



TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC
Khoa Công Nghệ Thông Tin

CHUYÊN ĐỀ CMS

BÁO CÁO ĐÖ ÁN : WEBSITE BÁN HÀNG

Mã lớp học phần: CNC10735002

Giảng viên hướng dẫn: Bùi Thị Phương Thảo

Họ tên sinh viên - Mã số sinh viên:

NHÓM 2			
STT	MSSV	HỌ VÀ TÊN	CHỨC VỤ
1	18211TT3316	NGÔ NGỌC ĐIỆN	THÀNH VIÊN
2	18211TT3078	CAO THÀNH ĐẠT	THÀNH VIÊN
3	18211TT5028	HỒ KHẮC LĨNH	THÀNH VIÊN
4	18211TT3018	NGUYỄN VĂN SƠN	NHÓM TRƯỞNG
5	18211TT3327	NGUYỄN VĂN THÁI	THÀNH VIÊN

THÁNG 12/2020

MỤC LỤC

DANH SÁCH HÌNH ẢNH.....	
LỜI MỞ ĐẦU	
CHƯƠNG I: GIỚI THIỆU CMS WORDPRESS	1
1. Giới thiệu Wordpress	1
2. Ưu điểm và nhược điểm của Wordpress.....	1
3. Quản trị website với Wordpress.....	2
4. Hướng dẫn cài đặt Wordpress.....	3
5. Cài đặt XAMPP	3
6. Cài đặt XAMPP trên MacOS	9
CHƯƠNG II: PHÁT TRIỂN PLUGIN.....	16
1. Giới thiệu về plugin	16
2. Hướng dẫn cách phát triển plugin	16
CHƯƠNG III: PHÁT TRIỂN GUTENBERG BLOCK.....	19
1. Cài đặt Theme	19
2. Giới thiệu gutenberg block	19
3. Hướng dẫn cách phát triển gutenberg block.....	21
CHƯƠNG IV: PHÁT TRIỂN WIDGET	29
1. Giới thiệu widget.....	29
2. Hướng dẫn cách phát triển widget.....	29
CHƯƠNG V: PHÁT TRIỂN THEME TRONG WORDPRESS.....	33
1. Giới thiệu theme.....	34
2. Hướng dẫn cách phát triển theme.....	34
3. Tạo Theme ‘thủ công’	34
CHƯƠNG VI: BÁO CÁO ĐỒ ÁN.....	41
1. Giới thiệu đồ án.....	41
2. Header và Footer	43
3. Footer.....	44
4. Trang chủ	45
5. Trang Sản Phẩm (Tất cả sản phẩm)	52
6. Trang Danh Mục (Danh mục sản phẩm)	54
7. Trang chi tiết sản phẩm	58
8. Trang Giỏ Hàng và Thanh Toán.....	60
TỔNG KẾT.....	71
KẾT LUẬN.....	73
TÀI LIỆU THAM KHẢO.....	

DANH SÁCH HÌNH ẢNH

Hình 1. Tiến trình thực hiện project	1
Hình 2. Logo wordpress	1
Hình 3. Chọn phiên bản download	3
Hình 4. Bỏ chọn phần “Learn more about Bitnami for XAMPP“.	6
Hình 5. Bảng điều khiển của XAMPP.....	7
Hình 6. Khởi động thành công Apache và MySQL	8
Hình 7. Giao diện XAMPP trên Macos	9
Hình 8. Khởi động MySQL và Apache	10
Hình 9. Tệp Wordpress sau khi download về.....	10
Hình 10. Folder wordpress tại folder Xampp	11
Hình 11.Yêu cầu cấu hình cho wordpress	11
Hình 12. Thiết lập database cho Wordpress	12
Hình 13. Lỗi chưa có Database	12
Hình 14. Tạo mới Database trong phpmyadmin	13
Hình 15. Cài đặt wordpress thành công.....	14
Hình 16. Đăng nhập vào wp-admin.....	14
Hình 17. Giao diện wp-admin	15
Hình 18. Create thư mục trong themes.....	16
Hình 19. Thiết lập thông tin cho plugin	16
Hình 20. Hiển thị plugin trong wp.....	17
Hình 21. Khởi tạo shortcode g2-rand-posts.....	17
Hình 22. Thêm shortcode vào bài post.....	18
Hình 23. Kết quả sau khi khởi tạo	18
Hình 24. So sánh Gutenberg và Current Editor.....	19
Hình 25. Nội dung cũ convert sang Block	20
Hình 26. Tạo folder Gutenberg Block	21
Hình 27. Cấu trúc file plugin.php	22
Hình 28. Plugin hiển thị trên trình duyệt wp-admin.....	22
Hình 29. Tạo file test-block.js	23
Hình 30. Code của file test-block.js	23
Hình 31. Sau khi click “Kích hoạt” plugin trên wp-admin (trình duyệt)	23
Hình 32. Truy cập vào Tạo bài viết mới.....	24
Hình 33. Hiển thị danh sách block của WordPress	24
Hình 34. Mở Block mới vừa tạo (Email G2).....	25
Hình 35. Kết quả khi mở block vừa tạo.....	25
Hình 36. Thêm email đúng định dạng	26
Hình 37. Kết quả thêm email đúng định dạng.....	26
Hình 38. Thêm email sai định dạng.....	27
Hình 39. Kết quả thêm email sai định dạng	27
Hình 40. Cập nhật email sai.....	28

Hình 41. Widget wordpress	29
Hình 42 . Thêm Widget cách 1	30
Hình 43.Thêm Widget cách 2.....	30
Hình 44. Custom widget và save	31
Hình 45. Kết quả sau khi thêm Widget	31
Hình 46. Hình ảnh giao diện chính và CSS khi trỏ chuột	32
Hình 47. Chi tiết sản phẩm	32
Hình 48. Danh mục và menu sản phẩm.....	33
Hình 49. Khởi tạo file chứa toàn bộ theme trong themes.....	35
Hình 50. File giới thiệu thông tin	36
Hình 51. Giao diện sau khi tạo file	36
Hình 52. Thêm logo vào theme đã tạo.....	37
Hình 53. Kết quả sau khi thêm	37
Hình 54. Kích hoạt theme	38
Hình 55. Sau khi kích hoạt hành công.....	38
Hình 56. Show kết quả sau khi kích hoạt	39
Hình 57. Code bối cảnh của một trang	39
Hình 58. Cấu trúc thư mục hoàn chỉnh.....	42
Hình 59. Header của trang web	43
Hình 60. Tạo giao diện header trong header.php (part 1 - header-top)	43
Hình 61. Tạo giao diện header trong header.php (part 2 - header-top)	44
Hình 62. Giao diện footer	44
Hình 63. Hiển thị giao diện footer trong footer.php (part 1 - logo)	45
Hình 64. Hiển thị giao diện footer trong footer.php (part 2)	45
Hình 65. Hiển thị sản phẩm mới nhất bằng slide (10 sản phẩm)	46
Hình 66. Cấu trúc thư mục template-parts	46
Hình 67. Gọi file trong template-parts vào index.php.....	46
Hình 68. Khởi tạo slide hiển thị sản phẩm mới nhất trong content.php (part 1).....	47
Hình 69. Hàm lấy 10 sản phẩm mới nhất trong functions.php.....	48
Hình 70. Các button di chuyển trong content.php	48
Hình 71. Giao diện phần hiển thị 6 sản phẩm theo danh mục	49
Hình 72. Hiển thị 6 sản phẩm theo danh mục trong functions.php (part 1)	49
Hình 73. Hiển thị 6 sản phẩm theo danh mục (part 2)	50
Hình 74. Hiển thị 6 sản phẩm theo danh mục (part 3)	50
Hình 75. Hàm lấy sản phẩm theo danh mục trong functions.php	51
Hình 76. Hàm tính phầm trăm giảm giá sản phẩm trong functions.php	51
Hình 77. Gọi hàm lấy 6 sản phẩm theo danh mục vào index.php	51
Hình 78. Giao diện tất cả sản phẩm (part 1).....	52
Hình 79. Giao diện tất cả sản phẩm (part 2).....	53
Hình 80. Khởi tạo trang all-product.php	53
Hình 81. Thêm trang mới trong wp-admin.....	54
Hình 82. Tạo trang tất cả sản phẩm trong wp-admin	54

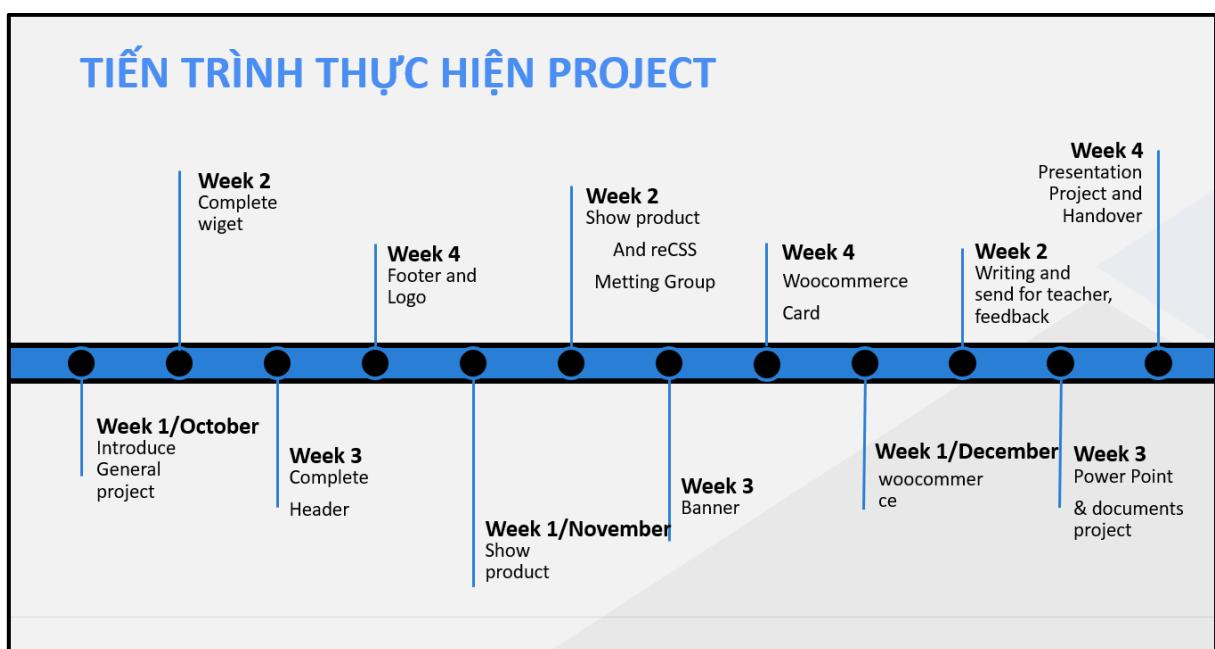
Hình 83. Cấu trúc thư mục đồ án.....	55
Hình 84. Cấu trúc file singular.php	55
Hình 85. Cấu hình file archive.php (hiển thị sản phẩm danh mục).....	56
Hình 86. Hàm lấy slug danh mục theo id trong functions.php.....	56
Hình 87. Giao diện sản phẩm khi nhấp vào danh mục (menu - part 1).....	57
Hình 88. Giao diện sản phẩm khi nhấp vào danh mục (part 2 - phân trang).....	57
Hình 89. Giao diện sản phẩm khi nhấp vào danh mục (part 3 - phân trang).....	58
Hình 90. Giao diện trang single.php (chi tiết sản phẩm).....	58
Hình 91. Giao diện single trong single.php (part 1 - ảnh và tên sản phẩm).....	59
Hình 92. Giao diện single trong single.php (part 2 - Giá tiền sản phẩm).....	59
Hình 93. Giao diện single trong single.php (part 3 - Thêm giỏ hàng và Thông tin)....	59
Hình 94. Giao diện trang giỏ hàng khi chưa có sản phẩm	60
Hình 95. Hướng dẫn thêm sản phẩm vào giỏ hàng	60
Hình 96. Kết quả khi thêm vào giỏ hàng	61
Hình 97. Trang giỏ hàng khi mới thêm sản phẩm	61
Hình 98. Tiến hành cập nhật giỏ hàng (tăng số lượng sản phẩm)	62
Hình 99. Kết quả update giỏ hàng thành công	62
Hình 100. Cấu trúc file g2-gio-hang.php.....	63
Hình 101. Tạo Trang Giỏ Hàng (sử dụng trang có sẵn của Woocommerce).....	63
Hình 102. Giao diện thanh toán mặc định (part 1)	64
Hình 103. Giao diện thanh toán mặc định (part 2)	65
Hình 104. Tạo Trang Thanh Toán (sử dụng trang có sẵn của Woocommerce)	66
Hình 105. Cấu trúc file g2-thanh-toan.php.....	66
Hình 106. Thông báo lỗi khi để trống form trong Thanh Toán.....	66
Hình 107. Điền thông tin thanh toán (part 1)	67
Hình 108. Điền thông tin thanh toán (part 2)	67
Hình 109. Điền thông tin thanh toán (part 3)	68
Hình 110. Gửi thông tin thanh toán thành công.	69
Hình 111. Thông tin thanh toán của khách hàng (website - wp-admin)	69
Hình 112. Control thông tin thanh toán khách hàng	70

LỜI MỞ ĐẦU

Trước sự bùng nổ của thế giới công nghệ nói chung và thương mại điện tử nói riêng. Môi trường internet đã đa dạng hoá các mô hình và xâm chiếm hầu hết các ngành nghề trong cuộc sống. Theo đó, Ngành kinh doanh theo hướng online đang nở rộ và thu lượm được nhiều kết quả tích cực. Nhận thấy được tiềm năng phát triển từ mô hình kinh doanh này. Nhóm chúng em quyết định thực hiện, triển khai, xây dựng Website bán hàng.

Đi kèm theo đó, các hệ thống website cũng phát triển không ngừng với đủ các lĩnh vực: văn hóa, thời sự, khoa học công nghệ, thể thao, ca nhạc, phim ảnh, giáo dục, y tế,... Tuy nhiên để tạo ra một trang web có sức sống bền lâu thì bản thân nó phải đem lại lợi ích cho nhiều người. Giao diện bắt mắt, là yếu tố quan trọng để người xem click chuột đến trang web của bạn.

Trong quá trình phát triển, Nhóm chúng em chân thành cảm ơn tới các thầy, cô giáo bộ môn và đặc biệt là Giảng viên: Bùi Thị Phương Thảo đã tạo điều kiện và hướng dẫn tận tình cho nhóm. Nhờ đó, nhóm chúng em đã hoàn thành đúng hạn. Dưới đây là quá trình mà nhóm đã nghiên cứu và xây dựng. Các nội dung (Chương) được sắp xếp theo thứ tự phía dưới cũng chính là quá trình ‘nuôi’ trang web từ lúc khai sinh cho tới lúc này.



