

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

-----o0o-----



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

**TÌM HIỂU KỸ THUẬT LÀM GIAO DIỆN CHO HỆ
QUẢN TRỊ NỘI DUNG WORDPRESS VÀ ỨNG DỤNG**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

TÌM HIỂU KỸ THUẬT LÀM GIAO DIỆN CHO HỆ QUẢN TRỊ NỘI DUNG WORDPRESS VÀ ỨNG DỤNG

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Trịnh Doãn Khiển

Giáo viên hướng dẫn: ThS Đỗ Văn Chiêu

Mã số sinh viên: 1212101001

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

-----oOo-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Trịnh Doãn Khiển

Mã số: 1212101001

Lớp: CT1601

Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu kỹ thuật làm giao diện cho hệ quản trị nội dung Wordpress và ứng dụng.

NHIỆM VỤ ĐỀ TÀI

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

a. Nội dung:

- Tìm hiểu về hệ quản trị nội dung mã nguồn mở WordPress.
- Cài đặt trên máy website xây dựng từ WordPress.
- Kỹ thuật tạo giao diện người dùng cho WordPress.
- Quản trị nội dung trang tin trên WordPress.
- Đưa website lên Internet.

b. Các yêu cầu cần giải quyết

- Hiểu được mục tiêu và cách thức thực hiện.
- Xây dựng được một website với giao diện riêng và đưa lên Internet.

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: Đỗ Văn Chiểu

Học hàm, học vị: Thạc sĩ

Cơ quan công tác: Đại học Dân lập Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu về hệ quản trị nội dung mã nguồn mở WordPress.
- Cài đặt trên máy website xây dựng từ WordPress.
- Kỹ thuật tạo giao diện người dùng cho WordPress.
- Quản trị nội dung trang tin trên WordPress.
- Đưa website lên Internet.

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 18 tháng 04 năm 2016

Yêu cầu phải hoàn thành trước ngày 9 tháng 07 năm 2016

Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N
Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngàytháng.....năm 2016

HIỆU TRƯỞNG

GS.TS.NGŨT Trần Hữu Nghị

PHẦN 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 2016

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

**PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHĂM 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 2016
Cán bộ chăm phản biện
(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết em xin bày tỏ lòng biết ơn đối với Th.S Đỗ Văn Chiêu – Bộ môn Công nghệ thông tin – Trường Đại học Dân Lập Hải Phòng, người đã dành cho em rất nhiều thời gian quý báu, trực tiếp hướng dẫn tận tình giúp đỡ, chỉ bảo em trong suốt quá trình làm luận văn tốt nghiệp.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong Bộ môn Công nghệ thông tin - Trường ĐHDL Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường, đã đọc và phản biện luận văn của em giúp em hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành luận án này.

Em xin cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, Ban giám hiệu nhà trường, Bộ môn tin học, các Phòng ban nhà trường đã tạo điều kiện tốt nhất trong suốt thời gian học tập và làm tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình học tập, trong thời gian thực tập cũng như trong quá trình làm luận văn nhưng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy giáo, cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải Phòng, Ngày ... tháng ... năm 2016

Sinh viên

Trịnh Doãn Khiển

MỤC LỤC

LỜI CẢM ƠN	1
DANH MỤC HÌNH	4
GIỚI THIỆU	6
CHƯƠNG 1: HỆ QUẢN TRỊ NỘI DUNG WORDPRESS	7
1.1. Giới thiệu mã nguồn mở	7
1.1.1 Khái niệm mã nguồn mở	7
1.1.2 Phân loại phần mềm nguồn mở	8
1.1.3 Lợi ích mã nguồn mở	8
1.1.4 Một số loại mã nguồn mở thường gặp	8
1.2. Tìm hiểu về WordPress	11
1.2.1 Giới thiệu về hệ quản trị nội dung WordPress	11
1.2.2 Các giai đoạn phát triển của WordPress	12
1.2.3 Những thành tựu của WordPress	14
1.2.4 Những nét nổi bật của WordPress	15
CHƯƠNG 2: CÀI ĐẶT TRÊN MÁY WEBSITE ĐƯỢC XÂY DỰNG TỪ WORDPRESS	17
2.1 Những yêu cầu cài đặt cho WordPress	17
2.2 Cài đặt Wordpress trên Localhost	17
CHƯƠNG 3: KỸ THUẬT TẠO GIAO DIỆN NGƯỜI DÙNG TRONG WORDPRESS	25
3.1 Khái niệm cơ bản về giao diện người dùng trong WordPress (Theme Wordpress)	25
3.2. Cấu trúc của Theme WordPress	26
3.2.1 Cấu trúc thư mục Theme WordPress	26
3.2.2 Theme Stylesheet	27
3.2.3 Tệp tin tùy chọn chức năng	27
3.2.4 Template files	28
3.2.6 Vòng lặp (Loop)	32
3.3. Thiết kế theme WordPress	33
3.3.1 Ý tưởng thiết kế Theme	33
3.3.2 Tạo cấu trúc theme	34
3.3.3 Viết code cho tệp tin function.php	35

3.3.4	Viết code cho tệp tin header.php.....	38
3.3.5	Viết code cho tệp tin footer.php.....	40
3.3.6	Viết code cho tệp tin index.php	42
3.3.7	Viết code cho tệp tin content.php	43
3.3.8	Viết code cho Post Format.....	46
3.3.9	Code cho tệp tin single.php và page.php	48
3.2.10	Viết code cho các trang lưu trữ, trang tìm kiếm và trang 404	49
3.3.11	Viết code cho tệp tin sidebar.php.....	52
3.3.12	Viết CSS cho theme	52
CHƯƠNG 4: QUẢN TRỊ NỘI DUNG TRANG TIN TRÊN WORDPRESS		58
4.1.	Các thành phần cơ bản trong trang quản trị WordPress	58
4.1.1	Truy cập vào trang quản trị WordPress	58
4.1.2	Các thành phần trong DashBoard	59
4.2.	Quản trị website WordPress	60
4.2.1	Đăng bài viết	60
4.2.2	Tạo Page.....	63
4.2.3	Thiết lập và quản lý bình luận.....	64
4.2.4	Hướng dẫn sử dụng Widget	66
4.2.5	Làm việc với Menu trong WordPress	68
4.2.6	Hướng dẫn Plugin trong WordPress	72
4.2.7	Quản trị người dùng (users) trên WordPress	74
4.2.8	Hướng dẫn mục Cài đặt	76
KẾT LUẬN		83
TÀI LIỆU THAM KHẢO.....		.84

DANH MỤC HÌNH

Hình 1.2.1: WordPress thuở mới thành lập vào năm 2003.....	11
Hình 2.2.1: Giao diện XAMPP.....	17
Hình 2.2.2: Giao diện phpMyadmin.....	18
Hình 2.2.3: Giao diện Database trong phpMyadmin.....	18
Hình 2.2.4 Thư mục sau khi giải nén mã nguồn.....	18
Hình 2.2.5: Các tệp tin và thư mục mã nguồn của WordPress.....	19
Hình 2.2.6: Mã nguồn wordpress trong thư mục website.....	19
Hình 2.2.7: Giao diện Database trong phpMyadmin.....	20
Hình 2.2.8: Tạo user account.....	20
Hình 2.2.9: Tạo database cùng với user.....	21
Hình 2.2.10: Chọn ngôn ngữ khi cài đặt WordPress.....	21
Hình 2.2.11: Wordpress yêu cầu nhập thông tin database.....	22
Hình 2.2.12: Nhập thông tin database.....	22
Hình 2.2.13: Thiết các thông tin quan trọng của website.....	23
Hình 2.2.14: Cài đặt website thành công.....	23
Hình 2.2.15: Trang quản trị của WordPress sau khi cài đặt.....	24
Hình 2.2.16: Trang chủ website sau khi cài WordPress.....	24
Hình 3.2.1: Cấu trúc template trong Theme WordPress.....	28
Hình 3.2.2: Bố cục website dự kiến.....	33
Hình 3.2.3: Thư mục chứa theme.....	35
Hình 3.2.4: Theme vừa tạo trong Giao diện của trang Quản trị.....	35
Hình 3.2.5: Thêm Menu cho website.....	40
Hình 3.2.6: Theme Widget cho sidebar.....	52
Hình 3.4.1: Trang chủ website Công nghệ 24h.....	56
Hình 3.4.2: Một bài viết trong website Công nghệ 24h.....	57
Hình 4.1.1: Đăng nhập vào trang quản trị.....	58
Hình 4.1.2: Trang Quản trị (DashBoard).....	59
Hình 4.2.1: Tạo bài viết mới.....	61
Hình 4.2.2: Khung soạn thảo bài viết.....	61
Hình 4.2.3: Thêm thẻ và chuyên mục cho bài viết.....	62
Hình 4.2.4: Thêm ảnh đại diện cho bài viết.....	62
Hình 4.2.5: Chọn ảnh đại diện cho bài viết.....	63
Hình 4.2.6: Đăng bài viết.....	63
Hình 4.2.7: Thêm trang mới.....	64
Hình 4.2.8: Khung bình luận trong website.....	64
Hình 4.2.9: Bình luận chờ xét duyệt.....	65
Hình 4.2.10: Duyệt bình luận.....	65
Hình 4.2.11: Hình thị bình luận không cần chờ duyệt.....	66
Hình 4.2.12: Cấm bình luận nhạy cảm.....	66
Hình 4.2.13: Quản lý widget.....	67
Hình 4.2.14: Thêm Widget.....	67
Hình 4.2.15: Loại bỏ widget.....	68
Hình 4.2.16: Thiết lập widget.....	68
Hình 4.2.17: Giao diện Menu.....	69

<i>Hình 4.2.18: Tạo Menu</i>	69
<i>Hình 4.2.19: Thêm Trang vào Menu</i>	70
<i>Hình 4.2.20: Thêm Liên kết vào Menu</i>	70
<i>Hình 4.2.21: Thêm Chuyên mục vào Menu</i>	71
<i>Hình 4.2.22: Tạo Menu Location</i>	71
<i>Hình 4.2.23: Menu xuất hiện trên trang chủ</i>	72
<i>Hình 4.2.23: Cài plugin mới</i>	73
<i>Hình 4.2.24: Danh sách plugin</i>	73
<i>Hình 4.2.25: Giao diện cài đặt Plugin</i>	74
<i>Hình 4.2.26: Giao diện thêm người dùng mới</i>	75
<i>Hình 4.2.27: Danh sách người dùng</i>	75
<i>Hình 4.2.28: Cài đặt tổng quan</i>	76
<i>Hình 4.2.29: Cài đặt Viết</i>	78
<i>Hình 4.2.30: Cài đặt đọc</i>	79
<i>Hình 4.2.31: Trang khu vực Cài đặt -> thảo luận</i>	80
<i>Hình 4.2.32: Trang khu vực tùy chỉnh Media</i>	81
<i>Hình 4.2.33: Trang khu vực Cài đặt -> Đường dẫn tĩnh</i>	82

GIỚI THIỆU

Phần mềm nguồn mở là phần mềm với mã nguồn được công bố và sử dụng một giấy phép nguồn mở. Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi, cải tiến phần mềm, sao chép và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi mà không phải trả tiền bản quyền cho người lập trình trước.

Thuật ngữ "phần mềm nguồn mở" có nghĩa gần tương đương với "mã nguồn mở" nhưng với độ bao hàm cao hơn. Ngày nay có rất nhiều dạng mở (không đóng) bao gồm: phần cứng, phần mềm nguồn mở, tài liệu/học liệu mở, thiết kế mở... Phần mềm nguồn mở đang phát triển với tốc độ khá cao, cho thấy nó có nhiều động lực hơn so với mã đóng.

Ở Việt Nam mã nguồn mở cũng không còn xa lạ. Việc sử dụng và phát triển mã nguồn mở ở đất nước đang phát triển như nước ta rất được quan tâm, nó giúp giảm thiểu nguồn tài chính khi chi trả bản quyền và nhiều lợi ích khác. Hiện nay, có rất nhiều sản phẩm nguồn mở đang được sử dụng như: WordPress, Joomla, Magento, NukeViet,... Trong đó, hệ quản trị nội dung mã nguồn mở WordPress đang được quan tâm và sử dụng rộng rãi bởi tính đơn giản, dễ sử dụng và đặc biệt là hoàn toàn miễn phí. Khó khăn duy nhất trong việc sử dụng WordPress làm website là việc tạo giao diện theo yêu cầu người dùng. Đồ án này của em sẽ nghiên cứu kỹ thuật tạo giao diện người dùng cho WordPress, đưa website lên trên Internet và quản trị nội dung của website.

CHƯƠNG 1: HỆ QUẢN TRỊ NỘI DUNG WORDPRESS

1.1. Giới thiệu mã nguồn mở

1.1.1 Khái niệm mã nguồn mở

Mã nguồn mở (Open Source) là thuật ngữ chỉ các sản phẩm phần mềm có mã nguồn mở. Phần mềm nguồn mở là các phần mềm được cung cấp dưới dạng cả mã và nguồn, không chỉ là miễn phí về giá mua mà chủ yếu là miễn phí về bản quyền. Người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung quy định trong giấy phép phần mềm nguồn mở mà không cần xin phép ai, điều mà họ không được làm đối với các phần mềm đóng.

Nhà cung cấp phần mềm nguồn mở có quyền yêu cầu người dùng trả một số chi phí về dịch vụ bảo hành, huấn luyện, nâng cấp, tư vấn... tức là những dịch vụ thực sự đã thực hiện để phục vụ người dùng, nhưng không được bán các sản phẩm nguồn mở vì nó là tài sản của trí tuệ chung.

Một điều kiện hay được áp dụng nhất là GPL: GNU General Public Licence của tổ chức Free Software Foundation.

GPL có hai đặc điểm phân biệt đó là:

- Tác giả gốc giữ bản quyền về phần mềm nhưng cho phép người quyền tìm hiểu, phát triển, công bố cũng như khai thác thương mại sản phẩm...
- Tác giả sử dụng luật bản quyền để bảo đảm các quyền đó không bị vi phạm đối với tất cả mọi người, trên mọi phần mềm có sử dụng mã nguồn mở của mình.

Các điều khoản phân phối của phần mềm mã nguồn mở phải tuân thủ các tiêu chuẩn sau:

- Tự do tái phân phối.
- Mã nguồn.
- Các chương trình phát sinh.
- Tính toàn vẹn của mã nguồn cung cấp bởi tác giả.
- Không có sự phân biệt đối xử giữa cá nhân hay nhóm người.
- Không phân biệt đối xử với bất cứ một lĩnh vực công việc nào.
- Việc phân phối bản quyền.

- Giấy phép phải không được giành riêng cho một sản phẩm.
- Bản quyền không được cản trở các phần mềm khác.

1.1.2 Phân loại phần mềm nguồn mở

- **Application**

Là các loại phần mềm ứng dụng, ví dụ như hệ điều hành, phần mềm văn phòng, phần mềm kế toán, phần mềm chỉnh sửa hình ảnh, công cụ lập trình (IDE), Web server,... Ví dụ:

- + Hệ điều hành: Windows, MacOS, Linux, Free BSD.
- + Phần mềm văn phòng: Open Office, King Office.
- + Công cụ lập trình: Adobe Dreamweaver, phpDesign.

- **Software Framework**

Là tập hợp những phần mềm (Software Package) giúp cho lập trình viên dùng để viết phần mềm nhanh hơn, khỏi phải viết lại code cho những vấn đề đã có người viết rồi.

1.1.3 Lợi ích mã nguồn mở

- Phần mềm có thể được sao chép hoàn toàn miễn phí, người dùng hoàn toàn an tâm khi chia sẻ một chương trình với bạn bè.
- Các định dạng file không hoàn toàn bị kiểm soát bởi một vài nhà cung cấp. Do yêu cầu công việc, người dùng muốn sử dụng dữ liệu cho một ứng dụng khác nhưng chương trình bản quyền không cho phép, với Open Source người dùng có thể gặp nhiều nhà cung cấp đáp ứng yêu cầu của mình.
- Hầu hết các sản phẩm Open Source đều có khả năng bảo mật tuyệt vời, khi một vết nứt được tìm thấy nó thường được sửa nhanh hơn phần mềm có bản quyền.
- Các hệ thống Open Source, nhất là các hệ thống dựa trên UNIX, thường rất linh hoạt vì chúng được xây dựng từ nhiều khối thống nhất và được miêu tả cặn kẽ, rất dễ để người dùng thay thế nhiều phần của hệ thống với phần có giao diện tương tự.
- Có một cộng đồng hỗ trợ lớn, không bị phụ thuộc vào bất kì công ty nào.

1.1.4 Một số loại mã nguồn mở thường gặp

Hiện nay, trên thế giới có rất nhiều loại nguồn mở (hay còn gọi là mã nguồn mở) khác nhau, mỗi loại nguồn mở có một ứng dụng riêng. Dưới đây là một số loại nguồn mở được sử dụng rộng rãi:

- Ubuntu: là hệ điều hành mã nguồn mở dùng cho máy tính xách tay, máy tính để bàn và cả máy chủ, Ubuntu chứa tất cả các chương trình ứng dụng cần thiết cho công việc tại nhà, ở trường hay tại văn phòng công ty.
- Vbulletin: là mã nguồn mở xây dựng các diễn đàn trực tuyến được sử dụng phổ biến nhất hiện nay. Hầu hết các diễn đàn lớn tại Việt Nam đều được xây dựng bằng Vbulletin.
- Apache Tomcat: là hệ thống mã nguồn mở của hãng Apache Software, ứng dụng Apache Tomcat có thể xử lý được số lượng lớn các yêu cầu bao gồm ứng dụng web trực tuyến, các gói dữ liệu lưu thông giữa Server – Client, tùy biến dễ dàng theo nhu cầu, quy mô của từng tổ chức, doanh nghiệp.
- Linux: là một hệ điều hành mã nguồn mở dạng Unix được xây dựng bởi Linus Torvalds. Linux có mọi đặc tính của một hệ điều hành hiện đại: hệ thống đa nhiệm, đa tuyến đoạn, bộ nhớ ảo, thư viện độ, tải theo nhu cầu, quản lý bộ nhớ, các module driver thiết bị, video frame buffering, và mạng internet bằng giao thức TCP/IP.
- Open Office: là chương trình mã nguồn mở thay thế cho Microsoft Office, ưu điểm của nó là dung lượng nhỏ và có thể tương thích được với Microsoft Office nhưng chưa thể thân thiện bằng Microsoft Office.
- WordPress: là một hệ quản trị nội dung mã nguồn mở được ứng dụng để thiết kế các website hay blog cá nhân.
- Ngoài ra còn rất nhiều loại mã nguồn mở khác như: Eclipse, Webwork, WebGUI, OpenCMS, Fedora, Centos, Joomla, Xenforo,...

Trên thị trường phần mềm, có nhiều loại giấy phép. Có thể chia các giấy phép này thành các loại sau:

- Phần mềm thương mại (Commercial Software). Là phần mềm thuộc bản quyền của tác giả hoặc nhà sản xuất, chỉ được cung cấp ở dạng mã nhị phân, người dùng phải mua và không có quyền phân phối lại.

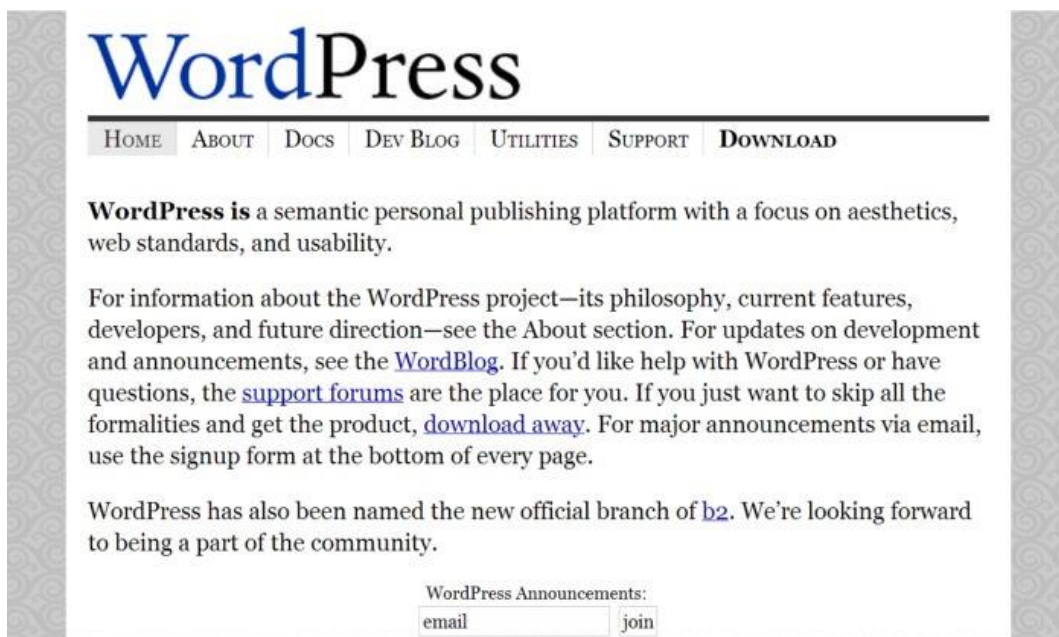
- Phần mềm thử nghiệm giới hạn (Limited Trial Software). Là những phiên bản giới hạn của các phần mềm thương mại được cung cấp miễn phí nhằm mục đích thử nghiệm, giới thiệu sản phẩm và kích thích người dùng quyết định mua. Loại phần mềm này bị giới hạn về tính năng và thời gian sử dụng (thường là 60 ngày).
- Phần mềm “chia sẻ” (Shareware). Loại phần mềm này có đủ các chức năng và được phát hành tự do, nhưng khuyến cáo các cá nhân hoặc tổ chức mua, tùy tình hình cụ thể. Nhiều tiện ích Internet (như “WinZip” dùng các thuận lợi của Shareware như một hệ thống phân phối).
- Phần mềm sử dụng phi thương mại (Non-commercial Use). Loại phần mềm này được sử dụng tự do và có thể phát hành lại bởi các tổ chức phi lợi nhuận. Nhưng các tổ chức kinh tế, thí dụ các doanh nghiệp,... muốn dùng phải mua. Netscape Navigator là một thí dụ của loại phần mềm này.
- Phần mềm không phải trả phần trăm cho nhà sản xuất (Royalties Free Binaries Software). Phần mềm được cung cấp dưới dạng nhị phân và được dùng tự do. Ví dụ: Bản nhị phân của các phần mềm Internet Explorer và NetMeeting.
- Thư viện phần mềm không phải trả phần trăm (Royalties Free Software Libraries) là những phần mềm mà mã nhị phân cũng như mã nguồn được dùng và phân phối tự do, nhưng người dùng không được phép sửa đổi. Ví dụ: các thư viện lớp học, các tệp “header”.
- Phần mềm mã nguồn mở kiểu BSD - (Open Source BSD-style). Một nhóm nhỏ khép kín (closed team) đã phát triển các PMNM theo giấy phép phân phối Berkely (BSD – Berkely Software Distribution) cho phép sử dụng và phân phối lại các phần mềm này dưới dạng mã nhị phân và mã nguồn. Về nguyên tắc nhóm phát triển không cho phép người dùng tự do lấy mã nguồn từ kho mã ra sửa (gọi là check-out) và đưa mã đã sửa vào lại kho mã mà không được họ kiểm tra trước (gọi là các “check-in”).
- Phần mềm mã mở kiểu Apache (Open Source Apache-style). Chấp nhận nguồn mở kiểu BSD nhưng cho phép những người ngoài nhóm phát triển xâm nhập vào lõi của mã nền (core codebase), tức là được phép thực hiện các “check-in”.

- Phần mềm mã mở kiểu CopyLeft hay kiểu Linux (Open Source CopyLeft, Linux-style). Phần mềm mã mở kiểu CopyLeft (trò chơi chữ của Free Software Foundation – FSF – và GNU – Gnu’s Not Unix, để đối nghịch hoàn toàn với CopyRight!) hay còn gọi là giấy phép GPL (General Public Licence) là một bước tiến quan trọng theo hướng tự do hóa của các giấy phép phần mềm. Giấy phép GPL yêu cầu không những mã nguồn gốc phải được phân phối theo các qui định của GPL mà mọi sản phẩm dẫn xuất cũng phải tuân thủ GPL.

1.2. Tìm hiểu về WordPress

1.2.1 Giới thiệu về hệ quản trị nội dung WordPress

WordPress là một phần mềm nguồn mở (*Open Source Software*) được viết bằng ngôn ngữ lập trình website PHP (Hypertext Preprocessor) và sử dụng hệ quản trị cơ sở dữ liệu MySQL. WordPress được ra mắt lần đầu tiên vào ngày 27/5/2003 bởi tác giả Matt Mullenweg và Mike Little.



Hình 1.2.1: WordPress thuở mới thành lập vào năm 2003

WordPress là một mã nguồn mở bằng ngôn ngữ PHP để hỗ trợ tạo blog cá nhân được rất nhiều người sử dụng ủng hộ về tính dễ sử dụng, nhiều tính năng hữu ích. Qua thời gian, số lượng người sử dụng tăng lên, các cộng tác viên là những lập trình viên cũng tham gia đông đảo để phát triển mã nguồn WordPress có thêm những tính năng tuyệt vời. Và cho đến năm 2016, WordPress đã được xem như là một hệ quản trị nội dung (CMS – Content Management System) vượt trội để hỗ trợ người dùng tạo ra

nhiều thể loại website khác nhau như blog, website tin tức/tạp chí, giới thiệu doanh nghiệp, bán hàng – thương mại điện tử, thậm chí với các loại website có độ phức tạp cao như đặt phòng khách sạn, thuê xe, đăng dự án bất động sản,... Hầu như mọi hình thức website với quy mô nhỏ và vừa đều có thể triển khai trên nền tảng WordPress. Nhưng như thế không có nghĩa là WordPress chỉ thích hợp với các dự án nhỏ, mà hiện nay có tới khoảng 25% website trong danh sách 100 website lớn nhất thế giới sử dụng mã nguồn WordPress. Ví dụ như trang tạp chí TechCrunch, Mashable, CNN, BBC America, Variety, Sony Music, MTV News, Bata, Quartz,...

