giao thức.....	17
1.2.1. OSI.....	17
1.2.2. TCP/IP	17
1.2.3. Giao thức HTTP	18
1.3. HTML	19
1.4. CSS.....	20
1.5. Javascript.....	23
1.6. Apache và IIS	25
1.7. Tổng quan về MySQL.....	25
1.7.1. Khởi động và sử dụng	25
1.7.2. Một số thuật ngữ	26
1.7.3. Loại dữ liệu trong MySQL.....	26
1.7.4. Những cú pháp cơ bản.....	26
1.8. Tổng quan về PHP	28
1.8.1. Lịch sử phát triển.....	28

1.8.2. Cấu trúc cơ bản.....	29
1.8.3. Xuất giá trị ra trình duyệt	30
1.8.4. Biến, hằng, chuỗi và các kiểu dữ liệu	30
1.8.5. Các phương thức được sử dụng trong lập trình PHP	32
1.8.6. Cookie và Session trong PHP	32
1.8.7. Hàm	34
1.9. Kết hợp PHP và MySQL trong ứng dụng website.....	35
CHƯƠNG 2: OPENCART	36
2.1. Kiến trúc của Opencart	36
2.1.1. Tầng mô hình (<i>Business process layer</i>)	39
2.1.2. Tầng biểu diễn (<i>Presentation layer</i>)	39
2.1.3. Tầng điều khiển (<i>Control layer</i>).....	39
2.2. Cấu trúc thư mục tệp tin Opencart	40
2.3. Cài đặt và việt hóa Opencart	43
2.3.1. Cài đặt XAMPP.....	43
2.3.2. Cài đặt Opencart.....	44
2.3.3. Việt hóa Opencart.....	47
CHƯƠNG 3: ỨNG DỤNG OPENCART.....	49
Bài toán	49
3.1. Lý do chọn Opencart.....	49
3.1.1. Opencart là mã nguồn mở	49
3.1.2. Opencart là một hệ thống giỏ hàng	50
3.1.3. Opencart có các tính năng phong phú	50
3.1.4. Opencart thân thiện với người dùng.....	50
3.2. Sử dụng Opencart xây dựng trang web.....	51
3.2.1 Đăng kí tên miền và Hosting.....	51
3.2.2. Cài đặt lên host.....	52

3.2.3. Cài đặt module(extensions).....	52
3.2.4. Nhập dữ liệu và quản lý đơn đặt hàng.....	54
3.2.5. Thêm giao diện.....	54
3.3. Một số kết quả thử nghiệm	56
KẾT LUẬN:	59
TÀI LIỆU THAM KHẢO:	60

DANH SÁCH CÁC HÌNH

Chương 1:

Hình 1. 1: Mô hình mạng bus	13
Hình 1. 2: Mô hình mạng vòng	13
Hình 1. 3: Mô hình mạng sao.....	14

Chương 2:

Hình 2. 1: Kiến trúc mô hình MVC	36
Hình 2. 2: Mô hình tuần tự của MVC	37
Hình 2. 3: Mô hình Three Layer	38
Hình 2. 4: So sánh MVC với Three Layer	39
Hình 2. 5: Cấu trúc thư mục tệp tin Opencart.....	40
Hình 2. 6: Thư mục admin	41
Hình 2. 7: Thư mục language.....	41
Hình 2. 8: Thư mục catalog	42
Hình 2. 9: Thư mục image	42
Hình 2. 10: Quá trình cài đặt XAMPP đã thành công	43
Hình 2. 11: Tạo database.....	44
Hình 2. 12: Bước 1 cài đặt Opencart.....	45
Hình 2. 13: Bước 2 cài đặt Opencart.....	45
Hình 2. 14: Bước 3 cài đặt Opencart.....	46
Hình 2. 15: Bước 4 cài đặt Opencart.....	47
Hình 2. 16: Giao diện trang admin đã Việt hóa	48
Hình 2. 17: Giao diện trang chủ đã Việt hóa	48

Chương 3:

Hình 3. 1: Cài đặt lên host.....	52
Hình 3. 2: Các module	53
Hình 3. 3: Module khảo sát lượt truy cập	53
Hình 3. 4: Đăng nhập vào host.....	54
Hình 3. 5: Upload giao diện	55
Hình 3. 6: Thiết lập	55
Hình 3. 7: Đổi giao diện.....	56
Hình 3. 8: Giao diện trang chủ giới thiệu sản phẩm	56
Hình 3. 9: Giao diện khách hàng đăng nhập.....	57
Hình 3. 10: Giao diện đặt hàng	57
Hình 3. 11: Hóa đơn.....	58
Hình 3. 12: Giao diện đơn đặt hàng	58

BẢNG DANH SÁCH VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ	Mô tả
1	HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
2	LAN	Local area network	Mạng cục bộ
3	MAN	Metropolitan area network	Mạng đô thị
4	WAN	Wide area network	Mạng diện rộng
5	IEEE	Institute of Electrical and Electronics Engineers	Viện kỹ nghệ Điện và Điện tử
6	IBM	International Business Machines	Tập đoàn công nghệ máy tính đa quốc gia
7	IP	Internet Protocol	Giao thức hướng dữ liệu được sử dụng bởi các máy chủ nguồn và đích để truyền dữ liệu trong một liên mạng chuyển mạch gói.
8	UDP	User Datagram Protocol	Một trong những giao thức cốt lõi của giao thức TCP/IP
9	OSI	Open Systems Interconnection Reference Model	Một thiết kế dựa vào nguyên lý tầng cấp, lý giải một cách trừu tượng kỹ thuật kết nối truyền thông giữa các máy vi tính và thiết kế giao thức mạng giữa chúng
10	TCP	Transmission Control Protocol	Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin

STT	Tên viết tắt	Tên đầy đủ	Mô tả
11	DNS	Domain Name System	Một hệ thống cho phép thiết lập tương ứng giữa địa chỉ IP và tên miền
12	HTTP	HyperText Transfer Protocol	Giao thức liên hệ thông tin giữa máy cung cấp dịch vụ và máy sử dụng dịch vụ
13	HTTPS	Hypertext Transfer Protocol Secure	Kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet.
14	URL	Uniform Resource Locator	Được dùng để tham chiếu tới tài nguyên trên Internet
15	WWW	World Wide Web	Một không gian thông tin toàn cầu mà mọi người có thể truy nhập (đọc và viết) qua các máy tính nối với mạng Internet
16	XHTML	Extensible HyperText Markup Language	Một ngôn ngữ đánh dấu có cùng các khả năng như HTML, nhưng có cú pháp chặt chẽ hơn
17	CSS	Cascading Style Sheets	Các tập tin định kiểu theo tầng
18	IIS	Internet Information Services	Một dịch vụ tùy chọn của Windows NT Server cung cấp các tính năng về Website
19	MVC	Model View Controller	Mô hình-biểu diễn-điều khiển
20	GUI	Graphical User Interface	Giao diện đồ họa người dùng

STT	Tên viết tắt	Tên đầy đủ	Mô tả
21	SEO	Senior Executive Officer	

LỜI CẢM ƠN

Trong lời đầu tiên của báo cáo đồ án tốt nghiệp “Xây dựng trang Web bán hàng điện tử bằng Opencart” này, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Trước hết, em xin chân thành cảm ơn Thầy Giáo - Ths. Nguyễn Trinh Đông, Giảng viên Khoa Công Nghệ Thông Tin, Trường ĐHDL Hải Phòng, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện đồ án.

Xin chân thành cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, ban giám hiệu nhà trường, các thầy cô trong Khoa Công Nghệ Thông Tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em cũng như các bạn khác trong suốt thời gian học tập và làm tốt nghiệp.

Cuối cùng em xin gửi lời cảm ơn đến gia đình, bạn bè, người thân đã giúp đỡ động viên em rất nhiều trong quá trình học tập và làm Đồ án Tốt Nghiệp.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên Đồ án thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy cô giáo và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện đồ án của mình.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2012

Sinh viên

Nguyễn Bá Tú

MỞ ĐẦU

Hiện nay trên thế giới thương mại điện tử đang phát triển rất mạnh mẽ. Kỹ thuật số giúp chúng ta tiết kiệm đáng kể các chi phí như chi phí vận chuyển trung gian, chi phí giao dịch, và đặc biệt là giúp tiết kiệm thời gian để con người đầu tư vào các hoạt động khác. Hơn nữa thương mại điện tử còn giúp con người có thể tìm kiếm tự động theo nhiều mục đích khác nhau, tự động cung cấp thông tin theo nhu cầu và sở thích của con người. Giờ đây, con người có thể ngồi tại nhà để mua sắm mọi thứ theo ý muốn và các website bán hàng trên mạng sẽ giúp ta làm được điều đó. Chính vì vậy các công nghệ mã nguồn mở trở lên được chú ý vì các tính năng của nó. Giá thành rẻ và được hỗ trợ rất nhiều trên mạng sẽ giúp ta nhanh chóng xây dựng các website bán hàng thân thiện và dễ sử dụng với người dùng.

Chính vì vậy trong đề án này em chọn đề tài về: **“Xây dựng một trang bán hàng trên mạng trên nền mã nguồn mở là Opencart”**. Đây là 1 hệ thống đơn giản nhưng đủ mạnh để cho phép nhanh chóng xây dựng các ứng dụng bán hàng trên Internet

CHƯƠNG 1: CÔNG NGHỆ WEB

1.1. Giới thiệu Web

Web là công nghệ *Internet* cho phép thể hiện các thông tin một cách sinh động, gắn gũi với con người hơn trên một trang thông tin gọi là trang *Web*. Trang *Web* được trình bày trên các bộ duyệt *Web (Browser)* trên các máy *Client*. Trang *Web* chính là các tệp văn bản dạng *Text* được cấu trúc hóa theo ngôn ngữ HTML.

Để thiết lập và đưa vào hoạt động một *Website* phải đảm bảo ba yếu tố:

- Tên *website* _ *Domain name*.
- *Webhosting* _ Nơi lưu trữ trên máy chủ *Internet*.
- Các trang *web*.

1.1.1. Phân loại

Web tĩnh

Web tĩnh(HTML,DHTML)thường được dùng để thiết kế các trang *web* có nội dung ít cần thay đổi và cập nhật. *Website* tĩnh là *website* chỉ bao gồm các trang *web* tĩnh và không có cơ sở dữ liệu đi kèm.

Ưu điểm: thiết kế đồ họa đẹp, tốc độ truy cập nhanh, thân thiện khi tìm kiếm, chi phí đầu tư thấp.

Nhược điểm: Khó khăn trong việc thay đổi và cập nhật thông tin, khó tích hợp, nâng cấp và mở rộng.

Web động

Web động là thuật ngữ được dùng để chỉ những *website* có cơ sở dữ liệu và được hỗ trợ bởi các phần mềm phát triển *web*.

Với *web* động, thông tin hiển thị được gọi ra từ một cơ sở dữ liệu khi người dùng truy vấn tới một trang *web*. Trang *web* được gửi tới trình duyệt gồm những câu chữ, hình ảnh, âm thanh hay những dữ liệu số hoặc ở dạng bảng hoặc ở nhiều hình thức khác nữa.

Web động thường được phát triển bằng các ngôn ngữ lập trình tiên tiến như PHP, ASP, ASP.NET, Java, CGI, Perl, và sử dụng các cơ sở dữ liệu quan hệ mạnh như Access, My SQL, MS SQL, Oracle, DB2.

Thông tin trên *web* động luôn luôn mới vì nó dễ dàng được thường xuyên cập nhật thông qua việc sử dụng các công cụ cập nhật của các phần mềm quản trị *web*. Thông tin luôn được cập nhật trong một cơ sở dữ liệu và người dùng *Internet* có thể xem những chỉnh sửa đó ngay lập tức. Vì vậy *website* được hỗ trợ bởi cơ sở dữ liệu là phương tiện trao đổi thông tin nhanh nhất với người dùng *Internet*. Điều dễ nhận thấy là những *website* thường xuyên được cập nhật sẽ thu hút nhiều khách hàng tới thăm hơn những *website* ít có sự thay đổi về thông tin.

Web động có tính tương tác với người sử dụng cao.

1.1.2. Các mô hình mạng

Trong kỹ thuật mạng, việc quan trọng nhất là vận chuyển dữ liệu giữa các máy. Nói chung sẽ có hai phương thức là:

Mạng quảng bá (*broadcast network*)

Mạng quảng bá bao gồm một kênh truyền thông được chia sẻ cho mọi máy trong mạng. Mẫu thông tin ngắn gọi là gói (*packet*) được gửi ra bởi một máy bất kỳ thì sẽ tới được tất cả máy khác. Trong gói sẽ có một phần ghi địa chỉ gói đó muốn gửi tới. Khi nhận các gói, mỗi máy sẽ kiểm tra lại phần địa chỉ này. Nếu một gói là dành cho đúng máy đang kiểm tra thì sẽ được xử lý tiếp, bằng không thì bỏ qua.

Mạng điểm nối điểm (*point-to-point network*)

Mạng điểm nối điểm bao gồm nhiều mối nối giữa các cặp máy tính với nhau. Để chuyển từ nguồn tới đích, một gói có thể phải đi qua các máy trung gian. Thường thì có thể có nhiều đường di chuyển có độ dài khác nhau (từ máy nguồn tới máy đích với số lượng máy trung gian khác nhau). Thuật toán để định tuyến đường truyền giữ vai trò quan trọng trong kỹ thuật này.

1.1.3. Mô hình vật lý

LAN

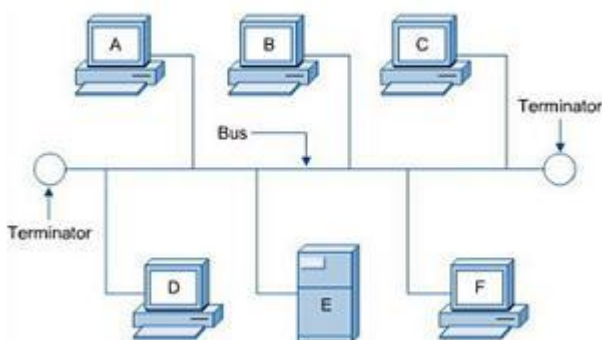
LAN (*local area network*), hay còn gọi là "mạng cục bộ", là mạng tư nhân trong một toà nhà, một khu vực (trường học hay cơ quan) có cỡ chừng vài trăm mét. Chúng nối các máy chủ và các máy trạm trong các văn phòng và nhà máy để chia sẻ tài nguyên và trao đổi thông tin. *LAN* có 3 đặc điểm:

1. Giới hạn về tầm cỡ phạm vi hoạt động từ vài mét cho đến vài trăm mét.

2. Thường dùng kỹ thuật đơn giản chỉ có một đường dây cáp (*cable*) nối tất cả máy. Vận tốc truyền dữ liệu thông thường là 10 Mbps, 100 Mbps.

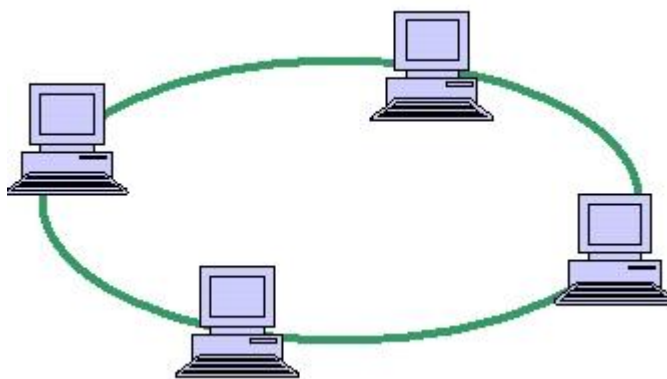
3. Ba kiến trúc mạng kiểu LAN thông dụng bao gồm:

- Mạng bus hay mạng tuyến tính. Các máy nối nhau một cách liên tục thành một hàng từ máy này sang máy kia. Ví dụ của nó là *Ethernet* (chuẩn IEEE802.3).