Hình 1. Tiến trình thực hiện project

CHƯƠNG I: GIỚI THIỆU CMS WORDPRESS

1. Giới thiệu Wordpress



Hình 2. Logo wordpress

Wordpress là một mã nguồn mở thường được dùng để xuất bản ra các blog hoặc website được viết bằng ngôn ngữ lập trình PHP và hệ cơ sở dữ liệu MySQL. Với Wordpress thì người dùng có thể tạo ra bất cứ một loại trang web nào với các Theme và Plugin có sẵn để nâng cao tính thẩm mỹ và trải nghiệm cho người dùng. Wordpress là một hệ quản trị nội dung. Giúp người không cần biết kiến thức lập trình vẫn có thể tạo ra một trang web và hiện thực hóa nó thành một website theo mục đích cá nhân của họ.

2. Ưu điểm và nhược điểm của Wordpress

2.1 Ưu điểm

Khả năng sử dụng wordpress: Wordpress luôn nổi tiếng là một công cụ tạo ra một website dễ dùng nhất với quy trình đăng ký đơn giản, giao diện dễ nhìn cho dù bạn không phải là người phải am hiểu về kiến thức lập trình cũng có thể dễ dàng tạo ra cho mình một website.

Hỗ trợ nhiều loại website: Bạn có thể tạo ra bất cứ một loại website nào mà bạn

muốn với Wordpress: tạp chí, tin tức, doanh nghiệp, blog các nhân, bán hàng,...

Đa dạng giao diện có sẵn: Wordpress có hỗ trợ các giao diện miễn phí cũng như tính phí phù hợp với yêu cầu cho từng website và nếu có kiến thức về lập trình bạn cũng có thể chỉnh sửa các giao diện theo thẩm mỹ riêng của mình để website được đẹp hơn nâng cao trải nghiệm người dùng hơn.

Nhiều plugin hỗ trợ: Nhắc đến Wordpress thì phải nhắc đến các plugin hỗ trợ đi kèm, có hơn 34000 plugin trong thư viện Wordpress giúp cho bạn có thể thoái mái lựa chọn cho mình các plugin phù hợp từ đó tạo điều kiện cho bạn có thể tạo ra một website chuyên nghiệp và chất lượng.

Tính phổ biến: Với cộng đồng đông đảo và được nhiều ông lớn lựa chọn (Coca cola, Sony, BBC,...) thì việc các bạn được giải đáp một số vấn đề gấp phải trong quá trình tạo ra website cũng được hỗ trợ rất nhanh chóng.

Tiết kiệm chi phí: Việc thiết kế một trang website riêng cho bạn với Wordpress là hoàn toàn miễn phí với rất nhiều theme miễn phí.

2.2 Nhuộc điểm

Tính bảo mật: Việc bảo mật cho một website là rất quan trọng khi được đưa vào sử dụng, sự đa dạng các plugins trong Wordpress rất dễ xâm nhập nhưng bạn có thể khắc phục được tình trạng này bằng việc cập nhật liên tục plugin.

Khả năng quản lý người dùng: Vai trò quản lý người dùng vẫn còn là hạn chế đối với Wordpress. Nhưng để khắc phục được tình trạng này bạn có thể cài đặt Wordpress multisite plugins và phân chia vai trò quản lý.

Tốc độ xử lý dữ liệu: Đối với các website có dung lượng lớn thì việc tốc độ load web chậm là điều không thể tránh khỏi chứ không riêng gì với Wordpress. Để khắc phục tình trạng này bạn có thể sử dụng các thủ thuật như dùng bộ nhớ đệm, xoá đi các plugins không cần thiết, giảm kích thước hình ảnh,...

3. Quản trị website với Wordpress

Khi tạo website với Wordpress thì bạn sẽ phải biết các quản trị một website như thế nào. Ví dụ:

Đăng nhập vào trang quản trị.

Tạo, chỉnh sửa bài viết.

Upload ảnh, chèn video vào website.

Tạo, chỉnh sửa, xoá các trang.

Tạo, chỉnh sửa, xoá các user.

Thay đổi email quản trị và người dùng.

Tạo, chỉnh sửa menu.

4. Hướng dẫn cài đặt Wordpress

Với độ phổ biến rộng rãi như hiện nay thì bạn có thể cài đặt Wordpress trên nhiều nền tảng khác nhau (Hosting, VPS), hay cài đặt trên localhost máy của bạn. Ở đây mình sẽ hướng dẫn cài đặt Wordpress trên localhost trên Mac và Windows.

5. Cài đặt XAMPP

* Cài đặt XAMPP trên Windows

Tải phiên bản XAMMP 1.8.3 tại đây:

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/1.8.3/xampp-win32-1.8.3-5-VC11-installer.exe/download>



Hình 3. Chọn phiên bản download

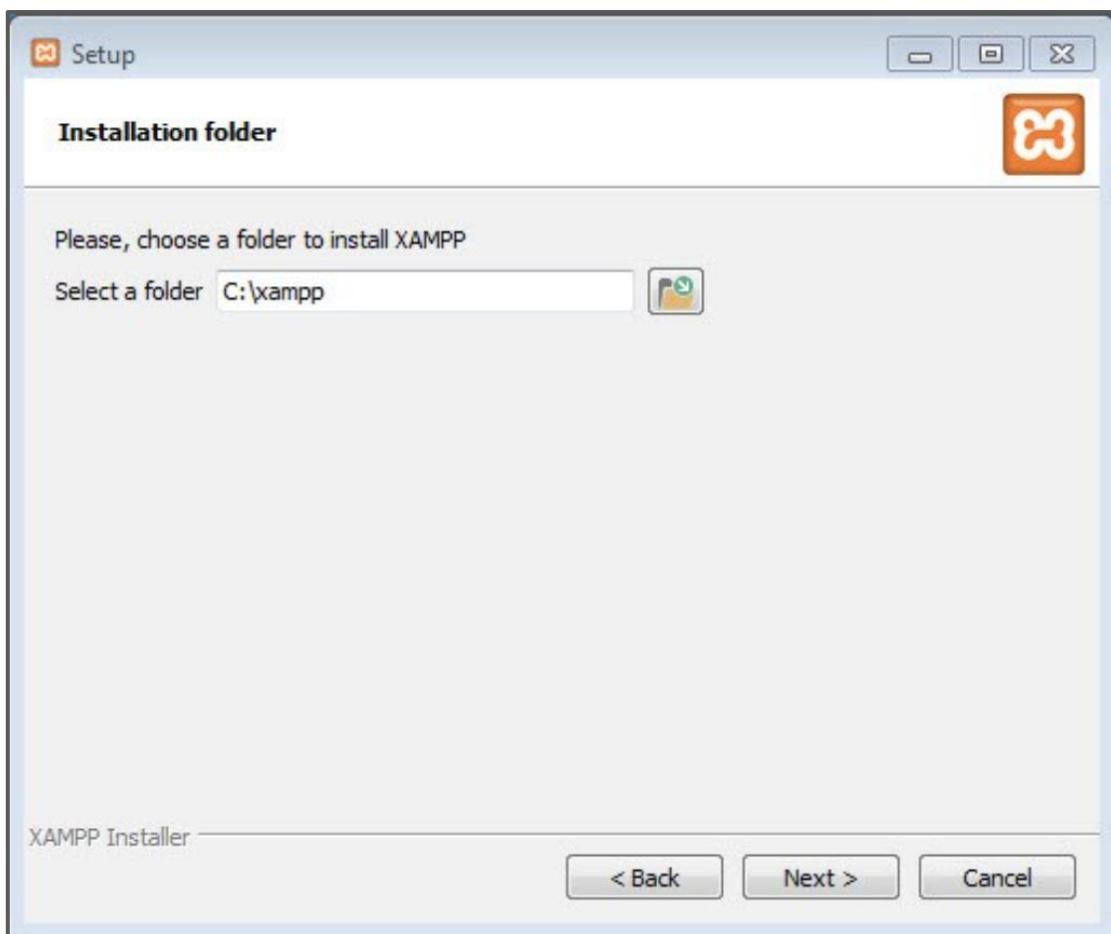
Sau khi tải file cài đặt về xong, hãy chạy nó, sau đó chọn Next.



Hình 4. Install XAMPP

Ở phần chọn đường dẫn, bạn hãy chọn đường dẫn cần lưu cài đặt của XAMPP. Lưu ý rằng đường dẫn này bạn phải nhớ vì khi cài đặt web lên localhost, bạn phải truy cập vào thư mục này. Bạn nên để mặc định là c:\xampp.

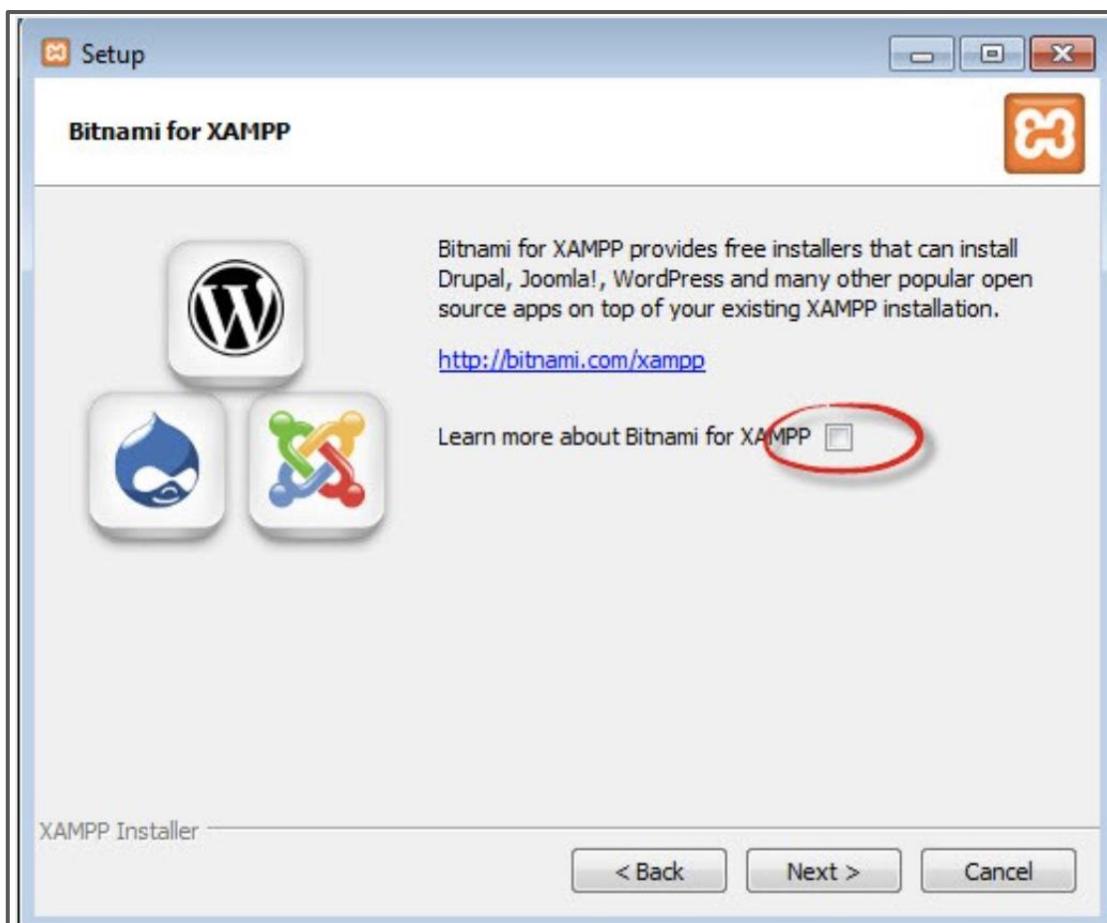
Tiếp tục ấn Next.



Hình 5. Chọn nơi lưu trữ

Ở trang kế tiếp, bạn bỏ chọn phần “Learn more about Bitnami for XAMPP“. Và ấn Next 2 lần nữa để bắt đầu quá trình cài đặt XAMPP.

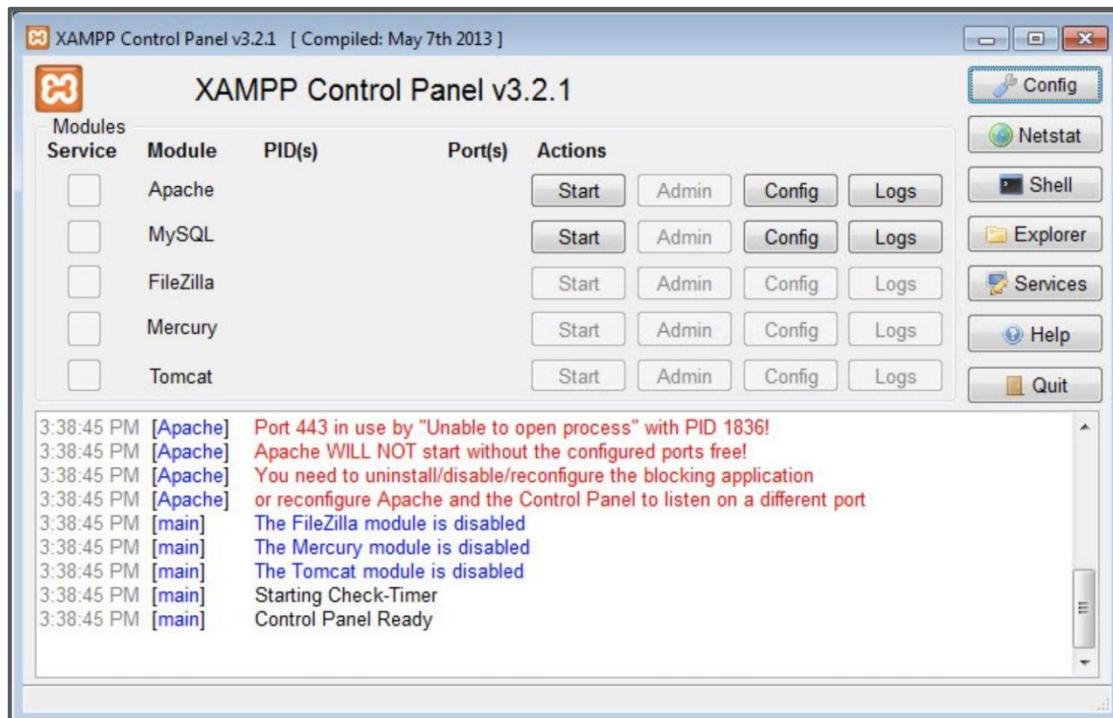
Kết quả sẽ hiển thị như sau:



Hình 4. Bỏ chọn phần “Learn more about Bitnami for XAMPP“.

Sau khi cài xong, ấn nút Finish để kết thúc cài đặt và mở bảng điều khiển của XAMPP. Tuy nhiên, hãy khởi động lại máy sau khi cài đặt xong để tránh tình trạng không khởi động được localhost.