1.2.2 Các giai đoạn phát triển của WordPress

- **2003:** WordPress được ra mắt lần đầu bởi Matt Mullenweg và Mike.
- **2004:** Plugin được giới thiệu với phiên bản **1.2 (Mingus)**.
- **2005:**
 - + Hệ thống theme và các trang tĩnh được giới thiệu ở phiên bản **1.5 (Strayhorn)**.
 - + Bộ nhớ đệm tồn tại lâu (persistent caching) và một giao diện người dùng backend mới được ra mắt trong phiên bản **2.0 (Duke)**.
- **2007:**
 - + Một giao diện người dùng mới, tự động lưu, chính tả kiểm tra và các tính năng mới khác được giới thiệu trong phiên bản **2.1 (Ella)**.
 - + Widget, hỗ trợ Atom tốt hơn và tối ưu hóa tốc độ xuất hiện trong phiên bản **2.2 (Getz)**.
 - + Tagging, cập nhật các thông báo và một hệ thống phân loại (taxonomy) mới đã được giới thiệu trong phiên bản **2.3 (Dexter)**.
- **2008:**
 - + Phiên bản **2.5 (Breckner)** đã được phát hành với giao diện người dùng (UI) quản trị mới được thiết kế bởi Happy Cog và giới thiệu bảng điều khiển và API shortcode.
 - + Phiên bản **2.6 (Tyner)** được xây dựng trên 2.5 và giới thiệu tính năng revisions post (sửa bài viết).
 - + Phiên bản **2.7 (Coltrane)** được ra mắt với giao diện quản lý mới để cải thiện khả năng sử dụng và thực hiện các công cụ quản trị tùy biến hơn. Phiên bản 2.7 cũng giới thiệu tính năng nâng cấp tự động, tích hợp cài đặt plugin, sticky post,

luồng bình luận (comment), phân trang và một API mới quản lý số lượng lớn, và các tài liệu nội tuyến.

- **2009:**

- + Phiên bản **2.8 (Baker)** giới thiệu một giao diện được cài sẵn và cải thiện giao diện người dùng và API.
- + Phiên bản **2.9 (Carmen)** giới thiệu tính năng chỉnh sửa hình ảnh, Trash/ Undo, số lượng lớn các plugin được cập nhật và hỗ trợ oEmbed.

- **2010:** Phiên bản **3.0 (Thelonious)** được phát hành chính thức. Phiên bản này giới thiệu tính năng phân loại bài viết, thêm quản lý menu, giới thiệu một giao diện mặc định mới với tên là "Twenty ten" và cho phép quản lý nhiều trang web (MultiSite).

- **2011:**

- + Phiên bản **3.1 (Gershwin)** giới thiệu post format và admin bar.
- + Phiên bản **3.2 (Reinhardt)** cải thiện WordPress nhanh hơn và nhẹ hơn, phiên bản này nâng cấp yêu cầu tối thiểu để PHP 5.2.4 và MySQL 5.0.15 và giới thiệu một giao diện mặc định mới với tên là: "Twenty Eleven".
- + Phiên bản **3.3 (Sonny)** ra mắt giúp cho WordPress thân thiện hơn cho người mới bắt đầu với tin nhắn chào mừng và con trỏ tính năng.

- **2012:**

- + Phiên bản **3.4 (Green)** giới thiệu các tùy biến giao diện và xem trước giao diện.
- + Phiên bản **3.5 (Elvin)** giới thiệu hệ thống quản lý media mới và một giao diện mặc định mới: "Twenty Twelve".

- **2013:**

- + Phiên bản **3.6 (Peterson)** giới thiệu một giao diện mặc định mới với tên là "Twenty Thirteen", hỗ trợ Audio và Video dựng sẵn, cải thiện tính năng Auto Save và Post Locking.
- + Phiên bản **3.7 (Basie)** giới thiệu bản cập nhật tự động để bảo trì và cập nhật bảo mật, đồng hồ đo độ mạnh mật khẩu, kết quả tìm kiếm được cải thiện và hỗ trợ toàn cầu tốt hơn.
- + Phiên bản **3.8 (Parker)** giới thiệu thiết kế quản trị mới và giao diện mặc định mới: "Twenty Fourteen".

- **2014:**
 - + Phiên bản **3.9 (Smith)** đã cải thiện media, giới thiệu live widget và xem trước tiêu đề.
 - + Phiên bản **4.0 (Benny)** giới thiệu tính năng “grid view” cho các thư viện media và xem trước hình ảnh cho nội dung nhúng.
 - + Phiên bản **4.1 (Dinah)** giới thiệu tính năng cài đặt ngôn ngữ từ màn hình Settings và một giao diện mặc định mới rất đẹp với tên: "Twenty Fifteen".
- **2015:**
 - + Phiên bản **4.2 (Powell)** thêm hỗ trợ biểu tượng cảm xúc, thêm hỗ trợ ký tự mở rộng và chuyển mã hóa cơ sở dữ liệu từ utf8 thành utf8-mb4.
 - + Phiên bản **4.3 (Billie)** thêm hỗ trợ các biểu tượng trang web dựng sẵn và giới thiệu định dạng phím tắt trong trình soạn thảo văn bản.
 - + Phiên bản **4.4 (Clifford)** thêm hình ảnh, bài nhúng và một giao diện mặc định mới: "Twenty Sixteen".
- **2016:** Hiện tại, phiên bản **4.5 (Coleman)** thêm liên kết nội tuyến, thêm phím tắt định dạng và xem trước Responsive trong Customizer.

1.2.3 Những thành tựu của WordPress

Wordpress có những thành tựu rất vượt bậc và là một mã nguồn CMS mở phổ biến nhất hành tinh. Những thành tựu đáng kể nhất là:

- Trên thế giới, có khoảng 25 bài viết được đăng lên các website sử dụng WordPress mỗi giây.
- Số lượng website làm bằng WordPress chiếm 23% tổng số lượng website trên thế giới.
- Trong số 100% các website sử dụng mã nguồn CMS, WordPress chiếm 75%.
- Phiên bản WordPress 4.0 đạt hơn 16 triệu lượt tải chỉ sau khoảng hai tháng.
- Mã nguồn WordPress hiện đang có khoảng 785 lập trình viên cùng hợp tác phát triển.
- Chỉ tính các giao diện (hay còn gọi là theme) miễn phí trên thư viện WordPress thì đã có hơn 2.700 themes khác nhau.

1.2.4 Những nét nổi bật của WordPress

- Hệ thống Plugin phong phú và không ngừng cập nhật, ngoài ra người dùng có thể viết Plugin hoặc tích hợp code vào Wordpress.
- Được phát triển bằng nhiều ngôn ngữ (hỗ trợ tiếng Việt).
- Cập nhật phiên bản liên tục, cộng đồng hỗ trợ lớn.
- Có hệ thống Theme đồ sộ, nhiều theme chuyên nghiệp có khả năng SEO tốt.
- Việc quản lý blog, quản lý các bài viết rất thuận tiện giống như các phần mềm thiết kế website chuyên nghiệp.
- Thể hiện các tệp PDF, DOC, Powerpoint ngay trên nội dung bài viết. Đặc biệt tích hợp sẵn Latex - công cụ soạn thảo công thức toán học, giúp người sử dụng có thể viết công thức toán học ngay trên blog.
- WordPress có 23 Widget (ứng dụng tạo thêm) như Thống kê số truy cập blog, Các bài mới nhất, Các bài viết nổi bật nhất, Các comment mới nhất, Liệt kê các chuyên mục, Liệt kê các Trang, Danh sách các liên kết, Liệt kê số bài viết trong từng tháng... Có 79 theme để người dùng lựa chọn.
- Ngoài việc được áp dụng để xây dựng các Website dạng trang tin tức và Blog, WordPress còn được sử dụng để xây dựng nên các Website thương mại điện tử với mục đích chính là bán hàng Online. Tuy nhiên nếu xét trên phương diện này thì WordPress không thực sự nổi trội.
- Ngoài thống kê số truy cập của từng ngày cho blog, Wordpress còn thống kê số truy cập của từng ngày đối với mỗi bài viết của blog. Trên cơ sở đó chủ blog sẽ có định hướng nên viết vấn đề gì tiếp theo.
- Các comment có thể duyệt rồi mới cho đăng, comment nào có nội dung không phù hợp có thể xóa, nếu cho là spam thì sau này IP đó không có thể gửi comment vào blog được nữa.
- Admin (chủ blog) có thể cho 35 cộng tác viên gửi bài vào blog, có thể phân quyền cho các cộng tác viên theo các cấp độ khác nhau. Lưu giữ danh sách thành viên đã ghé thăm trang blog. Admin cũng có thể cho bất kỳ ai đăng bài qua email vào blog miễn là admin cho họ một địa chỉ email bí mật của blog (địa chỉ này có thể thay đổi bất kỳ lúc nào).

- Sao lưu dữ liệu nhằm khôi phục nội dung blog một cách dễ dàng nếu chẳng may blog bị hack, và cung cấp công cụ chuyên nhà từ các blog khác sang blog WordPress.
- WordPress hỗ trợ 3 GB để lưu trữ các tệp hình ảnh và văn bản.
- Hàng ngày WordPress có thống kê 100 bài trên các blog tiếng Việt của WordPress được nhiều người đọc nhất trong vòng 48 tiếng. Nhờ đó bạn biết được các thông tin quan trọng nhất đang diễn ra.

CHƯƠNG 2: CÀI ĐẶT TRÊN MÁY WEBSITE ĐƯỢC XÂY DỰNG TỪ WORDPRESS

2.1 Những yêu cầu cài đặt cho WordPress

- Cấu hình đề nghị:
 - + Máy chủ: Apache hoặc Nginx
 - + PHP: Phiên bản 5.6 hoặc mới hơn.
 - + MySQL: Phiên bản 5.6 hoặc mới hơn.
 - + Ngoài ra, máy chủ apache cần hỗ trợ module mod_rewrite

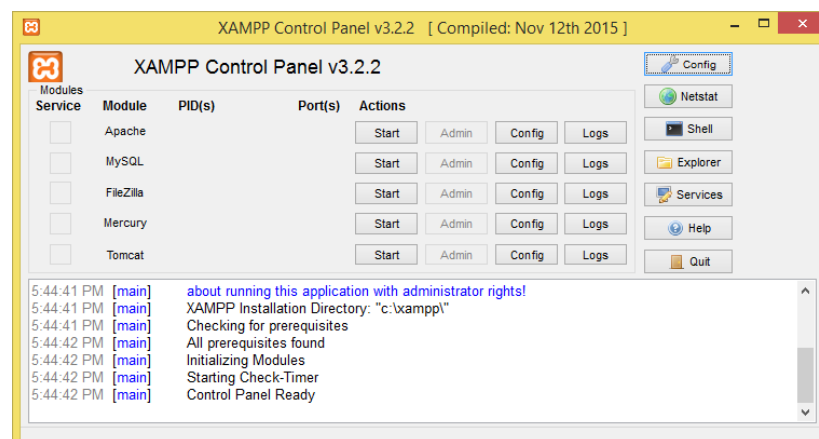
Wordpress cũng có thể hoạt động trên với PHP 5.2.4+ và MySQL 5.0+ nhưng rất dễ dẫn tới các lỗ hổng bảo mật.

- Máy tính người truy cập

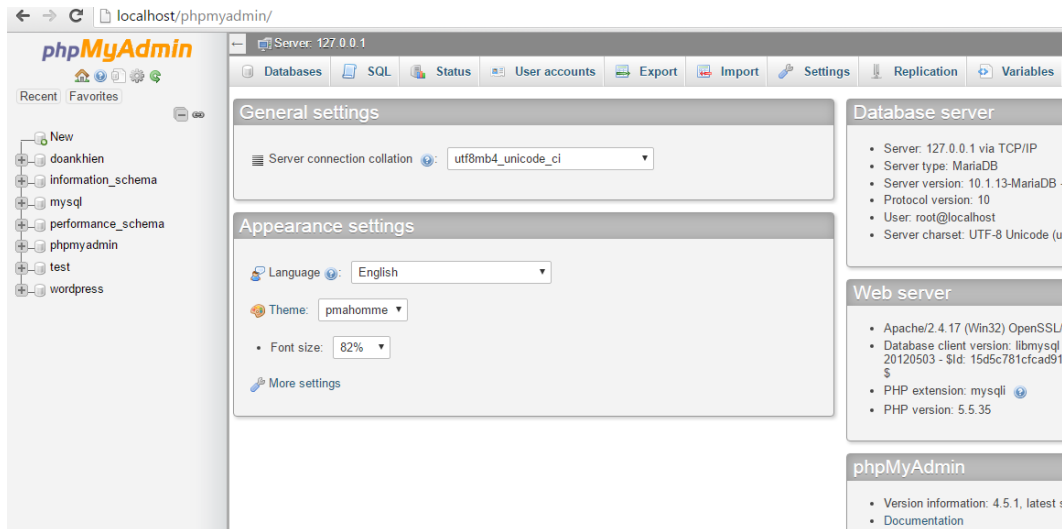
WordPress có thể truy cập tốt trên các phiên bản mới nhất của các trình duyệt Firefox, Internet Explore, Google Chrome, Opera... Máy tính người truy cập chỉ cần cài một trong các trình duyệt này là có thể tương tác với Wordpress thông qua Internet.

2.2 Cài đặt Wordpress trên Localhost

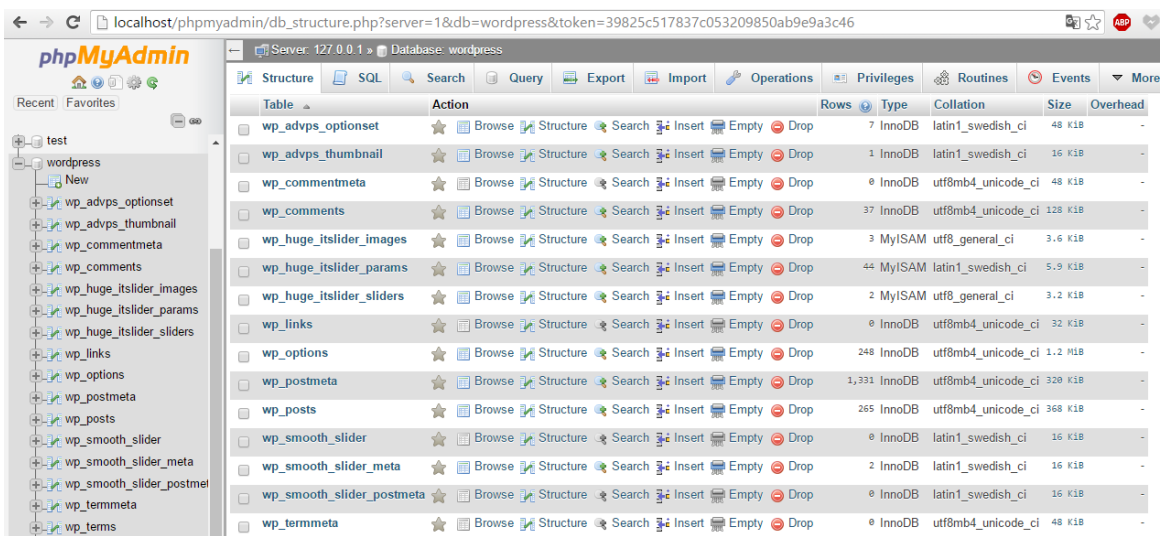
Trước khi cài đặt, máy tính cần phải cài sẵn hệ thống Apache và MySQL. Ví dụ trên XAMPP:



Hình 2.2.1: Giao diện XAMPP



Hình 2.2.2: Giao diện phpMyadmin

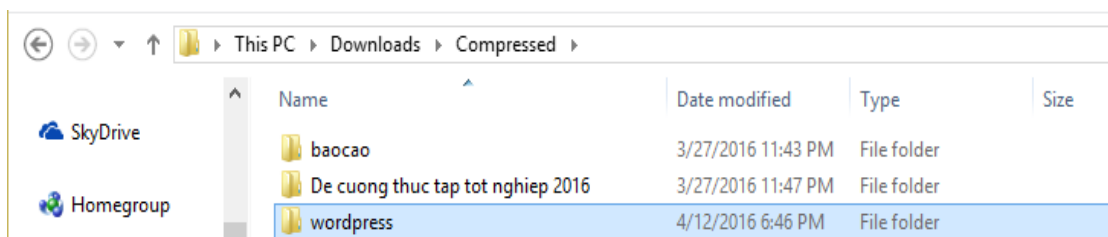


Hình 2.2.3: Giao diện Database trong phpMyadmin

- **Các bước cài đặt WordPress trên localhost**

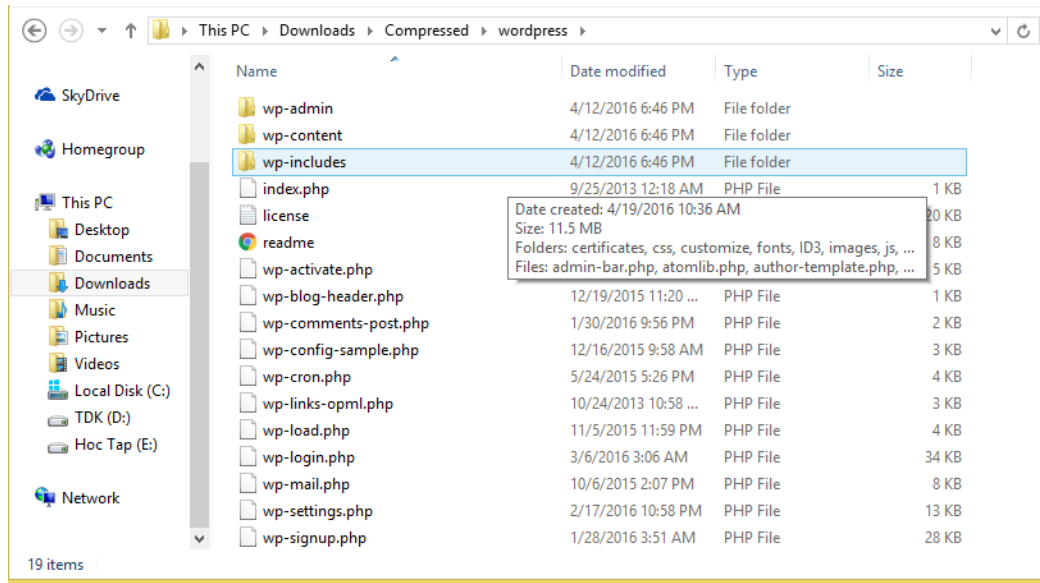
Bước 1: Tải mã nguồn từ website WordPress.org

Trước tiên, chúng ta tải phiên bản mới nhất của mã nguồn WordPress tại địa chỉ <https://wordpress.org/latest.zip>. Sau đó chúng ta giải nén tệp tin này ra sẽ được một thư mục mang tên “wordpress”.



Hình 2.2.4 Thư mục sau khi giải nén mã nguồn

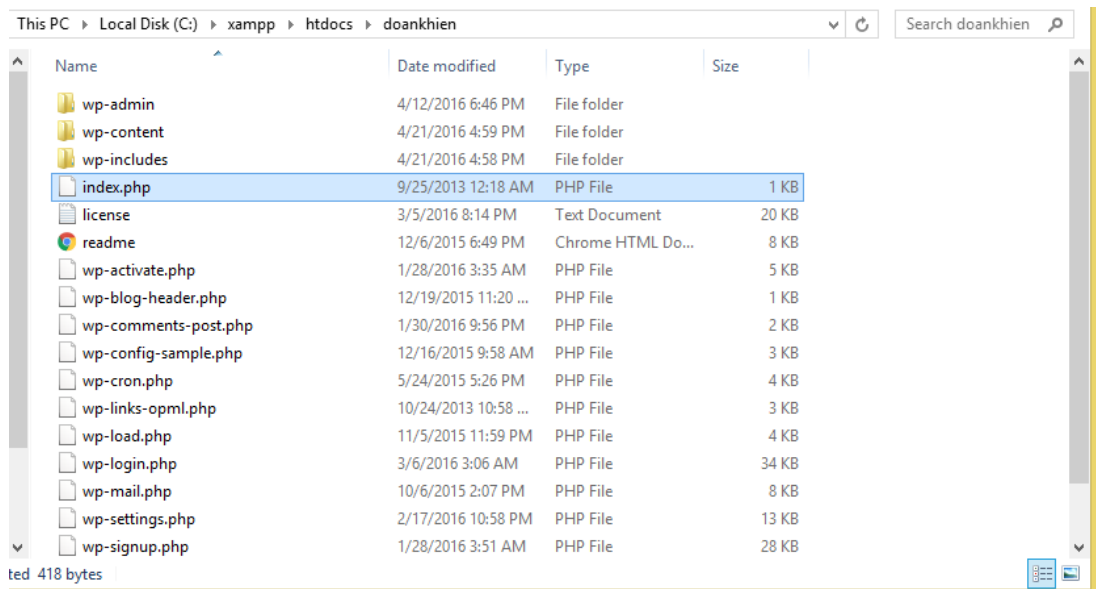
Trong thư mục của wordpress, chúng ta sẽ thấy có một số thư mục tên là *wp-admin*, *wp-includes*, *wp-content* và một số tệp tin tên là *index.php*, *wp-config-sample.php*,... Tất cả tệp tin và thư mục này là mã nguồn WordPress.



Hình 2.2.5: Các tệp tin và thư mục mã nguồn của WordPress.

Bước 2: Copy mã nguồn WordPress vào Localhost

Copy toàn bộ tệp tin và thư mục trong thư mục “wordpress” này vào thư mục website trong localhost (ví dụ: C:\xampp\htdocs\doankhien).

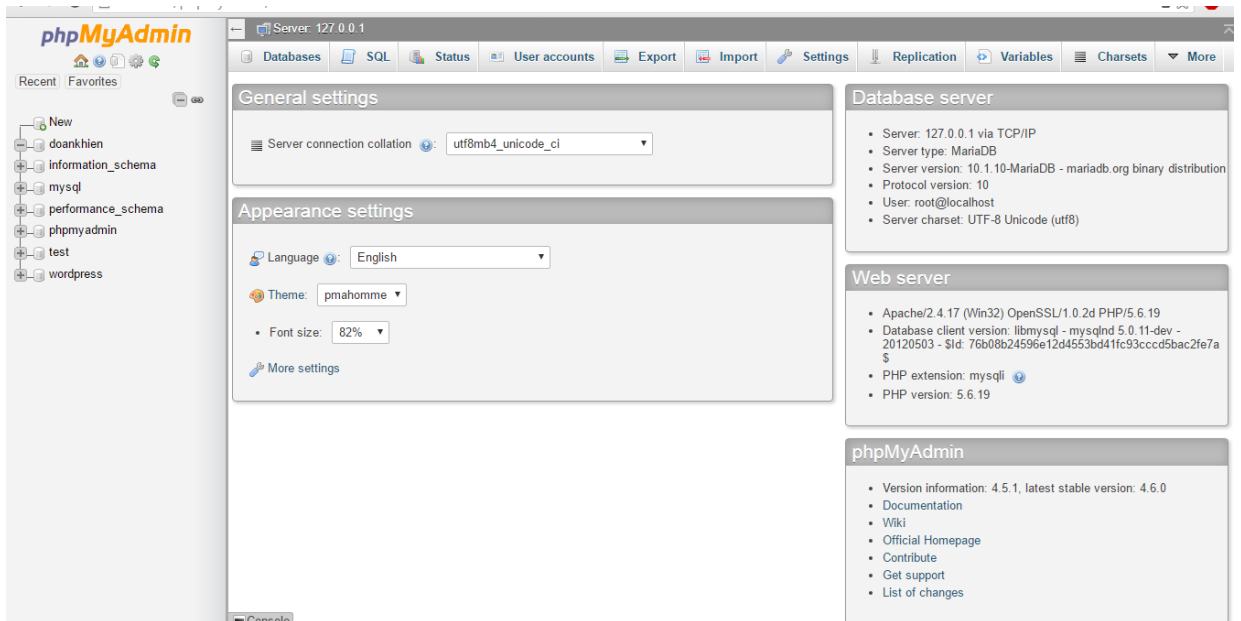


Hình 2.2.6: Mã nguồn wordpress trong thư mục website

Bước 3: Tạo mới một database

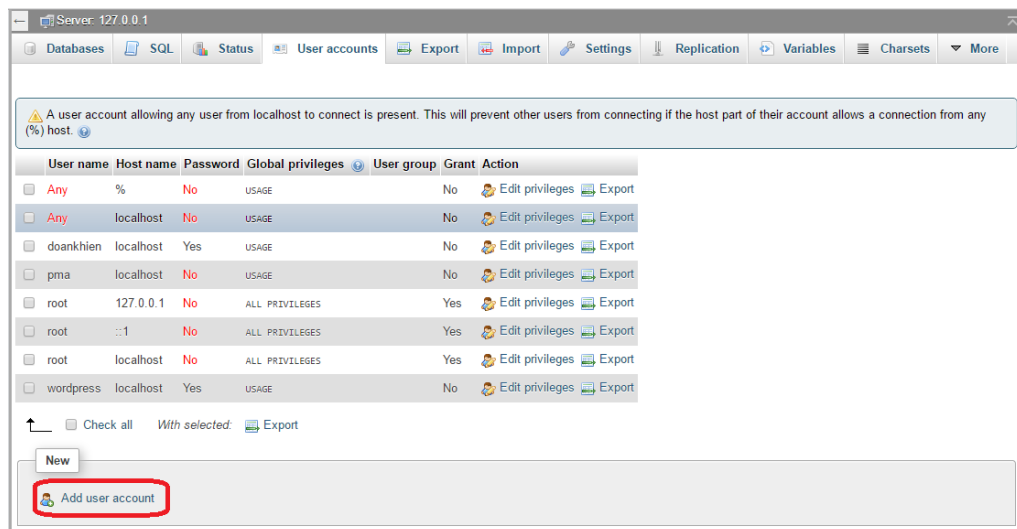
Để chạy được WordPress thì localhost phải có một database dùng MySQL để lưu các dữ liệu mềm vào đó như: bài viết, các thiết lập,...trên website. Để tạo một

database, chúng ta hãy truy cập vào localhost với đường dẫn: <http://localhost/phpmyadmin>. Sau đó, nhấn vào menu Database.