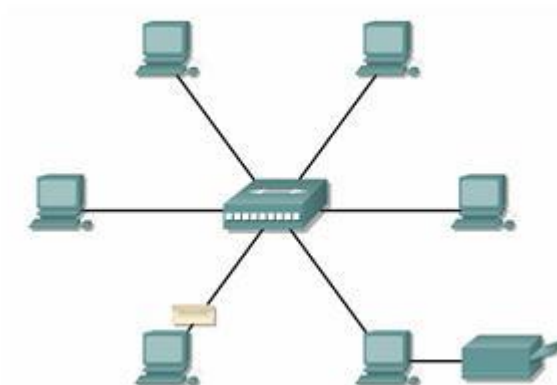
Hình 1. 1: Mô hình mạng bus

- Mạng vòng. Các máy nối nhau như trên và máy cuối lại được nối ngược trở lại với máy đầu tiên tạo thành vòng kín. Thí dụ mạng vòng thẻ bài IBM (*IBM token ring*).



Hình 1. 2: Mô hình mạng vòng

- Mạng sao.



Hình 1. 3: Mô hình mạng sao

MAN

MAN (*metropolitan area network*), hay còn gọi là "mạng đô thị", là mạng có cỡ lớn hơn LAN, phạm vi vài km. Nó có thể bao gồm nhóm các văn phòng gần nhau trong thành phố, nó có thể là công cộng hay tư nhân và có đặc điểm:

1. Chỉ có tối đa hai dây cáp nối.
2. Không dùng các kỹ thuật nối chuyên.
3. Có thể hỗ trợ chung vận chuyển dữ liệu và đàm thoại, hay ngay cả truyền hình. Ngày nay người ta có thể dùng kỹ thuật cáp quang (*fiber optical*) để truyền tín hiệu. Vận tốc có hiện nay thể đạt đến 10 Gbps.

Ví dụ của kỹ thuật này là mạng DQDB (*Distributed Queue Dual Bus*) hay còn gọi là *bus* kép theo hàng phân phối (tiêu chuẩn IEEE 802.6).

WAN

WAN (*wide area network*), còn gọi là "mạng diện rộng", dùng trong vùng địa lý lớn thường cho quốc gia hay cả lục địa, phạm vi vài trăm cho đến vài ngàn km. Chúng bao gồm tập hợp các máy nhằm chạy các chương trình cho người dùng. Các máy này thường gọi là máy lưu trữ (*host*) hay còn có tên là máy chủ, máy đầu cuối (*end system*). Các máy chính được nối nhau bởi các mạng truyền thông con (*communication subnet*) hay gọn hơn là mạng con (*subnet*). Nhiệm vụ của mạng con là chuyển tải các thông điệp (*message*) từ máy chủ này sang máy chủ khác.

Mạng con thường có hai thành phần chính:

1. Các đường dây vận chuyển còn gọi là mạch (*circuit*), kênh (*channel*), hay đường trung chuyển (*trunk*).

2. Các thiết bị nối chuyển. Đây là loại máy tính chuyên biệt hoá dùng để nối hai hay nhiều đường trung chuyển nhằm di chuyển các dữ liệu giữa các máy. Khi dữ liệu đến trong các đường vô, thiết bị nối chuyển này phải chọn (theo thuật toán đã định) một đường dây ra để gửi dữ liệu đó đi. Tên gọi của thiết bị này là nút chuyển gói (*packet switching node*) hay hệ thống trung chuyển (*intermediate system*). Máy tính dùng cho việc nối chuyển gọi là "bộ chọn đường" hay "bộ định tuyến" (*router*).

Hầu hết các WAN bao gồm nhiều đường cáp hay là đường dây điện thoại, mỗi đường dây như vậy nối với một cặp bộ định tuyến. Nếu hai bộ định tuyến không nối chung đường dây thì chúng sẽ liên lạc nhau bằng cách gián tiếp qua nhiều bộ định tuyến trung gian khác. Khi bộ định tuyến nhận được một gói dữ liệu thì nó sẽ chứa gói này cho đến khi đường dây ra cần cho gói đó được trống thì nó sẽ chuyển gói đó đi. Trường hợp này ta gọi là nguyên lý mạng con điểm nối điểm, hay nguyên lý mạng con lưu trữ và chuyển tiếp (*store-and-forward*), hay nguyên lý mạng con nối chuyển gói.

Có nhiều kiểu cấu hình cho WAN dùng nguyên lý điểm tới điểm như là dạng sao, dạng vòng, dạng cây, dạng hoàn chỉnh, dạng giao vòng, hay bất định.

Mạng không dây

Các thiết bị cầm tay hay bỏ túi thường có thể liên lạc với nhau bằng phương pháp không dây và theo kiểu LAN. Một phương án khác được dùng cho điện thoại cầm tay dựa trên giao thức CDPD (Cellular Digital Packet Data) hay là dữ liệu gói kiểu cellular số.

Các thiết bị không dây hoàn toàn có thể nối vào mạng thông thường (có dây) tạo thành mạng hỗn hợp.

Liên mạng

Liên mạng (*Internetwork* hoặc viết gọn thành *Internet*) là hai hay nhiều mạng máy tính nối với nhau bằng các thiết bị định tuyến (*router*) cho phép dữ liệu được gửi qua lại giữa chúng. Các thiết bị định tuyến có nhiệm vụ hướng dẫn giao thông dữ liệu theo đường đúng (trong số một số các đường có thể) đi qua liên mạng để tới đích. Một số người đã nhầm lẫn khi gọi việc liên kết các mạng với nhau bằng các cầu (*bridge*) là liên mạng. Thực ra hệ thống đó chỉ là kết nối của các mạng con và việc gửi dữ liệu qua nó không đòi hỏi các giao thức liên mạng, chẳng hạn giao thức IP.

Ban đầu, liên mạng là một cách để kết nối các kiểu công nghệ mạng khác nhau. Nhưng rồi nó đã trở nên phổ biến rộng rãi qua sự phát triển của nhu cầu kết nối hai hoặc nhiều mạng cục bộ với nhau thành một dạng mạng diện rộng. Hiện nay, định nghĩa của liên mạng bao hàm cả việc kết nối các mạng máy tính thuộc các kiểu khác, chẳng hạn các mạng cá nhân PAN.

Internet chính là ví dụ thực tế nổi tiếng nhất của liên mạng. Đó là một mạng gồm các mạng chạy nhiều giao thức bậc thấp khác nhau, được thống nhất bởi một giao thức liên mạng - giao thức IP.

Giao thức IP chỉ cung cấp một dịch vụ chuyển gói tin không đáng tin cậy qua một liên mạng. Để gửi dữ liệu một cách đáng tin cậy, các ứng dụng phải sử dụng một giao thức tầng giao vận, chẳng hạn giao thức TCP, giao thức này cung cấp một dòng đáng tin cậy (*reliable stream*). Thuật ngữ không đáng tin cậy ở đây không có nghĩa là IP không đáng tin cậy, mà là nó gửi các gói tin mà không liên lạc và thiết lập một kết nối với máy đích từ trước. Dịch vụ đáng tin cậy thì làm ngược lại.

Do TCP là giao thức giao vận được sử dụng rộng rãi nhất, người ta thường gọi TCP và IP liên nhau là "TCP/IP". Một số ứng dụng thường sử dụng một giao thức giao vận đơn giản hơn (giao thức UDP) cho các nhiệm vụ không đòi hỏi việc gửi dữ liệu một cách tuyệt đối đáng tin cậy, chẳng hạn như *video streaming*.

1.2. Các giao thức

1.2.1. OSI

OSI, hay còn gọi là "Mô hình liên kết giữa các hệ thống mở", là thiết kế dựa trên sự phát triển của ISO(Tổ Chức Tiêu Chuẩn Quốc Tế). Mô hình bao gồm 7 tầng:

Tầng ứng dụng: cho phép người dùng (con người hay phần mềm) truy cập vào mạng bằng cách cung cấp giao diện người dùng, hỗ trợ các dịch vụ như gửi thư điện tử truy cập và truyền file từ xa, quản lý cơ sở dữ liệu dùng chung và một số dịch vụ khác về thông tin.

Tầng trình diễn: thực hiện các nhiệm vụ liên quan đến cú pháp và nội dung của thông tin gửi đi.

Tầng phiên: đóng vai trò "kiểm soát viên" hội thoại (*dialog*) của mạng với nhiệm vụ thiết lập, duy trì và đồng bộ hóa tính liên tác giữa hai bên.

Tầng giao vận: nhận dữ liệu từ tầng phiên, cắt chúng thành những đơn vị nhỏ nếu cần, gửi chúng xuống tầng mạng và kiểm tra rằng các đơn vị này đến được đầu nhận.

Tầng mạng: điều khiển vận hành của mạng con. Xác định mở đầu và kết thúc của một cuộc truyền dữ liệu.

Tầng liên kết dữ liệu: nhiệm vụ chính là chuyển dạng của dữ liệu thành các khung dữ liệu (*data frames*) theo các thuật toán nhằm mục đích phát hiện, điều chỉnh và giải quyết các vấn đề như hư, mất và trùng lặp các khung dữ liệu.

Tầng vật lý: Thực hiện các chức năng cần thiết để truyền luồng dữ liệu dưới dạng bit đi qua các môi trường vật lý.

1.2.2. TCP/IP

TCP/IP cũng giống như OSI nhưng kiểu này có ít hơn ba tầng:

Tầng ứng dụng: bao gồm nhiều giao thức cấp cao. Trước đây người ta sử dụng các áp dụng đầu cuối ảo như TELNET,FTP, SMTP. Sau đó nhiều giao thức đã được định nghĩa thêm vào như DNS, HTTP.

Tầng giao vận: nhiệm vụ giống như phân giao vận của OSI nhưng có hai giao thức được dùng tới là TCD và UDP.

Tầng mạng: chịu trách nhiệm chuyển gói dữ liệu từ nơi gửi đến nơi nhận, gói dữ liệu có thể phải đi qua nhiều mạng (các chặng trung gian). Tầng liên kết dữ liệu thực hiện truyền gói dữ liệu giữa hai thiết bị trong cùng một mạng, còn tầng mạng đảm bảo rằng gói dữ liệu sẽ được chuyển từ nơi gửi đến đúng nơi nhận. Tầng này định nghĩa một dạng thức của gói và của giao thức là IP.

Tầng liên kết dữ liệu: Sử dụng để truyền gói dữ liệu trên một môi trường vật lý.

1.2.3. Giao thức HTTP

Giao thức HTTP (*HyperText Transfer Protocol*) là một giao thức được sử dụng chủ yếu trên *Internet* ngày nay. HTTP là một giao thức đi tiên phong trong sự phát triển *Web*.

HTTP là giao thức *Client/Server* nằm ở tầng ứng dụng của mô hình phân tầng *Internet*. Bằng việc mở rộng những phương thức hay dòng lệnh, người sử dụng có thể sử dụng HTTP cho nhiều chức năng khác nhau, kể cả chức năng quản lý hệ thống tên *server* và các đối tượng phân tán. HTTP không ngừng được cải tiến, dẫn đến việc W3C được đưa ra vào năm 1994 đã phát triển thành một tiêu chuẩn chung cho *Web*.

HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP.

HTTPS là viết tắt của "*Hypertext Transfer Protocol Secure*", Nó là một sự kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên *Internet*. Giao thức HTTPS thường được dùng trong các giao dịch nhạy cảm cần tính bảo mật cao.

Hoạt động của HTTP

- Các tiến trình xảy ra khi *Client* mở tài liệu HTML trên một WWW *server*:

Kết nối TCP: Mặc định HTTP *server* sẽ lắng nghe ở cổng 80. Trình duyệt lựa chọn một cổng cục bộ (>1024) và thực hiện một kết nối TCP đến cổng một *server* trước khi dữ liệu được gửi. Một HTTP *server* có thể lắng nghe ở một cổng khác, tuy nhiên *Client* cần phải cung cấp số thứ tự cổng chính xác ở URL để có thể kết nối.

- Yêu cầu phía *Client*:

Dòng yêu cầu: Chứa một câu lệnh yêu cầu gọi là phương thức địa chỉ URL của đối tượng được giải quyết yêu cầu bởi phương thức.

Phương thức: Được sử dụng để chỉ thị cho *server* thi hành một công việc đặc biệt. Hầu hết các *server* bao gồm cả IIS chỉ hỗ trợ 3 phương thức bởi vì các phương thức không được thi hành bởi hầu hết các trình duyệt.

GET: Phương thức này chỉ thị cho HTTP *server* gửi đến một đối tượng(các *Website*, *file*, hình ảnh) bằng địa chỉ URL. Phương thức này chỉ dùng trên HTTP *Client*.

HEAD: Phương thức này cũng giống như *GET*, tuy nhiên nó chỉ trả về thông tin *header* của đối tượng chứ không phải toàn bộ dữ liệu.

POST: Phương thức này được sử dụng bởi HTTP *Client* để gửi một đối tượng lên *server*.

- *Server* trả lời:

WWW *server* nhận một yêu cầu và quá trình này căn cứ trên phương thức yêu cầu chứa trong dòng lệnh yêu cầu. *Server* sau đó sẽ trả lời.

Dòng *Status*: Chỉ ra yêu cầu thành công hoặc bị lỗi.

Message Header Field: Nó cung cấp các thông tin về *server* và kiểu dữ liệu trả về.

Dữ liệu trả lời: Đây là dữ liệu yêu cầu ở dạng *bit*.

1.3. HTML

Ngôn ngữ siêu văn bản HTML(*Hyper Text Markup Language*) là ngôn ngữ biểu diễn văn bản cho phép ta đưa vào một văn bản nhiều thuộc tính cần thiết để có thể truyền thông quảng bá trên mạng WWW(*World Wide Web*). HTML cho phép ta thay đổi cách bày trí của văn bản, tạo những tài liệu siêu văn bản có khả năng đối thoại tương tác với người dùng.

HTML được định nghĩa như là một ứng dụng đơn giản của SGML(*Standard Generalized Markup Language*) và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn định dạng dữ liệu trên *Internet* do tổ chức *World Wide Web Consortium (W3C)* duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho *Web*.

1.4. CSS

- CSS là các tập tin định kiểu theo tầng (*Cascading Style Sheets-CSS*) được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML. Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho XML, SVG, XUL. Các đặc điểm kỹ thuật của CSS được duy trì bởi *World Wide Web Consortium (W3C)*. Thay vì đặt các thẻ qui định kiểu dáng cho văn bản HTML (hoặc XHTML) ngay trong nội dung của nó, nên sử dụng CSS.

- Tác dụng của CSS : Hạn chế tối thiểu việc làm rối mã HTML của trang *Web* bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang *Web* được gọn gàng hơn, tách nội dung của trang *Web* và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung. Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang *Web*, giúp tránh phải lặp lại việc định dạng cho các trang *Web* giống nhau.

- Sử dụng CSS:

Có 3 cách để sử dụng CSS

"Inline CSS" : Áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính style

Ví dụ :

```
<span style="font-weight:bold;text-decoration:underline;color:#FF0000;">Đoạn text cần in đậm, gạch chân, màu đỏ</span>
```

"Internal CSS" : Đặt CSS ở đầu trang *Web* để áp dụng kiểu dáng cho toàn bộ trang ấy, khi đó chỉ cần đặt đoạn CSS vào trong cặp thẻ <style> rồi đặt vào trong phần header của *Web* (giữa <head> và </head>)

Ví dụ :

```
<style type="text/css">
```

```
body {font-family:verdana;color:#0000FF;} /* Kiểu chữ
trong trang Web là "Verdana", màu chữ thông thường là màu xanh dương */
```

```
</style>
```

"External CSS" : Đặt các thuộc tính CSS vào một tệp tin riêng biệt (*.css), khi đó có thể tham chiếu đến từ nhiều trang *Web* khác nhau

Ví dụ về nội dung tệp style.css:

```
body {font-family:verdana;color:#0000FF;}
```

Tham chiếu tới tệp tin CSS trên từ trang *Web* bằng đoạn mã (mã có thể nằm ngoài thẻ <head>):

```
<link rel="stylesheet" type="text/css"
href="style.css">
```

- Mức độ ưu tiên khi sử dụng CSS

Mức độ ưu tiên khi áp dụng CSS như sau:

External CSS < *Internal CSS* < *Inline CSS*

Có thể hiểu rằng mã CSS nào "gần" với tag nhất thì sẽ được ưu tiên áp dụng hơn cả.