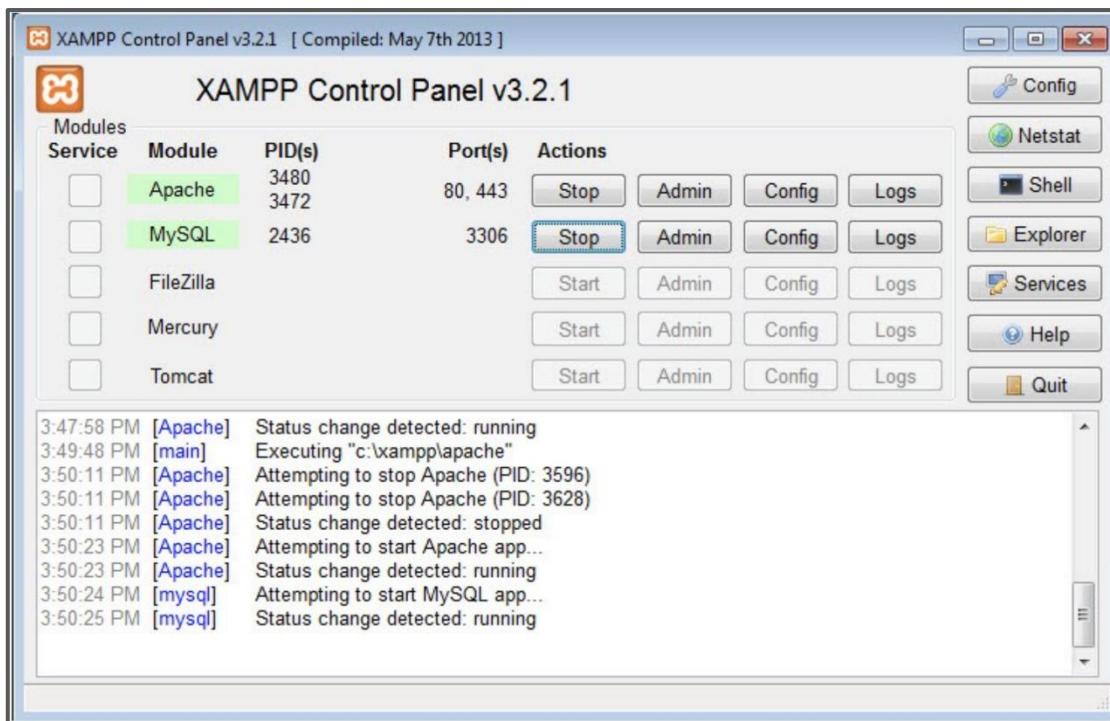
Bây giờ bạn hãy vào thư mục c:\xampp và mở file xampp-panel.exe lên để bật bảng điều khiển của XAMPP.



Hình 5. Bảng điều khiển của XAMPP

Bạn để ý sẽ thấy hai ứng dụng Apache và MySQL có nút Start, đó là dấu hiệu báo 2 ứng dụng này chưa được khởi động, hãy ấn vào nút Start của từng ứng dụng để khởi động Webserver Apache và MySQL Server lên thì mới chạy được localhost.

Nếu cả hai ứng dụng chuyển sang màu xanh như hình dưới là đã khởi động thành công.



Hình 6. Khởi động thành công Apache và MySQL

Sau khi khởi động xong, bạn hãy truy cập vào website với địa chỉ là <http://localhost> sẽ thấy nó hiển thị ra trang giới thiệu XAMPP như hình dưới.



Hình 9. Giao diện XAMPP trên browser

6. Cài đặt XAMPP trên MacOS

Tải phiên bản XAMPP 1.8.3 tại đây:

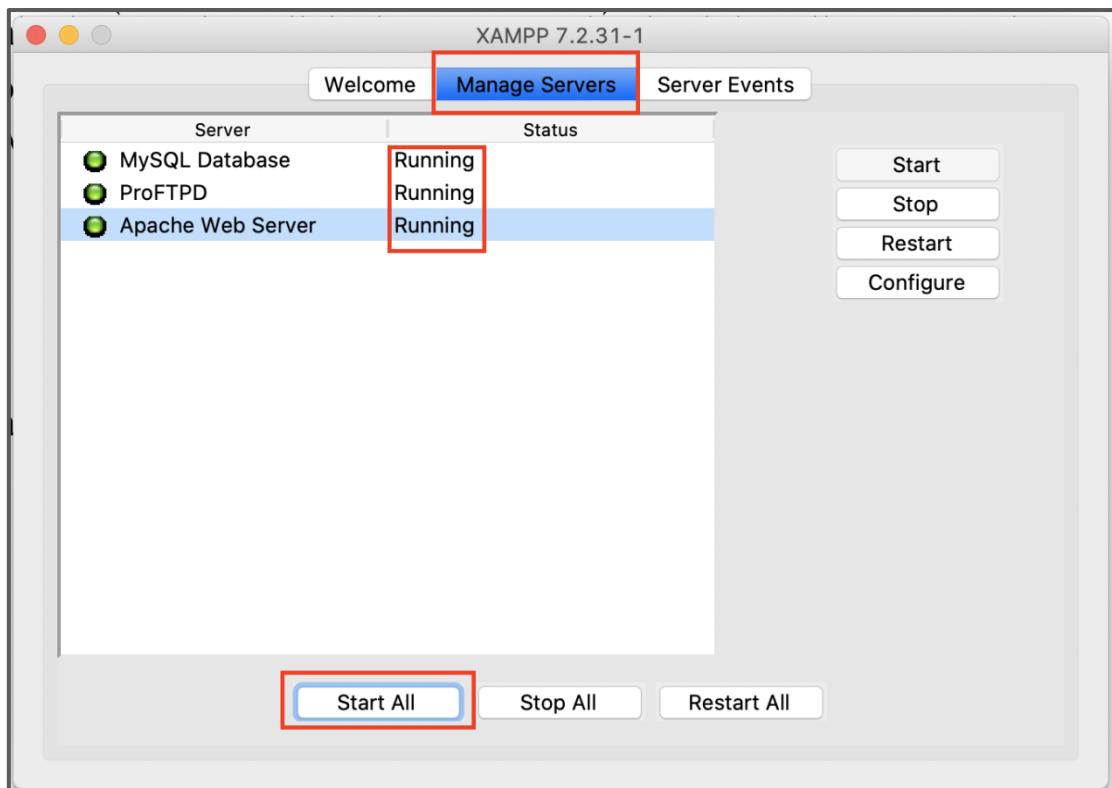
<https://www.apachefriends.org/blog/news-article-157383.html>

Sau khi tải về các bạn click chuột vào và tiến hành install tương tự như Windows. Sau khi start thành công thì các bạn có thể mở giao diện của XAMPP lên tương tự như Windows.



Hình 7. Giao diện XAMPP trên Macos

Sau đó chọn sang tab Manage Server để khởi động MySQL và máy chủ Apache.



Hình 8. Khởi động MySQL và Apache

4.2 Cài đặt wordpress

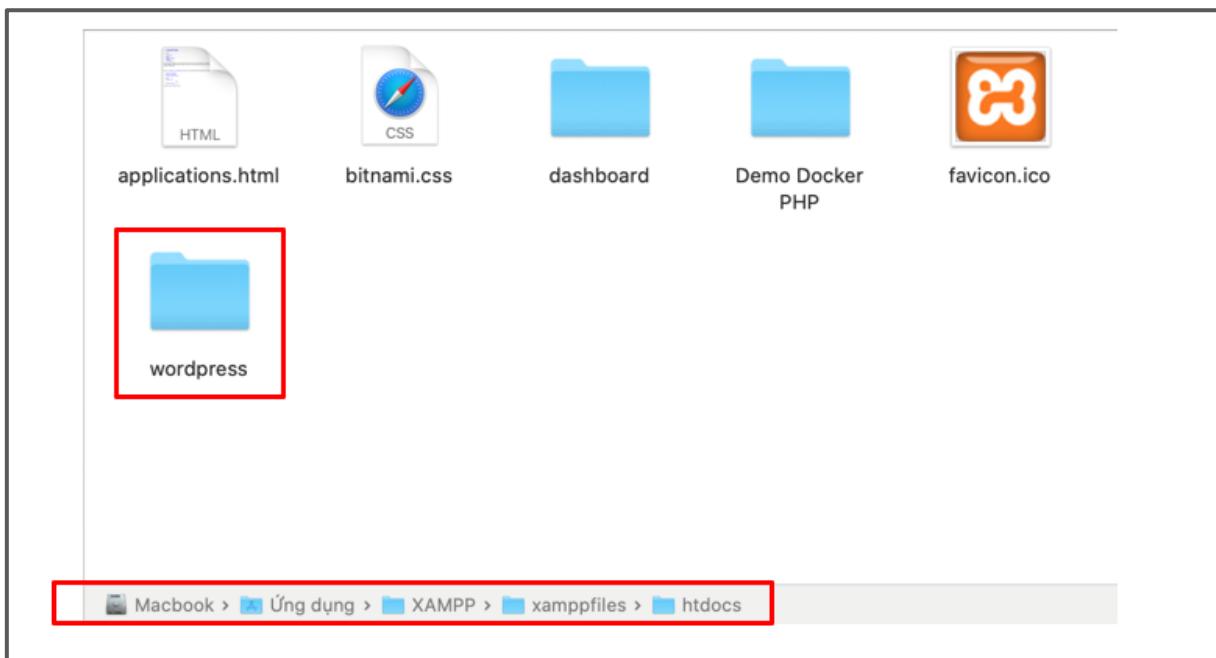
Sau khi đã khởi động thành công MySQL và máy chủ Apache thì sẽ download Wordpress về để tiến hành cài đặt vào máy của mình.

- Truy cập vào: <https://wordpress.org/download/> và download Wordpress về bạn sẽ có được một file wordpress.zip tại folder mà mình download về.



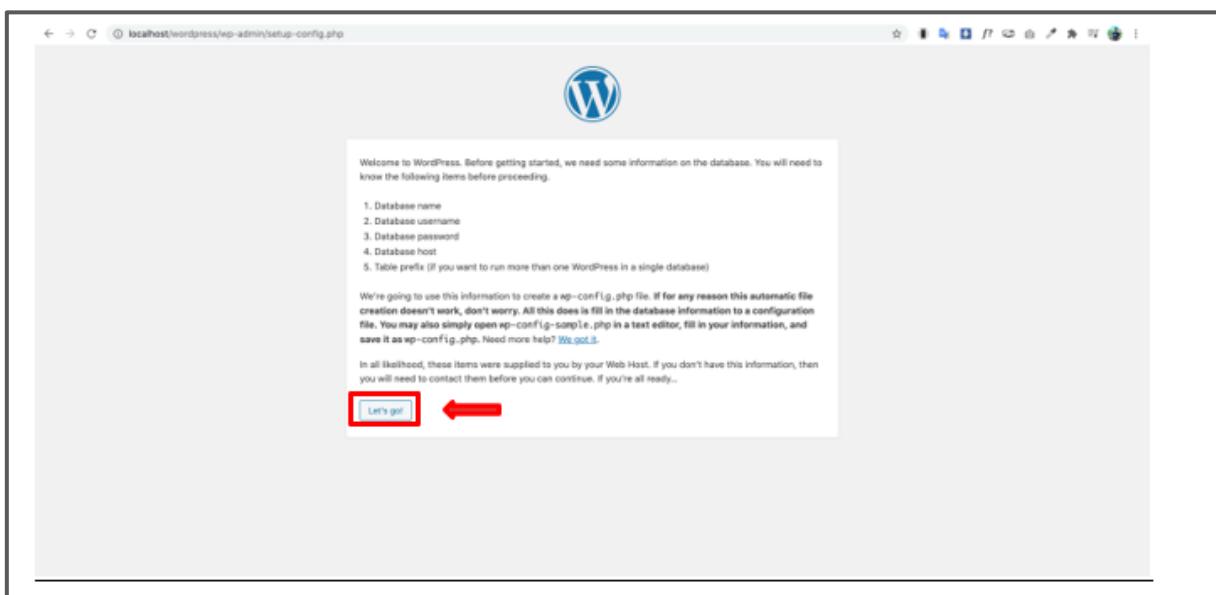
Hình 9. Tệp Wordpress sau khi download về

- Các bạn giải nén ra và copy sang file htdocs của Xampp.



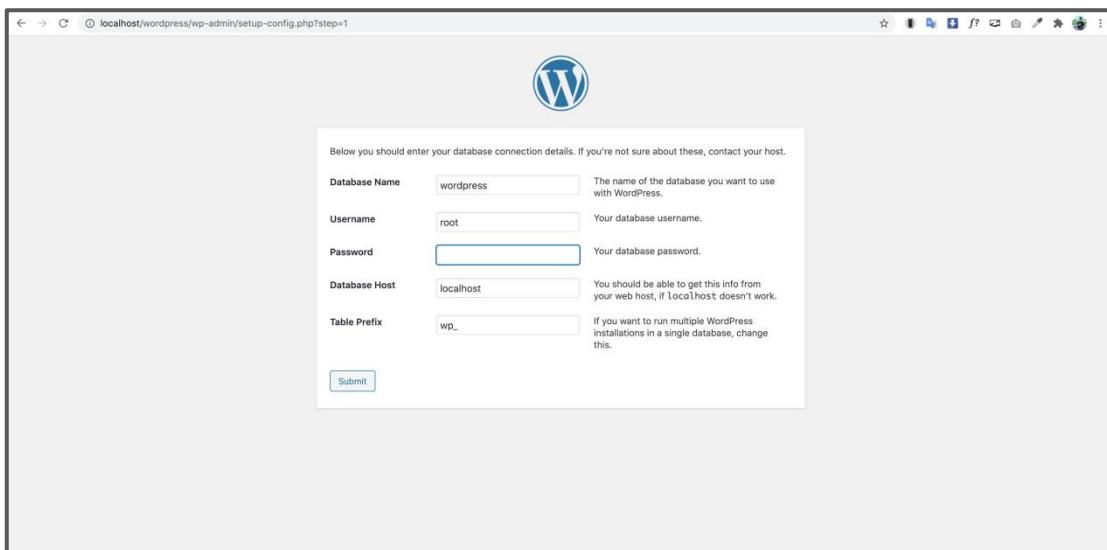
Hình 10. Folder wordpress tại folder Xampp

- Mở trình duyệt và truy cập vào đường dẫn: <http://localhost/wordpress/>, sau đó trình duyệt sẽ tự động chuyển hướng tới việc cài đặt Wordpress (wp-admin).