Hình 2.2.7: Giao diện Database trong phpMyadmin

Sau đó vào **User accounts**, chọn **Add user account**

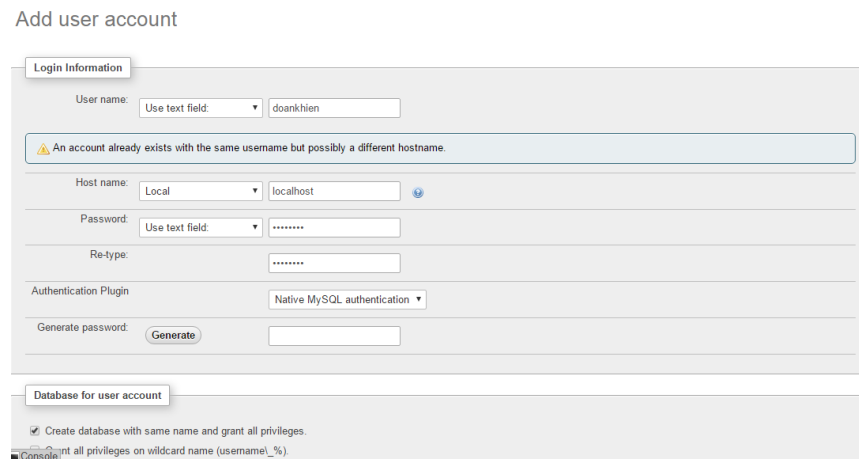


Hình 2.2.8: Tạo user account

Tại đây chúng ta cần điền những thông tin sau :

- User name : Tên User của database
- Password : Mật khẩu của user
- Hostname : Thường là localhost

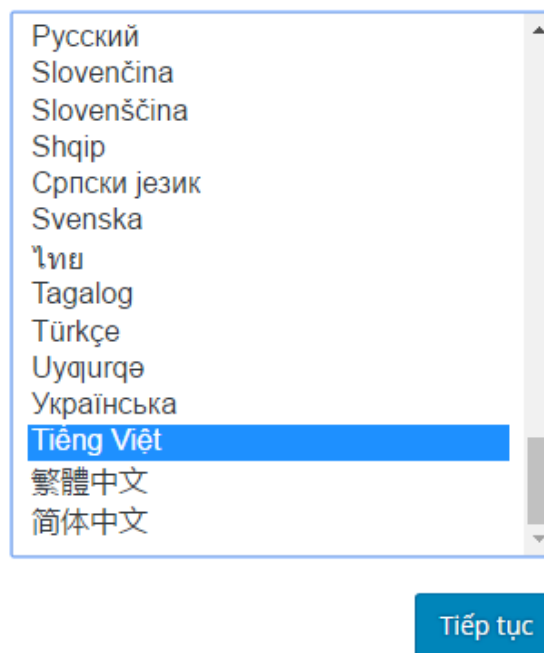
Sau đó đánh dấu tích vào ô **“Create database with same name and grant all privileges”** để tạo cơ sở dữ liệu với cùng tên và cấp tất cả các đặc quyền cho người dùng vừa tạo.



Hình 2.2.9: Tạo database cùng với user

Bước 4: Chạy website để cài đặt

Sau khi copy xong mã nguồn, mở bảng điều khiển của XAMPP lên và khởi động Apache và MySQL. Sau đó truy cập vào của chúng ta website theo đường dẫn: **http://localhost/tên-webiste** (Ví dụ: http://localhost/doankhien). Lúc này, giao diện trình duyệt của website sẽ hiện ra bảng chọn ngôn ngữ cần cài đặt cho WordPress, chúng ta chọn **Tiếng Việt** và ấn **Tiếp tục**.



Hình 2.2.10: Chọn ngôn ngữ khi cài đặt WordPress

Tiếp theo, hệ thống sẽ nhắc nhở cho chúng ta chưa tiến hành đổi tệp tin wp-config-sample.php thành wp-config.php và khai báo thông tin database vào đó. Hãy **ấn Thực hiện ngay** để tiếp tục quá trình cài đặt.

Chào mừng bạn đến với WordPress. Trước khi bắt đầu, chúng tôi cần thông tin để kết nối với cơ sở dữ liệu. Bạn sẽ cần biết những thông tin sau trước khi tiến hành.

1. Tên cơ sở dữ liệu
2. Tên truy cập cơ sở dữ liệu
3. Mật khẩu truy cập cơ sở dữ liệu
4. Địa chỉ máy chủ cơ sở dữ liệu
5. Tiền tố bảng (nếu bạn muốn chạy nhiều hơn một bản WordPress trong cùng một cơ sở dữ liệu)

We're going to use this information to create a wp-config.php file. If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php. Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

Thực hiện ngay!

Hình 2.2.11: Wordpress yêu cầu nhập thông tin database

Tiếp theo, chúng ta tiến hành nhập thông tin cho database. Trong đó, tên cơ sở dữ liệu và tên đăng nhập là cơ sở dữ liệu chúng ta đã tạo ở bước 3.

Bạn cần nhập thông tin chi tiết để kết nối với cơ sở dữ liệu của bạn. Nếu bạn không biết, hãy liên hệ với nhà cung cấp dịch vụ máy chủ của bạn.

Tên Cơ Sở Dữ Liệu	<input type="text" value="wordpress"/>	The name of the database you want to use with WordPress.
Tên đăng nhập	<input type="text" value="wordpress"/>	Your database username.
Mật khẩu	<input type="text" value="mật khẩu"/>	Your database password.
Địa Chỉ Máy Chủ Cơ Sở Dữ Liệu	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Tiền Tố Bảng Dữ Liệu	<input type="text" value="wp_"/>	Nếu bạn muốn chạy nhiều bản WordPress với cùng một cơ sở dữ liệu, hãy thay đổi giá trị này.

Gửi

Hình 2.2.12: Nhập thông tin database

Tiếp theo, chúng ta cần phải thiết lập các thông tin quan trọng cho website như Tên của website, tên tài khoản admin cùng mật khẩu,...Sau khi nhập xong hãy **ấn nút Cài đặt WordPress**.

Thông tin

Làm ơn cung cấp thông tin dưới đây. Đừng lo lắng, bạn luôn luôn có thể thay đổi những cài đặt này về sau.

Tiêu đề trang

Tên đăng nhập
Tên đăng nhập chỉ có thể bao gồm bằng chữ cái và số, khoảng trống, gạch dưới, gạch ngang, dấu chấm và kí tự @.

Mật khẩu
Rất yếu
Quan trọng: Bạn cần tài khoản này để đăng nhập. Hãy lưu nó vào một nơi an toàn


Xác nhận mật khẩu Xác nhận sử dụng mật khẩu yếu

Your Email
Hãy kiểm tra kỹ lưỡng địa chỉ thư điện tử trước khi tiếp tục.

Tương tác với công cụ tìm kiếm Ngăn chặn các công cụ tìm kiếm đánh chỉ mục website này
Việc tuân thủ yêu cầu này hoàn toàn phụ thuộc vào các công cụ tìm kiếm.

Hình 2.2.13: Thiết các thông tin quan trọng của website

Cuối cùng, WordPress sẽ thông báo cho chúng ta là đã cài đặt thành công. Nhấn vào nút **Đăng nhập** để đăng nhập vào bảng quản trị WordPress.



Thành công!

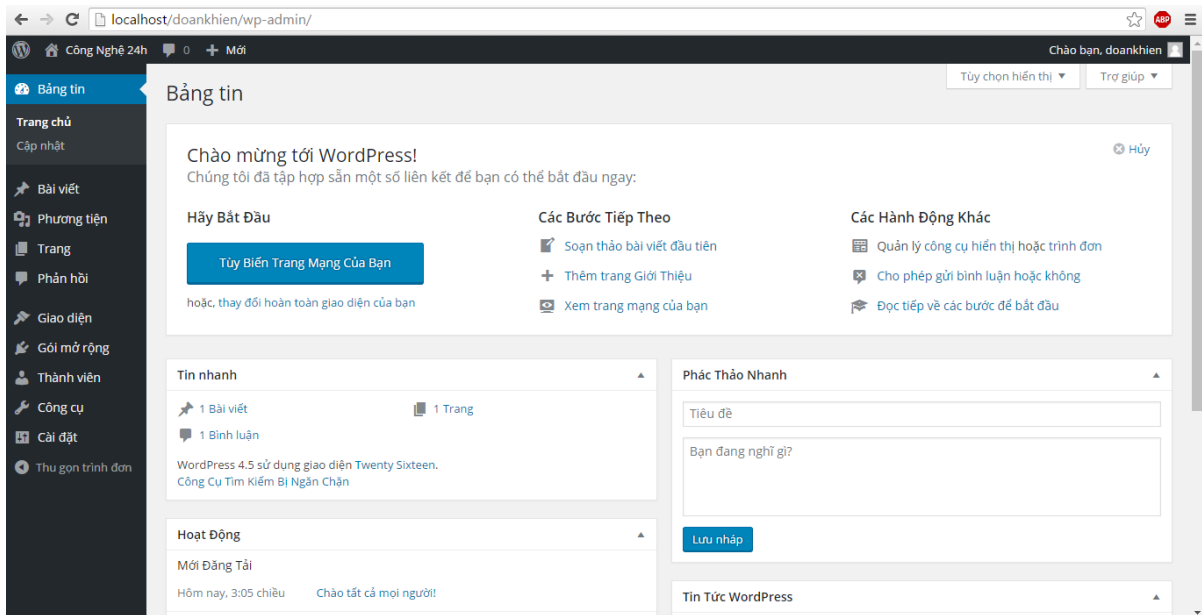
WordPress has been installed. Thank you, and enjoy!

Tên đăng nhập doankhien

Mật khẩu *Mật khẩu bạn đã chọn.*

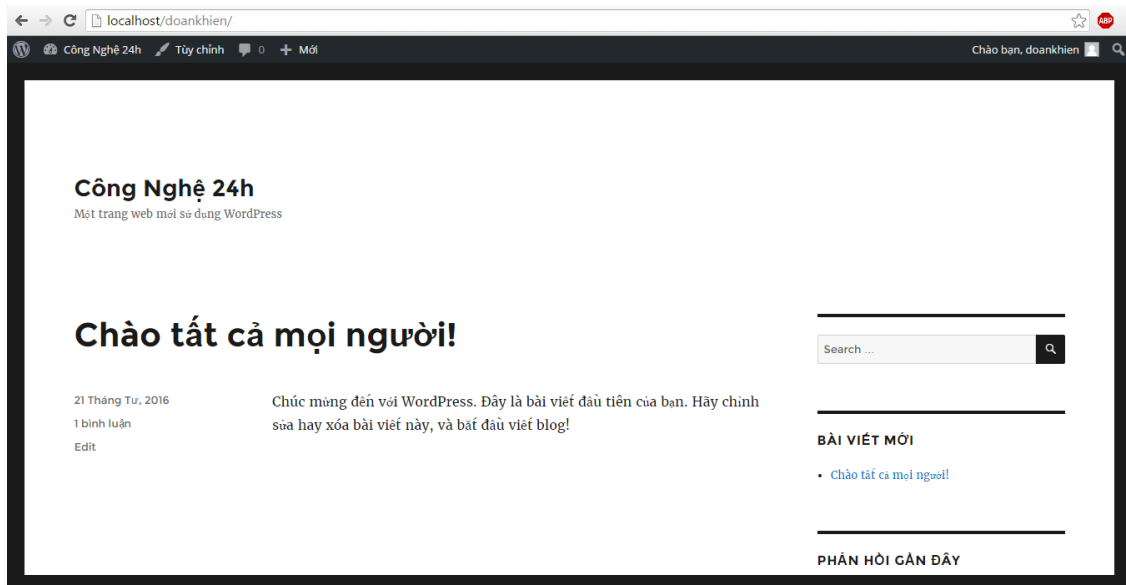
Hình 2.2.14: Cài đặt website thành công

Và đây là giao diện trang quản trị của WordPress.



Hình 2.2.15: Trang quản trị của WordPress sau khi cài đặt

Nếu truy cập vào tên miền chính như **http://localhost/doankhien**, sẽ ra trang chủ của website WordPress với giao diện mặc định.



Hình 2.2.16: Trang chủ website sau khi cài WordPress

Còn nếu chúng ta muốn truy cập vào trang quản trị Admin của WordPress thì sẽ thêm **/wp-admin** vào đường dẫn thành: “**http://localhost/doankhien/wp-admin**”.

CHƯƠNG 3: KỸ THUẬT TẠO GIAO DIỆN NGƯỜI DÙNG TRONG WORDPRESS

3.1. Khái niệm cơ bản về giao diện người dùng trong WordPress (Theme Wordpress)

Đối với website làm bằng WordPress thì giao diện người dùng (hay còn gọi là Theme) là thư mục chứa toàn bộ tệp tin thiết kế giao diện. Đó là nơi WordPress cho phép chúng ta thay đổi, chỉnh sửa giao diện. Mỗi website có một theme riêng biệt. Khi một nhà phát triển được một theme tốt, họ có thể bán nó trên thị trường. Hiện nay có hàng ngàn theme từ miễn phí đến trả phí được rao bán trên internet. Việc chọn theme sẽ quyết định đến hình thức, bố cục giao diện của chúng ta và vì vậy, đây là nội dung đầu tiên chúng ta cần quan tâm, lựa chọn.

Theme là thư mục bắt buộc có trong website của chúng ta. Tất cả những công ty làm website bằng WordPress muốn chỉnh sửa giao diện đều phải phải thông qua thư mục này. Khi cài đặt wordpress đã cho sẵn 3 theme miễn phí là Twentyfifteen, Twentyfourteen, Twentythirteen tuy nhiên những theme này khá đơn giản thường phải chỉnh sửa thêm mới có thể sử dụng.

- Ưu điểm của theme WordPress :

- + Cài đặt đơn giản chỉ cần vài thao tác là hoàn thành.
- + Dễ sửa chữa, chỉnh sửa, bảo trì.
- + Giao diện quản lý đơn giản dễ hiểu.
- + Cấu trúc rõ ràng
- + Thân thiện với các công cụ tìm kiếm như Google, ping...

- Phân biệt theme WordPress

- + **Theme WP thường:** Là một theme có cấu trúc code cơ bản trong thư viện API và các hàm mở trong WordPress hỗ trợ sẵn, một theme thường bao gồm các tệp tin chính như index.php, header.php, footer.php, single.php, sidebar.php, archive.php, category.php và thỉnh thoảng sẽ có thêm một vài tệp tin nhưng các tệp tin đó đều thuộc cấu trúc theme chuẩn WordPress.
- + **Theme Framework:** Là một loại theme nâng cao, có cấu trúc không giống như một theme thường và chỉ dành cho những người sử dụng có kinh nghiệm. Loại theme này thường có giao diện không được đẹp vì mục đích

nó sinh ra là để người sử dụng tiến hành tùy biến nó thành một giao diện mang chất riêng của mình.

- + **Builder Theme:** Đây là loại theme có cấu trúc cũng khác hoàn toàn với một theme thường nhưng lại có cách sử dụng khá đơn giản, chúng ta chỉ việc kéo thả những đối tượng nó có sẵn vào giao diện và tùy chỉnh luôn màu sắc để tự thiết kế cho mình một loại giao diện riêng. Loại này thường dùng cho những người dùng phổ thông vì khả năng tùy biến không được tốt. Một số Builder Theme nổi tiếng như Headway, Ultimatum, iBuilder, PageLines.
 - + **Starter Theme:** Loại này thì chỉ dành cho các lập trình viên hoặc các designer để họ tự thiết một theme hoàn toàn là của mình dựa trên cấu trúc đã được làm sẵn. Nếu những chúng ta muốn học cách làm theme thì nên bắt đầu với loại này.
 - + **Child Theme:** Đây là một loại theme thường nhưng nó được thừa kế các tính năng của theme mẹ, thường được dùng để tùy biến theme mà không can thiệp vào phần code của theme mẹ. Ngoại trừ một số theme đặc biệt như Headway, Thesis thì tất cả các theme còn lại đều hỗ trợ child theme, kể cả theme thường.
- **Kỹ năng và kiến thức cần có để tạo và chỉnh sửa theme**
- + Kỹ năng lập trình cơ bản
 - + Kỹ năng thiết kế
 - + Kiến thức html, css, javascript
 - + Kiến thức WordPress căn bản

3.2. Cấu trúc của Theme WordPress

3.2.1 Cấu trúc thư mục Theme WordPress

Thư mục của Theme gồm 3 thành phần chính:

- Theme stylesheet: là tệp tin style.css, kiểm soát việc trình bày (hình ảnh thiết kế và bố trí) của website.
- Các tệp tin Template: kiểm soát cách các trang web tạo ra các thông tin từ cơ sở dữ liệu WordPress để hiển thị ra bên ngoài.
- Tệp tin tùy chọn chức năng (functions.php).

Ngoài ra, thư mục theme còn có các thư mục con chứa hình ảnh (nếu có), các tệp tin javascripts (nếu có)...

3.2.2 Theme Stylesheet

Ngoài các thông tin về CSS cho theme, tệp tin *style.css* còn có chức năng khai báo thông tin của theme như tên theme, tên tác giả, số phiên bản,... để theme có thể hiển thị trong khu vực Themes của WordPress. Nếu theme của chúng ta không có tệp tin này thì theme sẽ không hợp lệ.

Ví dụ :

```
/*
*Theme Name: Doan Khiem
*Theme URI: https://congnghe24h.esy.es
*Author: Trịnh Doãn Khiển
*Phiên bản: 1.0
*License: GNU General Public License v2 or later
*License URI: http://www.gnu.org/licenses/gpl-2.0.html
*Tags: basic,two-colum,post-format
*Text Domain: doankhien
*Language Folder: /languages
*/
```

3.2.3 Tệp tin tùy chọn chức năng

Một theme cần sử dụng một tập các chức năng, nằm trong thư mục con của theme và được đặt tên là *functions.php*. Tệp tin này tuy không thuộc template của theme nhưng nó sẽ có vai trò chứa những đoạn mã PHP để khai báo các tính năng đặc biệt hoặc sử dụng hàm *add_theme_support()* để khai báo các tính năng trong theme. Tệp tin *function.php* không phải là template nên sẽ không hiển thị ra bên ngoài. Nhưng tất cả code PHP trong đây sẽ được thực thi khi website được tải ra.

Các chức năng thường được gọi trong *function.php*:

- Khai báo theme stylesheet và scrips.
- Kích hoạt tính năng của Theme như Sidebars, Navigation Menus , Post Thumbnail , Post Format , Headers Custom, Background ...
- Xác định hàm được sử dụng trong một số tệp tin template của Theme.

3.2.4 Template files

Các template có đuôi PHP được sử dụng để khai báo khuôn mẫu được định dạng sẵn nhằm mục đích hiển thị nội dung trên website và được xuất ra dưới dạng HTML. Các tệp tin template được tạo thành từ HTML, PHP và các WordPress Template Tags.



Hình 3.2.1: Cấu trúc template trong Theme WordPress

- index.php – Template gốc của theme

Tệp tin index.php không chỉ là template để sử dụng cho trang chủ, mà nó còn là template gốc của website nếu như các template khác chưa được khai báo. Ví dụ nếu theme của chúng ta không có tệp tin single.php để làm template cho trang nội dung của Post, thì nó sẽ sử dụng tệp tin index.php để hiển thị. Các template khác cũng vậy.

- header.php – Phần đầu của theme

Template này sử dụng để khai báo phần header của trang, bao gồm các thẻ mở như `<html>`, `<head>`, `<body>`,... Và sau đó ở các template khác, chúng ta sẽ gọi nó ra bằng hàm `get_header()`.

- footer.php – Phần chân trang của theme

Nó cũng giống như header.php đó là được sử dụng để khai báo phần chân trang của theme. Rồi sau đó ở các template khác ta sẽ gọi nó ra bằng hàm `get_footer()`.

- sidebar.php – Phần hiển thị sidebar

Chúng ta có thể khai báo sidebar trực tiếp vào các template khác với hàm `dynamic_sidebar()` nhưng nếu chúng ta sử dụng sidebar ở nhiều template khác nhau thì chúng ta nên viết code hiển thị sidebar vào tệp tin sidebar.php. Rồi sau đó sẽ dùng hàm `get_sidebar()` để gọi template này ra.

- Các template khác trong theme

Các template dưới đây sẽ không bắt buộc chúng ta phải tạo ra như 3 tệp tin ở trên, nhưng các template này sẽ được sử dụng nếu chúng ta có khai báo. Template nào không khai báo thì nó sẽ sử dụng template cấp cao hơn. Ví dụ nếu `single-123.php` không khai báo thì nó sẽ sử dụng `single.php`.

+ Template hiển thị trang lưu trữ

Template này sẽ sử dụng cho tất cả các trang lưu trữ trên website. Trang lưu trữ là các trang phân loại bài viết như `category`, `tag`, `custom taxonomy`,...

- `archive.php` – Định dạng hiển thị cho toàn bộ trang lưu trữ trên website như lưu trữ theo ngày tháng, `category` (chuyên mục), `tag` (thẻ), `custom taxonomy`,...
- `category.php` – Định dạng hiển thị cho toàn bộ `category` của website.
- `category-tin-tuc.php` – Định dạng hiển thị trang `category` có slug là `tin-tuc`.
- `category-123.php` – Định dạng hiển thị cho `category` mang ID 123.
- `tag.php` – Định dạng hiển thị toàn bộ `tag` của website.
- `tag-tin-tuc.php` – Định dạng hiển thị toàn bộ `tag` có slug là `tin-tuc`.
- `tag-123.php` – Định dạng hiển thị toàn bộ `tag` có ID là 123.
- `author.php` – Định dạng hiển thị cho trang toàn bộ các tác giả trong website.

- *author-doankhien.php* – Định dạng trang hiển thị tác giả tên doankhien.
- *author-123.php* – Định dạng trang hiển thị tác giả có ID là 123.
- *archive-product.php* – Định dạng trang hiển thị danh sách các bài viết thuộc post type tên product.
- *taxonomy-product-category.php* – Định dạng trang hiển thị danh sách các bài viết thuộc custom taxonomy tên product-category.

+ **Template hiển thị trang nội dung**

Template này sẽ định dạng cho trang hiển thị nội dung của Post hoặc Page hoặc một Custom Post Type nào đó.

- *single.php* – Định dạng trang hiển thị nội dung của tất cả các Post.
- *single-product.php* – Định dạng trang hiển thị nội dung tất cả các post trong post type tên product.
- *single-hello.php* – Định dạng trang hiển thị nội dung của post có slug là hello.
- *single-123.php* – Định dạng trang hiển thị nội dung của post mang ID là 123.
- *page.php* – Định dạng hiển thị toàn bộ Page trong website.
- *page-123.php* – Định dạng hiển thị page có ID là 123.

+ **Template trang chủ**

Các template này sẽ được sử dụng cho việc định dạng hiển thị của trang chủ:

- *index.php*
- *front-page.php*
- *home.php*

+ **Template trang 404**

Template này sẽ hiển thị lỗi 404 trong website, và nó chỉ có 1 tệp tin duy nhất là *404.php*.

+ **Template trang kết quả tìm kiếm**

Khi sử dụng chức năng tìm kiếm trên website, kết quả tìm kiếm sẽ được hiển thị bằng template *search.php*. Nếu *search.php* không tồn tại kết quả sẽ dùng template *archive.php*.

3.2.5 Template Tags

Nếu các tệp tin template là những tệp tin đại diện cho trang hiển thị các thành phần trong website thì **template tag** là những hàm được sử dụng để hiển thị một thành phần đặc biệt trong template. Các template tag chúng ta có thể sử dụng bằng cách lồng vào những thẻ HTML để nó hiển thị.

Các template tag là hàm PHP nên nó sẽ phải được nằm trong cặp thẻ `<?php ?>` của ngôn ngữ PHP.

Toàn bộ các template tag được khai báo ở các tệp tin sau :

- `wp-includes/general-template.php`
- `wp-includes/author-template.php`
- `wp-includes/bookmark-template.php`
- `wp-includes/category-template.php`
- `wp-includes/comment-template.php`
- `wp-includes/link-template.php`
- `wp-includes/post-template.php`
- `wp-includes/post-thumbnail-template.php`
- `wp-includes/nav-menu-template.php`

Một số Template Tag cơ bản :

- Template Tag tổng hợp

- + `get_header()` – Nối tệp tin header.php trong theme.
- + `get_sidebar()` – Nối tệp tin sidebar.php trong theme.
- + `get_footer()` – Nối tệp tin footer.php trong theme.
- + `bloginfo()` & `get_bloginfo()` – Hiển thị thông tin của website.
- + `wp_head()` – Hook mặc định của WordPress để hiển thị các thành phần trong thẻ `<head>`.
- + `wp_footer()` – Hook mặc định của WordPress để sử dụng ở footer.
- + `wp_nav_menu()` – Hiển thị một Menu Location.
- + `dynamic_sidebar()` – Hiển thị một sidebar.
- + `wp_link_pages()` – Hỗ trợ liên kết điều hướng trong mỗi bài viết.

- Template Tags cho Post

- + `the_permalink()` – Hiển thị đường dẫn tĩnh của post (bài viết) hiện tại.

- + *the_title()* – Hiển thị tiêu đề post hiện tại trong truy vấn.
- + *the_content()* – Hiển thị toàn bộ nội dung của post hiện tại trong truy vấn.
- + *the_excerpt()* – Hiển thị excerpt hoặc một phần đầu nội dung của post hiện tại trong truy vấn.
- + *the_ID()* – Hiển thị ID của post hiện tại trong truy vấn.
- + *the_tags()* – Hiển thị các tag có trong post hiện tại trong truy vấn.
- + *the_date()* – Hiển thị ngày đăng post hiện tại trong truy vấn.
- + *the_category()* – Hiển thị các category của post hiện tại trong truy vấn.
- + *post_class()* – Hiển thị các class HTML đặc trưng của post hiện tại trong truy vấn.

- **Template Tags cho ảnh thumbnail của Post**

- + *has_post_thumbnail()* – Kiểm tra xem post hiện tại trong truy vấn có thiết lập thumbnail (Featured Image) hay không.
- + *the_post_thumbnail()* – Hiển thị ảnh đại diện của post hiện tại trong truy vấn.

- **Template Tags cho Category và Tag**

- + *category_description()* – Hiển thị mô tả của category (chuyên mục).
- + *tag_description()* – Hiển thị mô tả của tag (thẻ).
- + *wp_dropdown_categories()* – Hiển thị danh sách category với dạng dropdown (đổ xuống).
- + *single_cat_title()* – Hiển thị tên category trong truy vấn.
- + *single_tag_title()* – Hiển thị tên tag trong truy vấn.
- + *wp_tag_cloud()* – Hiển thị mây thẻ.

3.2.6 Vòng lặp (Loop)

Vòng lặp là phần mạnh nhất của theme WordPress. Nó bắt đầu với một truy vấn (query) để xác định các bài viết hoặc các trang để lấy và kết thúc với một câu lệnh php “*endwhile*”.

Trong vòng lặp này, chúng ta có thể lấy ra: tiêu đề, nội dung bài viết, metadata (siêu dữ liệu), custom fields bên trong vòng lặp. Chúng là đầu ra cho mỗi bài viết hay trang khi các truy vấn được thực hiện.