- Cú pháp :

Cú pháp cơ bản:

```
css_selector_1 {
thuộc_tính_1: giá_trị_của_thuộc_tính_1;
thuộc_tính_2: giá_trị_của_thuộc_tính_2;
...
thuộc_tính_n: giá_trị_của_thuộc_tính_n;
}
```

```
css_selector_2 {
thuộc_tính_1: giá_trị_của_thuộc_tính_1;
thuộc_tính_2: giá_trị_của_thuộc_tính_2;
...
thuộc_tính_n: giá_trị_của_thuộc_tính_n;
}
```

...

```
css_selector_n {
thuộc_tính_1: giá_trị_của_thuộc_tính_1;
thuộc_tính_2: giá_trị_của_thuộc_tính_2;
...
thuộc_tính_n: giá_trị_của_thuộc_tính_n;
}
```

Chú thích :

```
/* Chú thích 1 */
```

```
/* Chú thích 2 */
```

```
/* Chú thích 3,
```

đây là một chú thích nhiều dòng

```
*/
```

- *CSS Selector* : *CSS Selector* dùng để xác định đoạn mã CSS tương ứng được bao trong phần từ { đến } sẽ được áp dụng cho những thành phần trong trang *Web*.

Ví dụ *CSS Selector* theo *tag body*:

```
body {
background: #ffffff;
/* trang Web sẽ có nền màu trắng */
font-family: Verdana;
/* font chữ mặc định là Verdana */
color: #ff0000;
/* màu chữ mặc định
là màu đỏ */
}
```

1.5. Javascript

Javascript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang *web*, nhưng cũng được dùng để tạo khả năng viết *script* sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi *Brendan Eich* tại Hãng truyền thông *Netscape* với cái tên đầu tiên *Mocha*, rồi sau đó đổi tên thành *LiveScript*, và cuối cùng thành *Javascript*. Giống *Java*, *Javascript* có cú pháp tương tự C. Js là phần mở rộng thường được dùng cho tập tin mã nguồn *Javascript*.

Cùng thời điểm *Netscape* bắt đầu sử dụng công nghệ *Java* trên trình duyệt *Netscape*, *LiveScript* đã được đổi tên thành *Javascript* để được chú ý hơn bởi ngôn ngữ lập trình *Java* lúc đó đang được coi là một hiện tượng. *Javascript* được bổ sung vào trình duyệt *Netscape* bắt đầu từ phiên bản 2.0b3 của trình duyệt này vào tháng 12 năm 1995. Trên thực tế, *Javascript* không được phát triển dựa từ *Java*. Do đó *Javascript* chỉ dựa trên các cách đặt tên của *Java*. *Java Script* gồm 2 mảng là *client-server* thực hiện lệnh trên máy của *end-user* và *web-server*.

Sau thành công của *Javascript*, *Microsoft* bắt đầu phát triển *JScript*, một ngôn ngữ có cùng ứng dụng và tương thích với *Javascript*. *JScript* được bổ sung vào trình duyệt *Internet Explorer* bắt đầu từ *Internet Explorer* phiên bản 3.0 được phát hành tháng 8 năm 1996.

DOM (*Document Object Model*), một khái niệm thường được nhắc đến với *Javascript* trên thực tế không phải là một phần của chuẩn *ECMAScript*, DOM là một chuẩn riêng biệt có liên quan chặt chẽ với XML.

Javascript là một ngôn ngữ lập trình dựa trên nguyên mẫu với cú pháp phát triển từ C. Giống như C, *Javascript* có khái niệm từ khóa, do đó *Javascript* gần như không thể được mở rộng.

Cũng giống như C, *Javascript* không có bộ xử lý xuất/nhập (input/output) riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, *Javascript* dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.

Trên trình duyệt, rất nhiều trang *web* sử dụng *Javascript* để thiết kế trang *web* động và một số hiệu ứng hình ảnh thông qua DOM. *Javascript* được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, *Javascript* còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang *Wikipedia* tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng *Javascript* khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng *Javascript* để tương tác với DOM bao gồm DHTML, *Ajax* và *SPA*.

Bên ngoài trình duyệt, *Javascript* có thể được sử dụng trong tập tin PDF của *Adobe Acrobat* và *Adobe Reader*. Điều khiển *Dashboard* trên hệ điều hành *Mac OS X* phiên bản 10.4 cũng có sử dụng *Javascript*. Công nghệ kịch bản linh động (*active scripting*) của *Microsoft* có hỗ trợ ngôn ngữ *JScript* làm một ngôn ngữ kịch bản dùng cho hệ điều hành. *JScript.NET* là một ngôn ngữ tương thích với CLI gần giống *JScript* nhưng có thêm nhiều tính năng lập trình hướng đối tượng.

Mỗi ứng dụng này đều cung cấp mô hình đối tượng riêng cho phép tương tác với môi trường chủ, với phần lõi là ngôn ngữ lập trình *Javascript* gần như giống nhau.

1.6. Apache và IIS

Apache hay là chương trình máy chủ HTTP là một chương trình dành cho máy chủ đối thoại qua giao thức HTTP. *Apache* chạy trên các hệ điều hành tương tự như *Unix*, *Microsoft Windows*, *Novell Netware* và các hệ điều hành khác. *Apache* đóng một vai trò quan trọng trong quá trình phát triển của mạng *web* thế giới (*World Wide Web*).

Khi được phát hành lần đầu, *Apache* là chương trình máy chủ mã nguồn mở duy nhất có khả năng cạnh tranh với chương trình máy chủ tương tự của *Netscape Communications Corporation* mà ngày nay được biết đến qua tên thương mại *Sun Java System WebServer*. Từ đó trở đi, *Apache* đã không ngừng tiến triển và trở thành một phần mềm có sức cạnh tranh mạnh so với các chương trình máy chủ khác về mặt hiệu suất và tính năng phong phú. Từ tháng 4 năm 1996, *Apache* trở thành một chương trình máy chủ HTTP thông dụng nhất.

IIS (*Internet Information Services*) là một dịch vụ tùy chọn của *Windows NT Server* cung cấp các tính năng về *Website*. IIS là một thành phần cơ bản để xây dựng một *Internet* hoặc *Intranet Server* trên nền *Windows NT 4.0*, *Workstation* và *Win 95*. IIS được tích hợp đầy đủ trong *Windows NT 4.0*. Với một bộ đầy đủ IIS và *Windows NT 4.0* người sử dụng sẽ nhận được sự thuận tiện khi xây dựng một cơ chế bảo mật trên *Windows NT server* và *Windows NT File System (NTFS)*.

1.7. Tổng quan về MySQL

MySQL là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với *Apache*, *PHP*. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên *MySQL* đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. *MySQL* cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ *SQL*. Nhưng *MySQL* không bao quát toàn bộ những câu truy vấn cao cấp như *SQL*. Về bản chất *MySQL* chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của *website* nhưng hầu hết có thể giải quyết các bài toán trong *PHP*.

1.7.1. Khởi động và sử dụng

Chúng ta sử dụng command như sau:

```
MySQL -hname -uuser -ppass
```

Để truy cập vào cơ sở dữ liệu.

Hoặc sử dụng bộ *appserv* để vào nhanh hơn theo đường dẫn sau:

Start/ Appserv/ *MySQL* command Line *client*

Sau đó nhập *password* mà chúng ta đã đặt vào.

1.7.2. Một số thuật ngữ

NULL : Giá trị cho phép rỗng.

AUTO_INCREMENT : Cho phép giá trị tăng dần (tự động).

UNSIGNED : Phải là số nguyên dương

PRIMARY KEY : Cho phép nó là khóa chính trong bảng.

1.7.3. Loại dữ liệu trong MySQL

Kiểu dữ liệu	Mô Tả
Char	Định dạng text có chiều dài từ 0->255
Varchar	Định dạng text có chiều dài từ 0->255
Text	Định dạng text có chiều dài 0->65535
Longtext	Định dạng text có chiều dài 0->4294967215
INT	Định dạng số có chiều dài từ 0->4294967215
Float	Định dạng số thập phân có chiều dài nhỏ
Double	Định dạng số thập phân có chiều dài lớn
Date	Định dạng thời gian theo định dạng: YYYY-MM-DD
DateTime	Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS

1.7.4. Những cú pháp cơ bản

- **Tạo một cơ sở dữ liệu:**

```
CREATE DATABASE tên_cơ_sở_dữ_liệu;
```

Cú pháp sử dụng cơ sở dữ liệu: Use tên_database;

Cú pháp thoát khỏi cơ sở dữ liệu: *Exit*

Tạo một bảng trong cơ sở dữ liệu:

```
CREATE TABLE user
(<tên_cột><mô_tả>, ..., <tên_cột_n>...<mô_tả_n>)
```

Hiển thị có bao nhiêu bảng: *show tables;*

Hiển thị có bao nhiêu cột trong bảng: *show columns from table;*

Thêm 1 cột vào bảng:

```
ALTER          TABLE          tên_bảng          ADD
<tên_cột><thuộc_tính> AFTER <tên_cột>
```

Thêm giá trị vào bảng:

```
INSERT          INTO          Tên_bảng(tên_cột)
VALUES (Giá_trị_tương_ứng);
```

Truy xuất dữ liệu:

```
SELECT tên_cột FROM Tên_bảng;
```

Truy xuất dữ liệu với điều kiện:

```
SELECT tên_cột FROM Tên_bảng WHERE điều_kiện;
```

Truy xuất dữ liệu và sắp xếp theo trình tự:

```
SELECT tên_cột FROM Tên_bảng
WHERE điều_kiện (có thể có where hoặc không)
ORDER BY Theo quy ước sắp xếp.
```

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

Truy cập dữ liệu có giới hạn:

```
SELECT tên_cột FROM Tên_bảng
WHERE điều_kiện (có thể có where hoặc không)
LIMIT vị trí bắt đầu, số record muốn lấy ra
```

Cập nhật dữ liệu trong bảng:

```
Update tên_bảng set tên_cột=Giá trị mới
WHERE (điều_kiện).
```

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các record trong bảng.

Xóa dữ liệu trong bảng:

```
DELETE FROM tên_bảng WHERE (điều_kiện).
```

Nếu không có ràng buộc điều kiện, chúng sẽ xóa toàn bộ giá trị của các record trong bảng.

1.8. Tổng quan về PHP

1.8.1. Lịch sử phát triển

PHP

Được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do *Rasmus Lerdorf* tạo ra năm 1995, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là '*Personal Home Page Tools*'.

PHP 3

PHP 3.0 là phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gũi với các phiên bản PHP mà chúng ta được biết ngày nay. Nó đã được *Andi Gutmans* và *Zeev Suraski* tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. PHP 3.0 đã chính thức được công bố vào tháng 6 năm 1998, sau thời gian 9 tháng được cộng đồng kiểm nghiệm.

PHP 4

Vào mùa đông năm 1998, ngay sau khi PHP 3.0 chính thức được công bố, *Andi Gutmans* và *Zeev Suraski* đã bắt đầu bắt tay vào việc viết lại phần lõi của PHP. Một động cơ mới, có tên '*Zend Engine*' (ghép từ các chữ đầu trong tên của *Zeev* và *Andi*), đã đáp ứng được các nhu cầu thiết kế này một cách thành công, và lần đầu tiên được giới thiệu vào giữa năm 1999. PHP 4.0, dựa trên động cơ này, và đi kèm với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố vào tháng 5 năm 2000, gần 2 năm sau khi bản PHP 3.0 ra đời.

PHP 5

Sự thành công hết sức to lớn của PHP 4.0 đã không làm cho nhóm phát triển PHP tự mãn. Cộng đồng PHP đã nhanh chóng giúp họ nhận ra những yếu kém của PHP 4 đặc biệt với khả năng hỗ trợ lập trình hướng đối tượng (OOP), xử lý XML, không hỗ trợ giao thức máy khách mới của *MySQL* 4.1 và 5.0, hỗ trợ dịch vụ *web* yếu. Những điểm này chính là mục đích để Zeev và Andi viết *Zend Engine* 2.0, lõi của PHP 5.0. Một thảo luận trên Slashdot đã cho thấy việc phát triển PHP 5.0 có thể đã bắt đầu vào thời điểm tháng 12 năm 2002 nhưng những bài phỏng vấn Zeev liên quan đến phiên bản này thì đã có mặt trên mạng *Internet* vào khoảng tháng 7 năm 2002. Ngày 29 tháng 6 năm 2003, PHP 5 Beta 1 đã chính thức được công bố để cộng đồng kiểm nghiệm. Đó cũng là phiên bản đầu tiên của *Zend Engine* 2.0. Phiên bản Beta 2 sau đó đã ra mắt vào tháng 10 năm 2003 với sự xuất hiện của hai tính năng rất được chờ đợi: *Iterators*, *Reflection* nhưng *namespaces* một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn. Ngày 21 tháng 12 năm 2003: PHP 5 Beta 3 đã được công bố để kiểm tra với việc phân phối kèm với Tidy, bỏ hỗ trợ *Windows* 95, khả năng gọi các hàm PHP bên trong XSLT, sửa chữa nhiều lỗi và thêm khá nhiều hàm mới. PHP 5 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.

1.8.2. Cấu trúc cơ bản

PHP cũng có thể bắt đầu và kết thúc giống với ngôn ngữ HTML. Chỉ khác, đối với PHP chúng ta có nhiều cách để thể hiện.

Cách 1 : Cú pháp chính:

```
<?php Mã lệnh PHP ?>
```

Cách 2: Cú pháp ngắn gọn

```
<? Mã lệnh PHP ?>
```

Cách 3: Cú pháp giống với ASP.

```
<% Mã lệnh PHP %>
```

Cách 4: Cú pháp bắt đầu bằng *script*

```
<script language=php>
```

.....

```
</script>
```

Mặc dù có 4 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chọn tối ưu.

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu ";"

Để chú thích 1 đoạn dữ liệu nào đó trong PHP ta sử dụng dấu "/*" cho từng dòng. Hoặc dùng cặp thẻ "/*.....*/" cho từng cụm mã lệnh.

Ví dụ: `<?php echo "Hello world!"; ?>`

1.8.3. Xuất giá trị ra trình duyệt

Để xuất dữ liệu ra trình duyệt chúng ta có những dòng cú pháp sau :

```
+ echo "Thông tin";
```

```
+ printf "Thông tin";
```

Thông tin bao gồm : biến, chuỗi, hoặc lệnh HTML

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu "."

1.8.4. Biến, hằng, chuỗi và các kiểu dữ liệu

Biến

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu "\$". Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.

1 biến được xem là hợp lệ khi nó thỏa các yếu tố :

+ Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.

+ Tên của biến không được phép trùng với các từ khóa của PHP.

Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khai báo vừa gán dữ liệu cho biến.

Bản thân biến cũng có thể gán cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.

Hằng

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm define theo cú pháp: `define (string tên_hằng, giá_trị_hằng)`.

Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố :

- + Hằng không có dấu "\$" ở trước tên.
- + Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- + Hằng chỉ được phép gán giá trị duy nhất 1 lần.
- + Hằng thường viết bằng chữ in để phân biệt với biến

Chuỗi

Chuỗi là một nhóm các kỹ tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Ví dụ:

'Hello'

"welcome to VietNam"

Để tạo 1 biến chuỗi, chúng ta phải gán giá trị chuỗi cho 1 biến hợp lệ.

Ví dụ:

```
$first_name= "Nguyen";
```

```
$last_name= `Van A` ;
```

Để liên kết 1 chuỗi và 1 biến chúng ta thường sử dụng dấu "."

Kiểu dữ liệu

Các kiểu dữ liệu khác nhau chiếm các lượng bộ nhớ khác nhau và có thể được xử lý theo cách khác nhau khi chúng được theo tác trong 1 *script*.