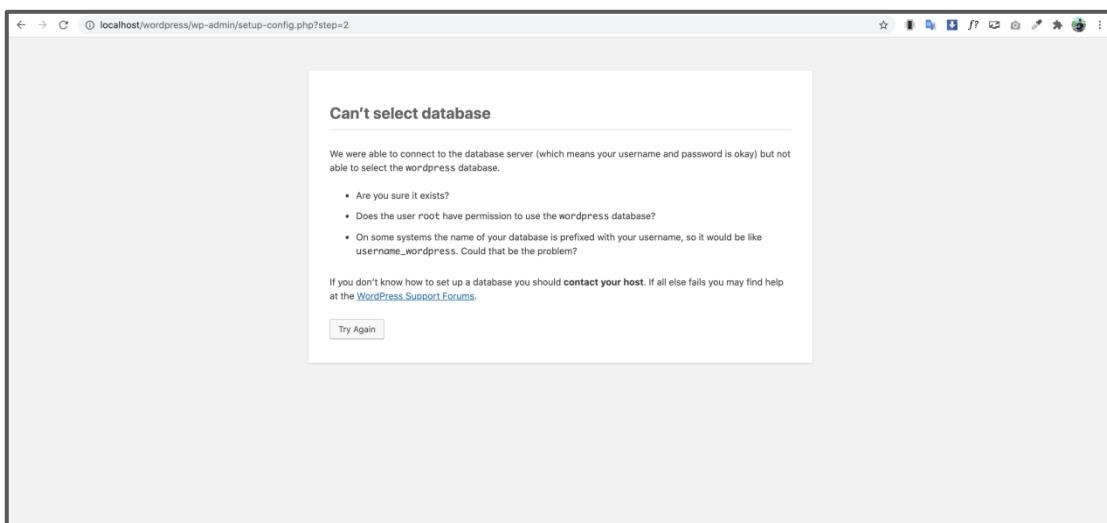


Hình 11.Yêu cầu cấu hình cho wordpress

- Thiết lập database cho wordpress với:
 - + Database Name: Tên database trong phpmyadmin
 - + Username và Password: Mặc định của Xampp là root và pass rỗng
 - + Database Host: localhost (mặc định)
 - + Table Prefix: Tiền tố của tên Table (mặc định ‘wp_’)

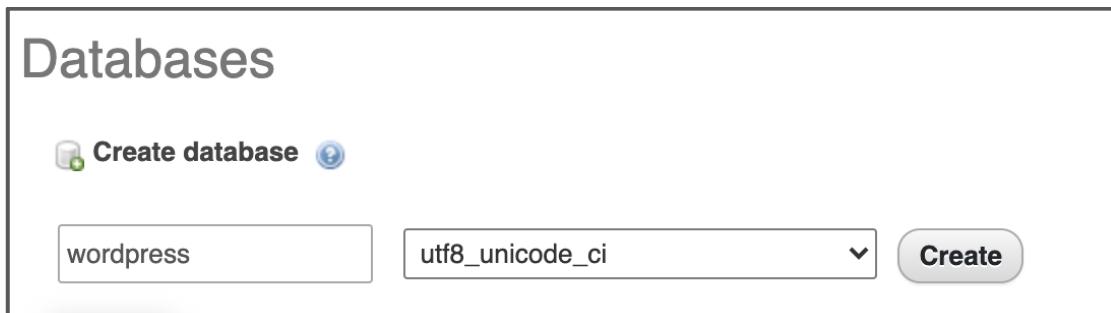


Hình 12. Thiết lập database cho Wordpress



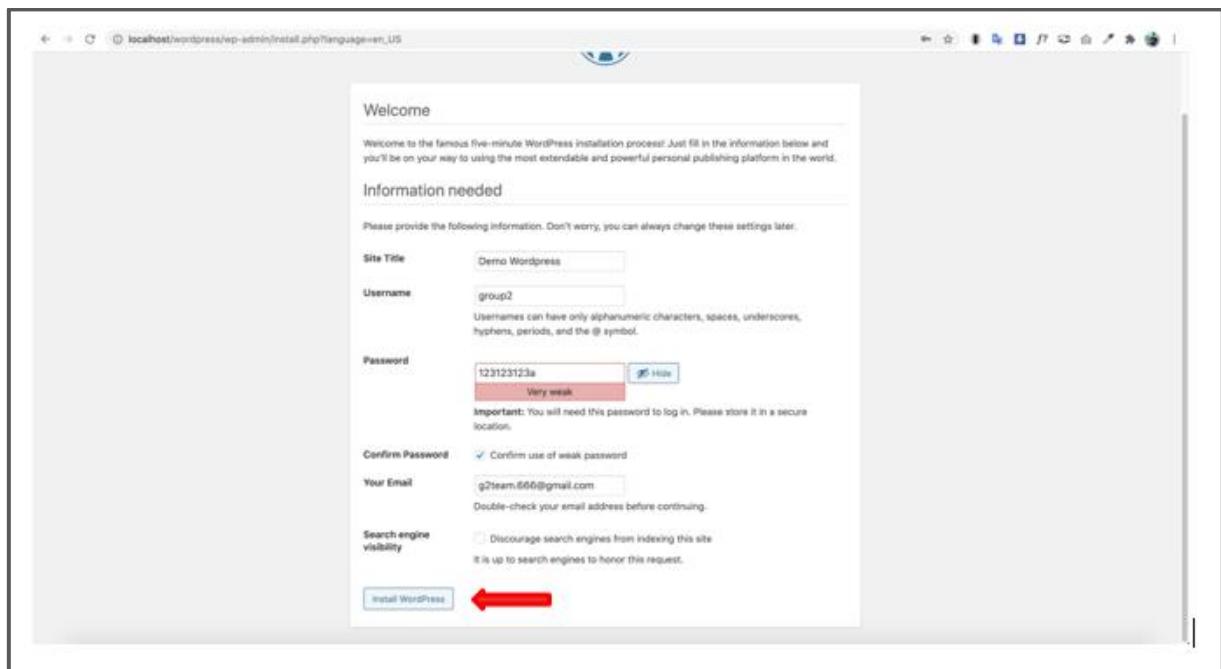
Hình 13. Lỗi chưa có Database

- Vì trong phpmyadmin(<http://localhost/phpmyadmin/>) chưa có database(ở đây là wordpress) nên bạn phải tiến hành tạo mới database với tên trùng với Database Name (Hình 13).



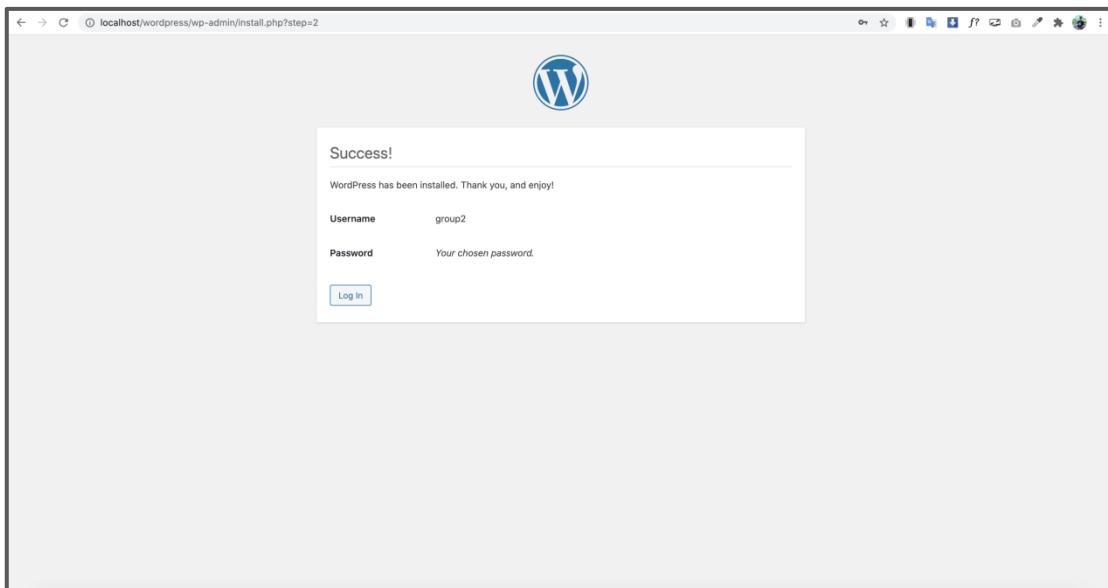
Hình 14. Tạo mới Database trong phpmyadmin

- Sau khi tạo xong Database cho Wordpress ta quay lại bước thiết lập Database ở Hình 13. Thế là bạn đã thiết lập thành công Database cho Wordpress.
- Tiếp theo, bạn đã sẽ được chuyển hướng đến bước cài đặt các thông tin cần thiết cho website của bạn.

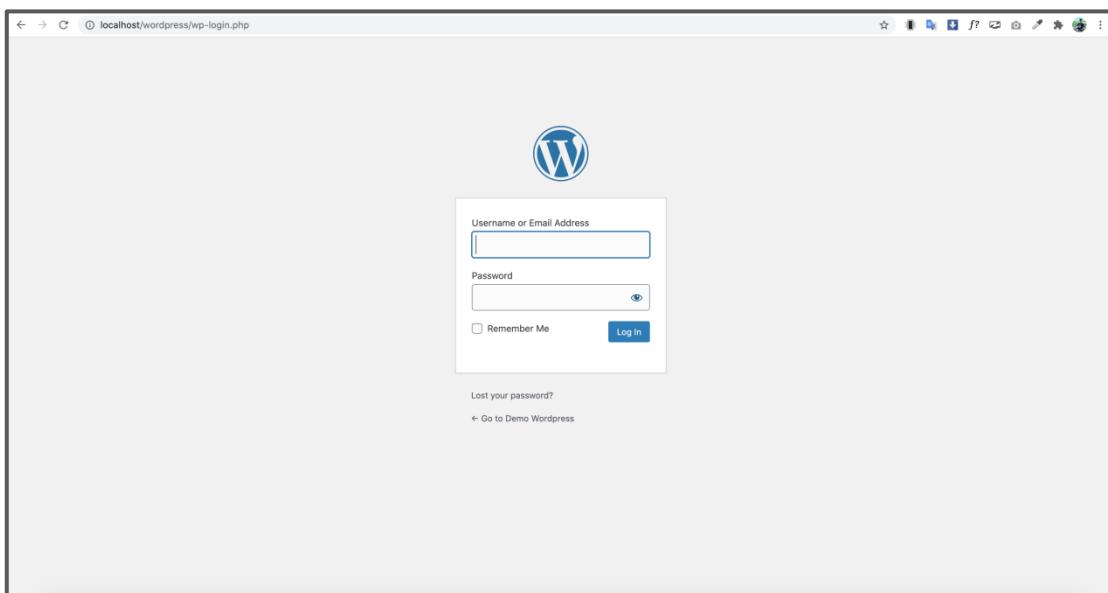


Hình 14.1. Thiết lập thông tin cho website

- Sau khi thiết lập đầy đủ thông tin theo yêu cầu bạn chọn “Install Wordpress” và sẽ thấy thông báo cài đặt thành công và bắt đầu đăng nhập bằng tài khoản vừa tạo.

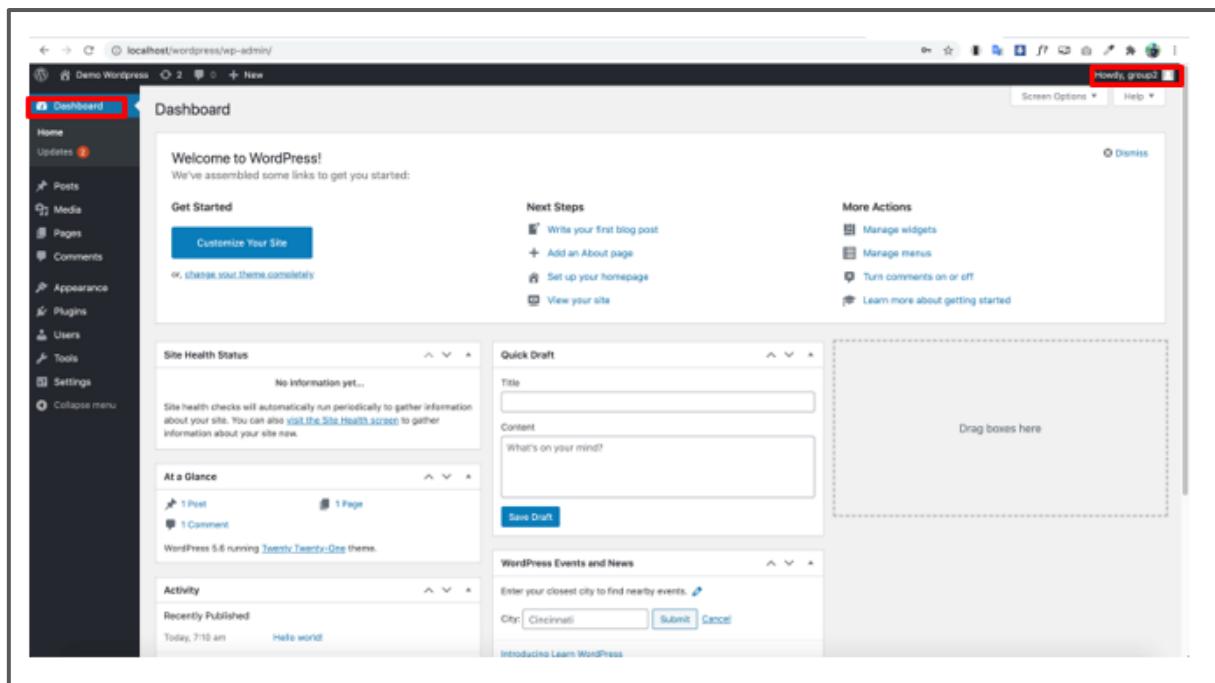


Hình 15. Cài đặt wordpress thành công



Hình 16. Đăng nhập vào wp-admin

- Giao diện wp-admin sẽ được hiển thị khi đăng nhập thành công.



Hình 17. Giao diện wp-admin

- Vậy là bạn đã tiến hành cài đặt thành công Wordpress trên máy và có thể tự tạo ra một website dựa vào các theme và plugin của Wordpress.

CHƯƠNG II: PHÁT TRIỂN PLUGIN

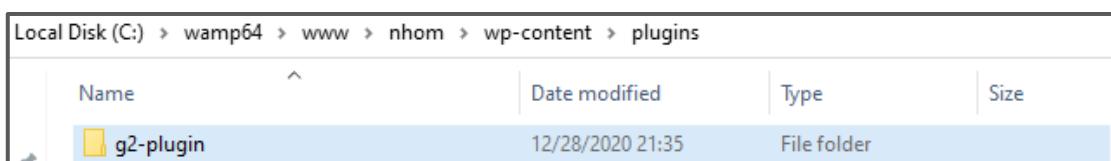
1. Giới thiệu về plugin

Plugin là phần mở rộng trong WordPress được các nhà phát triển xây dựng thêm một tính năng hữu ích khác mà WordPress mặc định không có, plugin được tạo ra dựa trên các API và những hàm mở được cung cấp từ WordPress.

Và có thể hiểu đơn giản rằng plugin là một mảnh ghép giúp bổ sung các tính năng còn thiếu nào đó mà bạn có thể cài vào thông qua Plugin >> Add new trong WordPress, còn ở đây thì mình sẽ phát triển một plugin đơn giản.

2. Hướng dẫn cách phát triển plugin

Đầu tiên để tạo plugin, mình truy cập vào thư mục wp-content\plugins, và tạo một thư mục có tên gì tùy thích, ở đây mình sẽ đặt tên là g2-plugin



Hình 18. Create thư mục trong themes

Sau đó vào thư mục g2-plugin vừa tạo, mình tạo file có tên là g2-plugin.php. Vì ở đây mình chỉ làm đơn giản nên chỉ tạo một file php đơn giản như thế này thôi.