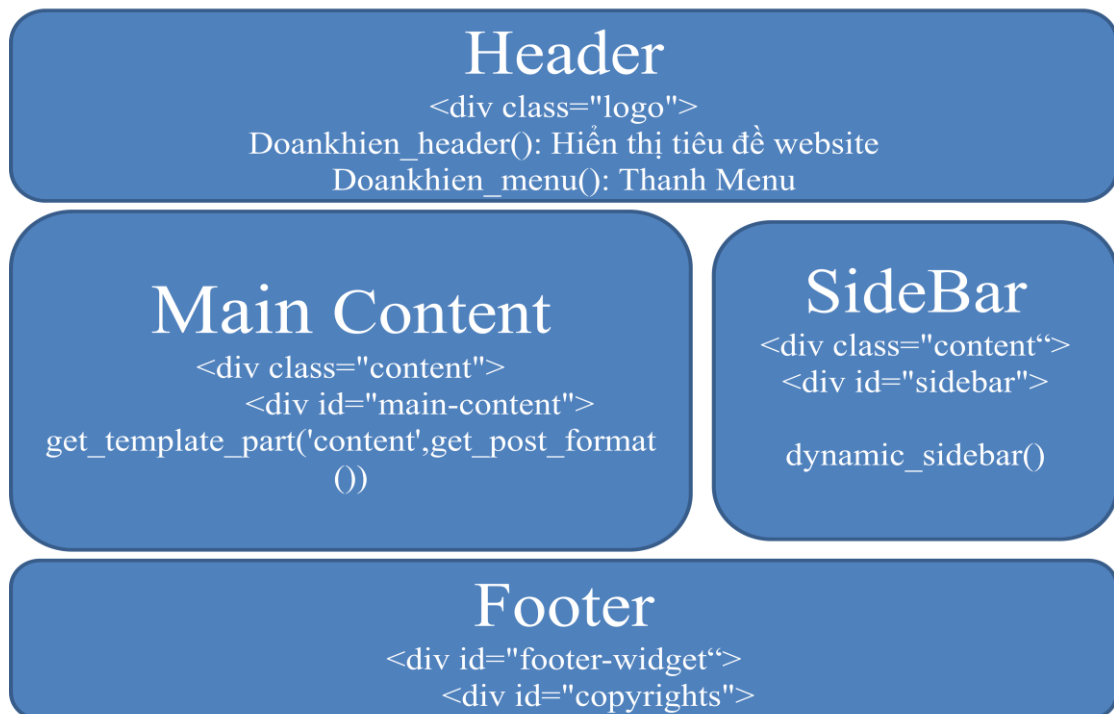
Chúng ta có thể thiết lập nhiều vòng lặp truy vấn trên một trang duy nhất. Ví dụ: trên trang *single.php* chúng ta có thể có các vòng lặp hiển thị toàn bộ nội dung của một bài duy nhất, với một vòng lặp xuất ra chỉ tiêu đề và hình thu nhỏ cho các bài viết liên quan.

```
<?php
    if( $wp_query->have_posts() ) {
        while( $wp_query->have_posts() ) { // Nếu have_posts() == TRUE thì nó
mới lặp, không thì ngừng
            $wp_query->the_post(); // Thiết lập số thứ tự bài viết trong chỉ
mục của query
            /*
             * Nội dung hiển thị bài viết
             */
            echo $post->post_title . '<br>';
        } }
?>
```

3.3. Thiết kế theme WordPress

3.3.1 Ý tưởng thiết kế Theme

- Ý tưởng: Xây dựng một website tin tức.
- Bố cục website:



Hình 3.2.2: Bố cục website dự kiến

Trang web mà chúng ta sắp xây dựng sẽ gồm 4 phần chính:

- + Header: Phần tiêu đề của website và thanh menu.
- + Main content: Hiển thị các bài viết hiện có.
- + Sidebar: Chứa các widget.
- + Footer: Phần chân trang của website gồm các widget và thông tin bản quyền.

3.3.2 Tạo cấu trúc theme

Trong một cấu trúc theme WordPress sẽ có các tệp tin (template) quan trọng như *functions.php*, *index.php*, *style.css*,... và đó là những tệp tin mà chúng ta sẽ làm. Ngoài ra chúng ta sẽ có thêm các tệp tin *content.php*, *content-none.php*,... để dễ dàng cho việc quản lý các vòng lặp để hiển thị các nội dung bài viết.

- Tạo thư mục chứa theme

Vào thư mục nguồn của website: **/wp-content/themes** và tạo một thư mục mà chúng ta muốn chứa theme (Ở đây em tạo thư mục với tên là: **doankhien**)

- + Tạo tệp tin *style.css* để khai báo thông tin của theme : Tên theme, mô tả, tên tác giả....

```

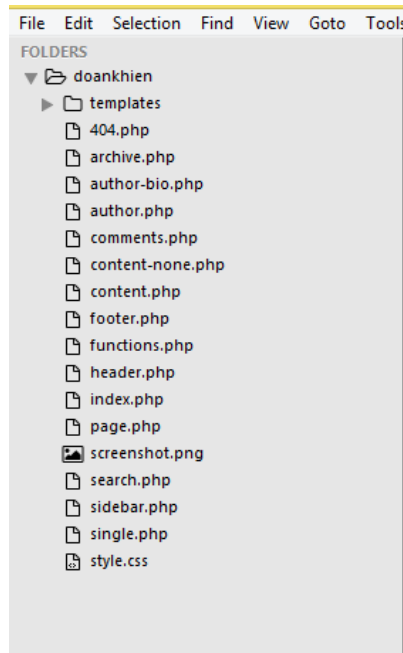
1.      /*
2.      *Theme Name: Doan Khien
3.      *Theme URI: https://congnghe24h.esy.es
4.      *Author: Trịnh Doãn Khiển
5.      *Author URI://congnghe24h.esy.es
6.      *Phiên bản: 1.0
7.      *License: GNU General Public License v2 or later
8.      *License URI: http://www.gnu.org/licenses/gpl-2.0.html
9.      *Tags: basic,two-colum,post-format
10.     *Text Domain: doankhien
11.     *Language Folder: /languages
12.     */

```

- + Tiếp theo đó, tạo lần lượt các tệp tin sau: *functions.php*, *index.php*, *header.php*, *footer.php*, *page.php*, *single.php*, *content.php*, *content-none.php*, *archive.php*, *search.php*, *author.php*, *404.php*, *sidebar.php*....

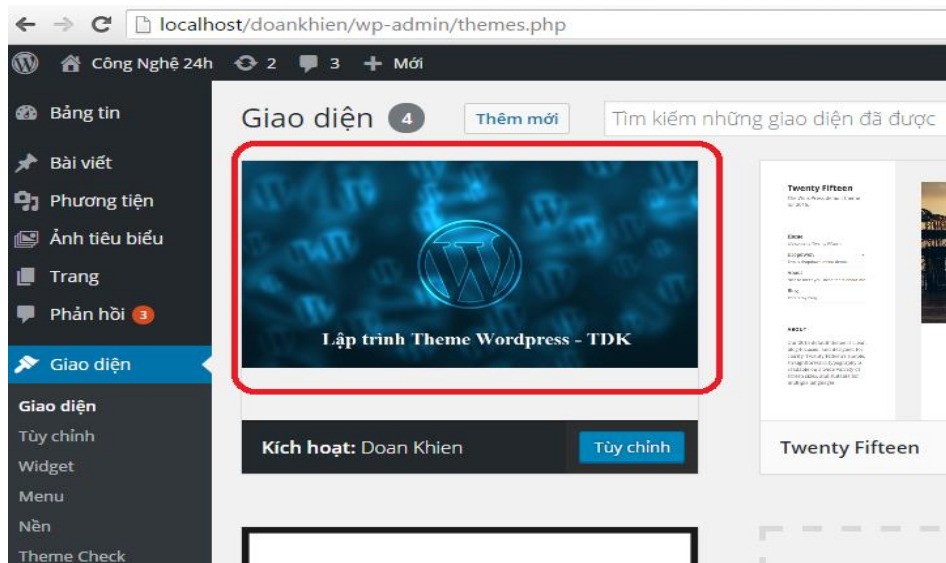
Tiếp tục, tạo một thư mục tên là **templates** trong theme và tạo các tệp tin sau vào thư mục đó, các tệp tin này sẽ là *custom page template*: *full-width.php*, *contact.php*.

Đây là hình ảnh về thư mục chúng ta vừa tạo :



Hình 3.2.3: Thư mục chứa theme

Bây giờ chúng ta vào **Bảng tin -> Giao diện -> Themes** sẽ thấy theme của chúng ta vừa tạo đã được hiển thị ra.



Hình 3.2.4: Theme vừa tạo trong Giao diện của trang Quản trị

3.3.3 Viết code cho tệp tin function.php

Một trong những tệp tin quan trọng nhất của một theme WordPress đó là tệp tin *functions.php*. Đây là một tệp tin bắt buộc trong theme và nó sẽ chứa các đoạn code nguồn mà chúng ta muốn nó luôn được load mỗi khi tải website. Toàn bộ code PHP cần thiết trong một theme (ngoại trừ các code hiển thị nội dung) thì sẽ đều được viết vào tệp tin này.

Trước hết chúng ta cần khai báo những hằng số và hàm cần thiết:

- Thiết lập các hằng dữ liệu quan trọng

Chúng ta sẽ thiết lập một số hằng thường dùng trong quá trình thiết kế theme như: đường dẫn tới thư mục theme (THEME_URL):

```
1. define( 'THEME_URL', get_stylesheet_directory() );
```

- Thiết lập chiều rộng nội dung (**\$content_width**):

```
if ( ! isset( $content_width ) ) {
    $content_width = 620;
}
```

Chúng ta thiết lập giá trị cho biến `$content_width` tức là thiết lập chiều rộng tối đa mà phần hiển thị nội dung (không tính sidebar) mà theme được phép sử dụng. Việc khai báo như vậy sẽ giúp cho các thành phần hiển thị trong nội dung như các mã nhúng oEmbed, hình ảnh,... sẽ không bị tràn ra ngoài khung nội dung.

- **Hàm thiết lập chức năng của Theme**

Hàm này sẽ có chức năng chèn vào điểm neo (hook) *init* của WordPress để khởi tạo các chức năng sẽ được theme hỗ trợ, như post format, customizer,...

```
if ( ! function_exists( 'doankhien_theme_setup' ) ) {
    function doankhien_theme_setup() {
    }
    add_action ( 'init', 'doankhien_theme_setup' );
}
```

Ở đây em đặt tên hàm này là: *doankhien_theme_setup()*. Trong hàm *doankhien_theme_setup()* chúng ta sẽ thiết lập một số tính năng quan trọng :

- + Thiết lập language cho theme:

Chúng ta sẽ khai báo thư mục chứa ngôn ngữ trong theme, và khai báo `textdomain` để load các chuỗi ngôn ngữ có trong theme nhằm mục đích cho theme có thể đọc được các tệp tin ngôn ngữ và người dùng có thể dịch ra nhiều ngôn ngữ khác nhau bằng việc sửa/tạo tệp tin **.po*.

```
$language_folder = THEME_URL . '/languages';
load_theme_textdomain( 'doankhien', $language_folder );
```

- + Thêm chức năng thumbnail cho post

Đây là chức năng để hiển thị ảnh đại diện (Featured Image) cho bài viết

```
add_theme_support( 'post-thumbnails' );
```

+ Thêm chức năng title-tag

```
add_theme_support( 'title-tag' );
```

+ Thêm chức năng Post Format

Chức năng Post Format nghĩa là chúng ta có thể tùy biến việc hiển thị post theo các định dạng như Video, Image, Gallery, Quote,...

```
add_theme_support( 'post-formats',  
    array(  
        'image',  
        'video',  
        'gallery',  
        'quote',  
        'link'    )  
);
```

+ Thêm chức năng custom background

Chức năng này sẽ giúp cho người dùng có thể đổi lại màu nền hoặc thêm ảnh nền cho website dễ dàng thông qua Customize.

```
$default_background = array(  
    'default-color' => '#e8e8e8',  
);  
add_theme_support( 'custom-background', $default_background );
```

+ Tạo menu location

```
register_nav_menu ( 'primary-menu', __( 'Primary Menu', 'doankhien' ) );
```

+ Tạo sidebar

```
$sidebar = array(  
    'name' => __( 'Main Sidebar', 'doankhien' ),  
    'id' => 'main-sidebar',  
    'description' => 'Main sidebar for Doankhien theme',  
    'class' => 'main-sidebar',  
    'before_title' => '<h3 class="widgettitle">',  
    'after_title' => '</h3>',  
);  
register_sidebar( $sidebar );
```

3.3.4 Viết code cho tệp tin header.php

Tại tệp tin này, chúng ta sẽ khai báo các thẻ HTML cần thiết mà trong một tài liệu HTML chuẩn đều có như <html>, <head>, <body>,.... Các thẻ này chỉ bao gồm phần thẻ mở còn phần thẻ đóng sẽ được viết ở tệp tin footer.php.

Trong tệp tin header.php chúng ta sẽ khai báo đoạn code HTML như sau:

```
<!DOCTYPE html>
<html "<?php language_attributes(); ?>" />
<head>
    <meta charset="<?php bloginfo('charset'); ?>" />
    <link rel="profile" href="http://gmpg.org/xfn/11"/>
    <link rel="pingback" href="<?php bloginfo('pingback_url');
?>"/>
    <?php wp_head(); ?>
</head>
<body <?php body_class(); ?> >
    <div id="container">
```

Trong đó :

- Hook *wp_head()*: Đây là điểm neo để giúp WordPress hiểu được đây là khu vực thẻ <head> của theme chúng ta để Wordpress có thể tự thêm các thành phần cần thiết lên, cũng như các plugin khác có muốn can thiệp vào khu vực này thì cũng sẽ dò qua hook *wp_head()*.
- *body_class()*: Đây là hàm trả về các class tượng trưng cho từng trang.
- Hàm *language_attributes()*: hiển thị thuộc tính ngôn ngữ cho thẻ <html>.
- Hàm *bloginfo()*: hiển thị thông tin về trang web dùng cho thẻ <meta>.

Tiếp theo, chúng ta có thể hiển thị tên website, mô tả website và thanh menu.

Lưu ý: Chúng ta nên tạo hàm riêng cho các phần hiển thị ở template trong tệp tin *function.php* thay vì viết code trực tiếp trong các template để thuận tiện cho việc quản lý và chỉnh sửa.

- Hàm hiển thị logo

Chúng ta mở tệp tin function.php và khai báo hàm hiển thị logo (Ví dụ, em đặt tên hàm này là *doankhien_header()*) ở trong đó với đoạn code sau:

```
if(!function_exists('doankhien_header')){
function doankhien_header(){ ?>
<div class="site-name">
```

```

<?php
if(is_home()){
    printf('<h1><a href="%1$s" title ="%2$s"> %3$s </a> </h1>',
        home_url(),
        get_bloginfo('description'),
        get_bloginfo('sitename') );
} else{
    printf('<p><a href="%1$s" title ="%2$s"> %3$s </a> </p>',
        home_url(),
        get_bloginfo('description'),
        get_bloginfo('sitename') );
}
?>
</div>
<div class="site-description">Information Technology for Your Life
</div>
<?php
}}

```

Các tham số như %1\$s, %2\$s, %3\$s là từng tham số tương ứng với 3 hàm dữ liệu nằm bên dưới nó, cụ thể:

- + %1\$s: home_url() – đường dẫn của website
- + %2\$s: get_bloginfo('description') – mô tả website
- + %3\$s: get_bloginfo('sitename') – tên website

Đoạn code trên có nghĩa là tên website sẽ được hiển thị với thẻ <h1> ở trang chủ, còn các trang khác thì nó sẽ được hiển thị bằng thẻ <p>.

- Hàm hiển thị menu

Chúng ta sẽ tạo ra một hàm ở trong tệp tin function.php để hiển thị menu (Ví dụ: đặt tên hàm là: *doankhien_menu(\$menu)*).

```

if(!function_exists('doankhien_menu')) {
    function doankhien_menu($menu) {
        $menu=array(
            'theme_location'=>$menu,
            'container'=>$menu,
            'container_class'=> $menu
        );
        wp_nav_menu($menu);
    }
}

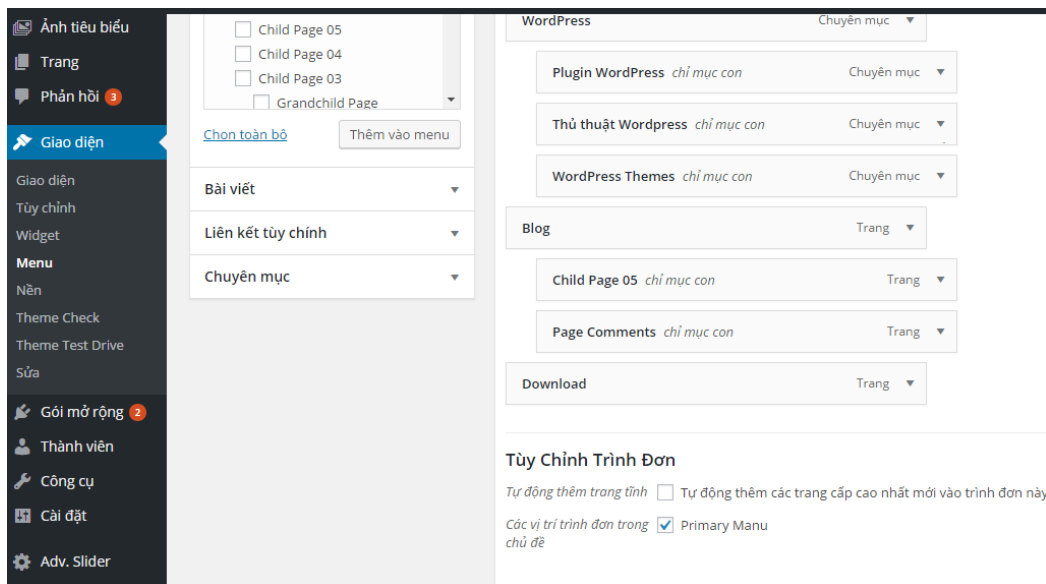
```

Trong đó, hàm `wp_nav_menu()` là hàm hiển thị menu. Hàm này sẽ được truyền tham số `$menu` là một mảng slug của menu mà chúng ta cần gọi ra và sẽ tự áp dụng tên slug đó vào làm class cho menu hiển thị.

Tiếp theo, chúng ta sẽ gọi 2 hàm `doankhien_header()` và `doankhien_menu()` vừa tạo vào trong phần **<container>** của tệp tin `header.php` để chúng hiển thị trong trên website.

```
<div id="container">
  <div class="logo">
    <?php doankhien_header(); ?>
    <?php doankhien_menu('primary-menu'); ?>
  </div>
```

Chúng ta truy cập vào trang wp-admin, **Giao diện -> Menu** để đăng ký menu



Hình 3.2.5: Thêm Menu cho website

Cuối cùng để gọi template `header.php` ở các template khác chúng ta sẽ thêm đoạn code sau: `<?php get_header(); ?>`

3.3.5 Viết code cho tệp tin footer.php

Đây là phần chân trang của website. Ở tệp tin `footer.php`, chúng ta sẽ có các thẻ đóng của các thẻ `<body>`, `<html>` và thẻ `<div class="container">`. Ngoài ra, chúng ta có thể thêm các widget, liên kết và thông tin bản quyền cho website.

Trong tệp tin `function.php` chúng ta dùng hàm `register_sidebar()` tương tự như phần tạo Sidebar chính để tạo thêm 4 widget cho footer: `first-footer-widget-area`, `second-footer-widget-area`, `third-footer-widget-area`, `fourth-footer-widget-are`.

Trong tệp tin footer.php, ta có đoạn code sau:

```
<div id="footer-widgets" class="clearfix">
  <aside class="fatfooter" role="complementary">
    <div class="first quarter left widget-area">
      <?php dynamic_sidebar( 'first-footer-widget-area' ); ?>
    </div>
    <div class="second quarter widget-area">
      <?php dynamic_sidebar( 'second-footer-widget-area' ); ?>
    </div>
    <div class="third quarter widget-area">
      <?php dynamic_sidebar( 'third-footer-widget-area' ); ?>
    </div>
    <div class="fourth quarter right widget-area">
      <?php dynamic_sidebar( 'fourth-footer-widget-area' ); ?>
    </div>
  </aside>
</div>
<div id="footer">
  <div id="copyrights">
    &copy; <?php echo date('Y'); ?> <a href="<?php echo
home_url(); ?>/"><?php bloginfo('name'); ?></a>
  </div>
</div><!-- #footer -->
</div> <!-- end container -->
<?php wp_footer(); ?>
</body><!-- end body -->
</html><!-- end html -->
```

- Hàm `wp_footer()` cũng được gọi để WordPress xác định được hook của phần footer.

Tương tự như phần header, chúng ta dùng hàm `get_footer()` để gọi phần footer ở các template khác.

3.3.6 Viết code cho tệp tin index.php

Đây là tệp tin mà nó sẽ làm trang chủ mặc định cho WordPress và thường ở phần này chúng ta sẽ hiển thị danh sách các bài viết mới nhất trên website. Các bài viết này có thể hiển thị nội dung đầy đủ hoặc rút gọn tùy theo ý muốn.

Trong tệp tin index.php, trước hết chúng ta sẽ dùng 2 hàm: `get_header()` để gọi phần header và `get_footer()` để gọi phần footer. Code hiển thị nội dung ra website sẽ được viết giữa hai đoạn này.

Tiếp theo, chúng ta có thể lựa chọn bố cục website tùy theo ý muốn của mình, miễn sao cân đối vào đẹp mắt là được. Ở đây, em chia phần nội dung ra thành 2 phần: phần hiển thị các bài viết nằm bên trái và phần sidebar nằm bên phải.

```
<?php get_header(); ?>
<div class="content">
  <div id="main-content">
  </div>
  <div id="sidebar">
  </div>
</div>
<?php get_footer(); ?>
```

Chúng ta có class `.content` dùng để bao bọc xung quanh phần hiển thị nội dung và sidebar. Sau đó, `#main-content` là khung hiển thị nội dung và `#sidebar` là khung hiển thị sidebar của website.

Trong phần `#main-content`, chúng ta sẽ một vòng lặp (loop) để hiển thị tất cả các bài viết đang có ra ngoài trang chủ của website:

```
<div id="main-content">
  <?php if(have_posts() ): while (have_posts() ) : the_post(); ?>
    <?php get_template_part('content',get_post_format()); ?>
  <?php endwhile?>
  <?php else : ?>
    <?php get_template_part('content','none'); ?>
  <?php endif;?>
</div>
```

- Hàm `get_template_part('content', get_post_format())` sẽ load tệp tin content `$format.php` trong thư mục theme. `$format` là tên định danh cho từng loại Post Format như video, audio, image,... Nếu bài viết đó chưa chọn Post Format thì nó sẽ load tệp tin `content.php`.

- Hàm `get_template_part('content', 'none')` sẽ load tệp tin `content-none.php` trong thư mục `theme` và tệp tin này chúng ta sẽ viết nội dung hiển thị thông báo query này chưa có dữ liệu.

Ở phần `#sidebar`, chúng ta sử dụng hàm `get_sidebar()` để hiển thị sidebar.

```
<div id="sidebar">
    <?php get_sidebar(); ?>
</div>
```

3.3.7 Viết code cho tệp tin `content.php`

Template `content.php` dùng để hiển thị nội dung của post (bài viết)/page (trang).

Trong tệp tin `content.php` chúng ta khai báo đoạn HTML như sau :

```
<article id="post-<?php the_ID();?>" <?php post_class(); ?> >
<div class="entry-thumbnail">

</div>
<div class="entry-header">

</div>
<div class="entry-content">

</div>
</article>
```

Trong phần này, chúng ta cho mỗi bài viết hiển thị ra đều nằm trong thẻ `<article>` với ID là `post-$id` của post cùng với các class (lớp) tượng trưng cho bài viết đó.

Tiếp đó, khu vực `.entry-thumbnail` sẽ dùng để hiển thị ảnh đại diện của bài viết. Phần `.entry-header` sẽ hiển thị tiêu đề và thông tin của bài viết. Phần `.entry-content` là hiển thị nội dung của bài viết.

- **Viết code cho `.entry-thumbnail`**

Trong tệp tin `function.php`, ta khai báo một hàm (ví dụ: `doankhien_thumbnail()`) để hiển thị ảnh đại diện cho một bài viết như sau:

```
if ( ! function_exists( 'doankhien_thumbnail' ) ) {
    function doankhien_thumbnail( $size ) {

        // Chỉ hiển thị thumbnail với post không có mật khẩu
```

```

        if ( ! is_single() && has_post_thumbnail() && !
post_password_required() || has_post_format( 'image' ) ) : ?>
            <figure class="post-thumbnail"><?php the_post_thumbnail( $size
); ?></figure><?php
            endif;
        }}

```

Tiếp theo, chúng gọi hàm vừa tạo vào khu vực hiển thị thumbnail trong tệp tin content.php và khai báo tên size ảnh cần hiển thị:

```

<div class="entry-thumbnail">
    <?php doankhien_thumbnail( 'thumbnail' ); ?>
</div>

```

- Viết code cho *.entry-header*

Phần này sẽ hiển thị tiêu đề của post và trong mỗi tiêu đề chúng ta sẽ dẫn một đường dẫn đến trang chi tiết của nó.

Trước tiên, chúng ta khai báo hàm hiển thị tiêu đề (Ví dụ: *doankhien_entry_header()* trong tệp tin function.php:

```

if(!function_exists('doankhien_entry_header')){
    function doankhien_entry_header(){ ?>
        <?php if(is_single()) : ?>
            <h1><a href="<?php the_permalink(); ?>" title="<?php
the_title(); ?>"><?php the_title(); ?></a></h1>
            <?php else: ?>
            <h2><a href="<?php the_permalink(); ?>" title="<?php
the_title(); ?>"><?php the_title(); ?></a></h2>
            <?php endif;?>
        <?php
    }
}

```

Và sau đó gọi hàm *doankhien_entry_header()* vào class *.entry-header*

```

<header class="entry-header">
    <?php doankhien_entry_header(); ?>
</header>

```

Tiếp theo đó, chúng ta tạo hàm để hiển thị thông tin của bài viết như: người đăng, thời gian đăng...(ví dụ: *doankhien_entry_meta()*):

```

if(!function_exists('doankhien_entry_meta ')) {
    function doankhien_entry_meta(){ ?>

```

```

<?php if(!is_page()) : ?>
<div class="entry-meta">
<?php
    printf(__('%<span class="author"> %1$s </span>', 'doankhien'),
        get_the_author() );
    printf(__('%<span class="date-published"> %1$s
</span>', 'doankhien'),
        get_the_date() );
    printf(__('%<span class="category"> %1$s</span>', 'doankhien'),
        get_the_category_list(', ' ) );
    echo '<br/>';
    if(comments_open()):
    echo '<span class="meta-reply">';
        comments_popup_link(
            __(' Leave a comment ', 'doankhien'),
            __(' One comment ', 'doankhien'),
            __(' % comment ', 'doankhien'),
            __(' Read all comment', 'doankhien') );
    echo '</span>';
    endif;
?>
        </div>
<?php endif; ?>
<?php }}

```

Sau đó gọi hàm này trong class entry-header ngay dưới hàm *doankhien_entry_header()*

```

<div class="entry-header">
    <?php doankhien_entry_header(); ?>
    <?php doankhien_entry_meta(); ?>
</div>

```

- **Viết code cho *.entry-content***

Chúng ta sẽ tạo hàm (ví dụ: *doankhien_entry_content()*) trong tệp tin *function.php* để hiển thị phần nội dung của bài viết và nút đọc thêm (readmore):

```

if(!function_exists('doankhien_entry_content')) {
function doankhien_entry_content () {
    if(!is_single() && !is_page()) {
        the_excerpt(); //hien thi noi dung rut gon
    } else {
        the_content(); // hien thi day du noi dung
    }
}
}

```

```

    }
    $link_pages=array(
        'before' => __('<p> Page ', 'doankhien'),
        'after' => '</p>',
        'nextpagelink'=> __('Next page', 'doankhien'),
        'previouslink' => __('Previous Page', 'doankhien')
    );
    wp_link_pages($link_pages);    }
}
function doankhien_readmore(){
return '<a class="read-more" href="'.get_permalink(get_the_ID()) .
'>'. __('[ ... Read more ]', 'doankhien'). '</a>';
}
add_filter('excerpt_more', 'doankhien_readmore');

```

Trong hàm này, chúng ta sẽ kiểm tra điều kiện:

- + Nếu không là trang single hoặc page: Hiển thị nội dung rút gọn bằng hàm *the_excerpt()*.
- + Ngược lại: Hiển thị nội dung đầy đủ bằng hàm *the_content()*.