Trong PHP chúng ta có 6 kiểu dữ liệu chính như sau :

Chúng ta có thể sử dụng hàm dựng sẵn *GETtype()* của PHP4 để kiểm tra kiểu của bất kỳ biến.

1.8.5. Các phương thức được sử dụng trong lập trình PHP

Có 2 phương thức được sử dụng trong lập trình PHP là *GET* và *POST*

Phương thức *GET*

Phương thức này cũng được dùng để lấy dữ liệu từ form nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung trang dữ liệu từ *webserver*.

Ví dụ:

Với url sau: *shownews.php?id=50*

Vậy với trang *shownews* ta dùng hàm `$_GET['id']` sẽ được giá trị là 50.

Phương thức *POST*

Phương thức này được sử dụng để lấy dữ liệu từ form nhập liệu. Và chuyển chúng lên trình chủ *webserver*.

1.8.6. Cookie và Session trong PHP

Cookie và *Session* là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống

Cookie

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại *server* mỗi khi *browser* tải 1 trang *web* từ *server*.

Những thông tin được lưu trữ trong *Cookie* hoàn toàn phụ thuộc vào *website* trên *server*. Mỗi *website* có thể lưu trữ những thông tin khác nhau trong *Cookie*, ví dụ thời điểm lần cuối ta ghé thăm *website*, đánh dấu ta đã *login* hay chưa, v.v...

Cookie được tạo ra bởi *website* và gửi tới *browser*, do vậy 2 *website* khác nhau (cho dù cùng *host* trên 1 *server*) sẽ có 2 *Cookie* khác nhau gửi tới *browser*. Ngoài ra, mỗi *browser* quản lý và lưu trữ *Cookie* theo cách riêng của mình, cho nên 2 *browser* cùng truy cập vào 1 *website* sẽ nhận được 2 *Cookie* khác nhau.

1. Để thiết lập *Cookie* ta sử dụng cú pháp:

SetCookie("tên *Cookie*", "giá trị", thời gian sống)

Tên *Cookie* là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên *Cookie*.

Ví dụ: *setCookie*("name", "admin", *time*()+3600);

2. Để sử dụng lại *Cookie* vừa thiết lập, chúng ta sử dụng cú pháp:

Cú pháp: `$_COOKIE["tên Cookies"]`

Tên *Cookie* là tên mà chúng ta thiết lập phía trên.

3. Để hủy 1 *Cookie* đã được tạo ta có thể dùng 1 trong 2 cách sau:

+ Cú pháp: *setCookie*("Tên *Cookie*")

Gọi hàm *setCookie* với chỉ duy nhất tên *Cookie* mà thôi

+ Dùng thời gian hết hạn *Cookie* là thời điểm trong quá khứ.

Ví dụ: *setCookie*("name", "admin", *time*()-3600);

Session

Một cách khác quản lý người sử dụng là *Session*. *Session* được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một *Session* được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi *Session* sẽ có được cấp một định danh (ID) khác nhau.

- Để thiết lập 1 *Session* ta sử dụng cú pháp: *Session_start*()

Đoạn *code* này phải được nằm trên các kịch bản HTML. Hoặc những lệnh *echo, printf*.

Để thiết lập 1 giá trị *Session*, ngoài việc cho phép bắt đầu thực thi *Session*. Chúng ta còn phải đăng ký 1 giá trị *Session*. Để tiện cho việc gán giá trị cho *Session* đó.

Ta có cú pháp sau: *Session_register*("Name")

4. Giống với *Cookie*. Để sử dụng giá trị của *Session* ta sử dụng mã lệnh sau:

Cú pháp: `$_SESSION["name"]`

Với Name là tên mà chúng ta sử dụng hàm `Session_register("name")` để khai báo.

5. Để hủy bỏ giá trị của *Session* ta có những cách sau:

```
Session_destroy() // Cho phép hủy bỏ toàn bộ giá trị của Session
```

```
Session_unset() // Cho phép hủy bỏ Session .
```

1.8.7. Hàm

Để giảm thời gian lặp lại 1 thao tác code nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong *website*. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc. Đồng thời có thể tùy biến ở mọi trang. Mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần

Hàm tự định nghĩa

Cú pháp:

```
function function_name()
{
//Lệnh thực thi
}
```

Tên hàm có thể là một tổ hợp bất kỳ những chữ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chữ cái và dấu gạch dưới.

Hàm tự định nghĩa với các tham số

Cú pháp:

```
function function_name($gt1, $gt2)
{
//Lệnh thực thi
}
```

Hàm tự định nghĩa với giá trị trả về

Cú pháp:

```
function function_name(Có hoặc không có đối số)
```

```
{
// Lệnh thực thi
return giatri;
}
```

Gọi lại hàm

PHP cung cấp nhiều hàm cho phép triệu gọi lại file. Như hàm *include*("URL đến file"), *require*("URL Đến file").

Ngoài hai cú pháp trên còn có *include_once*(), *require_once*(). Hai hàm này cũng có trách nhiệm gọi lại hàm. Những chúng sẽ chỉ gọi lại duy nhất 1 lần mà thôi.

1.9. Kết hợp PHP và MySQL trong ứng dụng website

Kết nối cơ sở dữ liệu

```
MySQL_connect("hostname","user","pass");
```

Lựa chọn cơ sở dữ liệu

```
MySQL_select_db("tên_CSDL");
```

Thực thi câu lệnh truy vấn

```
MySQL_query("Câu truy vấn ở đây");
```

Đếm số dòng dữ liệu trong bảng

```
MySQL_num_rows();
```

Lấy dữ liệu từ bảng đưa vào mảng

```
MySQL_fetch_array();
```

Đóng kết nối cơ sở dữ liệu

```
MySQL_close();
```

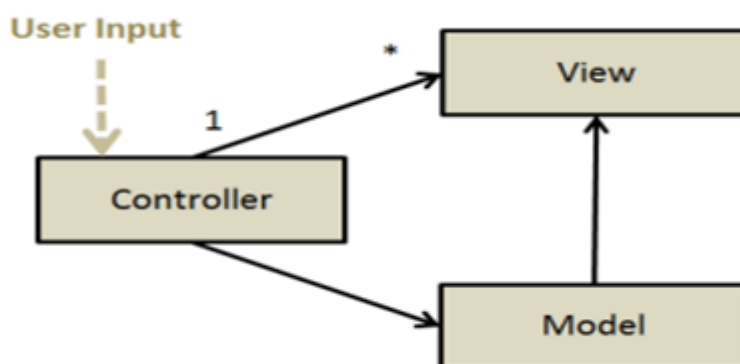
CHƯƠNG 2: OPENCART

2.1. Kiến trúc của Opencart

Opencart là một giải pháp thương mại điện tử dựa trên kiến trúc MVC (*Model View Controller*). MVC có nghĩa là mô hình (*Model*), xem (*View*), điều khiển (*Controller*).

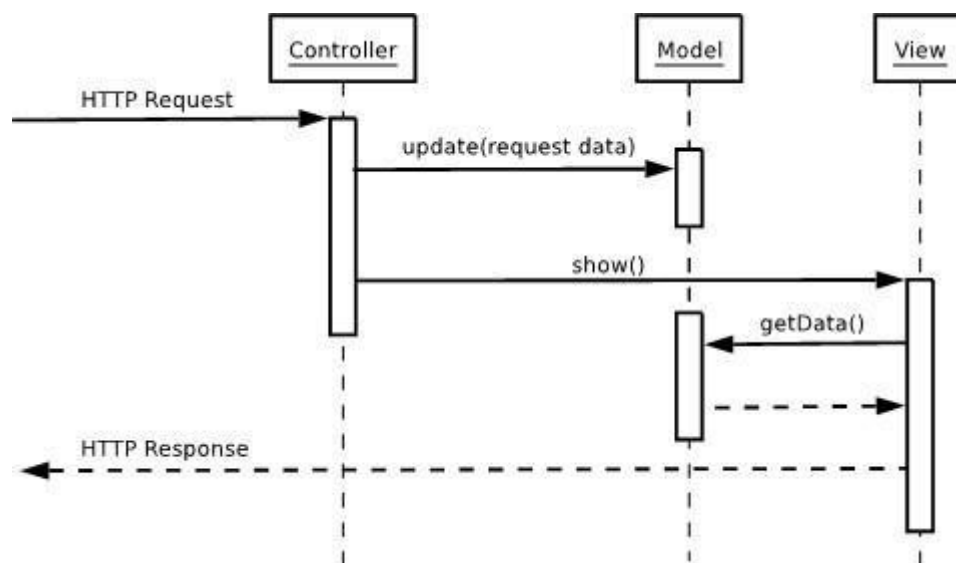
Các mô hình MVC (*Model View Controller*) là sự tách biệt của một ứng dụng giao diện người dùng đồ họa (GUI) từ cốt lõi *logic* của nó. Không có thiết kế tuyệt đối cho MVC, nhưng như với bất kỳ mẫu thiết kế nào, nên thích ứng với tình huống đang phải đối mặt khi phát triển.

MVC là một mẫu kiến trúc phần mềm trong kỹ nghệ phần mềm. Khi sử dụng đúng cách, mẫu MVC giúp cho nhà phát triển phần mềm tách biệt các pha phát triển phần mềm riêng rẽ với các nguyên tắc nghiệp vụ và giao diện người dùng. Phần mềm phát triển theo mẫu MVC tạo nhiều thuận lợi cho việc bảo trì vì các nguyên tắc nghệ nghiệp và giao diện ít liên quan với nhau.



Hình 2. 1: Kiến trúc mô hình MVC

Khi người dùng cần tương tác hoặc thay đổi trạng thái của đối tượng đồ họa thì sẽ tương tác thông qua *Controller* của đối tượng đồ họa. *Controller* sẽ thực hiện việc thay đổi trên *Model*. Khi có thay đổi ở *Model*, sẽ có một thông điệp được phát thông báo cho *View* và *Controller* biết. Nhận được thông điệp từ *Model*, *View* sẽ cập nhật lại thể hiện của mình, đảm bảo luôn thể hiện trực quan chính xác của *Model*. Còn *Controller* khi nhận được thông điệp từ *Model*, sẽ có những tương tác phản hồi lại người sử dụng.



Hình 2. 2: Mô hình tuần tự của MVC

Ưu điểm:

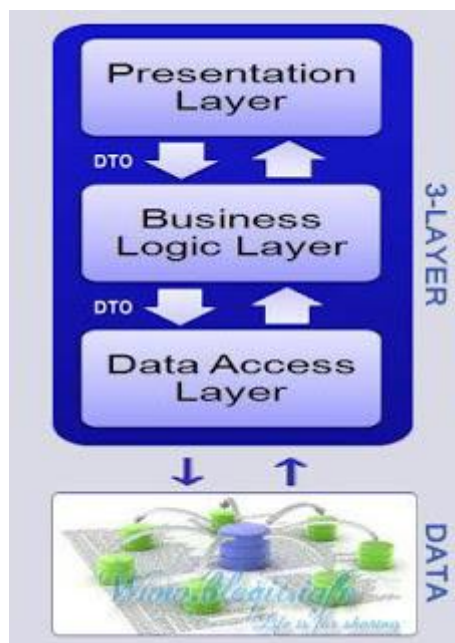
Ưu điểm mạnh nhất của MVC là nó giúp cho ứng dụng dễ bảo trì, *module* hóa các chức năng, và được xây dựng nhanh chóng. MVC tách các tác vụ của ứng dụng thành các phần riêng rẽ *Model*, *View*, *Controller* giúp cho việc xây dựng ứng dụng nhanh hơn. Dễ thêm các tính năng mới, dễ thay đổi các tính năng cũ. Có được ưu điểm đó vì MVC tách biệt sự phụ thuộc giữa các thành phần trong một đối tượng đồ họa, làm tăng tính linh động. Lúc đó có thể dễ dàng thay đổi giao diện bằng cách thay đổi thành phần *View* trong khi cách thức lưu trữ(*Model*) cũng như xử lý(*Controller*) không hề thay đổi. Tương tự có thể thay đổi *Model* và *Controller* mà những thành phần còn lại vẫn giữ nguyên.

Nhược điểm:

Đối với dự án nhỏ thì việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển, tốn thời gian chuyển dữ liệu giữa các tầng.

So sánh MVC và Three Layer:

Một mô hình không thể không nhắc đến khi nhắc đến MVC đó là mô hình *Three Layer*. Khá phổ biến và có nhiều điểm tương đồng với MVC.



Hình 2. 3: Mô hình Three Layer

Presentation Layer: Lớp này làm nhiệm vụ giao tiếp với người dùng để thu thập dữ liệu và hiển thị kết quả.

Business Logic Layer: Lớp này xử lý chính các dữ liệu trước khi chuyển xuống *Data Access Layer*. Đây là nơi kiểm tra các yêu cầu nghiệp vụ, tính toán, xử lý các yêu cầu và lựa chọn kết quả trả về cho *Presentation Layer*.

Data Access Layer: Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng.

Có những điểm tương đồng với MVC như đều có 3 lớp nhưng vẫn có những điểm khác nhau cơ bản giữa hai mô hình này.

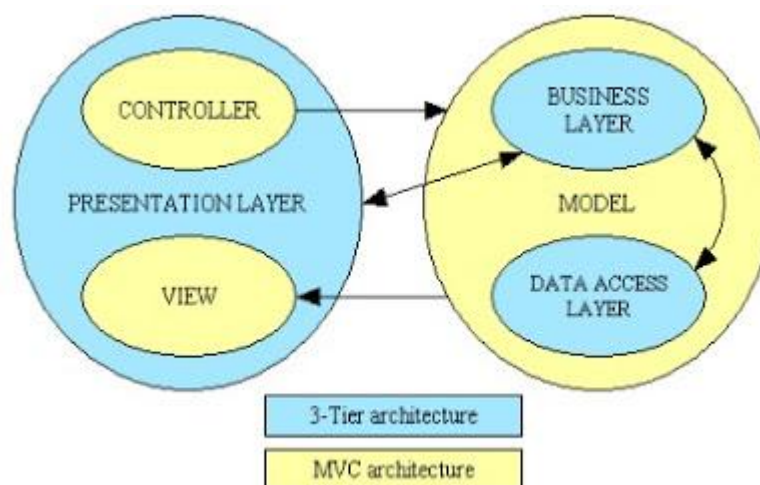
Sự khác biệt rõ nhất là cách xử lý yêu cầu của *Three Layer* khác so với MVC:

- Ở *Three Layer* yêu cầu được nhận từ *Presentation Layer* rồi chuyển qua *Business Logic Layer* để xử lý và được lưu trữ ở *Data Access Layer*.

- Còn MVC khi yêu cầu được gửi từ *Controller*, *Controller* sẽ thực hiện việc thay đổi trên *Model*. Lúc đó *Model* sẽ có thông điệp đồng thời cho *View* và *Controller* biết.

- Ở MVC thành phần *Model* giữ chức năng giống như của cả lớp *Business* và *Data Access* của *Three Layer*.

- *Three Layer* được sử dụng trong lập trình ứng dụng, còn MVC được sử dụng nhiều trong lập trình *Website*.



Hình 2. 4: So sánh MVC với Three Layer

2.1.1. Tầng mô hình (*Business process layer*)

Mô hình hóa dữ liệu và hành vi trong xử lý nghiệp vụ.

Chịu trách nhiệm:

- Thực hiện các truy vấn trong cơ sở dữ liệu.
- Tính toán trong các tiến trình xử lý nghiệp vụ.
- Đóng gói dữ liệu và hành vi, độc lập với tầng biểu diễn (*presentation layer*).

2.1.2. Tầng biểu diễn (*Presentation layer*)

- Hiển thị thông tin tùy thuộc vào loại *client*.
- Biểu diễn kết quả của tầng *business logic* (*Model*).
- Không cần quan tâm làm thế nào có được thông tin, hoặc thông tin ở đâu (*Model* chịu trách nhiệm).