Sau đó bạn mở file vừa tạo lên và thêm đoạn code này vào.

```

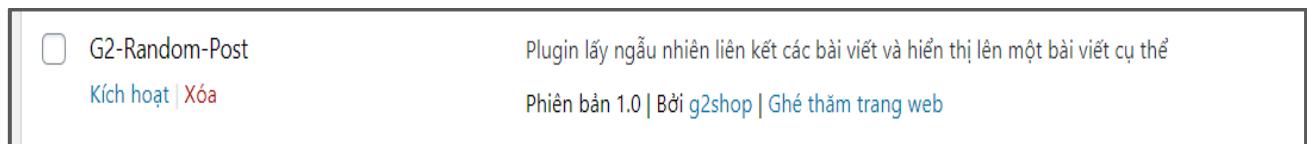
1  <?php
2  /*
3  Plugin Name: G2-Random-Post
4  Plugin URI: https://g2shop.com/
5  Description: Plugin lấy ngẫu nhiên kết các bài viết và hiển thị lên một bài viết cụ thể
6  Version: 1.0
7  Author: g2shop
8  Author URI: https://g2shop.com/
9  License: GPL
10 */

```

Hình 19. Thiết lập thông tin cho plugin

Bên trên thì chỉ là một đoạn comment mình dùng để khai báo plugin. Nếu bạn có nhiều file php trong mục này thì bạn khai báo đoạn này ở file nào thì WordPress sẽ lấy file đó làm tệp chính.

Sau khi tạo xong thì mình vào phần plugin sẽ thấy nó xuất hiện một thư mục plugin do mình tạo xuất hiện.



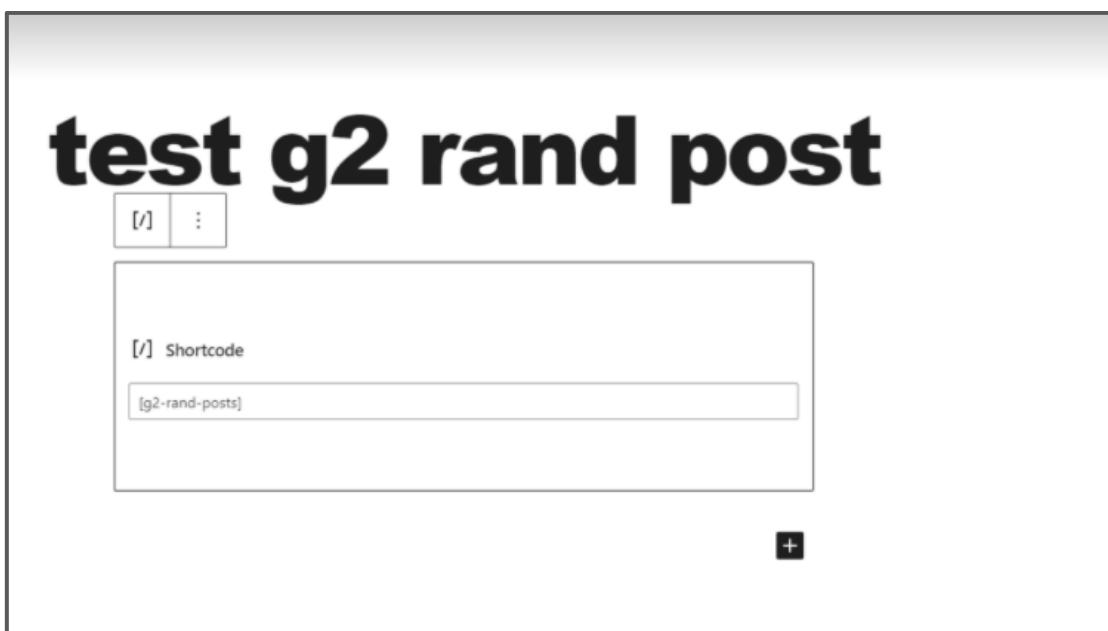
Hình 20. Hiển thị plugin trong wp

Mình sẽ viết một hàm để lấy những bài viết ngẫu nhiên, và mình sẽ add và một cái add_shortcode để sử dụng cho dễ. Code như sau

```
12 function g2_rand_posts() {
13     $args = array(
14         'post_type' => 'post',
15         'orderby' => 'rand',
16         'posts_per_page' => 5,
17     );
18     $the_query = new WP_Query( $args );
19     if ( $the_query->have_posts() ) {
20         $string .= '<ul>';
21         while ( $the_query->have_posts() ) {
22             $the_query->the_post();
23             $string .= '<li><a href="'. get_permalink() .'">'. get_the_title() .'
</a></li>';
24         }
25         $string .= '</ul>';
26         wp_reset_postdata();
27     } else {
28         $string .= 'no posts found';
29     }
30     return $string;
31 }
32 add_shortcode('g2-rand-posts','g2_rand_posts');
```

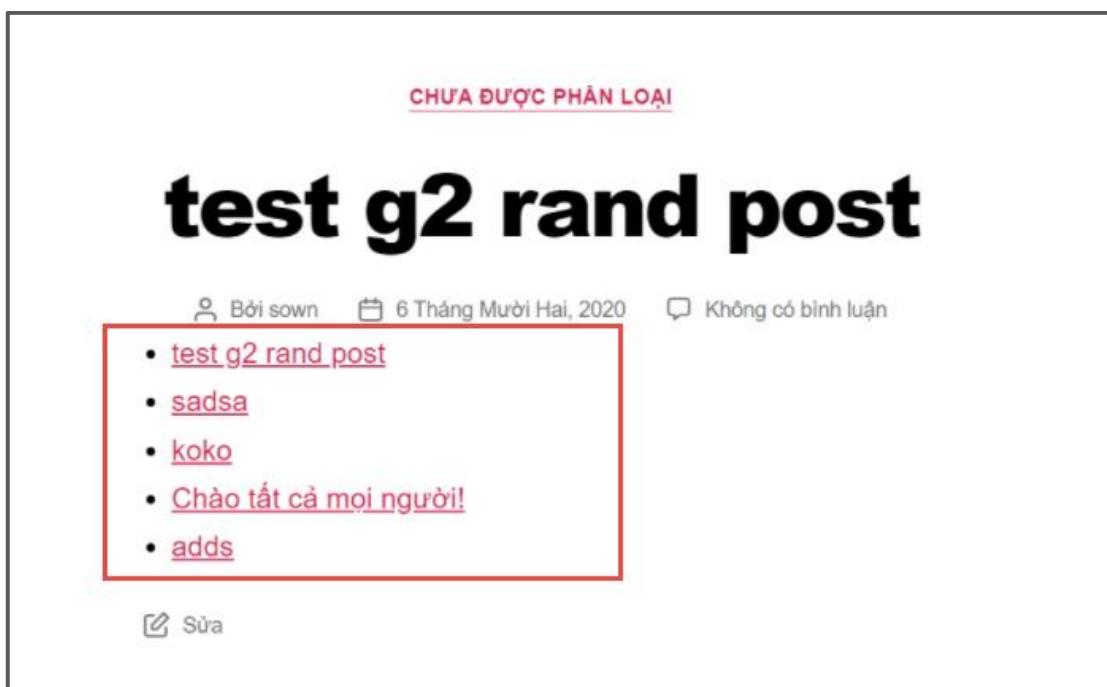
Hình 21. Khởi tạo shortcode g2-rand-posts

Sau đó mình sẽ thử mở một bài viết vào chèn đoạn shortcode đã tạo vào



Hình 22. Thêm shortcode vào bài post

Và kết quả như bên dưới là mình đã thành công



Hình 23. Kết quả sau khi khởi tạo

CHƯƠNG III: PHÁT TRIỂN GUTENBERG BLOCK

1. Cài đặt Theme

Theme là giao diện của Wordpress và cũng là thư mục chứa toàn bộ giao diện của Wordpress được phân bổ theo người dùng.

Có rất nhiều theme mẫu đẹp - độc - lạ nhưng tùy theo sở thích chúng ta có thể tự tạo cho mình một theme ưng ý và quan trọng là chính tay bạn làm theme đó.

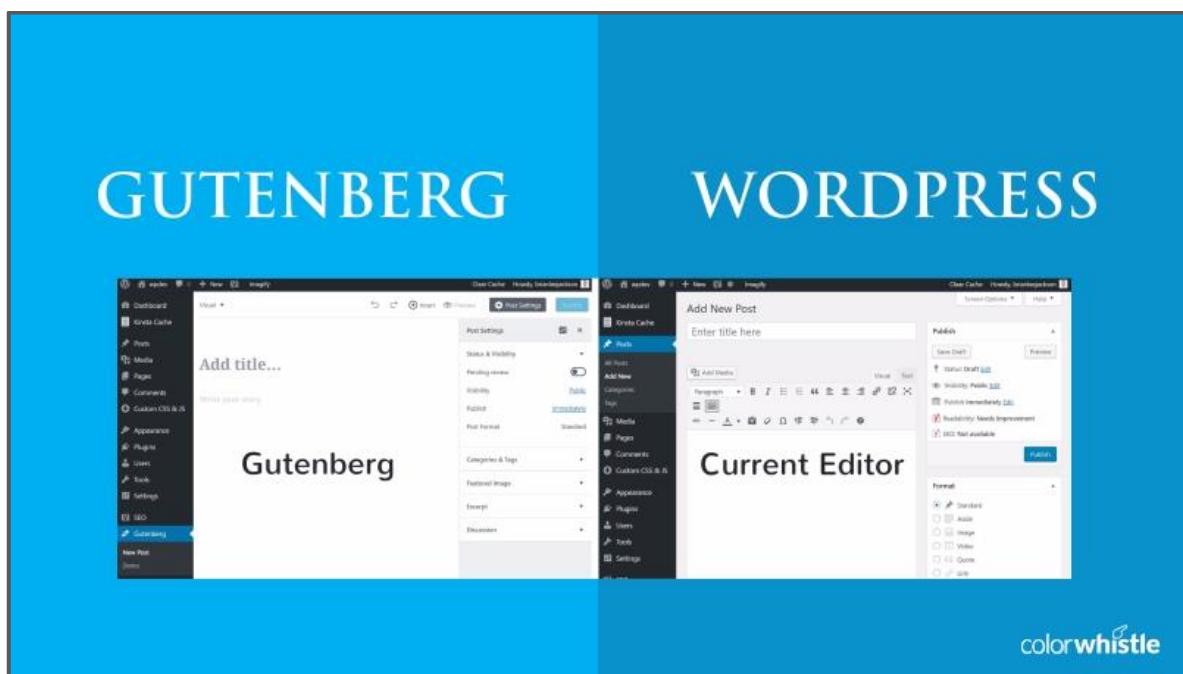
Phần chi tiết theme nhóm xin được đề cập chi tiết ở dưới đây.

2. Giới thiệu gutenberg block

Khái niệm:

Gutenberg hay với cái tên thông dụng Block Editor được biết đến từ ngày 6/12/2018 khi bản cập nhật 5.0 của WordPress được phát hành.

Gutenberg cung cấp một diện mạo (giao diện) mới cho người dùng, nhưng không chỉ thế, nó còn cho người dùng có thể tùy biến được nội dung mà mình tạo.



Hình 24. So sánh Gutenberg và Current Editor

Như thấy ở trên, các nội dung khi tạo một bài post như Title, Image, ... được bỏ vào một “Khối - Block”.

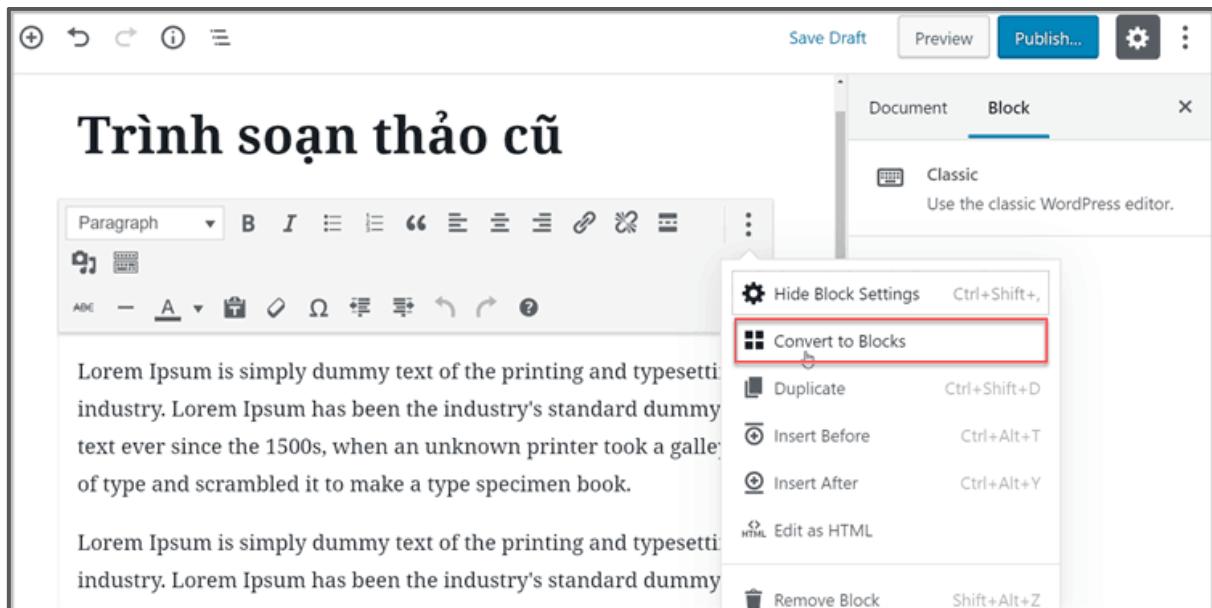
Các Khối có thể tạo thêm bằng plugin bởi vì mỗi Khối là một thành phần riêng biệt.

Vậy làm sao để sử dụng Gutenberg?

Rất giản đơn, chỉ cần cập nhật WordPress của bạn lên bản 5.0 hoặc cao hơn là được.

Bởi vì từ bản 5.0, Gutenberg đã trở thành trình soạn thảo mặc định của WordPress hay nói cách khác Gutenberg là một thành phần của WordPress.

Tất nhiên, toàn bộ nội dung của phiên bản cũ sẽ trở thành Classic Block. Nếu muốn tương tác thì chỉ cần Convert nó qua Gutenberg là được.



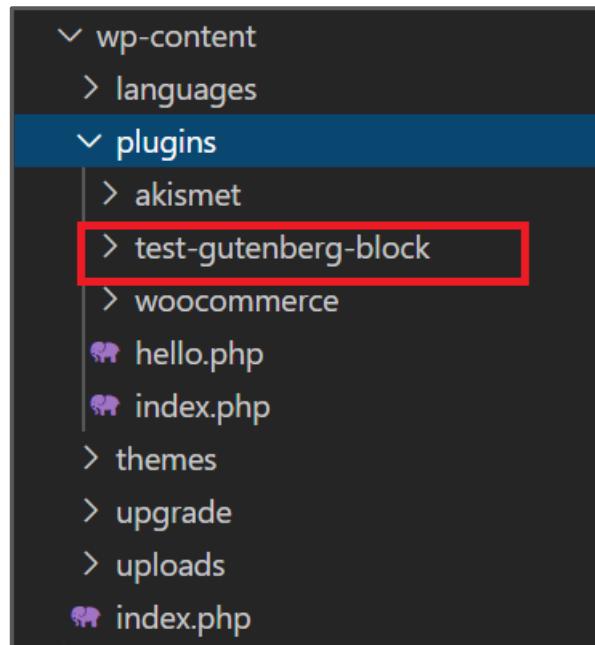
Hình 25. Nội dung cũ convert sang Block

Nếu muốn tắt Gutenberg bạn có thể tải plugin Classic Editor.