Sau đó, ta gọi hàm *doankhien_entry_content()* ra khu vực *.entry-content* :

```

<div class="entry-content">
    <?php doankhien_entry_content(); ?>
    <?php (is_single() ? doankhien_entry_tag() : '' ); ?>
</div>

```

Hàm *doankhien_entry_tag()* được tạo để hiển thị các tag hiện có của bài viết.

Trong hàm này chúng ta sửa dụng template tag *get_the_tag_list()*.

- Code cho tệp tin *content-non.php*

Tệp tin *content-none.php* hiển thị một thông báo chưa bài viết nào:

```

<div class="no-post">
    <?php _e('Nothing post found.', 'doankhien');
    ?>
</div>

```

3.3.8 Viết code cho Post Format

Post Format nhằm mục đích hiển thị bài viết theo các định dạng khác nhau: hình ảnh, video, link...

- Format content-image.php

```
<article id="post-<?php the_ID();?>" <?php post_class(); ?> >
<div class="entry-thumbnail">
    <?php doankhien_thumbnail('large'); ?>
</div>
<div class="entry-header">
    <?php doankhien_entry_header(); ?>
    <?php
        $attachment = get_children(array ('post_parent' => $post-
>ID));
        $attachment_number=count($attachment);
        printf(__('This image post contains %1$s
photos', 'doankhien'), $attachment_number);
    ?>
</div>
<div class="entry-content">
    <?php doankhien_entry_content(); ?>
    <?php (is_single() ? doankhien_entry_tag() : '' ); ?>
</div>
</article>
```

- Format content-video.php

```
<article id="post-<?php the_ID();?>" <?php post_class(); ?> >
<div class="entry-header">
    <?php doankhien_entry_header(); ?>
</div>
<div class="entry-content">
    <?php the_content(); ?> // Hiển thị toàn bộ nội dung
    <?php (is_single() ? doankhien_entry_tag() : '' ); ?>
</div>
</article>
```

- Format content-link.php

```
<article id="post-<?php the_ID();?>" <?php post_class(); ?> >
<div class="entry-thumbnail">
    <?php doankhien_thumbnail('thumbnail'); ?>
</div>
<div class="entry-header">
    <?php
```

```

        $link=get_post_meta($post->ID,'format_link_url',true);
        $link_description = get_post_meta($post->ID,
'format_link_description',true);
        if(is_single()) {
            printf( '<h1 class="entry-title"><a href="%1$s"
target="blank">%2$s </a></h1>' , $link,get_the_title());}
        else {
            printf( '<h2 class="entry-title"><a href="%1$s"
target="blank">%2$s </a></h2>' , $link,get_the_title());
        }?>
    </div>
    <div class="entry-content">
        <?php
            printf('<a href "%1$s"target="blank">%2$s</a>',
                $link,$link_description);
        ?>
        <?php (is_single() ? doankhien_entry_tag() : '' ); ?>
    </div>
</article>

```

3.3.9 Code cho tệp tin single.php và page.php

Đây là 2 template sẽ được tải khi chúng ta vào một post (bài viết) hay một page (trang) cụ thể.

- Viết code cho single.php

Cấu trúc của tệp tin single.php tương tự như tệp tin index.php. Chúng ta sẽ bổ sung thêm phần hiển thị tác giả và danh sách các bình luận hiện có.

Trước tiên, chúng ta tạo tệp tin author-bio.php để khai báo khung tác giả:

```

<div class="author-box">
    <div class="author-avatar ">
        <?php echo get_avatar(get_the_author_meta('ID')); ?>
    </div>
    <h3><?php printf('Written by <a href="%1$s">%2$s</a>',
        get_author_posts_url(get_the_author_meta('ID')),
        get_the_author());?></h3>
    <p><?php echo get_the_author_meta('description'); ?></p>
</div>

```

Trong tệp tin single.php chúng ta sẽ dùng code của tệp tin index.php và bổ sung thêm template author-bio.php và hàm `comments_template()` để gọi khung bình luận:


```

<?php get_header(); ?>
<div class="content">
  <div id="main-content">
    <?php if(have_posts()): while (have_posts()): the_post(); ?>
      <?php get_template_part('content',get_post_format()); ?>
      <?php get_template_part('author-bio'); ?>
      <?php comments_template(); ?>
    <?php endwhile?>
    <?php else : ?>
      <?php get_template_part('content','none'); ?>
    <?php endif;?>
  </div>
  <div id="sidebar">
    <?php get_sidebar(); ?>
  </div>
</div>
<?php get_footer(); ?>

```

- **Viết code cho page.php**

Hoàn toàn tương tự như tệp tin single.php. Chúng ta chỉ bỏ đi phần hiển thị danh sách bình luận.

3.2.10 Viết code cho các trang lưu trữ, trang tìm kiếm và trang 404

- Các trang lưu trữ bao gồm các trang hiển thị các bài viết theo phân loại như Tag, Category, Lưu trữ theo thời gian và trang riêng của từng tác giả.
 - + *archive.php* – Template chung dành cho các taxonomy (phân loại).
 - + *author.php* – Template dành cho trang riêng của từng tác giả, nếu không có tệp tin này website sẽ load tệp tin archive.php.
 - + *search.php* – Template cho trang hiển thị kết quả tìm kiếm.
 - + *404.php* – Template hiển thị thông báo lỗi 404 trên website WordPress.

- **Viết code cho archive.php**

Hoàn toàn tương tự với trang index.php. Chúng ta sẽ bổ sung thêm phần hiển thị tên trang lưu trữ hiện tại bằng cách sử dụng hàm điều kiện để kiểm tra xem query trên trang hiện tại là Tag, Category, Day, Month hay là Year.

```

<div class="archive-title">
<?php
  if(is_tag()) :

```

```

        printf(__('Posts tagged:%1$s', 'doankhien'),
single_tag_title('',false));
    elseif (is_category()):
        printf(__('Posts catagorized:
%1$s','doankhien'),single_cat_title('',false));
    elseif (is_day()) :
        printf(__('Daily Archives :
%1$s','doankhien'),get_the_time('l, F j, Y'));
    elseif (is_month()) :
        printf(__('Monthly Archives :
%1$s','doankhien'),get_the_time('F Y'));
    elseif (is_year()) :
        printf(__('Yearly Archives :
%1$s','doankhien'),get_the_time('Y'));
    endif;
?>
</div>

```

Kế tiếp, chúng ta viết thêm một đoạn code nữa để hiển thị mô tả của category và tag nếu có.

```

<?php if ( is_tag() || is_category() ) : ?>
    <div class="archive-description">
        <?php echo term_description(); ?>
    </div>
<?php endif; ?>

```

- **Viết code cho author.php**

Tệp tin này là một phần trong template archive.php, mục đích là hiển thị các bài viết của một tác giả dựa theo truy vấn đang truy cập. Nội dung của tệp tin author.php hoàn toàn tương tự như tệp tin index.php. Chúng ta sẽ bổ sung thêm class:"author-box" để hiển thị thông tin tác giả.

```

<div class="author-box">
    <div class="author-avator ">
        <?php echo get_avatar(get_the_author_meta('ID')); ?>
    </div>
    <h3><?php printf('Written by <a href="%1$s">%2$s</a>',
        get_author_posts_url(get_the_author_meta('ID')),
        get_the_author());?></h3>

```

```
<p><?php echo get_the_author_meta('description'); ?></p>
</div>
```

- **Viết code cho trang tìm kiếm (search.php)**

Đây là trang hiển thị kết quả tìm kiếm trên website nếu website đang sử dụng trình tìm kiếm mặc định của WordPress. Chúng ta sử dụng phần code của trang index.php và thêm phần hiển thị thông tin về truy vấn tìm kiếm:

```
<div class="search-info">
    <h2 class="page-title"><?php _e( 'Search Results for:',
    'doankhien' ); ?> <span><?php echo get_search_query();
    ?></span></h2><hr>
    <?php
        $search_query=new WP_Query('s='.$s.'&showpost=-1');
        $search_keyword=esc_html($s,1);
        $search_count=$search_query->post_count;
        printf(__('We found %1$s articles for your search
        query', 'doankhien'),$search_count); ?>
</div>
```

- **Viết code cho trang 404**

Ở trang này chúng ta sẽ hiển thị một thông báo nội dung của trang này không tồn tại. Ngoài ra, chúng ta có thể thêm khung tìm kiếm và danh sách các tags, danh sách các categories có trong website để cho người truy cập có thể lựa chọn.

```
<?php get_header(); ?>
<div class="content">
<div id="main-content">
    <?php
        _e('<h2> 404 NOT FOUND ERROR !!! </h2>', 'doankhien');
        _e('<p> The article you were looking for was not found, but
        maybe try looking again !', 'doankhien');
        get_search_form();
        _e('<h3> Content categories : </h3>', 'doankhien');
        echo '<div class="404-cat-list">';
        wp_list_categories(array('title-li'=> ''));
        echo '</div>';
        _e('Tag cloud ', 'doankhien');
        wp_tag_cloud();?>
    </div>
<div id="sidebar">
    <?php get_sidebar(); ?>
```

```

</div>
</div>
<?php get_footer(); ?>

```

3.3.11 Viết code cho tệp tin sidebar.php

Chúng ta đã đăng ký một sidebar với tham số là \$sidebar. Ở tệp tin sidebar.php chúng ta sẽ dùng hàm *dynamic_sidebar()* với tham số là 'main-sidebar'.

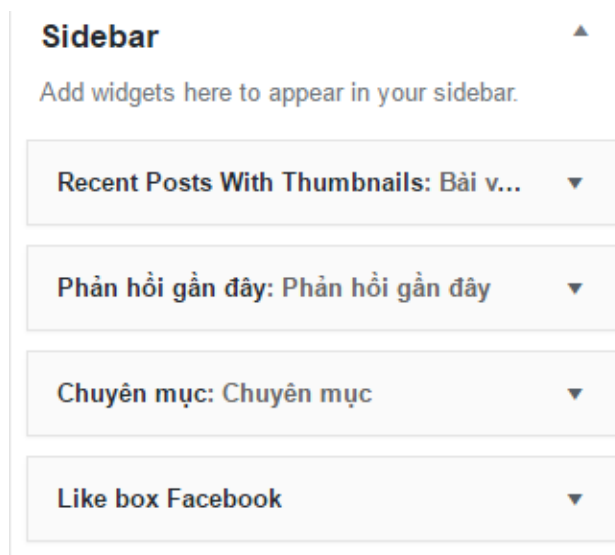
```

<?php
    if ( is_active_sidebar('main-sidebar') ) {
        dynamic_sidebar( 'main-sidebar' );
    } else {
        _e('This is widget area. Go to Appearance -> Widgets to
add some widgets.', 'doankhien');
    }?>

```

Hàm *is_active_sidebar()* dùng để kiểm tra xem main-sidebar đã có widget nào chưa, nếu sidebar đã được thêm widget vào thì hàm này sẽ hiển thị sidebar lên bằng hàm *dynamic_sidebar()*. Trường hợp nếu sidebar chưa có widget nào thì sẽ hiển thị một thông báo.

Chúng ta vào trang wp-admin, **Giao diện -> Widget** để thêm widget cho sidebar.



Hình 3.2.6: Theme Widget cho sidebar

3.3.12 Viết CSS cho theme

Đây là phần quan trọng nhất của theme. Nó sẽ quyết định xem theme của chúng ta có đẹp và hoàn hảo hay không. Để có thể viết CSS cho theme chúng ta cần có kiến thức cơ bản về HTML và CSS.

- Chèn CSS vào theme

Trong tệp tin `function.php`, chúng ta tạo một hàm mới. Ví dụ: `doankhien_style()`. Trong hàm này chúng ta dùng hàm `wp_register_style()` để đăng ký tệp tin `style.css` và danh sách chờ của WordPress. Sau đó dùng hàm `wp_enqueue_style()` để gọi tệp tin này ra giao diện.

```
function doankhien_style(){
    wp_register_style('main-style',
get_template_directory_uri()."/style.css",'all') ;
    wp_enqueue_style('main-style');
}
add_action('wp_enqueue_scripts','doankhien_style');
```

- Viết CSS cho Theme

Chúng ta sẽ viết CSS từ khái quát cho tới cụ thể từng vùng theo các vùng chọn dựa vào các `#id` và `.class` đã được quy định trong suốt quá trình tạo theme.

- + Viết CSS cho các phần tử quan trọng trong website như : kiểu chữ, màu chữ, cỡ chữ, màu liên kết...

```
/*Global Style*/
html{
    box-sizing: border-box;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
}
body{
    font-size: 16px;
    line-height: 1.4;
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;}
img{
    max-width: 100%;
    height: auto;}
a{
    color: #184C7D;
```

```
text-decoration: none;}
a:hover{
  color: #2771B6;
  text-decoration: none;}
input,input[type="submit"],button{
  border: 1px solid #c8c8c8 !important;
  padding: 5px 10px;}
h1{
  font-size: 2.2em;}
h2{
  font-size: 20px;}
h3{
  font-size: 1.7em;}
h4{
  font-size: 1.5em;}
h5{
  font-size: 1.3em;}
h6{
  font-size: 1.1em;}
table,table tr,table td {
  border: 2px solid #e7e7e7;
  padding: 5px;
}
```

+ CSS chia cột website

Chúng ta phân trang ra làm 2 cột đó là cột hiển thị nội dung #main-content và #sidebar. Và tất cả thành phần trong website được bao bọc trong một vùng chọn là #container.

Chúng ta sẽ đặt chiều rộng của website với kích thước là 960px cho toàn bộ website, chiều rộng của khung #main-content là 615px rồi nhảy qua trái với thuộc tính float: left, sau đó phần #sidebar sẽ có chiều rộng là 300px, cũng nhảy qua trái với thuộc tính float.

```
/*=== Chia cot website===*/
#container{
  width: 960px;
  margin: 0 auto;
}
.content{
  overflow: hidden;
```

```
padding: 15px;
background-color: #fff;
-moz-box-shadow: 0 0 9px 1px rgba(47,103,138,.20);
-webkit-box-shadow: 0 0 9px 1px rgba(47,103,138,.20);
box-shadow: 0 0 9px 1px rgba(47,103,138,.20);
}
#main-content{
overflow: hidden;
float: left;
width: 615px;
}
#sidebar{
overflow: hidden;
float: left;
width: 300px;
margin-left: 15px;
}
```

Chúng ta tiếp tục viết CSS cho các phần còn lại của website để tạo ra được giao diện theo ý muốn.

- Tạo ngôn ngữ cho theme

Chúng ta dùng phần mềm PoEdit để tạo tệp tin ngôn ngữ cho theme sau đó lưu lại với tên *[mã-ngôn-ngữ].po*. Ví dụ: *vi.po* vào thư mục language trong thư mục theme. Để kích hoạt ngôn ngữ tiếng việt, vào **WordPress -> Settings -> General**, phần Site Languages chọn là vi và lưu lại.

3.4 Ứng dụng

Sau quá trình thiết kế Theme cho WordPress, em đã ứng dụng để xây dựng một website tin tức và đưa lên trên Internet với các thông tin như sau:

- Tên website: Công Nghệ 24h
- Địa chỉ: <http://congnghe24h.esy.es/>
- Phiên bản WordPress: 4.5
- Mục đích: Cung cấp cho người đọc những tin tức mới nhất về Công nghệ.



Hình 3.4.1: Trang chủ website Công nghệ 24h

Công Nghệ 24h

Information Technology for Your Life

[Trang chủ](#) |
 [WordPress](#) |
 [Tin Tức](#) |
 [Thiết Bị Số](#) |
 [Hỏi đáp](#) |
 [Download](#)

[Công Nghệ 24h](#) » [Điện thoại](#) • [Featured](#) » [Chiêm ngưỡng 2 màn hình cong ấn tượng của Galaxy Note 7](#)

Chiêm ngưỡng 2 màn hình cong ấn tượng của Galaxy Note 7

doankhien 📅 02/07/2016 🏷️ Điện thoại, Featured 💬 Để lại bình luận

(Techz.vn) Hình ảnh thực tế đầu tiên về mẫu phablet “bom tấn” 2016 đã chính thức lộ diện.

Theo những thông tin trước đó, Galaxy Note 7 dự kiến sẽ trình làng trong một sự kiện lớn được tổ chức vào ngày 2/8. Và đây có lẽ là thời khắc hợp lý để người dùng lần lượt được chiêm ngưỡng mẫu phablet thế hệ mới của hãng.



Như hình ảnh trên, chúng ta có thể nhận ra rằng chiếc Galaxy Note 7 sở hữu thiết kế gần như tương đồng với Galaxy S7 edge, khi đi kèm 2 cạnh cong bóng bẩy. Tuy rằng kích thước có phần đồ sộ hơn khá nhiều. Ngoài ra, mặt trước của máy (phần cạnh trên) có khá nhiều cảm biến, không ngoại trừ khả năng khu vực này có chứa thêm bộ cảm biến quét mống mắt (iris scanner). Tức người dùng chỉ việc nhìn vào màn hình và thiết bị sẽ tự động mở khoá, thay vì thao tác truyền thống như gõ passcode hoặc quét vân tay. Thực ra công nghệ này không hoàn toàn mới mẻ, vì nó đã được áp dụng trên một số mẫu máy Windows Phone như Lumia 950 XL hay Lumia 950.

Thông số cấu hình của Note 7 bao gồm:

- Giao diện: Chạy Android 6.0 cùng giao diện mới Grace UX
- Màn hình: 5,7 inch hoặc 5,8 inch Super AMOLED độ phân giải 2K
- Ví xử lý: Snapdragon 820 cho thị trường Mỹ và Exynos 8893 cho thị trường quốc tế
- RAM: 6GB
- Camera chính: 12MP chính, 5MP phụ
- Bộ nhớ: UFS 2.0 dung lượng 64GB, hỗ trợ microSD và khe cắm thẻ nhớ mở rộng
- Pin: 3600mAh hoặc 4000mAh
- Tính năng: Màn hình cong, chống nước chứng chỉ IP68, Bút S-Pen mới, cổng USB Type-C.

Search

BÀI VIẾT GẦN ĐÂY



CÁN CẢNH CÁC MẪU GAMING LAPTOP CỦA MSI TẠI COMPUTEX 2016
 Bởi doankhien



TRÊN TAY MẪU LAPTOP MSI PRESTIGE PE60 6QD
 Bởi doankhien



Tổng hợp thông tin về Macbook Pro 2016: Thiết kế mới, cấu hình cũ
 Bởi doankhien



Nếu trở thành hiện thực, iPhone 7 space black sẽ đẹp đến mức nào?
 Bởi doankhien



Chiêm ngưỡng 2 màn hình cong ấn tượng của Galaxy Note 7
 Bởi doankhien

PHẢN HỒI GẦN ĐÂY

💬 doankhien trong [Cài Apache và NGINX proxy lên VPS tại DigitalOcean với ServerPilot](#)

💬 doankhien trong [Cài Apache và NGINX proxy lên VPS tại DigitalOcean với ServerPilot](#)

💬 Chris Ames trong [Page Comments](#)

CHUYÊN MỤC

Chọn chuyên mục ▼



Các bài viết có liên quan

Hình 3.4.2: Một bài viết trong website Công nghệ 24h

Trịnh Doãn Khiển – CT1601

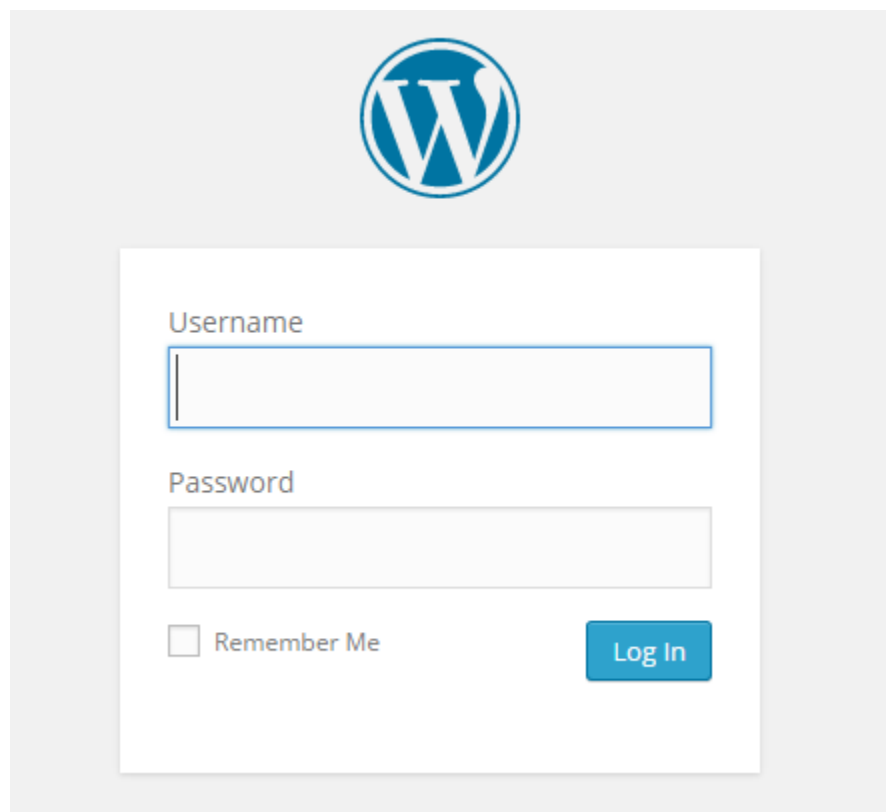
57

CHƯƠNG 4: QUẢN TRỊ NỘI DUNG TRANG TIN TRÊN WORDPRESS

4.1. Các thành phần cơ bản trong trang quản trị WordPress

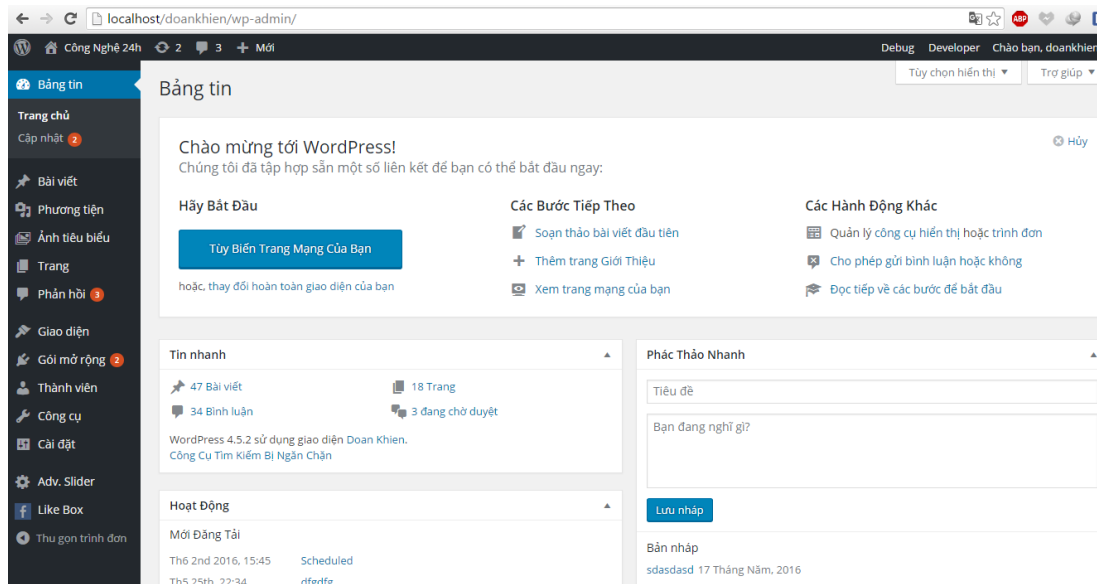
4.1.1 Truy cập vào trang quản trị WordPress

Để đăng nhập vào hệ thống quản trị của WordPress, chúng ta cần thêm **/wp-admin** sau đường dẫn đến website của chúng ta, ví dụ với website của em sẽ là: <http://localhost/doankhien/wp-admin>. Sau khi vào đường dẫn đó thì giao diện đăng nhập hiện ra:



Hình 4.1.1: Đăng nhập vào trang quản trị

Sau khi đăng nhập thành công, chúng ta sẽ thấy phần giao diện chính của trang quản trị (Dashboard) trong WordPress:



Hình 4.1.2: Trang Quản trị (Dashboard)

4.1.2 Các thành phần trong Dashboard

- **Cập nhật:** Thông báo về các chức năng cần updates như phiên bản của wordpress, plugins, themes,...
- **Bài viết:**
 - + All Post: Tất cả các bài viết.
 - + Post New: Viết một bài mới
 - + Categories: Chuyên mục, khi chúng ta vào phần này thì trong đó sẽ có các chức năng thêm, sửa, xóa chuyên mục bài viết.
 - + Tag: Thẻ, từ khóa cho bài viết.
- **Phương tiện:** Thư viện của chúng ta, chúng ta có thể thêm một hình ảnh, video hay bất kỳ tệp tin hỗ trợ nào lên thư viện này để sử dụng cho website. Trong Media có thể thêm, sửa và xóa
- **Trang:** Chúng ta có thể tạo mới một trang, sửa hoặc xóa trang đã tạo.
- **Phản hồi:** Đơn giản là dùng để quản lý các bình luận trên trang của chúng ta.
- **Giao diện:** Đây là phần rất quan trọng, nó quản lý rất nhiều phần. Hiểu một cách đơn giản về Giao diện nghĩa là nó sẽ quản lý những gì thuộc về cách thức hiển thị nội dung cho người dùng thấy như giao diện, các menu, widget hay màu nền trang... Trong Appearance có các phần sau:
 - + Themes: Thay đổi giao diện cho website, chúng ta có thể cài mới một giao diện hoặc dùng lại giao diện trước đây.