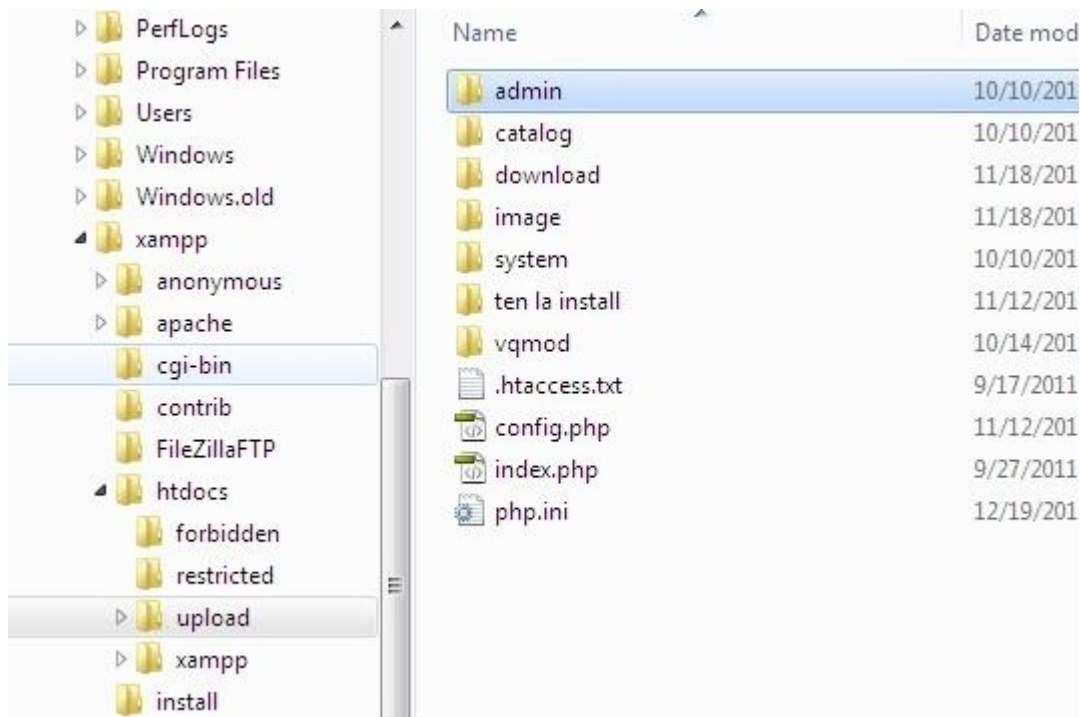
2.1.3. Tầng điều khiển (*Control layer*)

- Kết nối tương tác của người dùng với các nghiệp vụ cung cấp phía sau.
- Chọn ra cách biểu diễn phù hợp: ngôn ngữ, biến đổi định dạng thông tin theo vùng, quyền hạn người dùng.

- Một *request* tới ứng dụng sẽ chuyển cho tầng *Control*: Tầng này quyết định *request* được xử lý như thế nào, và thông tin sẽ được trả lại như thế nào.

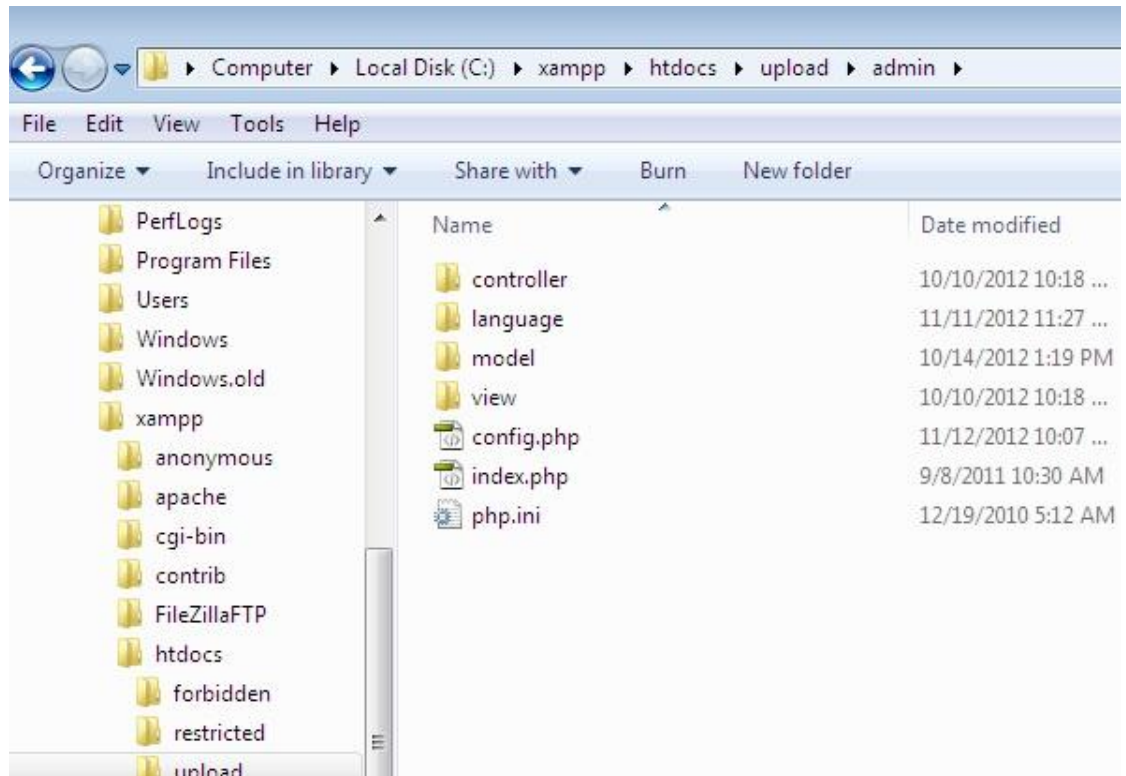
2.2. Cấu trúc thư mục tệp tin Opencart

- Thư mục *Opencart* là thư mục *upload* nằm trong thư mục `C:\xampp\htdocs`.



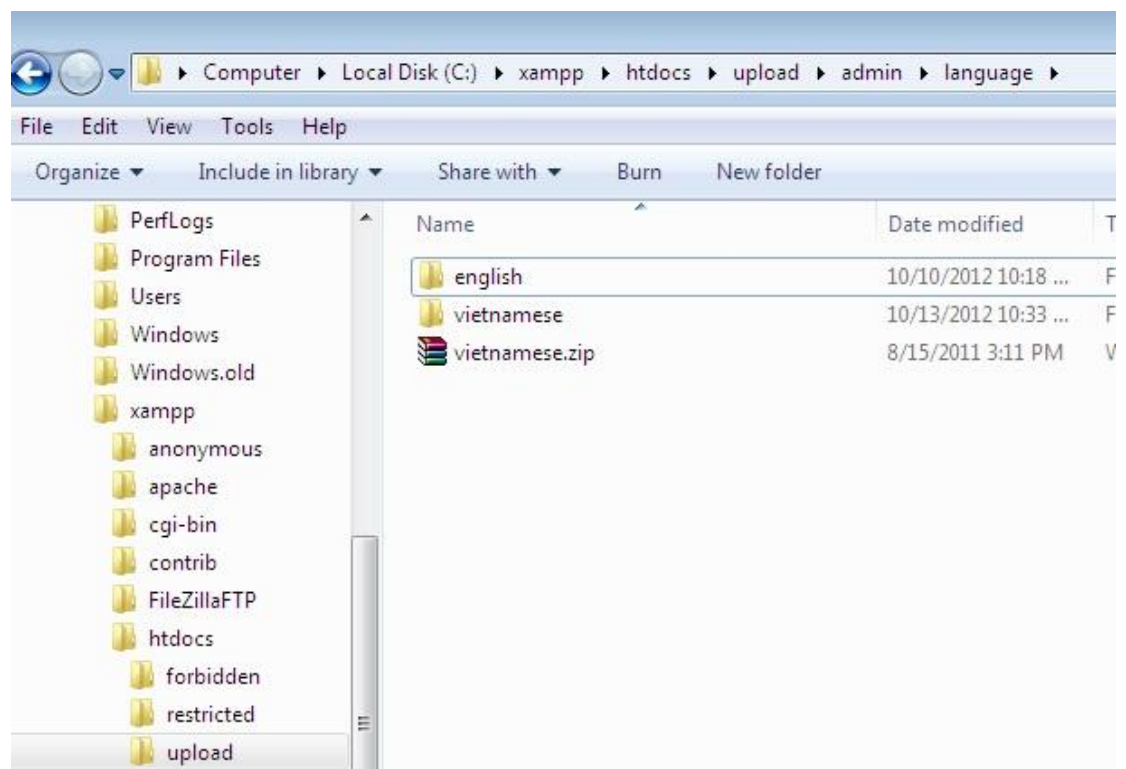
Hình 2. 5: Cấu trúc thư mục tệp tin Opencart

- Trong thư mục *upload* có chứa các thư mục: *admin*, *catalog*, *download*, *image*, *system*...
- Trong thư mục *admin* chứa các thư mục mô hình MVC: *Controller*, *Model*, *View*, *language*,...



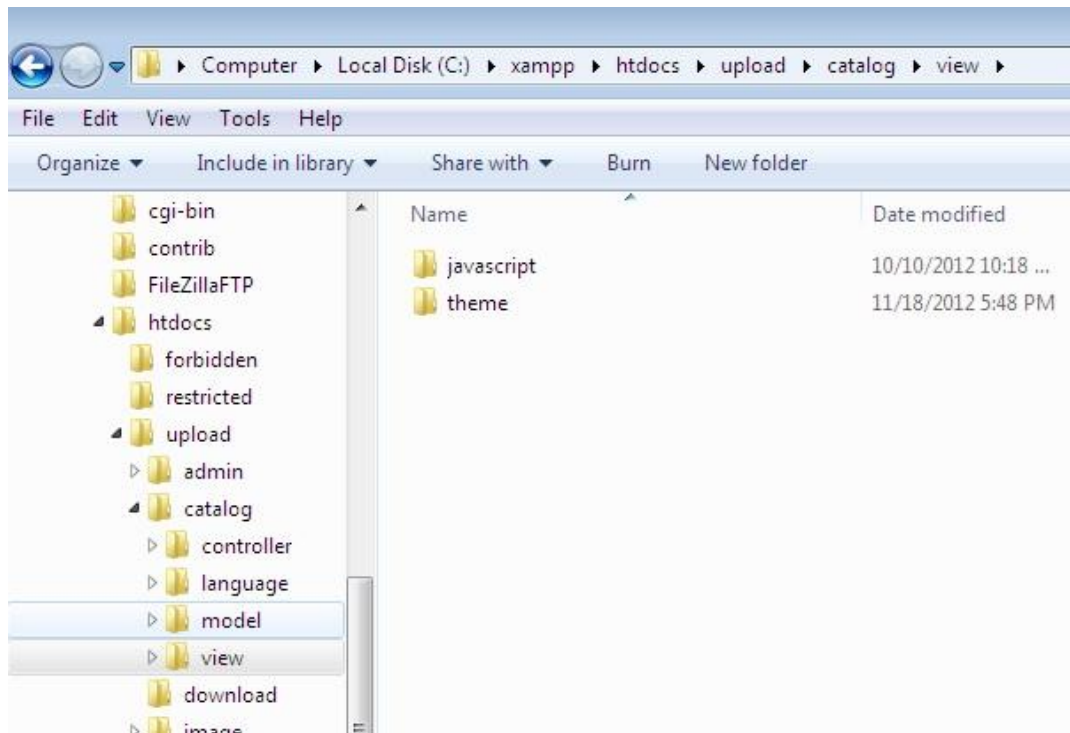
Hình 2. 6: Thư mục admin

- Thư mục *language* chứa các thư mục ngôn ngữ: *english*, *vietnamese*...



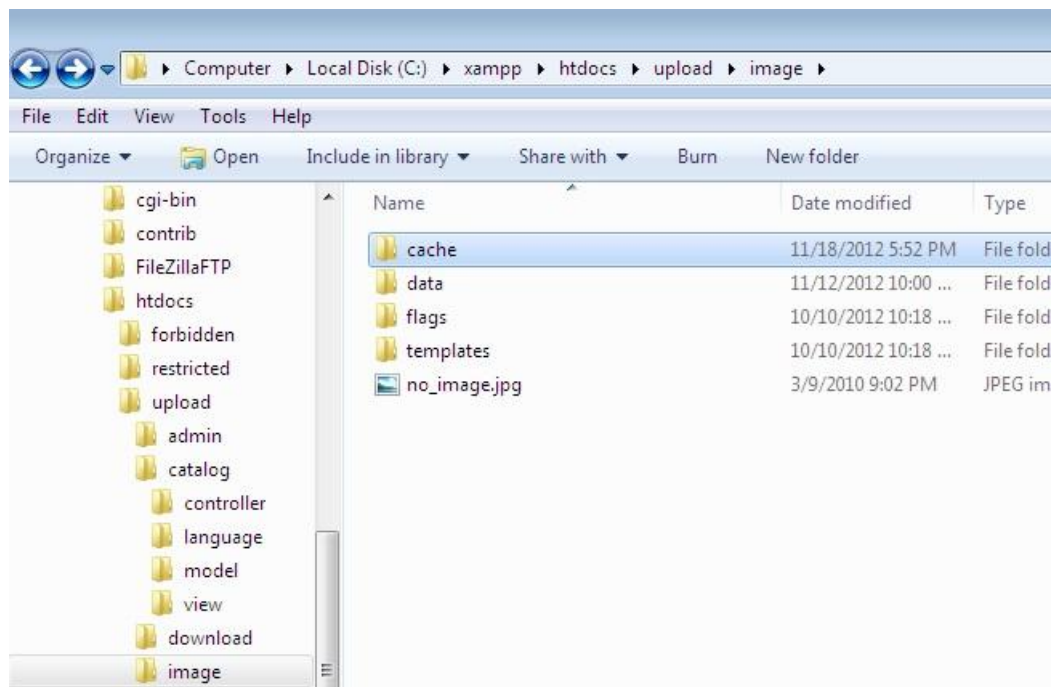
Hình 2. 7: Thư mục language

- Trong thư mục *catalog* chứa các thư mục: *Model*, *language*, *Controller* và thư mục *View* chứa các thư mục giao diện, *Javascript*



Hình 2. 8: Thư mục catalog

- Trong thư mục *image* chứa các thư mục *flags*, các *templates* và các *icon*...



Hình 2. 9: Thư mục image

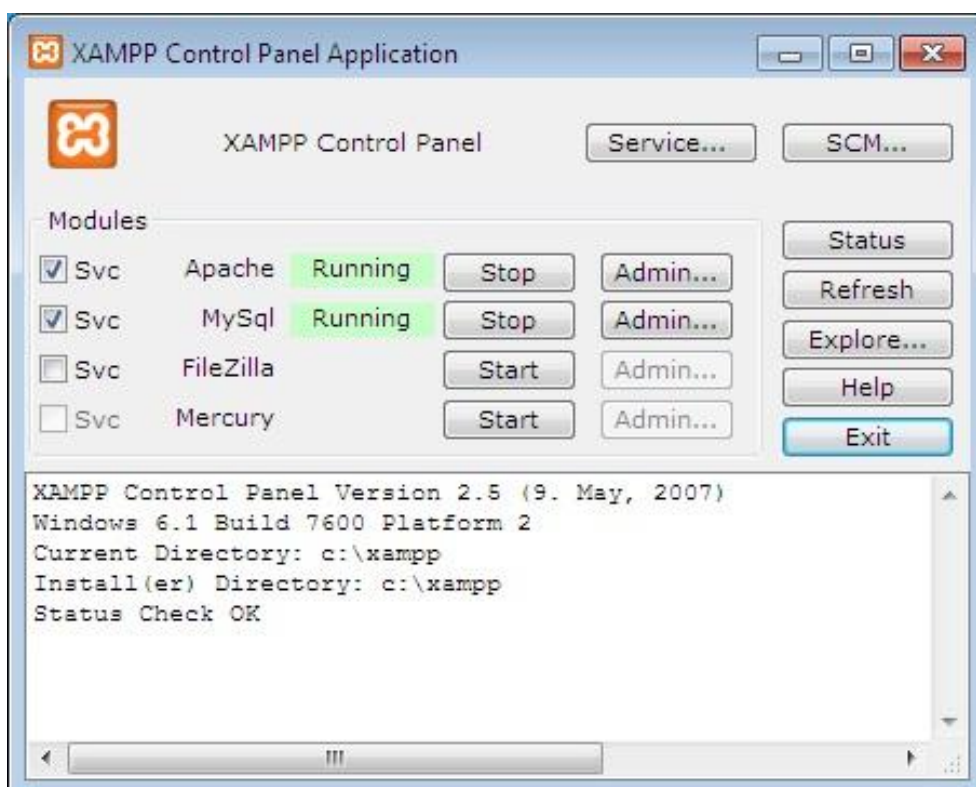
2.3. Cài đặt và việt hóa Opencart

2.3.1. Cài đặt XAMPP

Đầu tiên ta cần cài đặt XAMPP để thiết lập môi trường. XAMPP là một nguồn mở “*cross-platform*” miễn phí, gói phần mềm máy chủ *web*. Nó bao gồm chủ yếu là *Apache HTTP Server*, cơ sở dữ liệu *MySQL*, và trình biên dịch cho các kịch bản viết bằng các ngôn ngữ lập trình PHP và Perl. 'X' trong Xampp có nghĩa là nền tảng chéo. Vì vậy, XAMPP có sẵn cho *Microsoft Windows*, *Linux*, *Solaris*, và *Mac OS X*, và chủ yếu được sử dụng cho các dự án phát triển *web*.