Chúc các bạn thành công!

3. Hướng dẫn cách phát triển gutenberg block

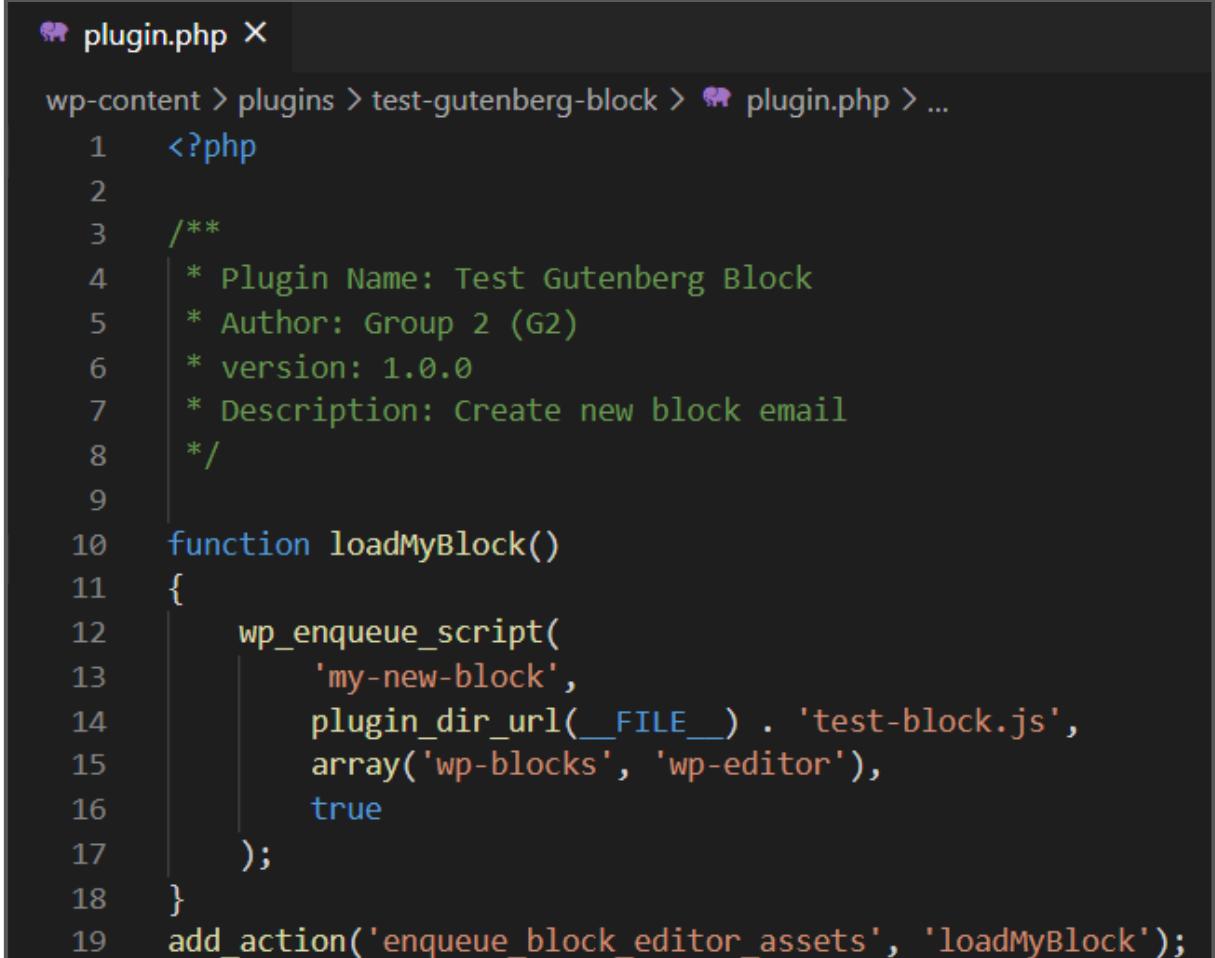
Đầu tiên, trong thư mục chứa các plugin của WordPress, ta tiến hành tạo một folder (ở đây mình tạo tên là **test-gutenberg-block**).



Hình 26. Tạo folder Gutenberg Block

Mở folder mới tạo và tạo một file mới đặt tên là **plugin.php** và cấu hình file đó như hình bên dưới.

Hiện tại mình đang trỏ vào thư mục plugin.php.



```

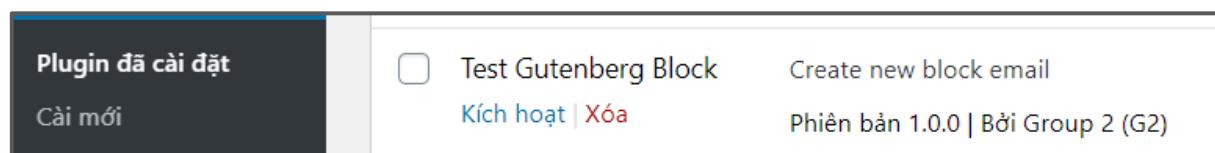
plugin.php ×

wp-content > plugins > test-gutenberg-block > plugin.php > ...
1   <?php
2
3   /**
4   * Plugin Name: Test Gutenberg Block
5   * Author: Group 2 (G2)
6   * version: 1.0.0
7   * Description: Create new block email
8   */
9
10  function loadMyBlock()
11  {
12      wp_enqueue_script(
13          'my-new-block',
14          plugin_dir_url(__FILE__) . 'test-block.js',
15          array('wp-blocks', 'wp-editor'),
16          true
17      );
18  }
19  add_action('enqueue_block_editor_assets', 'loadMyBlock');

```

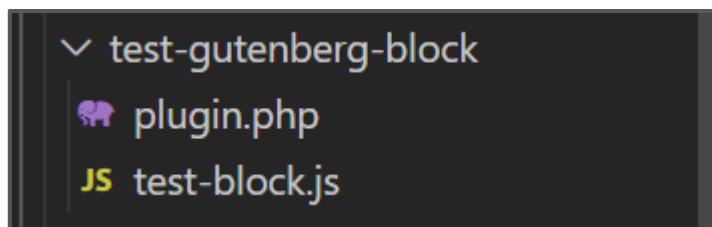
Hình 27. Cấu trúc file plugin.php

Ở dòng 11, truy cập vào file **test-gutenberg-block/g2-block.js**. Với việc sử dụng hàm **wp_enqueue_script()**, chúng ta có thể khai báo file mặc dù chúng chưa được khởi tạo. Ở dòng 13, khai báo để có thể sử dụng các block hay chức năng cơ bản của 2 file **wp-blocks**, **wp-editor**. Hay đơn giản nó như là *include* 2 file đó vào plugin của mình. Tiếp đến, mở giao diện **wp-admin** trên trình duyệt và chọn vào phần **Plugin** thì bạn sẽ hiển thị ra như thế này.



Hình 28. Plugin hiển thị trên trình duyệt wp-admin

Vì ở đây chỉ mới cấu hình file **plugin.php** nên tiếp đến chúng ta khởi tạo giao diện bằng cách tạo một file **test-block.js** (Dòng 10 hình 3)



Hình 29. Tạo file test-block.js

Trong file này, ta cấu hình nó như sau

```

wp-content > plugins > test-gutenberg-block > JS test-block.js > attributes >
1 wp.blocks.registerBlockType("brad/border-box", {
2   title: "Email G2",
3   icon: "email",
4   category: "common",
5   attributes: {
6     content: { type: "string" },
7   },
8   edit: function (props) {
9     function updateContent(event) {
10       props.setAttributes({ content: event.target.value });
11     }
12     return React.createElement("form",
13       {
14         name: "form",
15         action: "#",
16       },
17       null,
18       React.createElement("h3", null, "Email From G2"),
19       React.createElement("input", {
20         type: "text",
21         name: "emailG2",
22         id: "emailG2",
23       })
24     );
25   },
26   save: function (props) {
27     if (/^[\S+@\S+\.\S+]+$/.exec(props.attributes.content)) {
28       return wp.element.createElement(
29         "h3",
30           { style: { border: "3px solid #ccc" } },
31           props.attributes.content
32         );
33     } else {
34       return;
35     }
36   }
37 }
38 );
39 );
40 );
41 );

```

Hình 30. Code của file test-block.js

Ở dòng 30, hàm **save** để xuất kết quả ra khi đăng bài, ta kiểm tra giá trị người dùng nhập (dòng 32) với định dạng email. Nếu đúng, xuất ra **email** đó, ngược lại thì không làm gì cả.

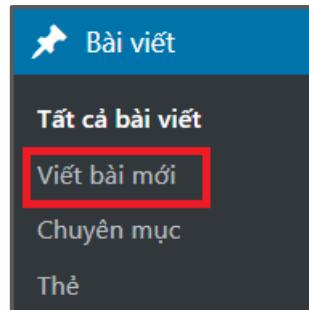
Tiếp đến, ta truy cập vào **wp-admin** (trình duyệt) để kích hoạt plugin này.



Hình 31. Sau khi click “Kích hoạt” plugin trên wp-admin (trình duyệt)

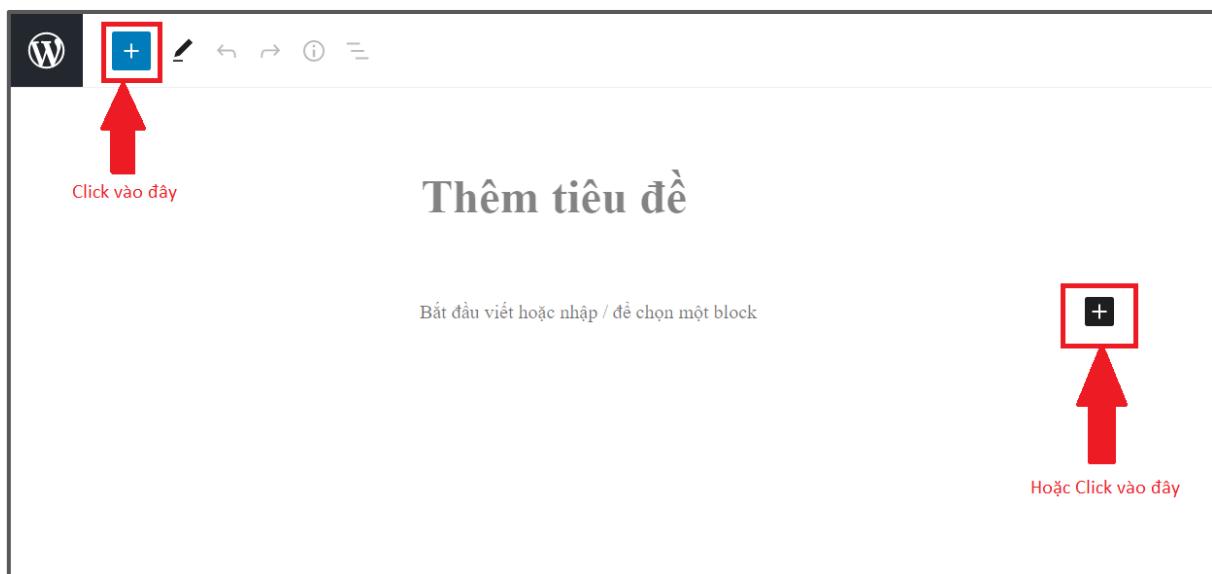
Vì đây là **block** nên nó sẽ xuất hiện trong “**Tạo mới Post/Page**” hoặc “**Chỉnh sửa Post/Page**”. Nhưng chúng đều giống nhau nên mình chỉ hướng dẫn “**Tạo mới Post**”

Đầu tiên, truy cập vào **tạo mới bài viết** của trang wp-admin



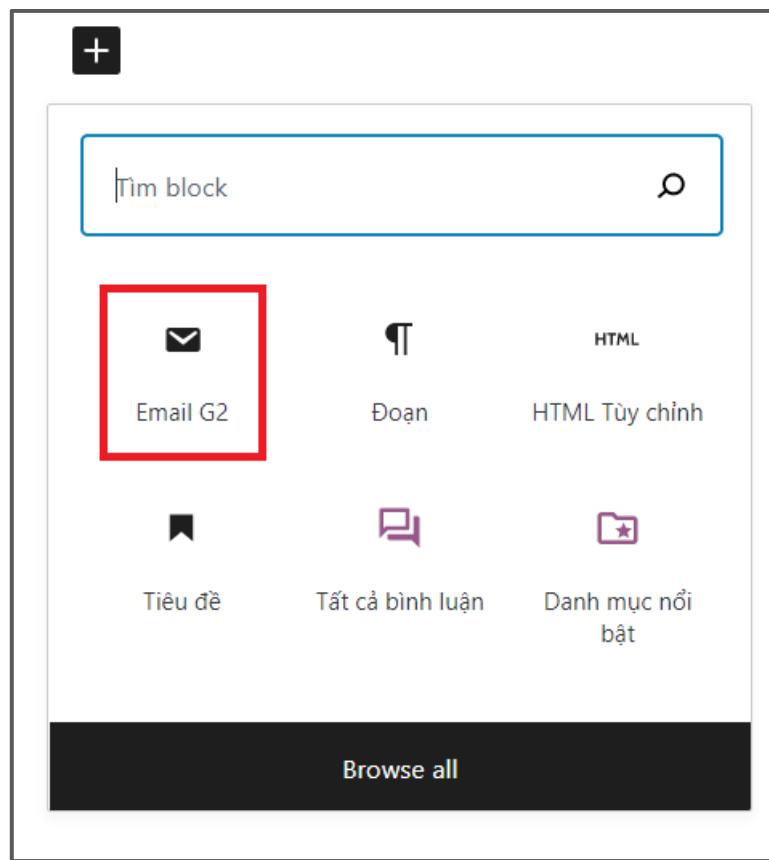
Hình 32. Truy cập vào Tạo bài viết mới

Màn hình sẽ hiển thị ra như bên dưới, tiếp tục click vào button gần Logo WordPress hoặc phía dưới ô nhập **Tiêu đề** (ở đây click vào dưới **Tiêu đề**)



Hình 33. Hiển thị danh sách block của WordPress

Giao diện sẽ hiển thị ra như thế này, sau đó click vào Block chúng ta vừa tạo ra



Hình 34. Mở Block mới vừa tạo (Email G2)



Hình 35. Kết quả khi mở block vừa tạo

Bây giờ, chúng ta tiến hành kiểm tra thử nhé!

Đầu tiên, sẽ sử dụng một Theme có sẵn của WordPress. Sau đó, tiến hành thêm block là email đúng và không đúng với định dạng trong phần code ta kiểm tra.