- + **Customize:** Tùy chỉnh trang web một cách trực quan hơn.
- + **Widgets:** Các tiện ích, chức năng mà bất kỳ trang web nào cũng cần đến như tìm kiếm, các bài viết gần đây, các bình luận mới nhất, danh sách thẻ (Tag), chuyên mục, lưu trữ.
- + **Menus:** Tạo, sửa, xóa menu, thêm các chuyên mục hoặc trang vào menu.
- + **Background:** Đơn giản là nền trang web, có thể chèn ảnh, video làm nền hoặc thay đổi màu nền cho website của chúng ta.
- + **Editor:** Trình soạn thảo, tức là đây là nơi chứa các tệp tin code của giao diện đang dùng, tùy mỗi giao diện mà các files code trong editor cũng khác nhau.
- **Gói Mở rộng:** Là một công cụ để cài đặt thêm các chức năng cho website của chúng ta. Trong Plugins chúng ta có thể tùy chỉnh thêm, hay xóa các plugins. Lưu ý: để chạy được plugins thì sau khi cài đặt cần phải kích hoạt cho nó. WordPress cung cấp các **Plugins miễn phí** và ngoài ra cũng có các **Plugins trả phí**.
- **Người dùng:** Quản lý người quản trị trang web, chúng ta có thể thêm một thành viên viết bài (Biên tập viên) cho website, thêm quản trị viên cho website....
- **Công cụ:** Các công cụ để quản lý dữ liệu
- **Cài đặt:** Cài đặt cấu hình cho trang Web

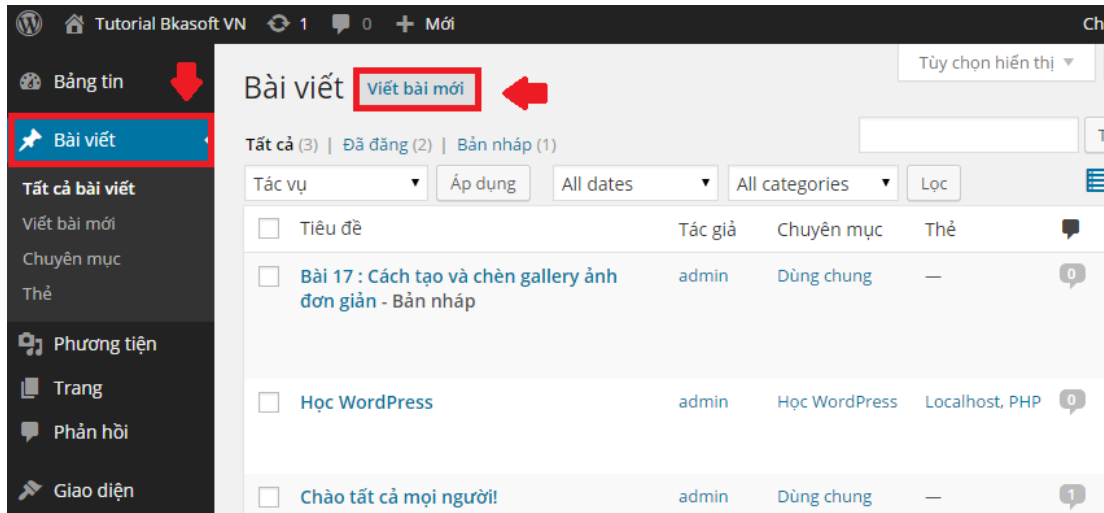
4.2. Quản trị website WordPress

4.2.1 Đăng bài viết

Bước 1: Truy cập vào trang Admin theo link:

- Trên website: [http://\[Tên-domain\].com/wp-admin/](http://[Tên-domain].com/wp-admin/)
- Trên Localhost: <http://localhost/wp-admin/>

Sau đó, vào **Bảng tin** -> **Bài viết** -> **Viết bài mới** để tạo bài viết mới.



Hình 4.2.1: Tạo bài viết mới

Bước 2: Soạn thảo bài viết

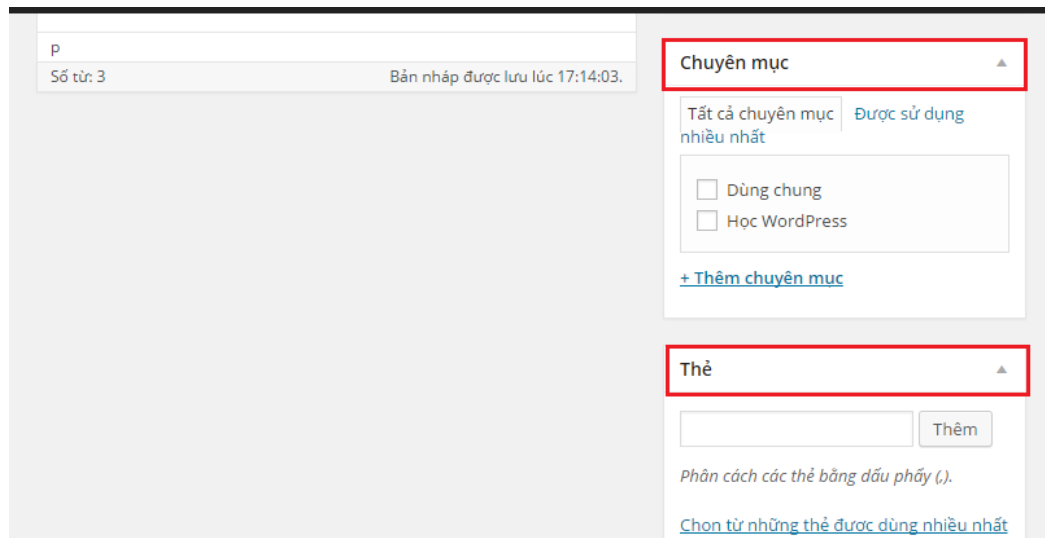
Chúng ta có thể gõ tiêu đề bài viết và nội dung bài viết theo ý của mình và mở khung soạn thảo nếu muốn thao tác với nhiều tính năng hơn.



Hình 4.2.2: Khung soạn thảo bài viết

Bước 3: Đưa bài viết vào chuyên mục phù hợp

- **Chuyên mục:** Chúng ta có thể tạo một chuyên mục mới hoặc tích chọn vào các chuyên mục đã có.
- **Thẻ:** Thêm các thẻ cho bài viết (dùng dấu phẩy “,” để ngăn cách giữa các thẻ).

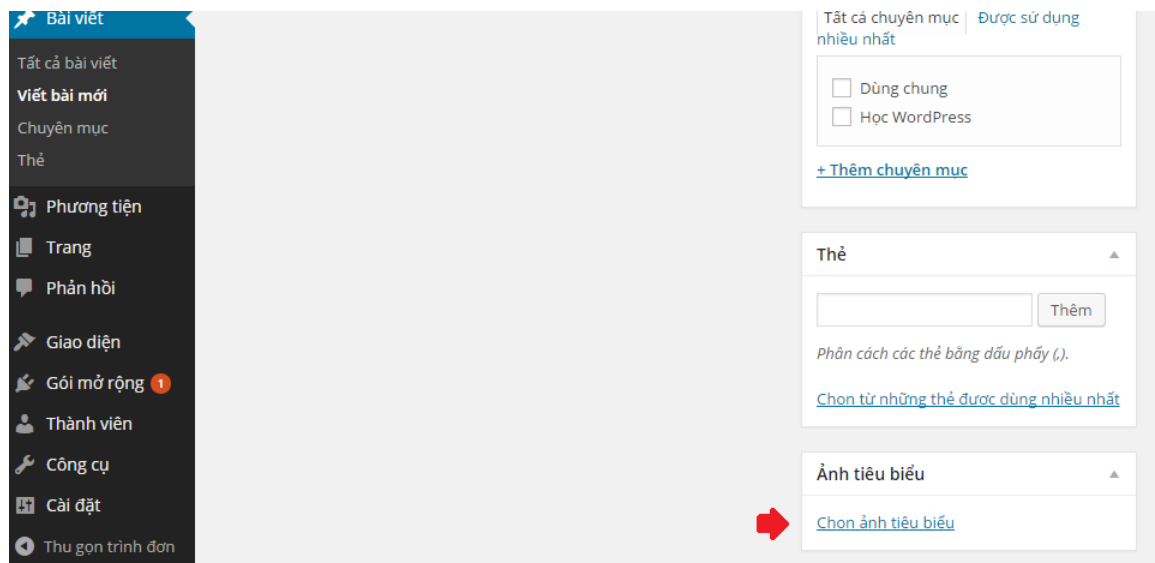


Hình 4.2.3: Thêm thẻ và chuyên mục cho bài viết

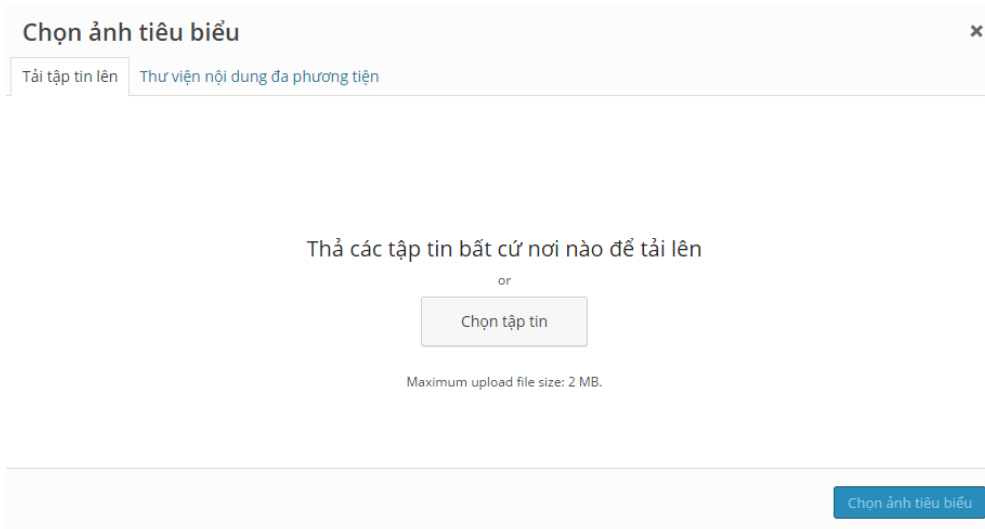
Bước 4: Chọn ảnh tiêu biểu

Đây là ảnh đại diện cho bài viết này, nó sẽ hiển thị ở trang chủ, trang chuyên mục và các trang được trả ra kết quả với ảnh đại diện và tiêu đề đi kèm. Chúng ta vào:

Chọn ảnh tiêu biểu -> Tải tệp tin lên -> Chọn ảnh tiêu biểu để tải lên.



Hình 4.2.4: Thêm ảnh đại diện cho bài viết



Hình 4.2.5: Chọn ảnh đại diện cho bài viết

Bước 5: Lưu bài viết hoặc xuất bản

Sau khi hoàn tất nội dung của bài viết và đưa vào chuyên mục phù hợp, chúng ta có thể tiến hành đăng bài viết để nó hiển thị lên trang chủ website hoặc lưu bản nháp bài viết đó:



Hình 4.2.6: Đăng bài viết

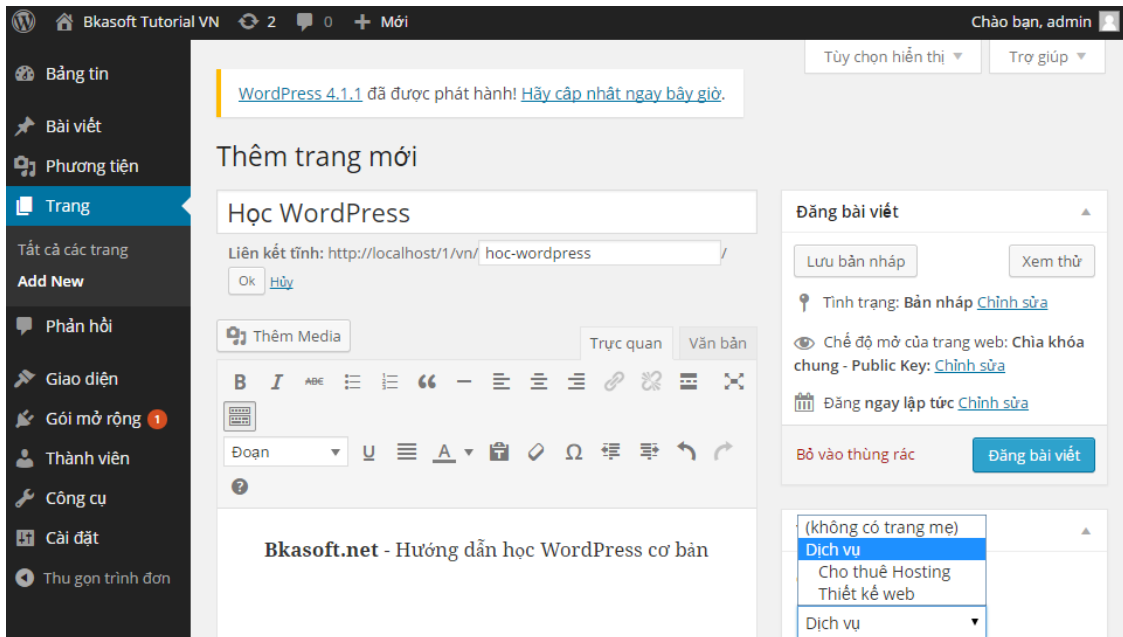
Sau khi đăng lên website xong, chúng ta truy cập ra trang chủ website sẽ thấy bài viết vừa đăng hoặc click vào **Xem bài viết** để xem bài viết.

4.2.2 Tạo Page

Một **Page** (trang) trong WordPress là một tính năng trong WordPress để tạo ra các trang độc lập hoặc cha con giống như thư mục cha và thư mục con, nhưng nó hiển thị như một bài viết độc lập không có các thẻ Tag hay các bài viết liên quan.

- **Các bước tạo Page**

Bước 1: Vào **Admin** -> **Pages** (Trang) -> **Add New** (Thêm trang mới)



Hình 4.2.7: Thêm trang mới

Bước 2: Nhập thông tin: Tên trang, mô tả, thuộc tính trang (trang mẹ, trang con), ảnh tiêu biểu như trong hình minh họa trên -> Chọn **Đăng bài viết**. Chúng ta có thể xem các trang hiện có và sửa chúng ở trong phần **Trang** -> **Tất cả các trang**.

4.2.3 Thiết lập và quản lý bình luận

Chúng ta thấy tính năng bình luận trong WordPress sẽ yêu cầu ta nhập: Họ và tên, email, website, viết bình luận và gửi bình luận.

Viết bình luận

Họ và tên (*)

Email (*)

Website

Viết bình luận...

Submit Comment

Hình 4.2.8: Khung bình luận trong website

- **Thiết lập quản lý bình luận trong WordPress**

+ Cách duyệt, xem và kiểm tra bình luận trong WordPress

Bước 1: Xem và kiểm tra các bình luận của độc giả ở chế độ chờ duyệt.

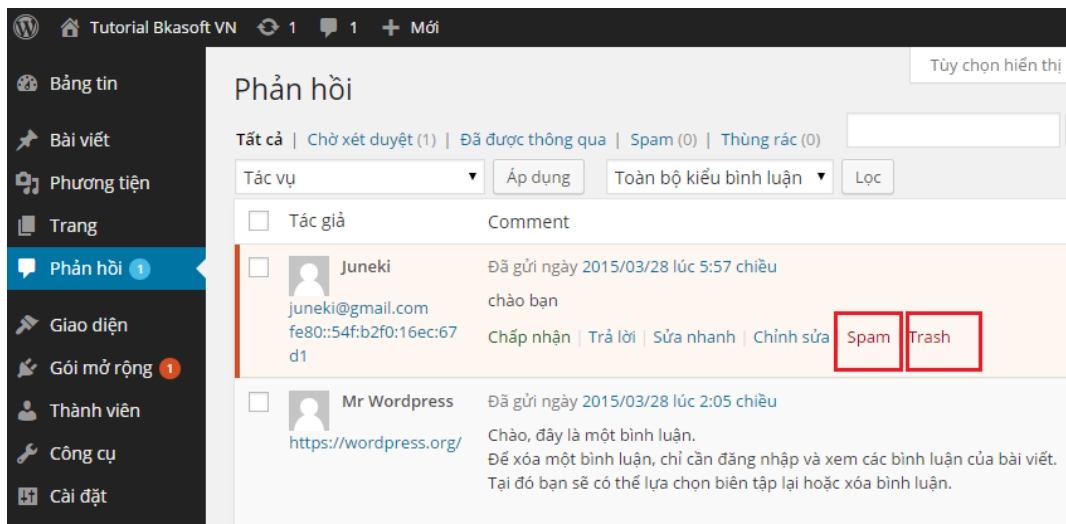
Chúng ta vào **Phản hồi > Chờ xét duyệt**



Hình 4.2.9: Bình luận chờ xét duyệt

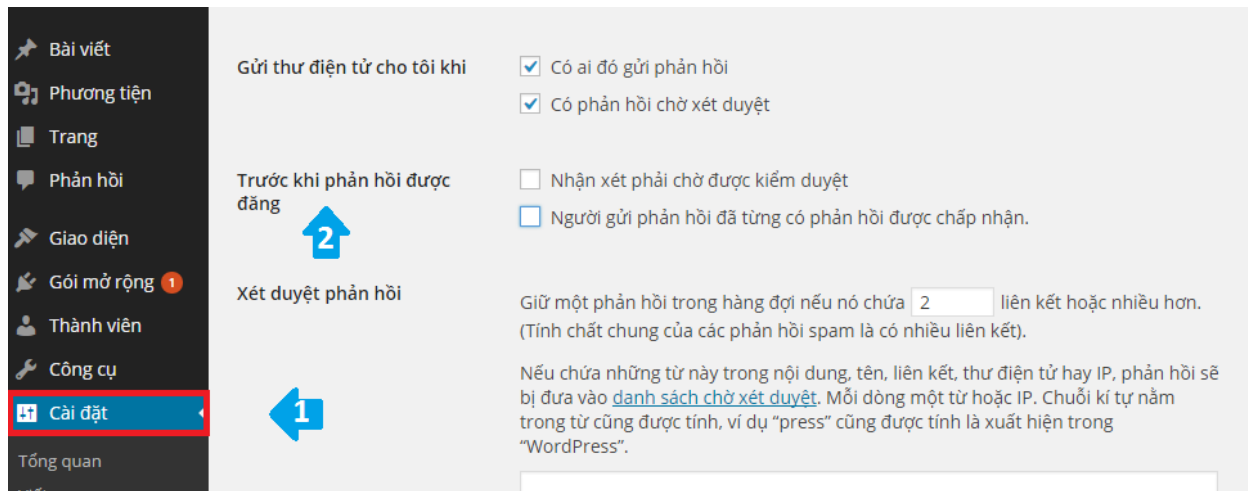
Bước 2 : Duyệt bình luận

Chúng ta ấn vào nút **Chấp nhận** để duyệt bình luận mà chúng ta muốn duyệt. Nếu chúng ta không muốn hiển thị hay xóa bình luận đó thì có thể ấn vào **Spam** hoặc **Thùng rác** (Trash).



Hình 4.2.10: Duyệt bình luận

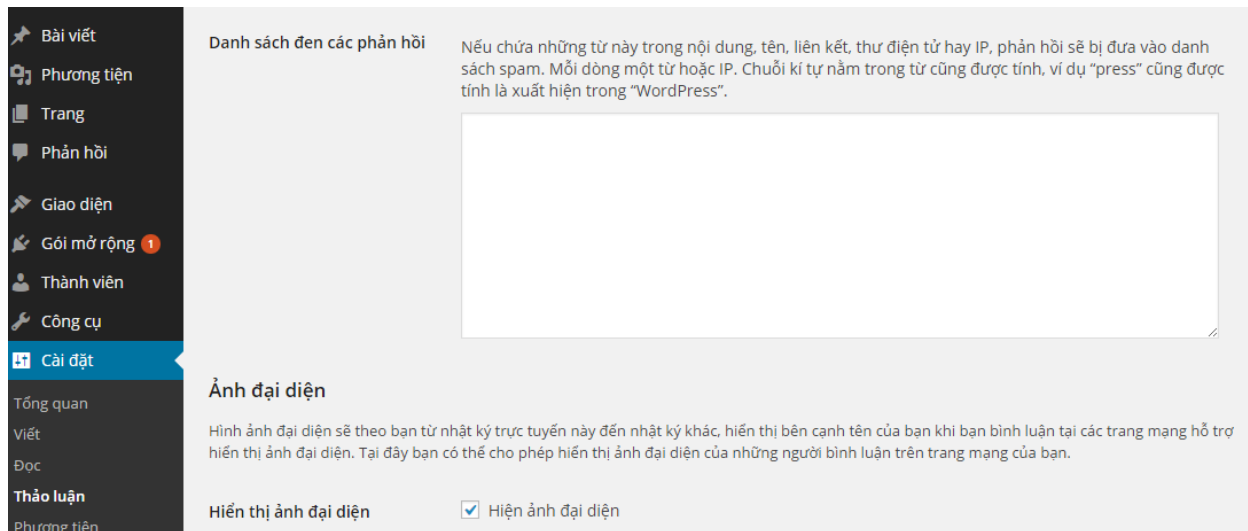
Nếu chúng ta muốn bình luận của họ sẽ được hiển thị ngay mà không cần chờ duyệt thì có thể vào: **Cài đặt -> Thảo luận -> Trước khi phản hồi được đăng** và bỏ chọn 2 dòng trong mục này.



Hình 4.2.11: Hình thị bình luận không cần chờ duyệt

- **Cấm các từ nhạy cảm khi bình luận trong WordPress**

Vào **Cài đặt > Thảo luận > mục Danh sách đen các phản hồi.**



Hình 4.2.12: Cấm bình luận nhạy cảm

4.2.4 Hướng dẫn sử dụng Widget

- **Widget là gì?**

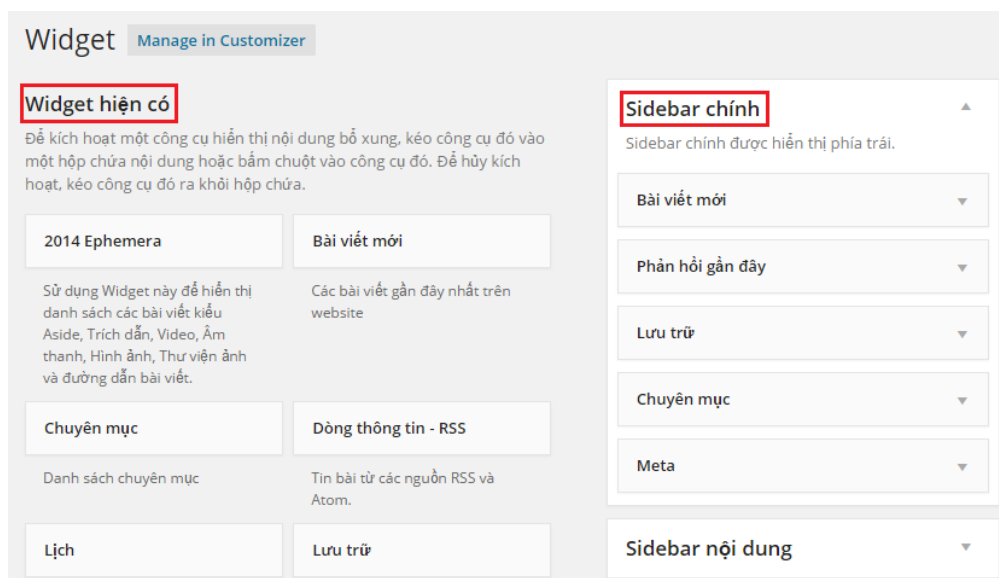
Widget có thể gọi là một tập hợp các chức năng mà bất kỳ một Website WordPress nào cũng phải cần dùng đến, mỗi chức năng tương ứng với một Widget để chúng ta chèn vào Sidebar chính của Theme.

- **Hướng dẫn sử dụng Widget**

+ **Giới thiệu khu vực Widget**

Để thao tác với Widget các chúng ta có thể vào theo đường dẫn sau:

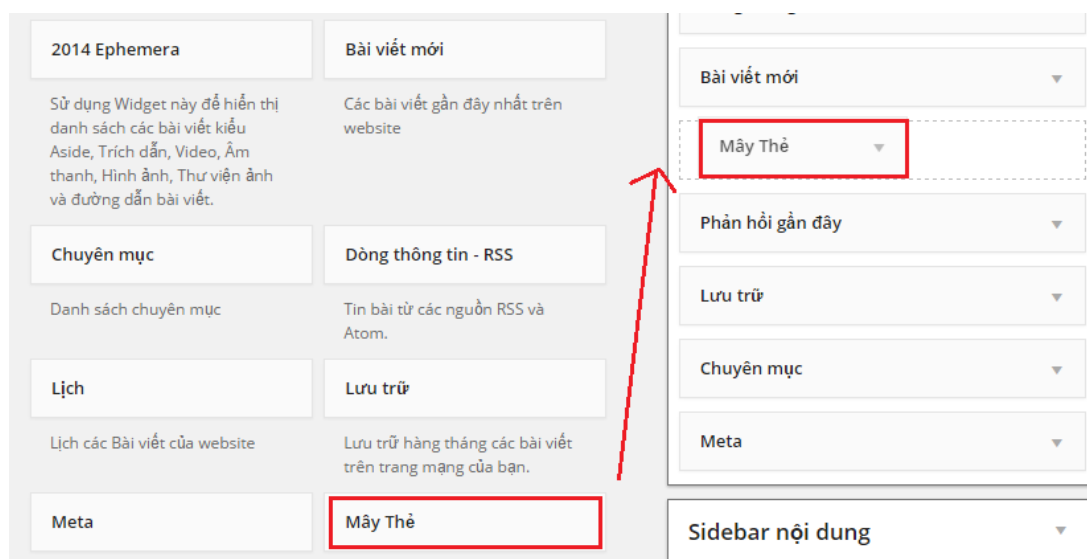
Bảng tin -> Giao Diện -> Widget



Hình 4.2.13: Quản lý widget

+ Sử dụng Widget

Để sử dụng Widget, chúng ta *click* và kéo Widget muốn sử dụng bên tay trái vào **Sidebar chính** để hiển thị Widget đó, như hình sau:



Hình 4.2.14: Thêm Widget

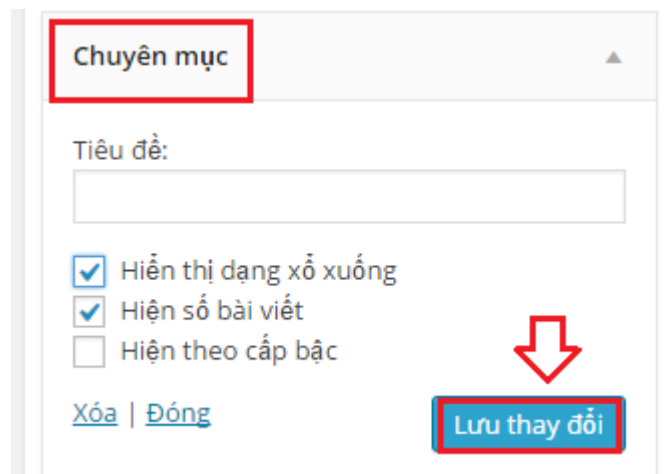
Và ngược lại, nếu chúng ta không muốn sử dụng một Widget nào đó trong Sidebar thì chúng ta có thể kéo Widget đó về mục **Widget không sử dụng** ở cuối trang, như hình sau:



Hình 4.2.15: Loại bỏ widget

+ Thiết lập Widget

Sau khi thêm xong Widget, mỗi Widget sẽ có các tùy chọn riêng biệt để chúng ta thiết lập nó nhằm hiển thị như ý chúng ta muốn, chúng ta bấm vào Widget cần thiết lập và chọn các tùy chọn, sau đó ấn **Lưu thay đổi** để lưu lại, ví dụ như hình dưới đây em thiết lập Widget *Chuyên Mục*.