Các bước cài đặt XAMPP:

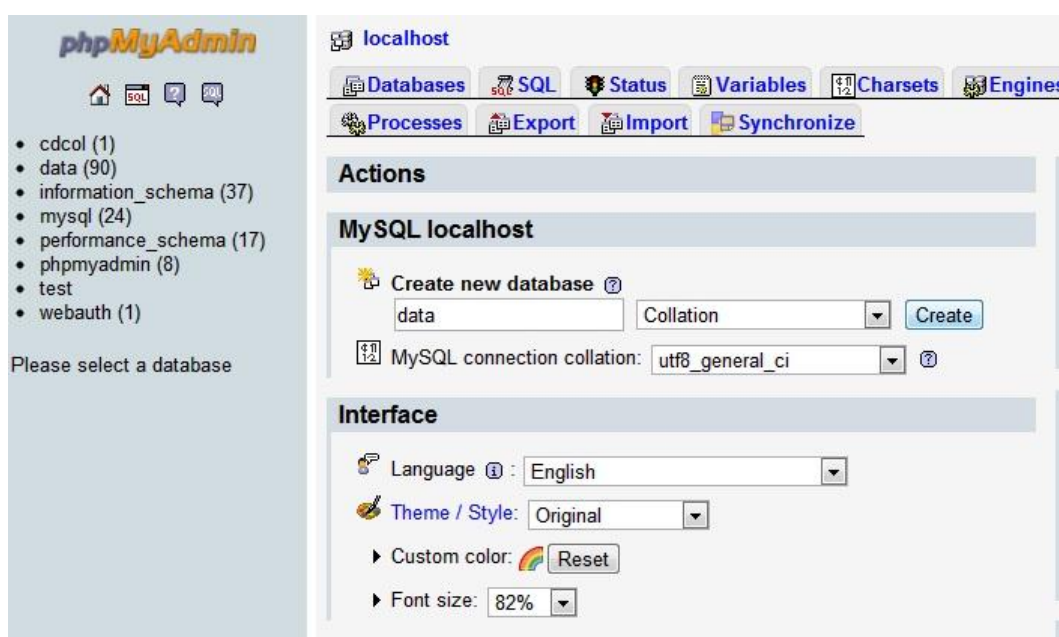
1. Nhấn đúp chuột vào trình cài đặt,
2. Chọn thư mục thích hợp để cài đặt.
3. Thực hiện theo các hướng dẫn trên cửa sổ cài đặt.
4. Mở bảng điều khiển XAMPP. Ở đó, chúng ta sẽ thấy điều khiển riêng biệt cho *Apache*, *MySQL*, *FileZilla*, và *Mercury*.
5. Nhấp vào *Startbutton* để bắt đầu tính năng tương ứng. Có một thông báo trạng thái của trạng thái hiện tại của các tính năng được lựa chọn.



Hình 2. 10: Quá trình cài đặt XAMPP đã thành công

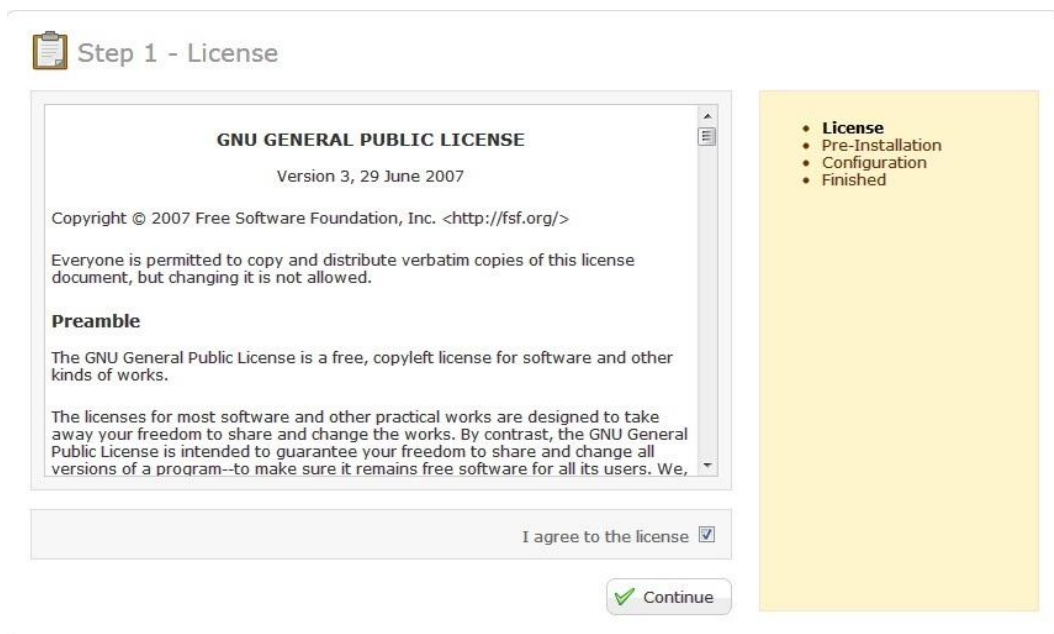
2.3.2. Cài đặt Opencart

- Tiến hành tải phần mềm mã nguồn mở *Opencart*:
- <http://www.Opencart.com/index.php?ro...nload/download>
- Sau khi tải xong tiến hành giải nén tập tin : nhấp chuột phải lên tập tin -> chọn *Extract Here*.
- Copy thư mục “ *Upload* ” vừa giải nén ra và *paste* vào “ C:\xampp\htdocs ”
- Chạy đường dẫn “ <http://localhost/phpmyadmin> ” tiến hành tạo một *database* như hình sau: điền tên *database* vào ô *Create new database* rồi ấn *Create*.



Hình 2. 11: Tạo database

- Bắt đầu các thao tác để đưa *web* lên *localhost*
- Chạy đường dẫn <http://localhost/upload/index.php>. Màn hình hiện lên như hình dưới đây.



Hình 2. 12: Bước 1 cài đặt Opencart

- Chọn “ *I agree to the license* ” -> “ *Continue* ”



Hình 2. 13: Bước 2 cài đặt Opencart

Tiếp tục nhấn “ *Continue* ”. Tùy chỉnh các thông số sau:

- *Databasehost: localhost* (để nguyên)
- *User: root*
- *Password* : (để trống)
- *Database name: data* (điền tên *database* vừa tạo lúc nãy)

- *Database prefix*: (để trống)
- *Username*: *admin* (tên truy cập vào trang quản trị)
- *Password*: 123456 (*pass* để đăng nhập trang quản trị)
- E-mail: 121239nguyentu@gmail.com (điền tên Email)

Nhấn “Continue”.

Step 3 - Configuration

1 . Please enter your database connection details.

* Database Host: localhost
 * User: root
 Password:
 * Database Name: data
 Database Prefix:

2. Please enter a username and password for the administration.

* Username: admin
 * Password: 123456
 * E-Mail: 121239nguyentu@

Continue

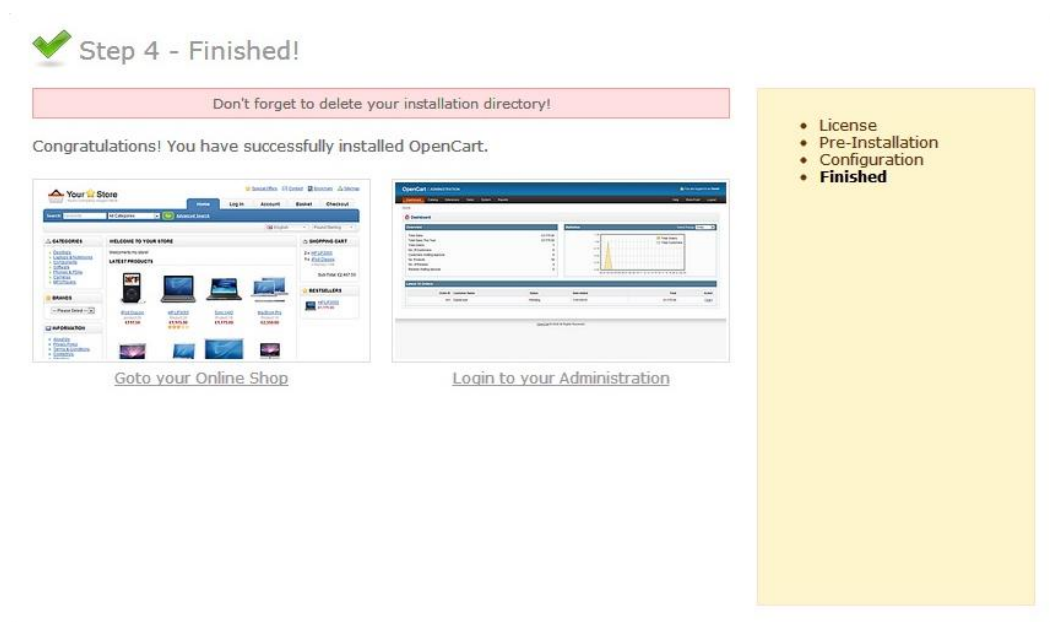
C:\xampp\htdocs\upload\config.php Writable
 C:\xampp\htdocs\upload\admin\config.php Writable

- License
- Pre-Installation
- **Configuration**
- Finished

Hình 2. 14: Bước 3 cài đặt Opencart

Vào “C:\xampp\htdocs\upload” xóa thư mục “Install” hoặc có thể đổi tên *file* thành tên khác.

- Chọn “Goto your Online Shop” để vào shop và chọn “Login to your Administration” để vào trang quản trị.



Hình 2. 15: Bước 4 cài đặt Opencart

2.3.3. Việt hóa Opencart

Tải thư mục ngôn ngữ tiếng Việt về. Giải nén và *copy* các thư mục vừa giải nén được vào thư mục `C:\xampp\htdocs\upload`.

Đăng nhập vào trang quản trị *Admin* bằng cách chạy đường dẫn:

“ <http://localhost/upload/admin> ”

User: admin

Pass : 123456

Chọn *Panel: System/Localisation/Languages/* chọn *Insert* để thêm ngôn ngữ. điền vào các thông tin sau:

Language Name: Vietnamese

Code: vi

Locale: vi_VN.UTF-8,vi_VN,vi-vn,vietnamese

Image: vn.png

Directory: vietnamese

Filename: vietnamese

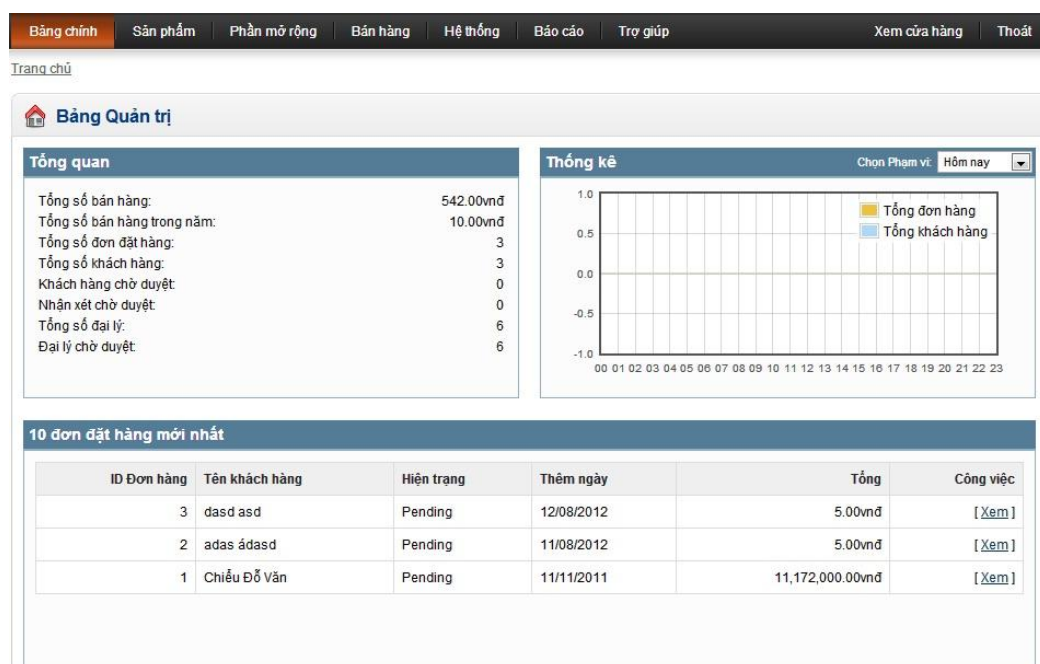
Status: Enabled

Chọn *save* để lưu lại

Vào System/Settings chọn *Edit*, chọn *tab Local: Language: vietnamese* (là ngôn ngữ bên ngoài site).

Administration Language: vietnamese (là ngôn ngữ trang quản lý).

Như vậy toàn bộ trang Web đã được việt hóa.



Hình 2. 16: Giao diện trang admin đã Việt hóa

The screenshot shows the Vietnamese storefront interface. At the top, there is a header with a logo **HD SHOP: 4B/132 AN ĐÃ**, phone number **ĐT: 01288428349**, and a language selector **Ngôn ngữ** with flags for Vietnamese and English. The **Giỏ hàng** (Shopping cart) shows **0 sản phẩm(s) - 0.00vnd**. A search bar **Tìm kiếm** is present. Below the header is a navigation bar with links: **Trang chủ**, **Danh sách yêu thích (0)**, **Tài khoản của tôi**, **Giỏ hàng**, and **Th**.

The main content area features a large product image of a **HD** device. Below it is a **Sản phẩm tiêu biểu** (Featured products) section with four items:

ASUS O!PLAY R1 0.00vnd Thêm vào giỏ	ASUS O!PLAY R3 0.00vnd Thêm vào giỏ	COEX M051 0.00vnd Thêm vào giỏ	COEX M052 0.00vnd Thêm vào giỏ
--	--	---	---

On the right side, there is a **Bán chạy** (Bestselling) section with two items:

HDX1000 0.00vnd Thêm vào giỏ
ASUS O!PLAY R1 0.00vnd Thêm vào giỏ

On the left side, there is a **Danh Mục** (Categories) section with links: **Đầu HD (59)**, **Thiết bị lưu trữ (25)**, **Tivi các loại (0)**, and **Phim HD (0)**. Below it is a **Mới nhất** (Newest) section with a small product image.