Ở đây, mình chỉ thêm cơ bản để hiển thị block ra chứ không đầy đủ.

- Thêm email đúng với định dạng



Hình 36. Thêm email đúng định dạng

Kết quả sẽ hiển thị ra sau khi đăng bài viết



Hình 37. Kết quả thêm email đúng định dạng

- Thêm email sai định dạng



Hình 38. Thêm email sai định dạng

Kết quả sẽ hiển thị ra sau khi đăng bài viết



Hình 39. Kết quả thêm email sai định dạng

Tất nhiên, bạn cũng có thể cập nhật bình thường **email** như tên hoặc thông tin bài viết



Hình 40. Cập nhật email sai

Nếu muốn tham khảo source code có thể nhấn [tại đây](#)

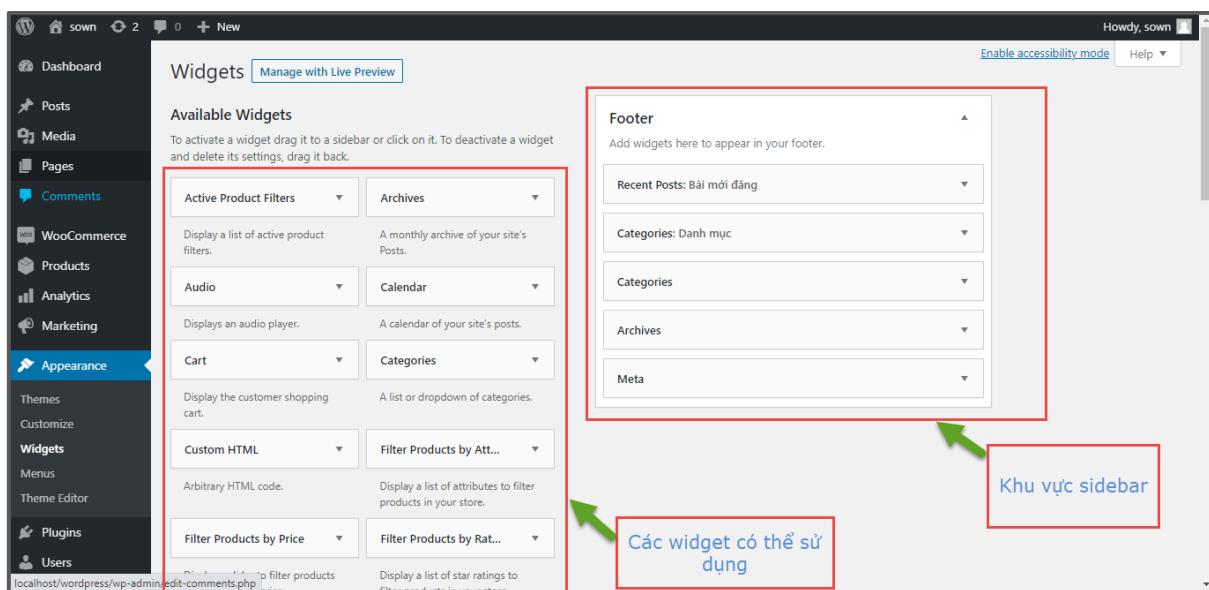
CHƯƠNG IV: PHÁT TRIỂN WIDGET

1. Giới thiệu widget

Khái niệm: Widget hay WordPress widget là một trong các thành phần của WordPress có chức năng: thêm nội dung và định hình cho sidebar hay footer. Ví dụ, widget có thể hiển thị những bài viết mới nhất lên thanh cạnh bên trái hoặc bên phải (sidebar), tạo lịch biểu, thanh tìm kiếm...

2. Hướng dẫn cách phát triển widget

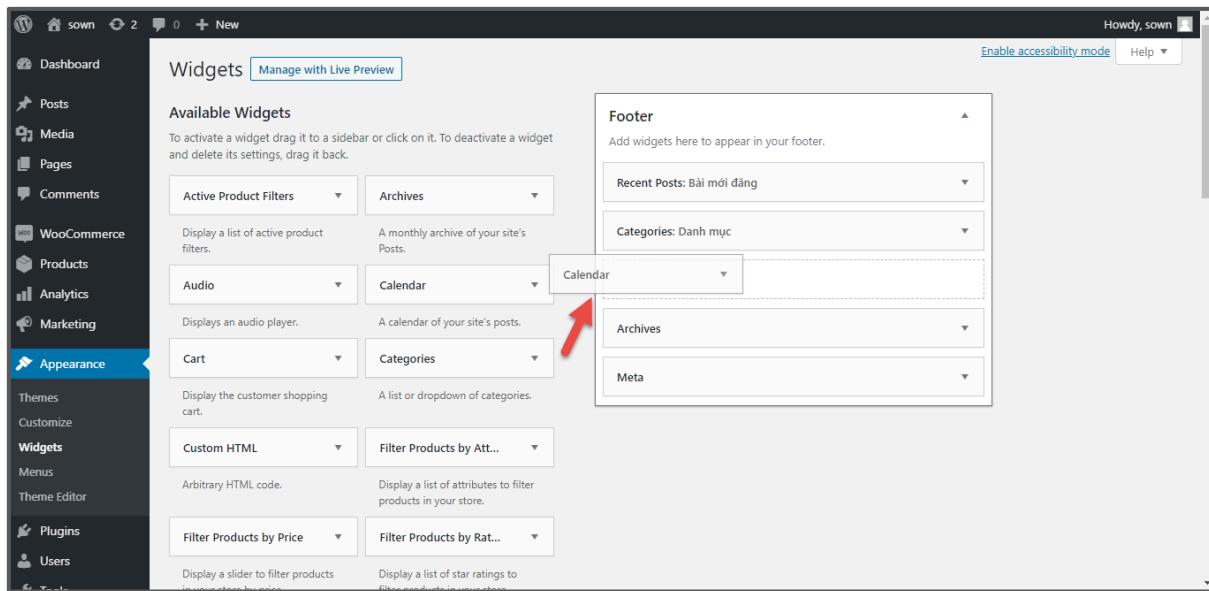
Đầu tiên, đăng nhập vào trình **wordpress dashboard**. Trong mục **Appearance** chọn vào phần **Widgets**.



Hình 41. Widget wordpress

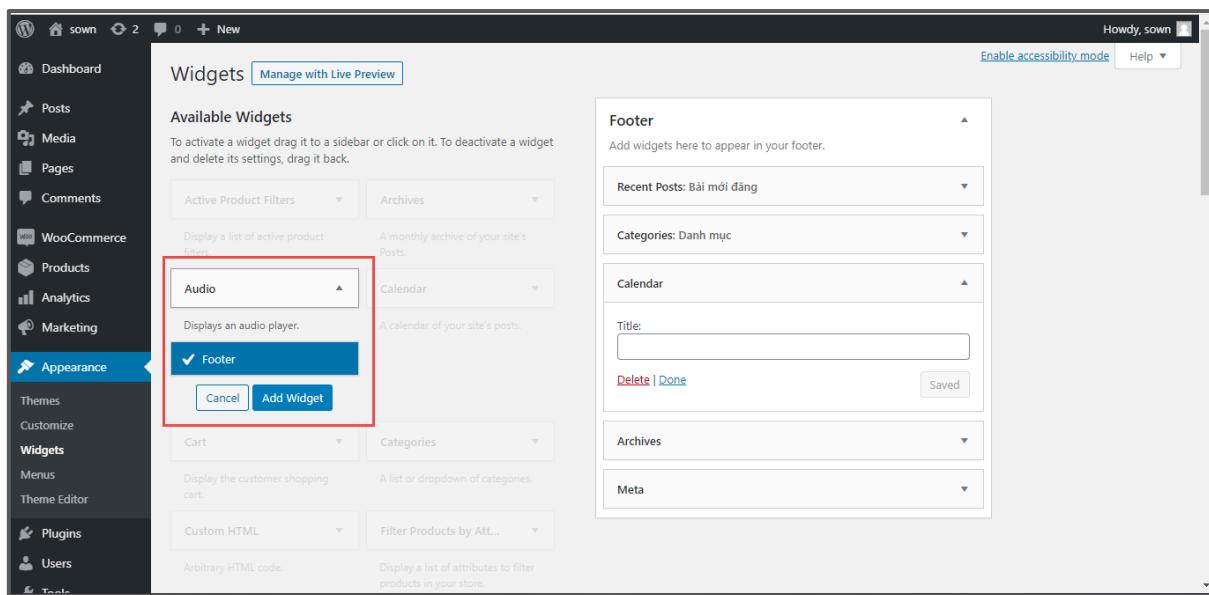
Trong đó, bên tay trái là danh sách các widget mà bạn có thể sử dụng và tay phải là các sidebar. Tùy vào theme, mà bạn sẽ có bao nhiêu sidebar, và có nhiều theme có sidebar dành riêng cho từng phần (Trang chủ, Single, Page...).

Để sử dụng widget, bạn chỉ cần click và kéo widget bên tay trái vào sidebar mà bạn muốn hiển thị widget đó mà thôi.



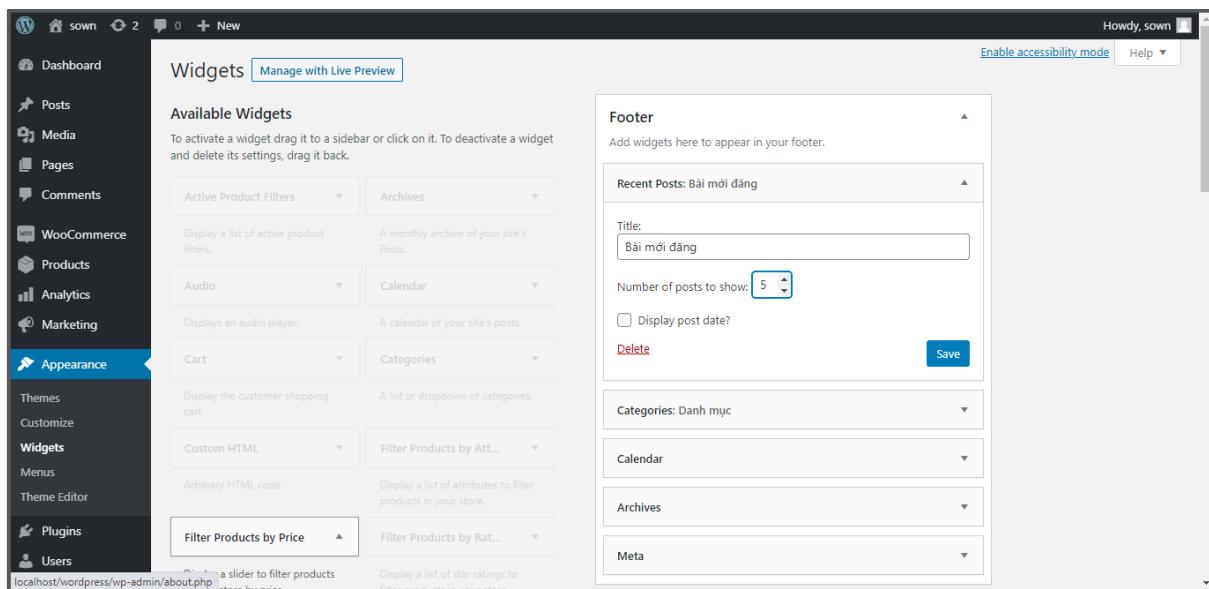
Hình 42 . Thêm Widget cách 1

Hoặc là click vào widget bên tay trái và chọn sidebar cần dùng và ấn nút Add Widget, nó sẽ tự thêm vào vị trí cuối cùng của sidebar.



Hình 43.Thêm Widget cách 2

Sau khi thêm xong widget, mỗi widget có thể sẽ có các tùy chọn riêng biệt để bạn thiết lập nó nhằm hiển thị như ý muốn, các bạn bấm vào và tùy chỉnh lại, sau đó ấn Save để lưu lại.