Hình 4.2.16: Thiết lập widget

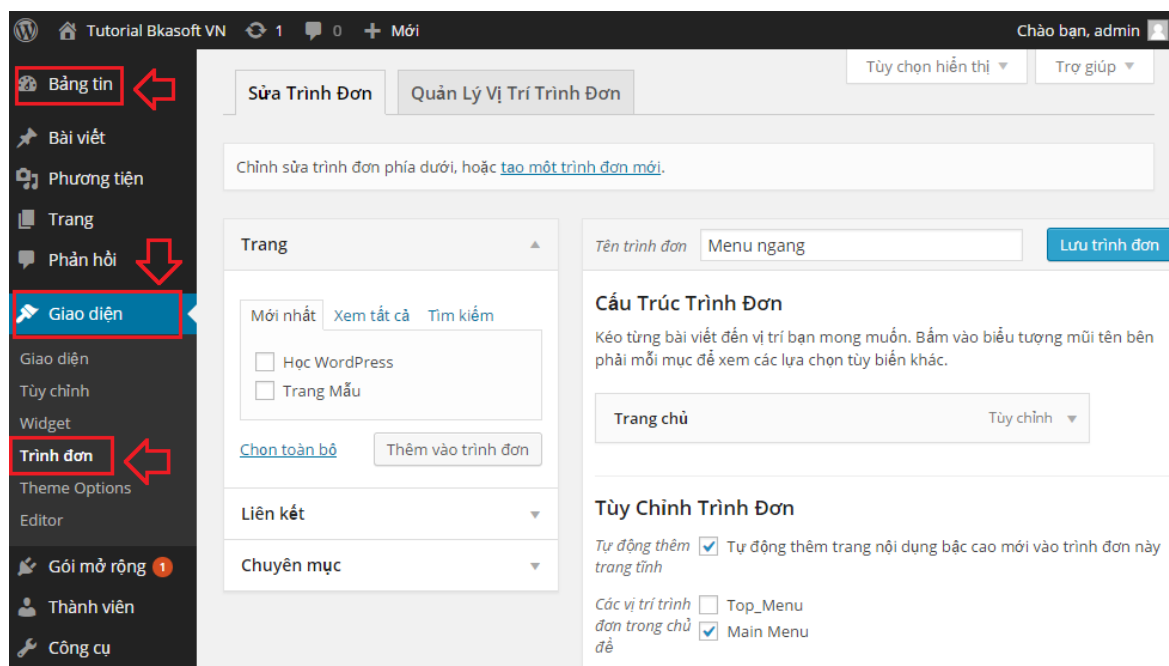
4.2.5 Làm việc với Menu trong WordPress

- Menu website là gì?

Menu website là thanh trình đơn hiển thị các liên kết trên Theme, do vậy số lượng Menu ít hay nhiều tùy thuộc vào mỗi Theme. Tùy theo Theme mà Menu sẽ hiển thị ở các vị trí khác nhau, ở trên hay ở dưới trang web.

- Cách truy cập vào trang quản lý Menu

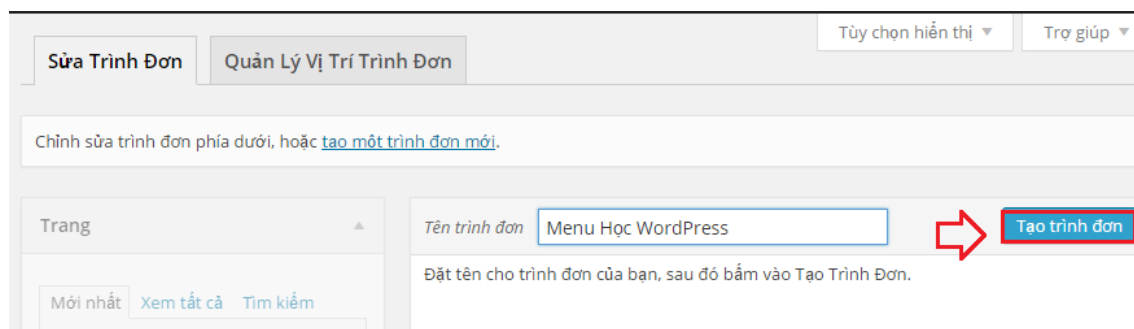
Trên **Bảng tin**, vào **Giao diện** -> **Trình đơn**



Hình 4.2.17: Giao diện Menu

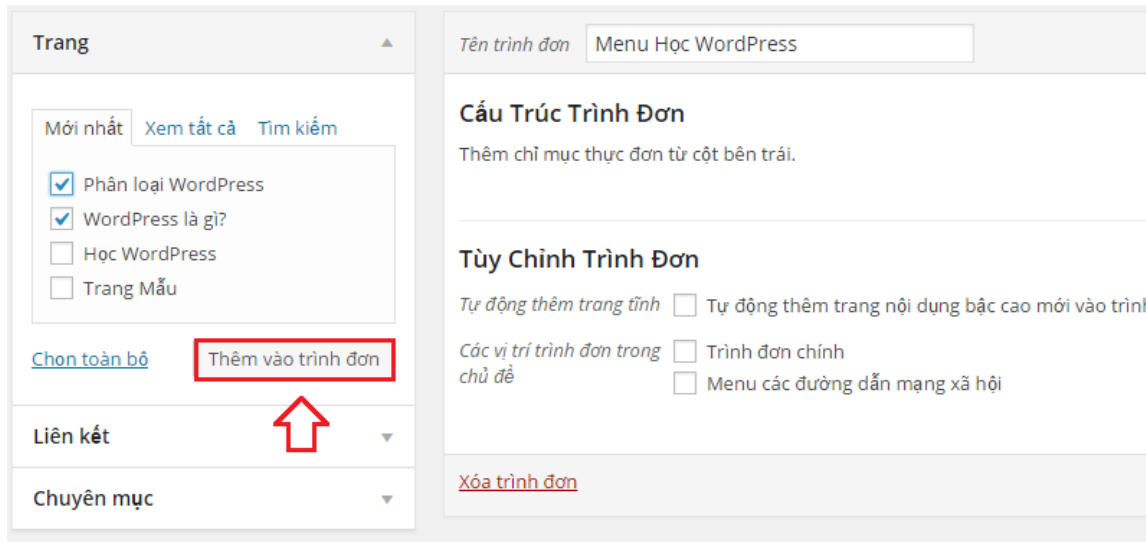
- Cách đưa Page, Link, Categories, Post vào một Menu

Bước 1: Tạo một Menu mới, chúng ta nhập tên tùy ý rồi ấn **Tạo trình đơn** như hình dưới đây:



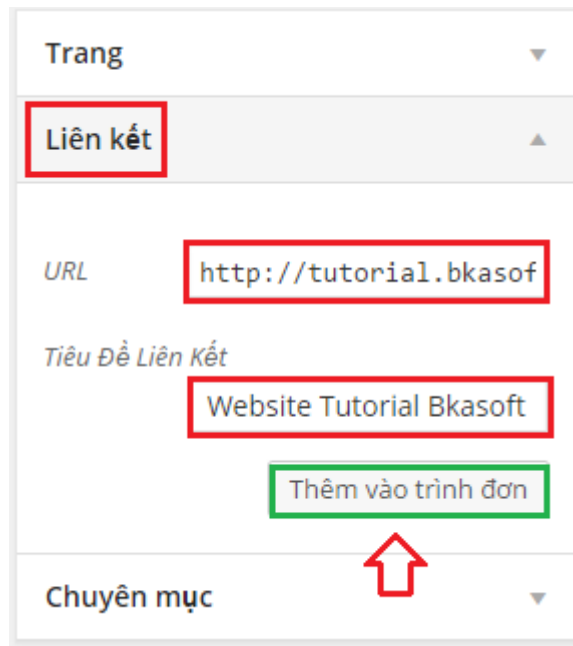
Hình 4.2.18: Tạo Menu

Bước 2: Đưa Page mà chúng ta cần thêm và ấn nút **Thêm vào trình đơn** để thêm Page đó vào **Menu Học WordPress**.



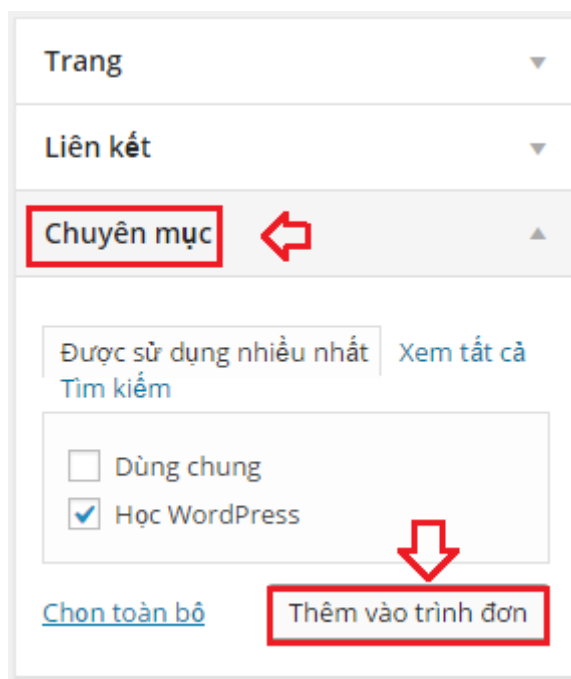
Hình 4.2.19: Thêm Trang vào Menu

Bước 3: Thêm một liên kết với tùy chọn **Link** chúng ta có thể xem hình dưới đây:



Hình 4.2.20: Thêm Liên kết vào Menu

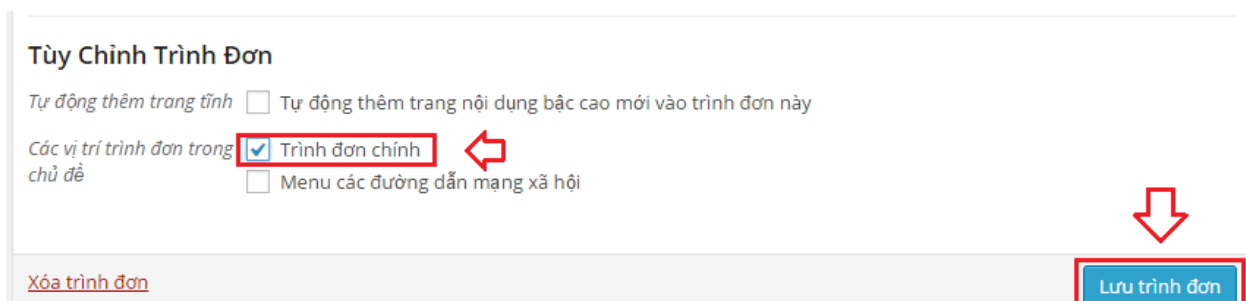
Bước 4: Thêm Catalog (Chuyên mục) vào Menu vừa tạo, xem hình sau:



Hình 4.2.21: Thêm Chuyên mục vào Menu

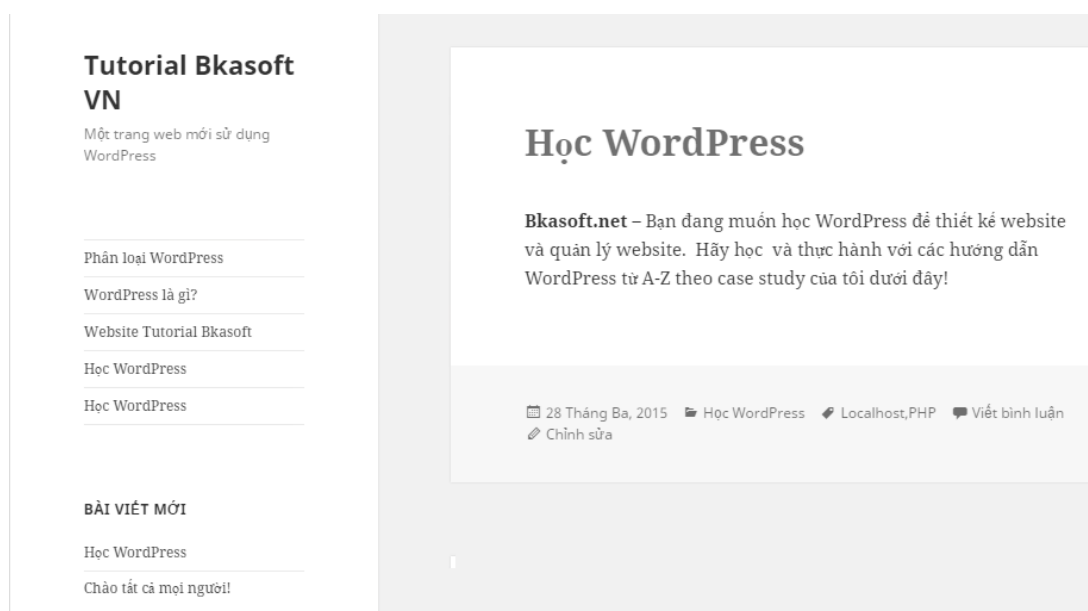
- Cách tạo một Menu Location

Sau khi chúng ta đã thêm đầy đủ các đối tượng vào Menu **Học WordPress** rồi, chúng ta có thể chọn Menu Location để áp dụng cho Menu này ở phần Menu Settings phía dưới và ấn **Lưu trình đơn**.



Hình 4.2.22: Tạo Menu Location

Bây giờ chúng ta hãy trở lại trang chủ và sẽ thấy website đã hiển thị Menu **Học WordPress**.



Hình 4.2.23: Menu xuất hiện trên trang chủ

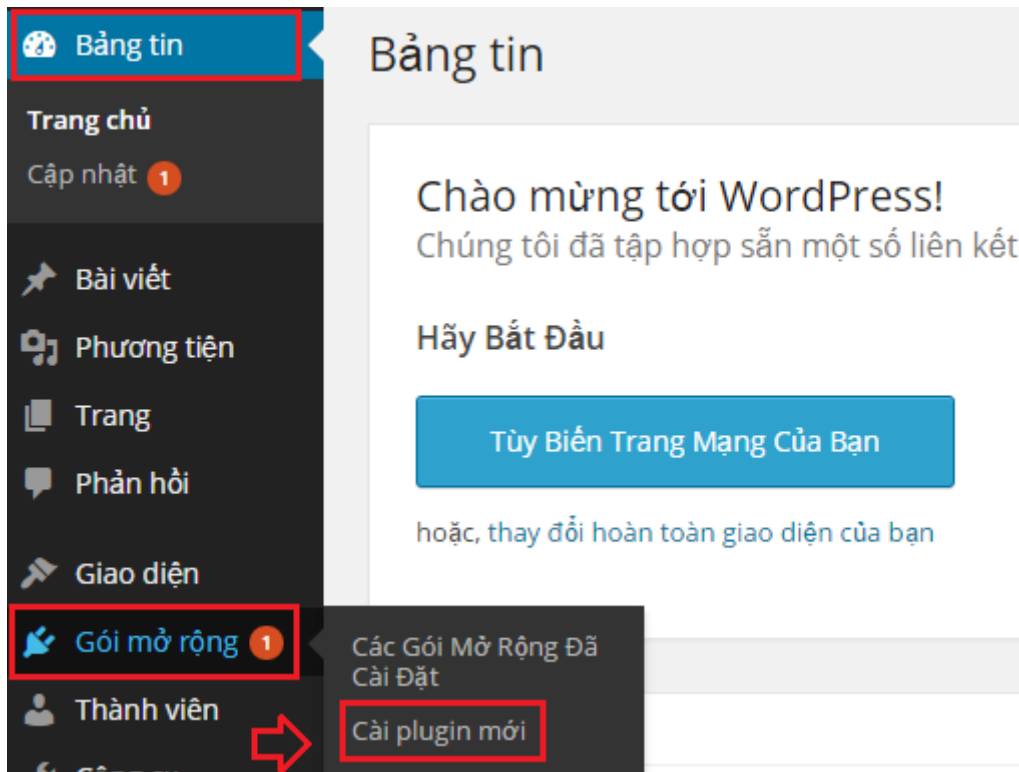
4.2.6 Hướng dẫn Plugin trong WordPress

- Plugin là gì?

Plugin được dịch theo nghĩa tiếng Việt là Ghim vào, dùng để bổ sung thêm một tính năng nào đó vào Website WordPress khi các tính năng trên WordPress không đáp ứng đủ nhu cầu của chúng ta. Chúng ta có thể cài thêm các Plugin để Website của mình trở nên hoàn hảo hơn vì một Website WordPress chuyên nghiệp không chỉ cần một Theme đẹp mà cần có cách sử dụng các Plugin hợp lý nữa. Hiện tại số lượng Plugin có trong thư viện WordPress.org rất nhiều và nó hoàn toàn miễn phí.

- Cách tìm và cài đặt Plugin

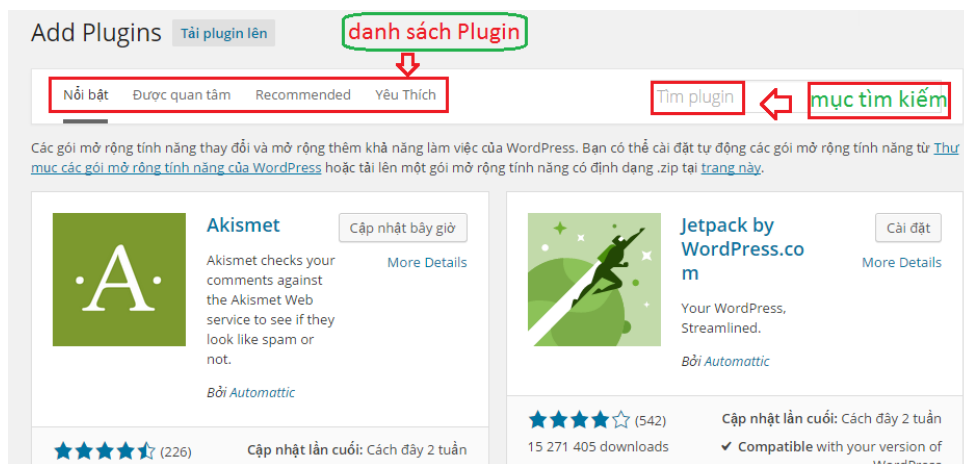
- + Cách tìm Plugin: Chúng ta truy cập vào **Bảng tin** → **Gói mở rộng** → **Cài Plugin mới**



Hình 4.2.23: Cài plugin mới

Chúng ta có thể tìm plugin bằng 2 cách:

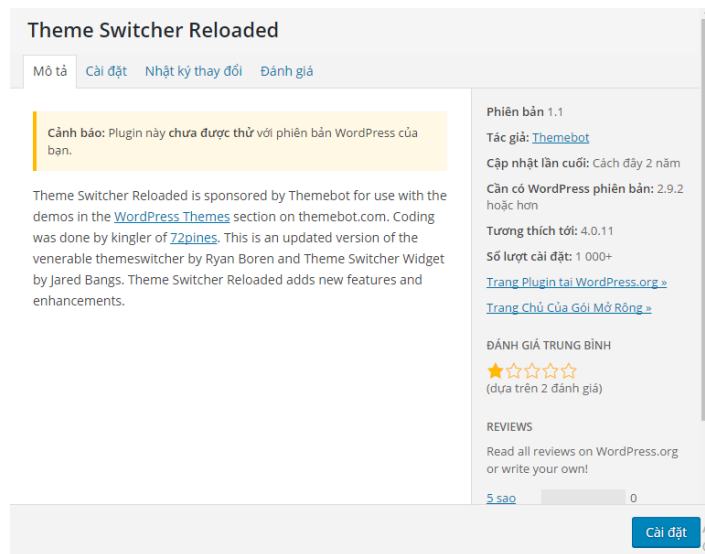
- Cách 1: Thông qua tên bằng cách điền tên ở khung Tìm Plugins bên tay phải rồi Enter
- Cách 2: Chúng ta xem danh sách các Plugin thông qua các bộ lọc như Nổi bật (Featured), Được quan tâm (Popular), Khuyến dùng (Recommended), Yêu thích (Favorites).



Hình 4.2.24: Danh sách plugin

+ Cách cài đặt Plugin

Sau khi tìm được plugin, chúng ta nhấn vào **Cài đặt** để tiến hành cài đặt Plugin:



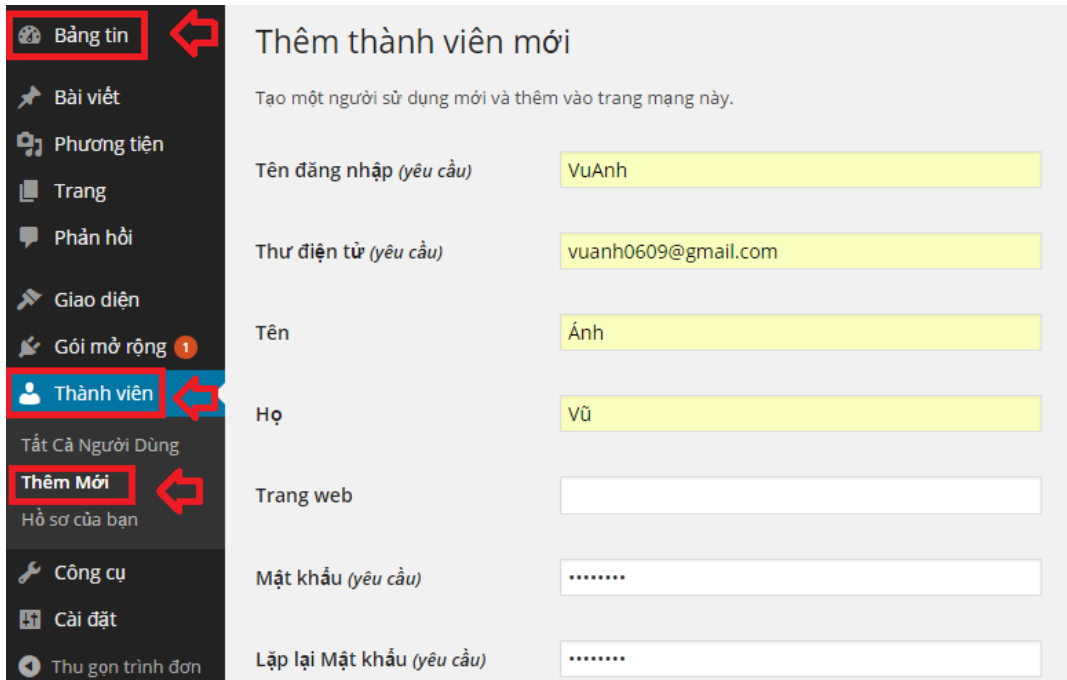
Hình 4.2.25: Giao diện cài đặt Plugin

4.2.7 Quản trị người dùng (users) trên WordPress

Mỗi nhóm người dùng có trong WordPress có thể được lập sẵn và có những quyền khác nhau, chúng ta có thể tạo ra các nhóm người dùng mới với các quyền do chúng ta tự thiết lập thông qua các plugin. Cụ thể là khi chúng ta muốn thêm một người dùng nào đó để họ đăng bài lên Website của chúng ta nhưng họ không có quyền sửa các bài khác thì chúng ta cho vào nhóm người dùng là **Thành viên đăng ký**, hoặc một người dùng chỉ được quyền sửa bài mà không có quyền tạo bài mới và đăng lên website thì chúng ta cho vào nhóm **Chỉnh sửa** chẳng hạn.

- Tạo người dùng (user) mới và quản lý

Chúng ta truy cập vào **Bảng tin -> Thành viên -> Thêm mới**

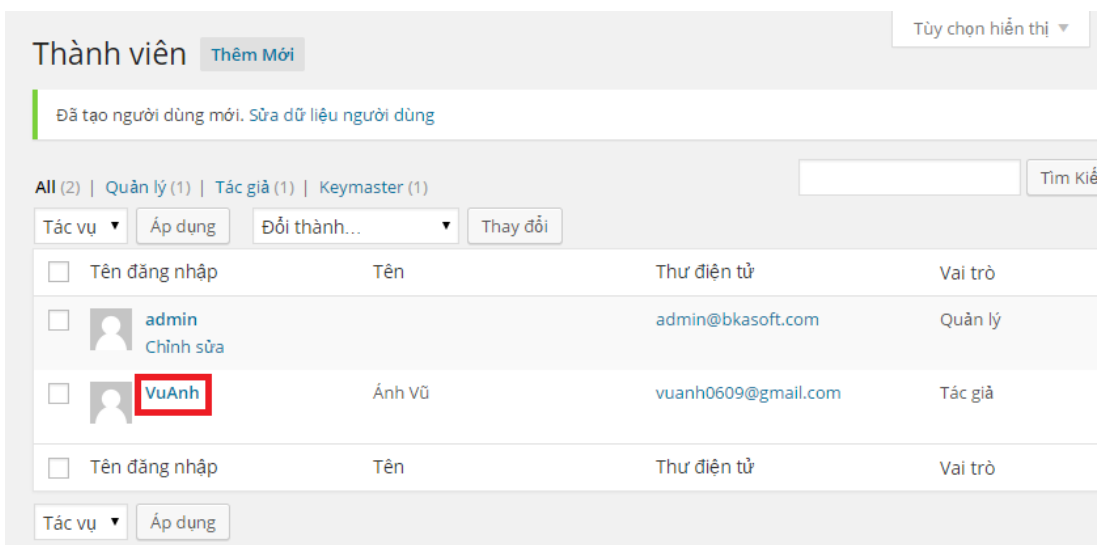


Hình 4.2.26: Giao diện thêm người dùng mới

Nhập xong các thông tin người dùng, chúng ta cũng ấn vào nút **Thêm thành viên mới** để bắt đầu tạo ra một người dùng mới.