Hình 2. 17: Giao diện trang chủ đã Việt hóa

CHƯƠNG 3: ỨNG DỤNG OPENCART

Bài toán

Cửa hàng HD Shop địa chỉ 4B/132 An Đà chuyên cung cấp thiết bị: đầu HD, thiết bị lưu trữ, Tivi các loại, phim HD các loại. Hiện nay cửa hàng có nhu cầu xây dựng *Website* giới thiệu các mặt hàng. Trong mỗi loại gồm có:

- Đầu HD: ASUS, COEX, DATAGE, DUNE, EAGET, XTREAMER, DVICO TVIX, TIZZBIRD, EGREAT, HD LIFE, HDX, HIMEDIA, IAMM, KAIBOER, MEDEBER, POPCORN, TYCO HD, WD.
- Thiết bị lưu trữ: HITACHI, IOMEGA, SASUNG, SEAGATE, WESTERN DIGITAL.
- Tivi các loại.
- Phim HD các loại.

Yêu cầu mỗi loại phải có ảnh đại diện và mô tả cụ thể. Hệ thống phải cho phép khách hàng đặt hàng và gửi đơn đặt hàng cho người quản lý. Trên *web* có các tiện ích như: sản phẩm tiêu biểu, sản phẩm bán chạy, sản phẩm mới nhất, thống kê lượt truy cập... Để xây dựng được *Website* này em sử dụng hệ mã nguồn mở *Opencart*, hiện đang là một hệ mã nguồn rất phổ biến và được hỗ trợ rất nhiều trên các cộng đồng mạng.

3.1. Lý do chọn Opencart

3.1.1. Opencart là mã nguồn mở

Opencart là một mã nguồn mở, sử dụng ngôn ngữ PHP, cung cấp các giải pháp thương mại điện tử mạnh mẽ với khả năng tạo kinh doanh trực tuyến.

Ngoài tính năng *free* thì *Opencart* được biết đến với các tính năng vốn có của một *Opensource*, là có thể sửa đổi, dễ hiệu chỉnh, dễ cải tiến theo hướng phát triển của bản thân người sử dụng. Trong lĩnh vực thiết kế *Website* bán hàng, *Opencart* quản lý hệ thống CMS (*Content Management System*) lớn với phần mua hàng, điểm thưởng. Điều mà các nhà phát triển tâm đắc nhất ở *OpenCart* là chế độ tự *build Theme*, tạo khả năng sáng tạo cho người dùng, cùng với chế độ quản lý điểm cộng, điểm thưởng, quản lý gian hàng cực kì rõ ràng và hiệu quả.

3.1.2. Opencart là một hệ thống giỏ hàng

Như đã nói ở trên, *Opencart* đầu tư khá công phu cho việc sử dụng, quản lý các gian hàng, và khách hàng. Nó chứa các *module* cho phép sử dụng các mục đích khác nhau, như giới thiệu sản phẩm, sản phẩm được đưa ra trưng bày...

3.1.3. Opencart có các tính năng phong phú

Các *Opensource* mỗi loại đều có thế mạnh riêng, không có loại nào hơn loại nào, nhưng từ khi nhà phát triển sử dụng *Opencart* thì thấy ngạc nhiên với các tính năng mà nó đem lại.

- Hệ thống tính điểm thưởng (*Coupon*)
- Thẻ quà tặng
- Xếp hạng các hạng mục sản phẩm
- SEO (*Senior Executive Officer*)
- Tiền tệ

3.1.4. Opencart thân thiện với người dùng

Đầu tiên, hãy xem qua trang quản lý *ADMIN*, các thẻ tab được sắp xếp dưới dạng trượt xuống, thuận tiện chúng được đóng gói lại từng khu vực với nhau, điều này có vừa tốt lại không tốt. Tốt ở việc truy cập nhanh, chỉ bằng các cú rê chuột, điều không tốt ở đây là sự phức tạp, chính sự kéo thả lại là điều khó sử dụng ở *Opencart*. Nhưng người viết cũng đánh giá cao cách bố trí trực quan và hợp lý của *Opensource* này.

Cộng đồng phát triển mạnh (cả miễn phí và thương mại), cộng đồng ngày càng phát triển mạnh ở Việt Nam, vì thế sự hỗ trợ là rất lớn và nhanh chóng. Đặc biệt nhà phát triển có thể tùy chỉnh những gì mình thích, có thể đào sâu nghiên cứu, và không giới hạn tập tin. Người sử dụng biết ít nhiều về *PhotoShop* và một ít lập trình có thể sử dụng được dễ dàng.

Cấu trúc *Opencart* dễ làm theo và hiểu là dễ dàng hơn so với các giải pháp thương mại điện tử. Đó là lý do nó đang phổ biến nhanh chóng.

Opencart là một nguồn giải pháp thương mại điện tử đang mở. Nó được phát triển với mẫu thiết kế MVC. *Opencart* là giải pháp thương mại điện tử mạnh mẽ cho các thương gia *Internet* với khả năng tạo kinh doanh trực tuyến của họ. Mã

Opencart sắp xếp hợp lý và dễ dàng theo dõi hơn so với một số mã nguồn mở công kênh, các giải pháp thương mại điện tử khác như *OsCommerce* và *Zencart*

Hiện nay, *Opencart* hỗ trợ 20 cổng thanh toán và 8 *methods, modules* vận chuyển.

Opencart đang được phổ biến từng ngày.

Các tính năng của Opencart:

- *Opencart* là mã nguồn mở.
- Hệ thống *Templatable* .
- *Opencart* hỗ trợ tải về các sản phẩm.
- *Opencart* mang lại xếp hạng và đánh giá sản phẩm.
- Tối ưu hóa công cụ tìm kiếm.
- Hỗ trợ tính năng tự động thay đổi kích thước hình ảnh.
- Có hỗ trợ cho nhiều cổng thanh toán và các *module* vận chuyển.

3.2. Sử dụng Opencart xây dựng trang web

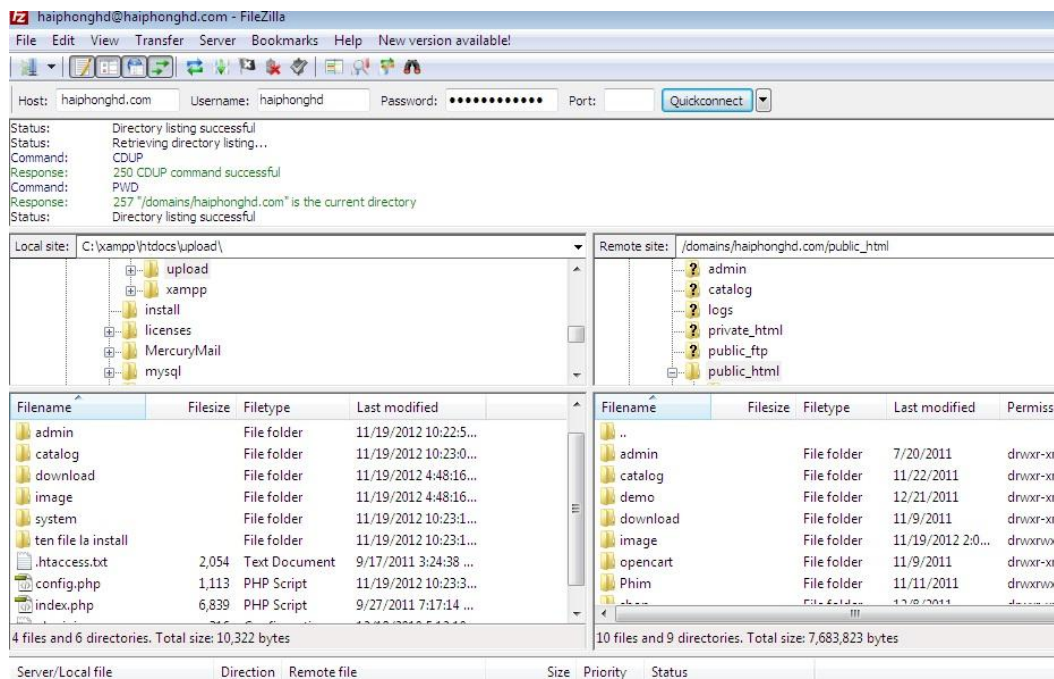
3.2.1 Đăng kí tên miền và Hosting

Đăng ký qua các dịch vụ cung cấp tên miền và hosting như <http://www.matbao.net> với nguyên tắc :

- Tên miền không được vượt quá 63 ký tự, bao gồm cả phần .com, .net, .org.
- Tên miền chỉ bao gồm các ký tự trong bảng chữ cái (a-z), các số (0-9) và dấu trừ (-).
- Các khoảng trắng và các ký tự đặc biệt trong tên miền khác đều không hợp lệ.
- Không thể bắt đầu bằng hoặc kết thúc tên miền bằng dấu trừ (-).
- Tên miền càng ngắn càng tốt, dễ nhớ, không gây nhầm lẫn, khó viết sai.
- Tên miền phải liên quan đến tên chủ thể hoặc lĩnh vực hoạt động của doanh nghiệp.
- Chủ động nộp phí duy trì trước khi tên miền hết thời hạn sử dụng, tự chịu trách nhiệm khi tên miền bị mất do hết hạn sử dụng mà không đóng phí để tránh thời gian gián đoạn sử dụng.

3.2.2. Cài đặt lên host

- Tiến hành *upload* *Opencart* đã lên *host* của <http://haiphonghd.com/>
- Tiến hành cài đặt và sử dụng phần mềm *Filezilla*.
- Nhấn *Filezilla* và tiến hành đăng nhập vào *host* của haiphonghd.



Hình 3. 1: Cài đặt lên host

- Tiến hành *upload* các thư mục của *Opencart* lên *host* bằng cách kéo thả.
- Đăng nhập vào *host* và *import* cơ sở dữ liệu.
- Gõ đường link haiphonghd.com/index.php và làm theo các hướng dẫn để cấu hình hệ thống và cài đặt cơ sở dữ liệu.
- Quá trình cài đặt *Opencart* lên *host* đã thành công.

3.2.3. Cài đặt module(extensions)

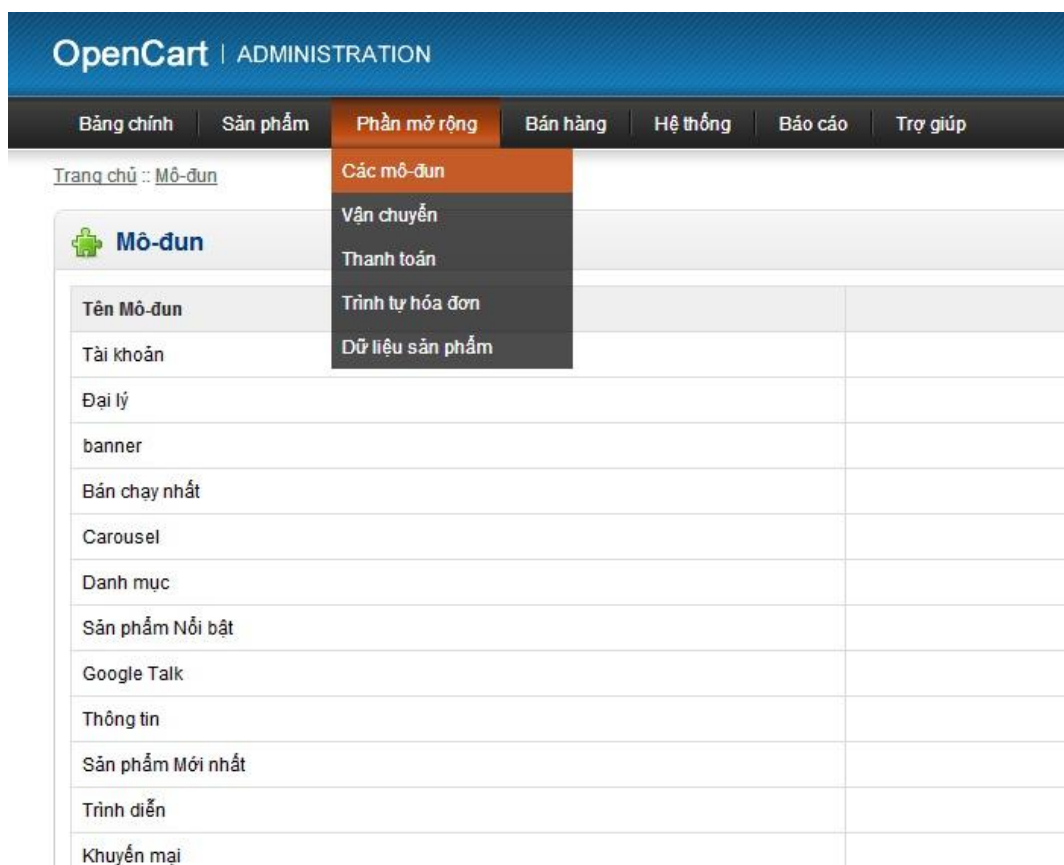
Sau khi cài đặt xong, ta có thể tiến hành thêm các *module* (*extensions*) như sau:

Ví dụ cài đặt *module* thống kê lượt truy cập.

Tải *module* hồng kê lượt truy cập *Visiter Counter* sau đó giải nén.

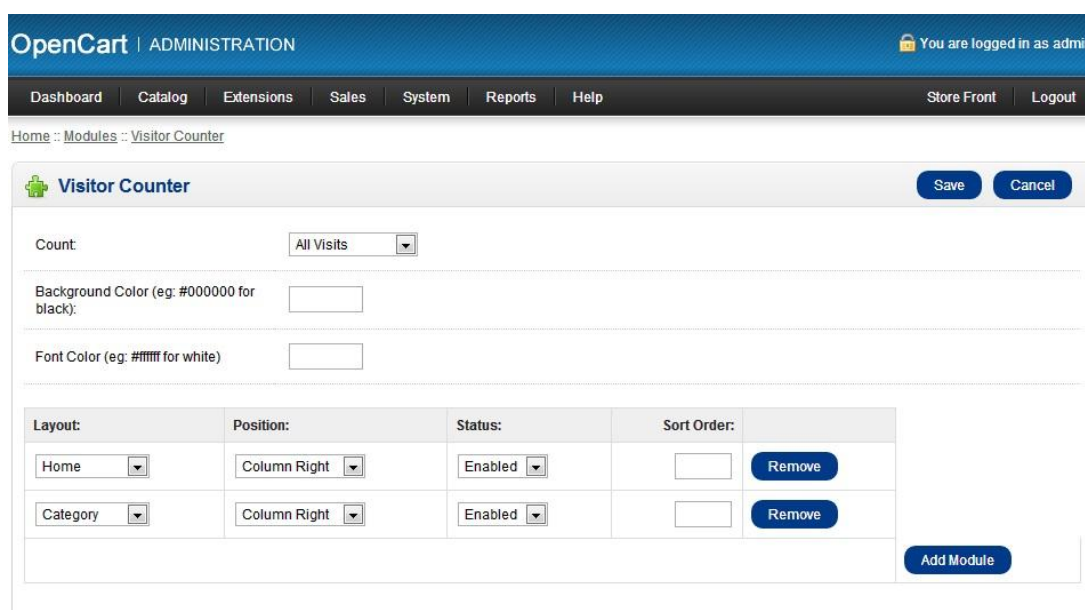
Dùng *Filezilla* để *upload* thư mục thống kê lượt truy cập lên *host* haiphonghd.com

Đăng nhập vào trang quản lý Admin <http://haiphonghd.com/admin>. Nhấn Phần mở rộng/Các mô đun.



Hình 3. 2: Các module

Nhấn “Cài đặt” để cài đặt tương ứng với *module Visitor Counter*. Nhấn “Sửa” để chỉnh sửa *module* thống kê lượt truy cập theo yêu cầu rồi nhấn “Lưu”.



Hình 3. 3: Module khảo sát lượt truy cập

Vậy là *module* thống kê lượt truy cập đã được cài đặt thành công.

3.2.4. Nhập dữ liệu và quản lý đơn đặt hàng.

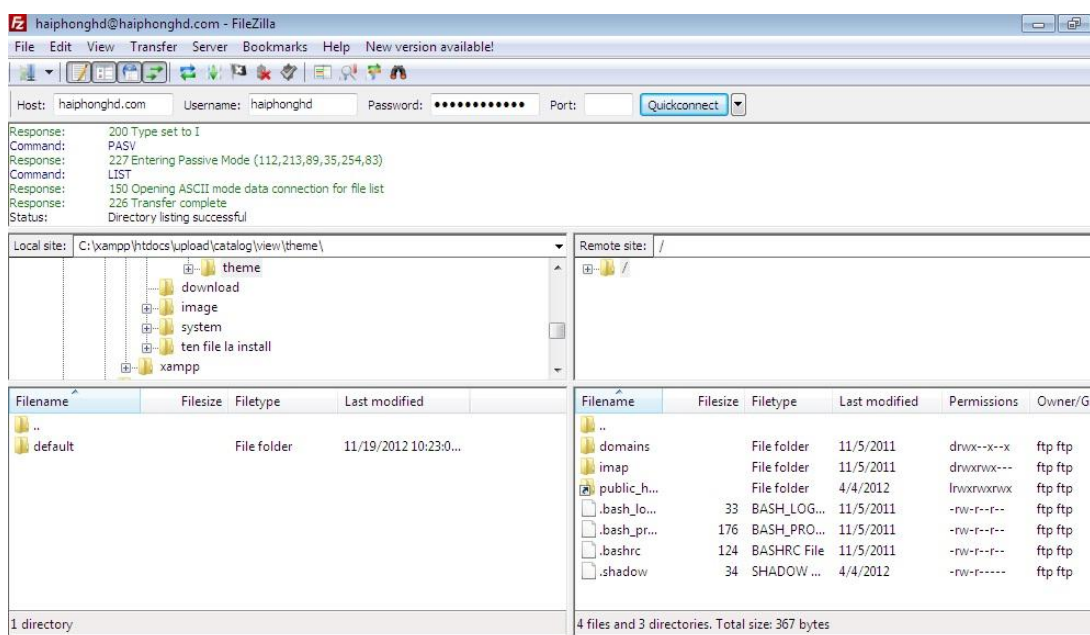
Tiến hành đăng nhập vào trang *admin*.

- Để nhập danh mục sản phẩm nhấn vào *menu* sản phẩm/danh mục, nhấn thêm.
- Để nhập sản phẩm nhấn vào *menu* sản phẩm/sản phẩm, nhấn thêm.
- Để nhập hãng sản xuất vào *menu* sản phẩm/hãng sản xuất, nhấn thêm.
- Quản lý đơn đặt hàng vào *menu* bán hàng/đơn đặt hàng.

3.2.5. Thêm giao diện

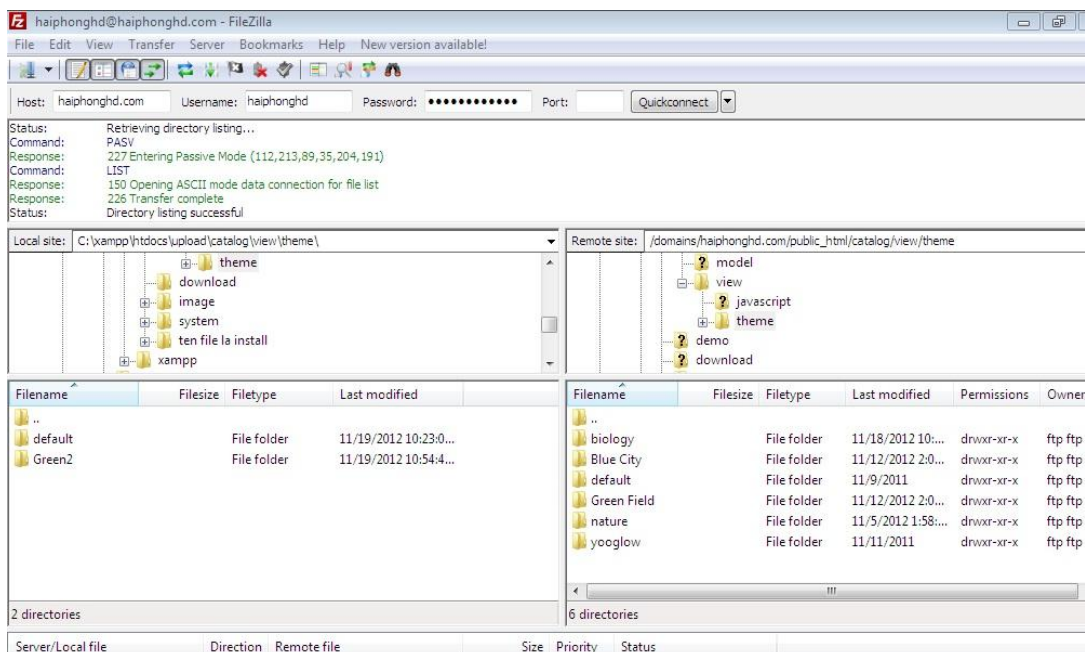
Tiến hành tải một giao diện trên mạng về. Tiến hành giải nén.

Dùng *Firezilla* đăng nhập vào *host* haiphonghd.com.



Hình 3. 4: Đăng nhập vào host

Nhấn vào thư mục *Public_html/catalog/View/theme*.

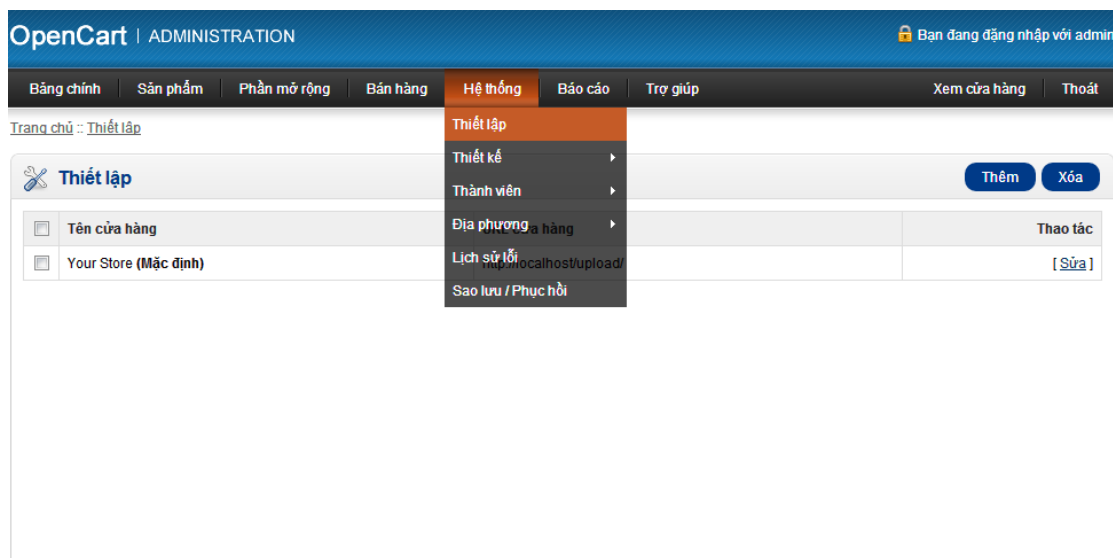


Hình 3. 5: Upload giao diện

Tiến hành kéo thư mục giao diện đã chỉnh sửa vào:

Public_html/catalog/View/theme.

Đăng nhập vào trang quản trị Admin <http://haiphonghd.com/admin>. Nhấn vào menu Hệ thống/Thiết lập.



Hình 3. 6: Thiết lập

Nhấn “Sửa” rồi chọn tab “Cửa hàng”.

Bảng chính Sản phẩm Phần mở rộng Bán hàng Hệ thống Báo cáo Trợ giúp Xem cửa hàng Thoát

Trang chủ :: Thiết lập


Thiết lập Lưu Hủy

Tổng quan Cửa hàng Khu vực Tùy chọn Hình ảnh Email Máy Chủ

* Tiêu đề:

Mô tả từ khóa:

Template:



Giao diện mặc định:

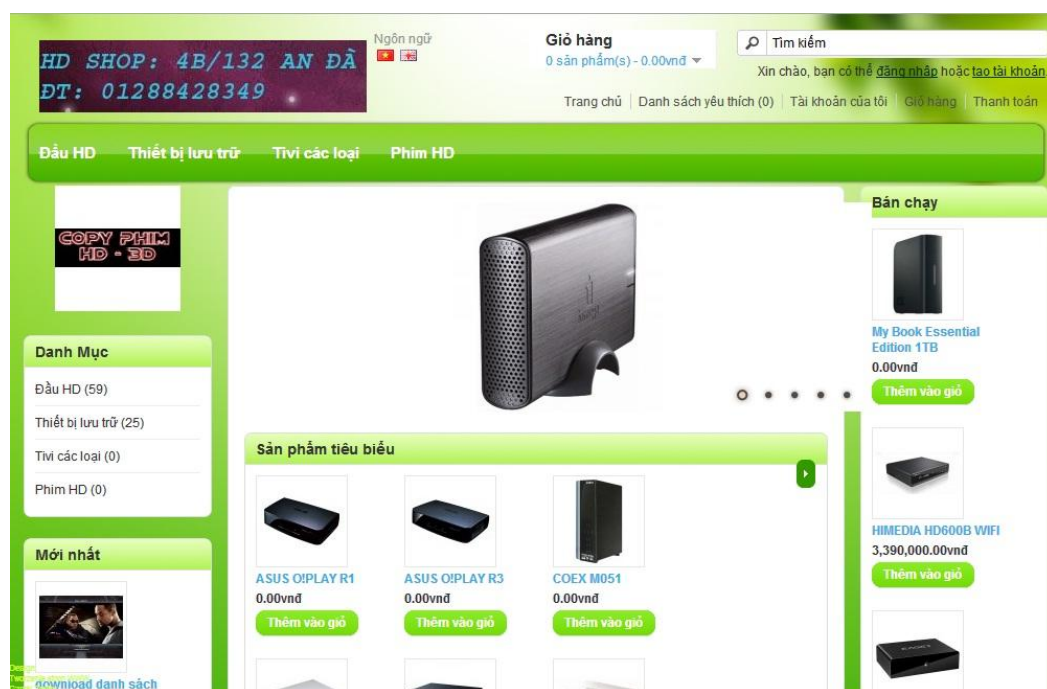
Hình 3. 7: Đổi giao diện

Nhấn chọn *Template* rồi chọn giao diện cần cài đặt. Nhấn “Lưu”.

Nhấn “Xem cửa hàng” để xem giao diện vừa thay đổi.

3.3. Một số kết quả thử nghiệm

Web haiphonghd.com



Hình 3. 8: Giao diện trang chủ giới thiệu sản phẩm

[Trang chủ](#) > [Tai khoản](#) > [Đăng nhập](#)

Đăng nhập tài khoản

Khách hàng mới

Đăng kí tài khoản

Bằng cách tạo tài khoản bạn sẽ có thể mua sắm nhanh hơn, cập nhật tình trạng đơn hàng, theo dõi những đơn hàng đã đặt.

[Tiếp tục](#)

Khách hàng cũ

Tôi là khách hàng cũ

Địa chỉ E-Mail:

Mật khẩu:

[Quên mật khẩu](#)

[Đăng nhập](#)



Hình 3. 9: Giao diện khách hàng đăng nhập

[Trang chủ](#) > [Đầu HD](#) > [ASUS](#)

ASUS

Hiện thị: [Danh sách](#) / [Lưới](#) Sắp xếp theo: [Mặc định](#) Hiện thị: [15](#)

So sánh sản phẩm (0)

	<p>ASUS O!PLAY R1 Bảo Hành : 12 tháng O!Play-R1 là dòng sản phẩm mới ra..</p>	<p>0.00vnđ Trước thuế: 0.00vnđ</p>	<p>Thêm vào giỏ <input type="checkbox"/> Thêm Yêu thích <input type="checkbox"/> Thêm so sánh</p>
	<p>ASUS O!PLAY R3 Bảo hành: 12 tháng R3 là người anh em cùng với R1 chỉ khác với R1 ..</p>	<p>0.00vnđ Trước thuế: 0.00vnđ</p>	<p>Thêm vào giỏ <input type="checkbox"/> Thêm Yêu thích <input type="checkbox"/> Thêm so sánh</p>

Hiện thị 1 đến 2 trong 2 (1 Trang)


Hình 3. 10: Giao diện đặt hàng

HD SHOP
4B/132 An Đà - Hải Phòng
Điện thoại: 01688468555
chieudv@gmail.com
http://haiphonghd.com

Ngày nhập: 11/11/2011
ID Đặt hàng: 1

Tới		Vận chuyển tới (nếu địa chỉ khác nhau)		
Chiều Đỗ Văn TK Company Hoàng Mai Đông Thái Hải Phòng 031 Hai Phong Viet Nam chieudv@gmail.com 0989747794		Chiều Đỗ Văn TK Company Hoàng Mai Đông Thái Hải Phòng 031 Hai Phong Viet Nam		
Sản phẩm	Model	Số lượng	Đơn giá	Tổng
HIMEDIA HD600B WIFI	HIMEDIA HD600B WIFI	1	3,507,000.00vnd	3,507,000.00vnd
EAGET M880 PLUS	EAGET M880 PLUS	1	0.00vnd	0.00vnd
HD PRO X2 PLUS	HD PRO X2 PLUS	1	2,898,000.00vnd	2,898,000.00vnd
download danh sách phim	download danh sách phim	1	0.00vnd	0.00vnd
HDD WD Essential Passport 500GB	HDD WD Essential Passpor	1	4,662,000.00vnd	4,662,000.00vnd
TYCO HD T8	TYCO HD T8	1	0.00vnd	0.00vnd
HDX1000	HDX1000	1	0.00vnd	0.00vnd
Thành tiền::				11,067,000.00vnd
Phí vận chuyển cố định::				105,000.00vnd
Tổng cộng ::				11,172,000.00vnd

Hình 3. 11: Hóa đơn

 **Đặt hàng**

Đơn hàng	ID Đặt hàng:	#1
Chi tiết Thanh toán	Không có hoá đơn.:	[text create invoice no]
Địa chỉ giao hàng	Tên cửa hàng:	HD SHOP
Sản phẩm	Url cửa hàng:	http://haiphonghd.com/
Lịch sử đặt hàng	Khách hàng:	Chiều Đỗ Văn
	Nhóm khách hàng:	Default
	Địa IP:	171.226.87.172
	E-Mail:	chieudv@gmail.com
	Điện thoại:	0989747794
	Tổng:	11,172,000.00vnd
	Tình trạng đặt hàng:	Pending
	Ngày nhập:	11/11/2011
	Ngày thay đổi:	11/11/2011

Hình 3. 12: Giao diện đơn đặt hàng

KẾT LUẬN:

Đồ án xây dựng một trang *Web* bán hàng điện tử bằng *Opencart* cho cửa hàng HD shop lên *Internet*.

Qua quá trình thực hiện đồ án, em đã tổng hợp lại được các kiến thức trong thời gian đã học tại Trường. Đồng thời, em đã tìm hiểu và nắm được quy trình cài đặt và ứng dụng *Opencart*, có thể chủ động trong việc tìm kiếm các thông tin để phục vụ công việc.

Sau khi hoàn thành đồ án, đồ án tập trung xử lý các kỹ thuật liên quan đến thương mại điện tử, trong đó, cho phép đặt yêu cầu dịch vụ trực tuyến. Đây là một mặt hàng đặc biệt trong lĩnh vực thương mại điện tử mà các kỹ thuật về bán hàng trên mạng hiện có chưa đáp ứng được. Đồ án cơ bản đã xử lý hết các chức năng theo yêu cầu của cửa hàng và đã đưa lên mạng thử nghiệm thật và cho kết quả như mong muốn.

Hướng phát triển tiếp theo của đồ án là tích hợp việc thanh toán trực tuyến trên *Internet* thông qua các nhà cung cấp dịch vụ tại Việt Nam như: Onepay, SmartLink,...

TÀI LIỆU THAM KHẢO:

1. Opencart 1.4 Template Design Cookbook.
2. HTTP Pocket Reference by Clinton Wong.
3. HTML_and_CSS,_Design_and_Build_Websites_by_Jon_Duckett.
4. Sams.Ajax.for.Web.Application.Developers.Oct.2006byKris Hadlock.
5. Web Application Architecture: Principles, Protocols and PracticesbyLeon Shklarand Rich Rosen