- **Cách xem và sửa thông tin người dùng**

Sau khi tạo xong người dùng, chúng ta có thể xem danh sách các người dùng đang có trong website bằng cách truy cập vào **Bảng tin** -> **Thành viên** -> **Tất cả người dùng**.



Hình 4.2.27: Danh sách người dùng

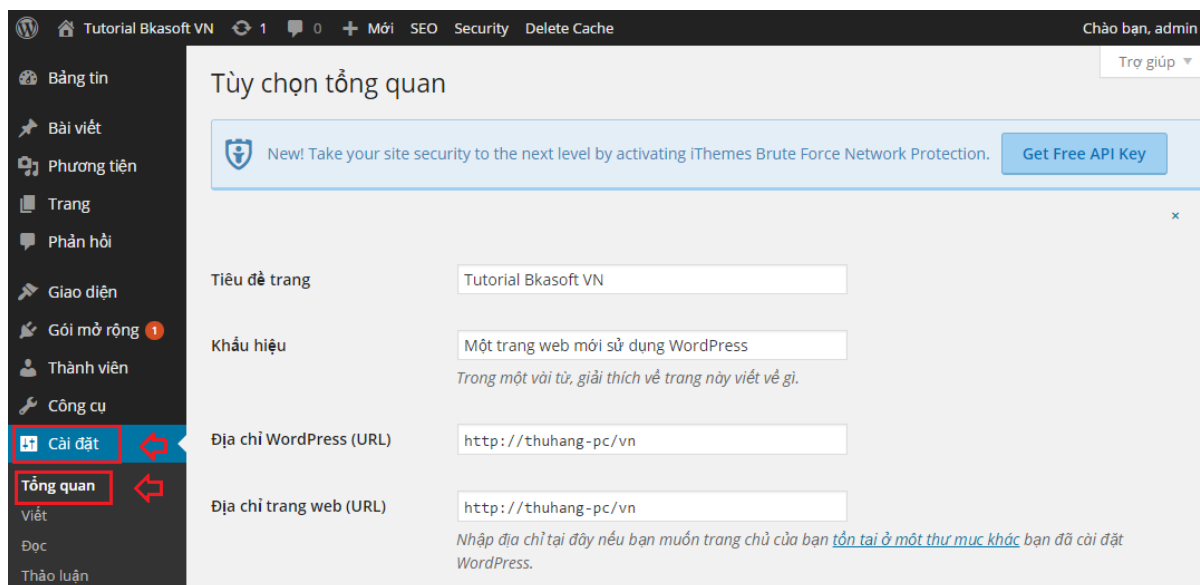
Để chỉnh sửa thông tin người dùng, chúng ta nhấn vào **Chỉnh Sửa** ở bên dưới tên người dùng và tiến hành chỉnh sửa và nhấn vào nút **Cập nhật** để lưu thay đổi.

Vai trò của mỗi nhóm người dùng:

- + **Biên tập viên** – Nhóm này có quyền đăng bài viết lên website (publish) và quản lý các post khác của những người dùng khác.
- + **Tác giả** – Nhóm này sẽ có quyền đăng bài lên website và quản lý các post của họ.
- + **Cộng tác viên** – Nhóm này sẽ có quyền viết bài mới nhưng không được phép đăng lên mà chỉ có thể gửi để xét duyệt (Save as Review) và quản lý post của họ.
- + **Thành viên đăng ký** – Người dùng trong nhóm này chỉ có thể quản lý thông tin cá nhân của họ.
- + **Quản lý** – Người dùng trong nhóm này đóng vai trò chủ chốt

4.2.8 Hướng dẫn mục Cài đặt

- Cài đặt -> Tổng quan

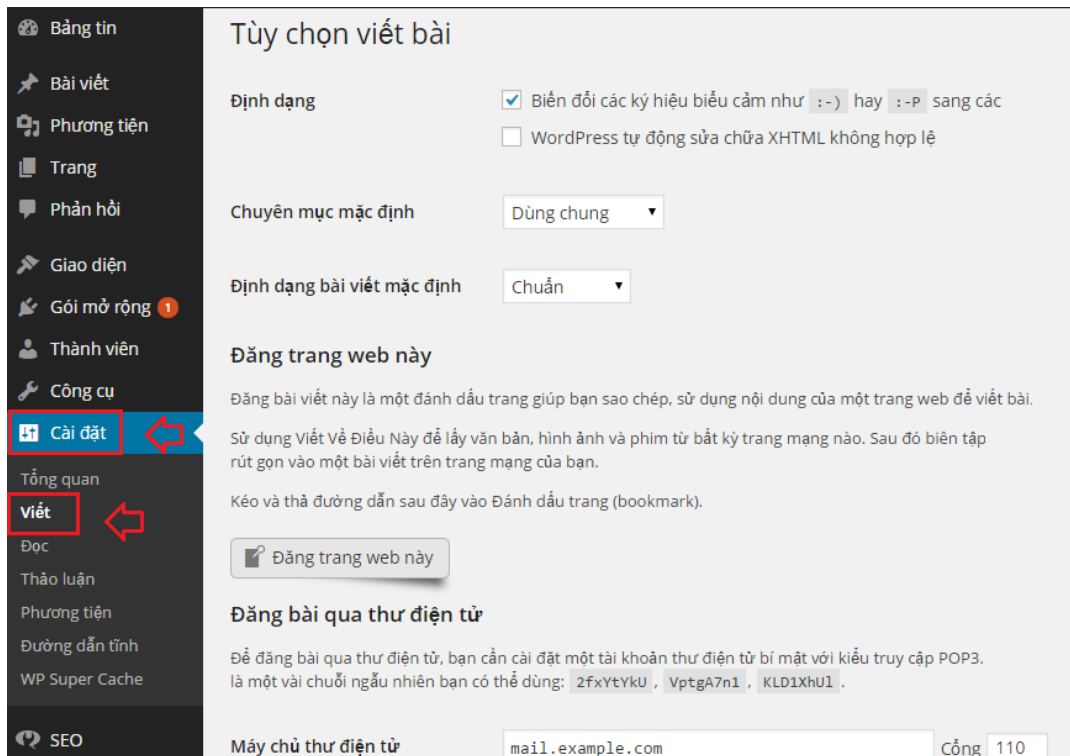


Hình 4.2.28: Cài đặt tổng quan

Trong khu vực này chúng ta có các thiết lập sau:

- + **Tiêu đề trang** (Site Title): Tên của website, tên này sẽ hiển thị mặc định trên tiêu đề website.
- + **Khẩu hiệu** (Tagline): Mô tả – slogan của website.
- + **Địa chỉ WordPress** (URL) [WordPress Address (URL)]: Địa chỉ của website WordPress hiện tại của chúng ta. Địa chỉ này sẽ tác động đến đường dẫn của Post và Page trên website.

- + **Địa chỉ trang Web (URL) [Site Address (URL)]**: Địa chỉ của website trang chủ của chúng ta, nếu chúng ta cài website WordPress làm trang chủ thì nên để giống với WordPress Address.
 - + **Địa chỉ E-mail (E-mail Address)**: Địa chỉ email của người quản trị website, các thông báo quan trọng về website sẽ gửi về đây.
 - + **Thành viên (Membership)**: Nếu đánh dấu vào mục Anyone can register, khách có thể tự đăng ký tài khoản người dùng trên website của chúng ta tại địa chỉ <http://domain/wp-login.php?action=register>.
 - + **Vai trò của thành viên mới (New User Default Role)**: Nhóm người dùng mà những người dùng mới đăng ký sẽ được đưa vào mặc định sau khi họ đăng ký xong.
 - + **Múi giờ (Timezone)**: Múi giờ mà chúng ta muốn sử dụng trên website, Việt Nam là GMT + 7.
 - + **Định dạng ngày (Date Format)**: Định dạng ngày tháng năm chúng ta muốn hiển thị trên website.
 - + **Tuần bắt đầu vào (Week Start On)**: Ngày mà chúng ta muốn nó là ngày đầu tiên của tuần.
 - + **Ngôn ngữ của trang (Site Language)**: Ngôn ngữ mà chúng ta muốn dùng trên website.
- **Cài đặt -> Viết**



Hình 4.2.29: Cài đặt Viết

Ý nghĩa của các thiết lập:

+ **Định dạng**(Formatting):

- **Biến đổi các biểu tượng chuyên đổi...(Convert emotions...):** Tự động chuyển đổi các ký hiệu biểu cảm sang dạng hình ảnh được hỗ trợ sẵn bởi WordPress.
- **WordPress tự động sửa chữa XHTML không hợp lệ(WordPress should correct...):** Nếu chúng ta đánh dấu vào phần này thì các nội dung trong post của chúng ta đều phải được tuân thủ theo cấu trúc cú pháp XHTML hợp lệ.

+ **Chuyên mục mặc định** (Default Post Category): Category mặc định của một bài Post nếu chúng ta quên chọn category khi đăng.

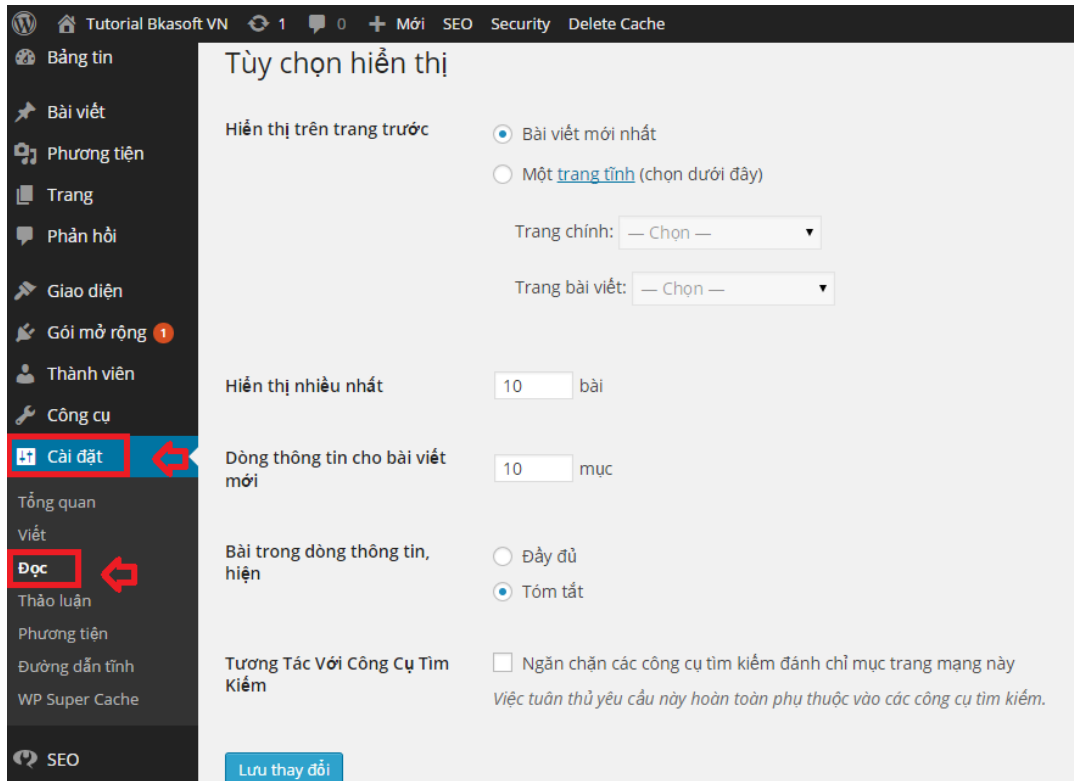
+ **Định dạng bài viết mặc định** (Default Post Format): Loại định dạng post mặc định khi đăng nếu chúng ta quên chọn.

+ **Đăng trang web này** (Press This): Tính năng copy nội dung của một địa chỉ nào đó và tự đăng lên website WordPress.

+ **Đăng bài qua thư điện tử** (Post via e-mail): Tính năng đăng bài thông qua e-mail, mình sẽ có hướng dẫn cụ thể phần này ở một bài riêng.

- + **Dịch vụ cập nhật (Update Service):** Các dịch vụ ping mà chúng ta muốn WordPress tự động gửi tín hiệu ping khi có bài mới.

- **Cài đặt -> Đọc**



Hình 4.2.30: Cài đặt đọc

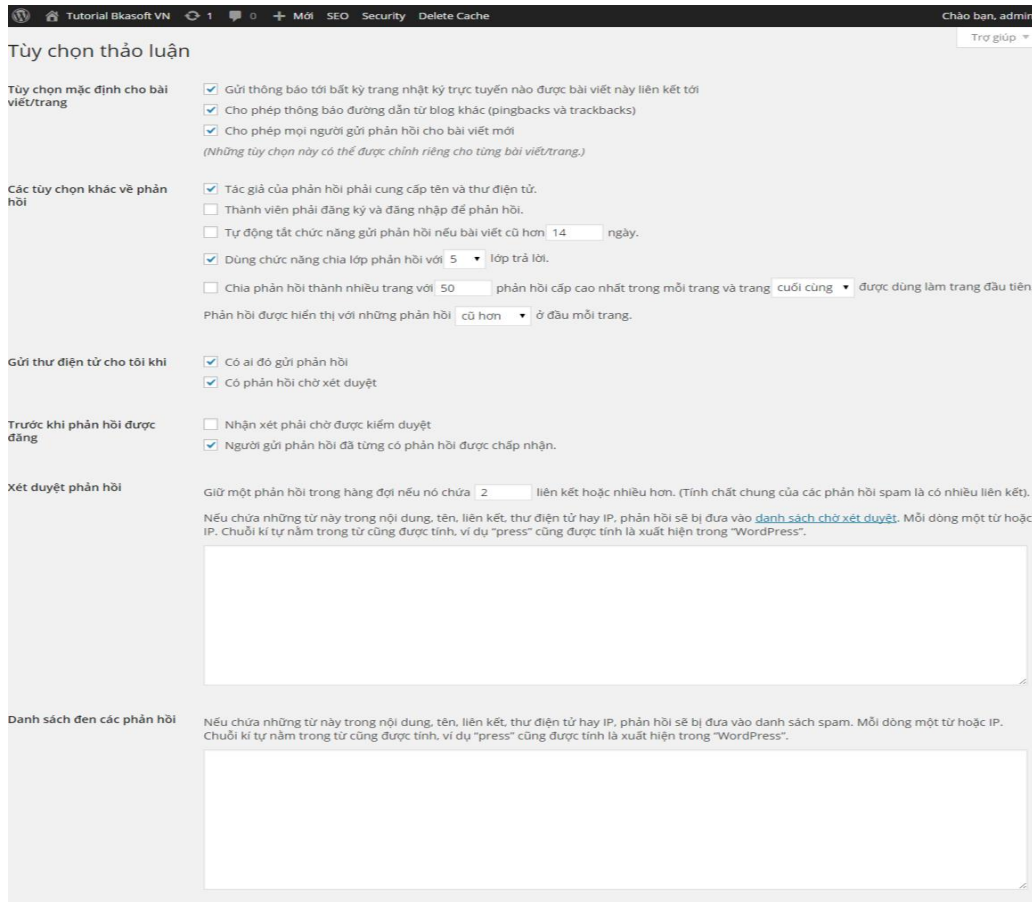
Ý nghĩa của các thiết lập:

- + **Hiển thị nhiều nhất (Blog pages show at most):** Số lượng post hiển thị ra trang blog. Hiện tại chúng ta cứ hiểu trang blog nghĩa là một trang hiển thị danh sách các post mới nhất trên website.
- + **Dòng thông tin cho bài viết mới (Syndication feeds show the most recent):** Số lượng post mới được hiển thị tại trang RSS Feed của website (<http://domain/feed>).
- + **Bài trong dòng thông tin, hiện (For each article in a feed, show):**
- + **Đầy đủ (Full text):** hiển thị nội dung trên RSS Feed với toàn nội dung.
- + **Tóm tắt (Summary):** hiển thị nội dung trên RSS Feed với bản rút gọn.
- + **Tương tác với công cụ tìm kiếm (Search Engine Visibility):** Nếu chúng ta đánh dấu vào phần này, nghĩa là các bot của các cỗ máy tìm kiếm (Google chẳng hạn) không thể đánh chỉ mục nội dung của chúng ta, từ đó website của chúng ta không hiển thị trên kết quả tìm kiếm tại Google.

- **Cài đặt -> Thảo luận**

Để truy cập vào khu vực này chúng ta làm theo hướng dẫn sau: **Cài đặt ->**

Thảo luận



Hình 4.2.31: Trang khu vực Cài đặt -> thảo luận

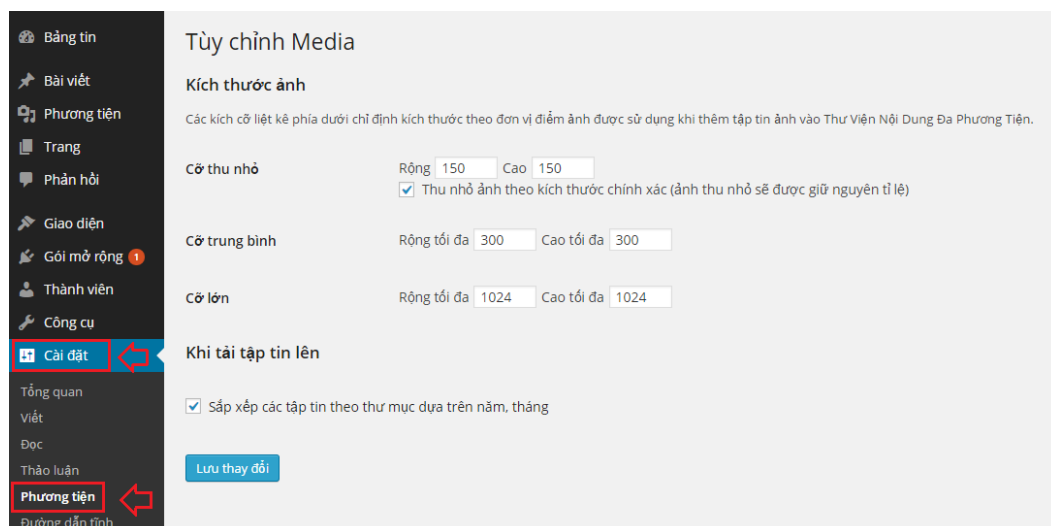
Ý nghĩa của các thiết lập:

- + **Tùy chọn mặc định cho bài viết/trang:** Các thiết lập trong đây sẽ chỉ đến việc bật tính năng liên quan tới bình luận ở dạng mặc định. Các thiết lập này có thể được sửa lại ở từng bài post/page riêng lẻ.
- + **Các tùy chọn khác về phản hồi:** Các thiết lập khác liên quan tới việc gửi bình luận.
- + **E-mail me whenever:** Thiết lập nhận e-mail thông báo về bình luận.
- + **Trước khi phản hồi được đăng:** Áp dụng trước khi bình luận được hiển thị lên.
- + **Xét duyệt phản hồi:** Tự động đưa bình luận vào trạng thái chờ duyệt nếu bình luận đó chứa từ khóa, liên kết, email hoặc địa chỉ IP có trong danh sách này. Mỗi quy tắc chặn đều phải được đặt ở một dòng riêng.

+ **Danh sách đen các phản hồi:** Các từ cấm bình luận. Mỗi từ cấm sẽ được khai báo bằng một dòng. Nếu bình luận nào đó chứa từ cấm thì sẽ bị đánh dấu là Spam.

+ **Ảnh đại diện:** Tùy chọn hiển thị ảnh avatar của người gửi bình luận.

- Cài đặt -> Phương tiện

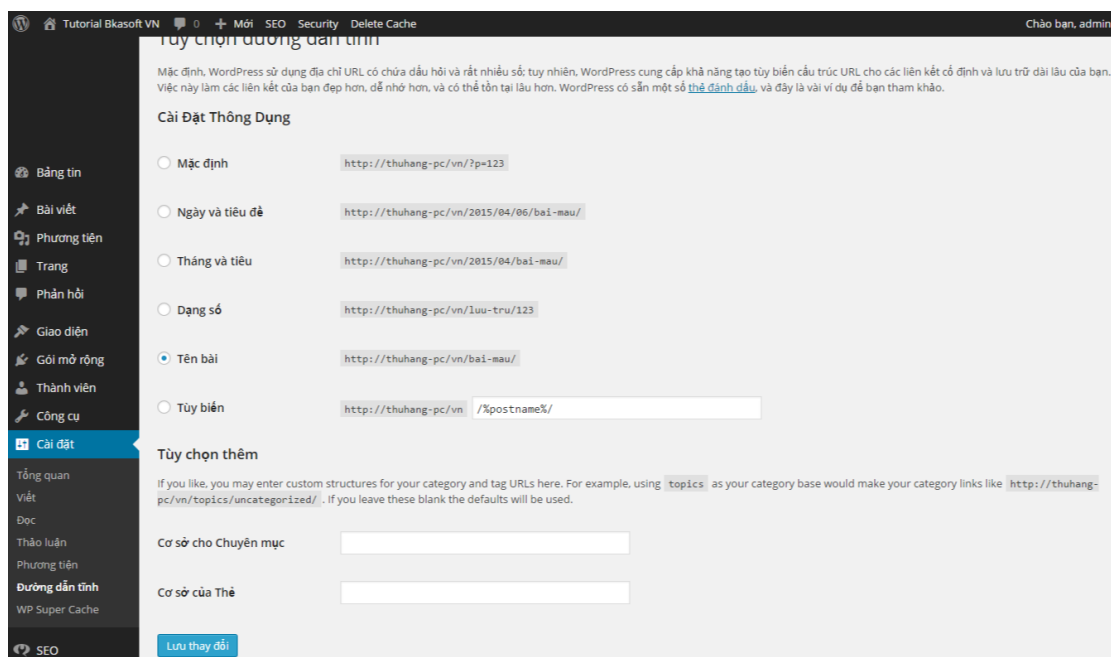


Hình 4.2.32: Trang khu vực tùy chỉnh Media

Các thiết lập có ý nghĩa như sau:

- + **Kích thước ảnh (Image sizes):** Các thiết lập trong này sẽ xác định kích thước ảnh mặc định của WordPress được sinh ra sau khi upload một tấm ảnh lên thư viện, bao gồm 3 loại sau:
 - **Cỡ thu nhỏ (Thumbnail size):** Kích thước ảnh loại nhỏ.
 - **Cỡ trung bình (Medium size):** Kích thước ảnh loại trung bình.
 - **Cỡ lớn (Large size):** Kích thước ảnh loại lớn.
- + **Khi tải tập tin lên (Uploading Files):** Thiết lập liên quan tới việc upload tập tin.
 - **Sắp xếp các tập tin theo thư mục dựa trên năm, tháng (Organize my uploads into month – and year – based folder):** Tự động đưa các tập tin được upload lên vào thư mục với cấu trúc ngày tháng so với thời gian upload.

- Cài đặt -> Đường dẫn tĩnh



Hình 4.2.33: Trang khu vực Cài đặt -> Đường dẫn tĩnh

Các thiết lập trong khu vực này có ý nghĩa như sau:

- + **Cài đặt thông dụng** (Common Settings): Các thiết lập thông dụng.
 - **Mặc định** (Default): Cấu trúc đường dẫn mặc định (đường dẫn động).
 - **Ngày và tiêu đề** (Day and name): Cấu trúc đường dẫn với kiểu hiển thị đầy đủ ngày tháng đăng post và tên post.
 - **Tháng và tiêu đề** (Month and name): Cấu trúc đường dẫn với kiểu hiển thị tháng, năm và tên post.
 - **Dạng số** (Numeric): Cấu trúc đường dẫn hiển thị ID của post thay vì tên.
 - **Tên bài** (Post name): Chỉ hiển thị tên post trên đường dẫn
 - **Tùy biến** (Custom Structure): Tùy chỉnh cấu trúc đường dẫn tùy ý, xem thêm phần cuối bài viết.
- + **Tùy chọn thêm** (Optional): Các thiết lập tùy chọn không bắt buộc.
 - **Cơ sở cho chuyên mục** (Category base): Tên đường dẫn mẹ của các đường dẫn tới trang category.
 - **Cơ sở của thẻ** (Tag base): Tên đường dẫn mẹ của đường dẫn tới các trang tag.

KẾT LUẬN

Đề tài “Tìm hiểu kỹ thuật làm giao diện người dùng cho hệ quản trị nội dung WordPress và ứng dụng” là một đề tài đang được quan tâm nhiều trong thời gian này do Hệ quản trị nội dung WordPress đang phát triển rất mạnh mẽ. Trong quá trình khảo sát và tìm hiểu để phân tích đề tài này dưới sự chỉ dẫn của giảng viên Th.S Đỗ Văn Chiếu, em đã hiểu được phần nào và thấy được các công việc cần phải làm để thiết kế giao diện cho website sử dụng hệ quản trị nội dung WordPress. Tuy gặp phải nhiều khó khăn nhưng em đã cố gắng để hoàn thành đề tài các tiêu chí sau:

- Hiểu được cấu trúc theme trong WordPress
- Tạo được một website với giao diện riêng theo ý muốn
- Đưa website lên Internet và quản trị.

Giao diện được xây dựng đảm bảo đủ tiêu chuẩn của một Theme trong WordPress với những Template cần thiết và cơ bản. Mặc dù đã cố gắng rất nhiều trong quá trình khảo sát tìm hiểu thiết kế giao diện, nhưng sẽ không tránh khỏi những thiếu sót. Vì vậy em mong quý thầy cô cũng như những ai quan tâm đến đề tài này chỉ dẫn và góp ý kiến cho em, để em hoàn thiện giao diện một cách đầy đủ.

CÁC TÀI LIỆU THAM KHẢO

- [1] Kỹ Thuật Và Thủ Thuật Lập Trình PHP, Th.S Nguyễn – Lương Phúc và nhóm tin học thực dụng, NXB Hồng Đức.
- [2] Tạo Website Hướng Database Bằng PHP&MySQL, VN-GUIDE (Tổng hợp và biên dịch), NXB Thống Kê .
- [3] Giáo Trình - Lập Trình Ứng Dụng Web Với PHP (Tập 1, Tập 2), Khuất Thùy Dương, NXB Đại Học Quốc Gia TP.HCM .
- [4] Sổ Tay PHP & MySQL, Nguyễn Trường Sinh chủ biên, NXB Lao Động – Xã Hội.