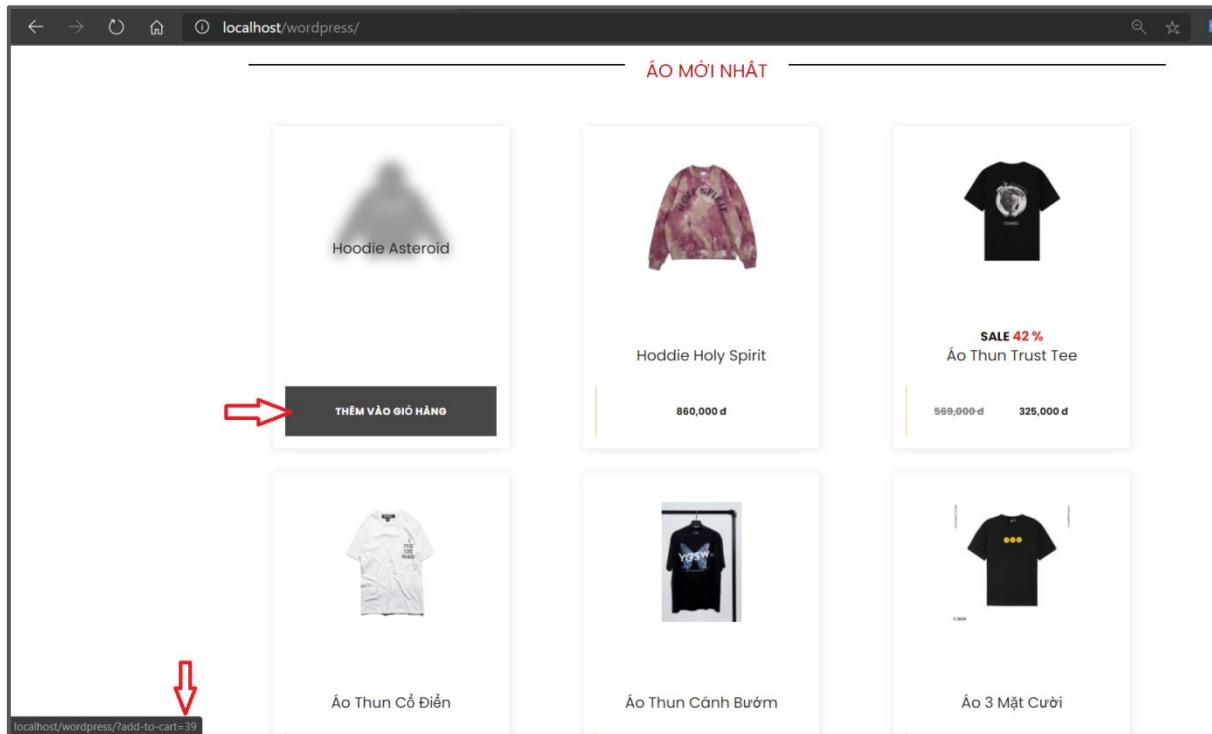
Hình 44. Custom widget và save

Kết quả sau khi thêm widget

Hình 45. Kết quả sau khi thêm Widget

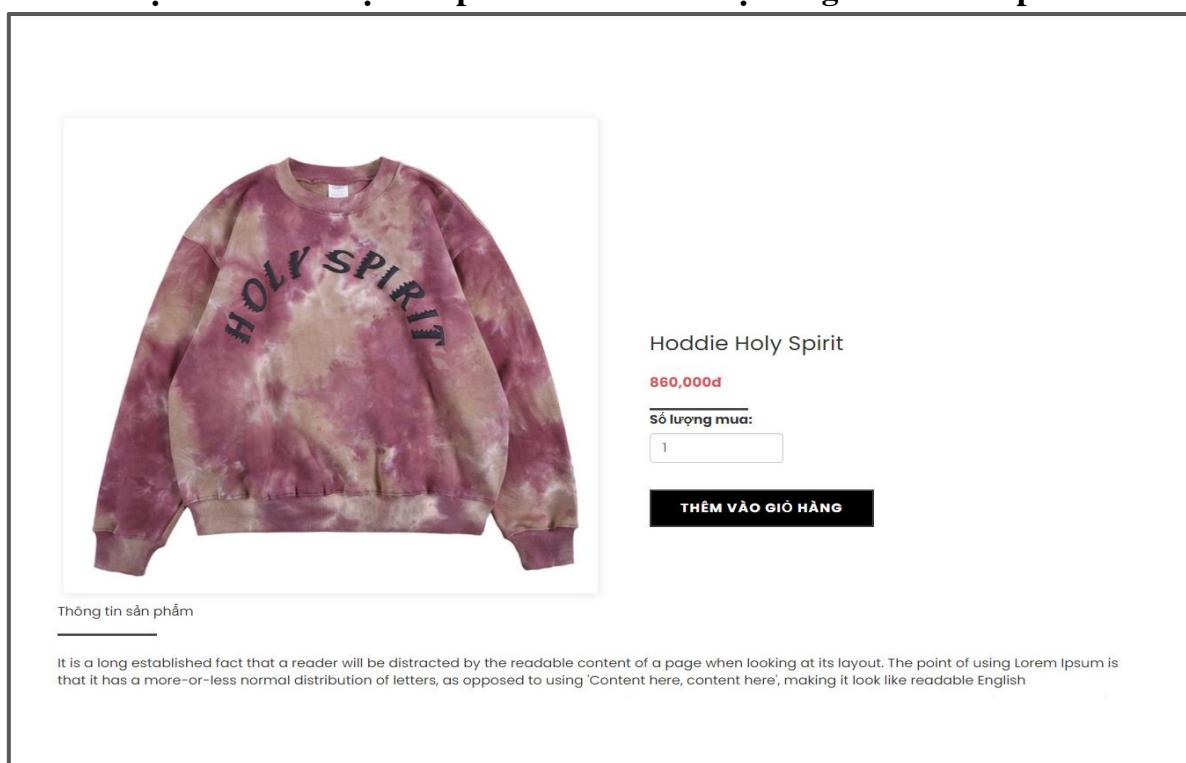
3. Giao diện tương tác:

Sau đây mình xin trình bày một vài giao diện tương tác trên website.



Hình 46. Hình ảnh giao diện chính và CSS khi trỏ chuột

Sau khi bạn click vào một sản phẩm thì sẽ hiển thị trang chi tiết sản phẩm:



Hình 47. Chi tiết sản phẩm

Khi đã tạo được widget thì bạn đã chia rõ được các components riêng biệt phục vụ cho các chức năng riêng của website trong một thẻ thông nhất. chẳng hạn như dưới đây là component menu và danh mục sản phẩm.



Hình 48. Danh mục và menu sản phẩm

Vậy là mình đã trình bày đầy đủ phần widget và đã áp dụng vào đồ án. Hy vọng các bạn cũng có thể làm được và làm đẹp hơn mình.

CHƯƠNG V: PHÁT TRIỂN THEME TRONG WORDPRESS

1. Giới thiệu theme

Theme là ‘bộ mặt’ của Wordpress và cũng là thư mục chứa toàn bộ giao diện của Wordpress được phân bổ theo người dùng.

Có rất nhiều theme mẫu đẹp - độc - lạ nhưng tùy theo sở thích chúng ta có thể tự tạo cho mình một theme ưng ý và quan trọng là chính tay bạn làm theme đó.

Để sử dụng được tính hiệu quả cao nhất thì chúng ta cần làm, hiểu và phải biết mường tượng kết quả trước khi kết quả hoàn thành.

Cũng như quá trình con người mới sinh cho tới trưởng thành, cần đổi mới và học hỏi các cơ chế mới.

Mình sẽ cung cấp cho bạn hàng ngàn theme mẫu theo link dưới đây:

<https://wordpress.org/themes/>

Trước khi mình hướng dẫn phát triển theme thì các bạn nên chú ý tới một khái niệm nữa là template để nhìn ra sự giống và khác nhau giữa template và theme tránh sự hiểu nhầm.

Theme là file chứa toàn bộ của một trang web trong mã nguồn mở wordpress. Còn template bối cảnh của một trang đơn lẻ trong toàn bộ giao diện hay nói cách khác là ‘con’ của themes vì template source code nằm trong file themes.

2. Hướng dẫn cách phát triển theme

Mình sẽ làm tuần tự để đưa tới kết quả như mong muốn.

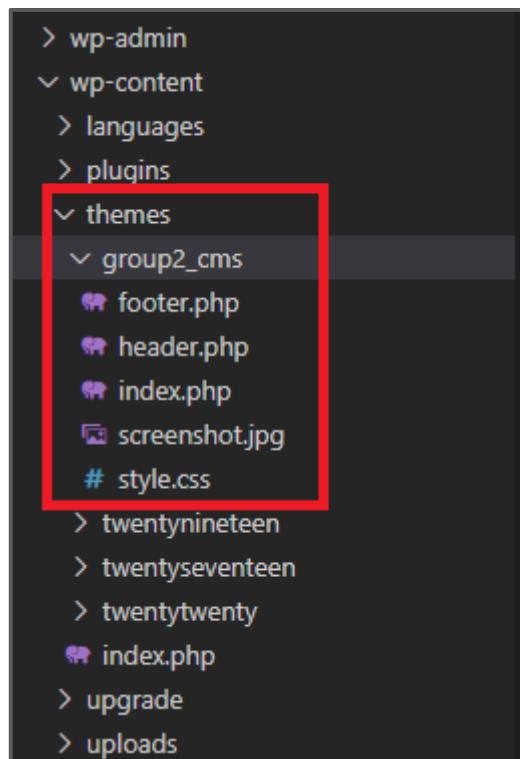
3. Tạo Theme ‘thủ công’

Có thể các bạn tạo bằng cách sử dụng plugin theme Start nhưng như mình đã nhận mạnh từ trước từ mục một. Mình chú trọng tới sản phẩm mình tự tạo mà không dùng hỗ trợ hay nói cách khác là ‘code’ chạy.

Sau khi đã thiết lập và mở được wordpress, chúng ta cần truy cập vào mục C:\wamp64\www\wordpress\wp-content\themes

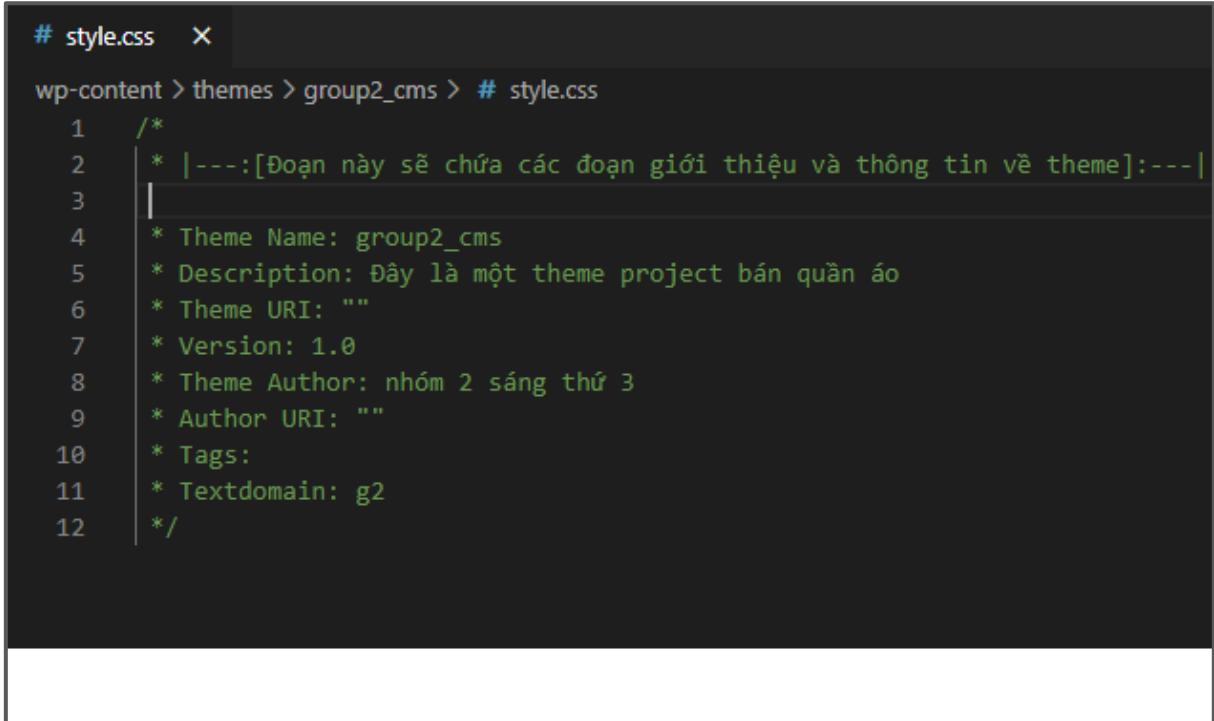
Khởi tạo Project ‘group2_cms’ trong thư mục /wp-content/themes/.

- Trong thư mục group2_cms này sẽ chứa toàn bộ các file php mà trang web xử lý (Hôm nay, nhóm chỉ thực hiện giao diện Header và Footer nên chỉ có 4 file header.php, footer.php và style.css, index.php được xử lý).



Hình 49. Khởi tạo file chứa toàn bộ theme trong themes

- Trong thư mục group2_cms, Tạo 1 file style.css (file này sẽ giới thiệu thông tin chi tiết về theme như : Tên, version, author,...).

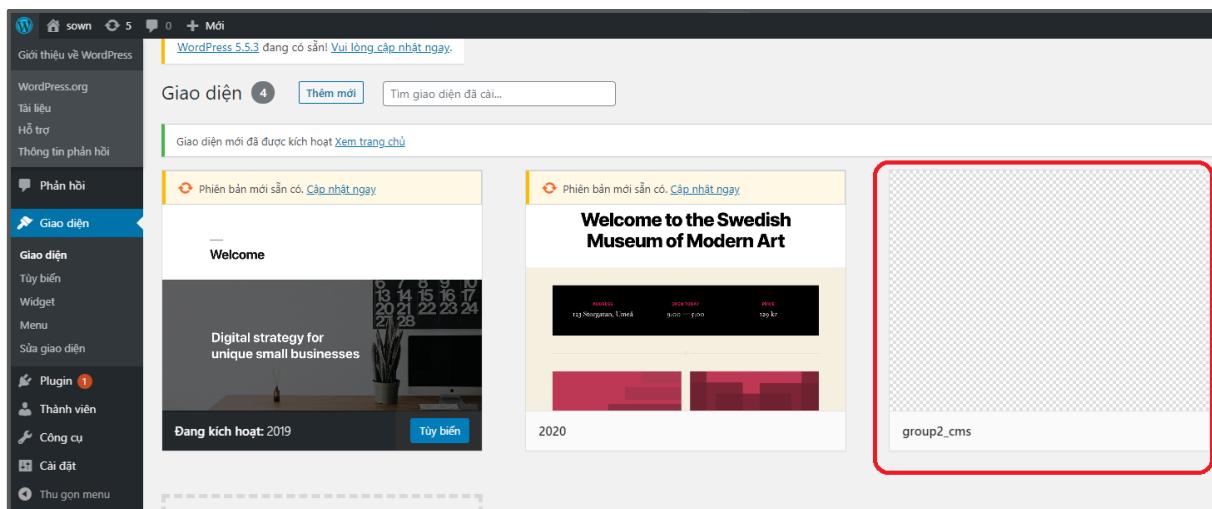


```
# style.css  X
wp-content > themes > group2 cms > # style.css
1  /*
2  * |---:[Đoạn này sẽ chứa các đoạn giới thiệu và thông tin về theme]---|
3  |
4  * Theme Name: group2 cms
5  * Description: Đây là một theme project bán quần áo
6  * Theme URI: ""
7  * Version: 1.0
8  * Theme Author: nhóm 2 sáng thứ 3
9  * Author URI: ""
10 * Tags:
11 * Textdomain: g2
12 */

```

Hình 50. File giới thiệu thông tin

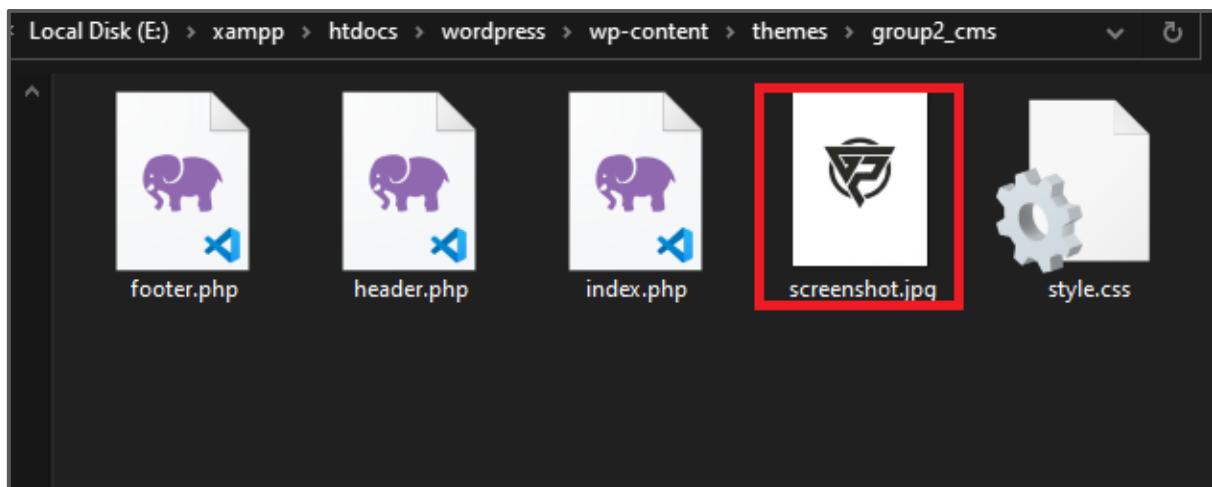
- Tạo giao diện **Header** trong file **header.php** và nhúng thêm thư viện bootstrap vào.
- Tạo file **index.php** để theme hoạt động trong wordpress vì đây là file làm trang chủ mặc định cho wordpress.
- Tạo file **index.php** để theme hoạt động trong wordpress vì đây là file làm trang chủ mặc định cho wordpress.



Hình 51. Giao diện sau khi tạo file

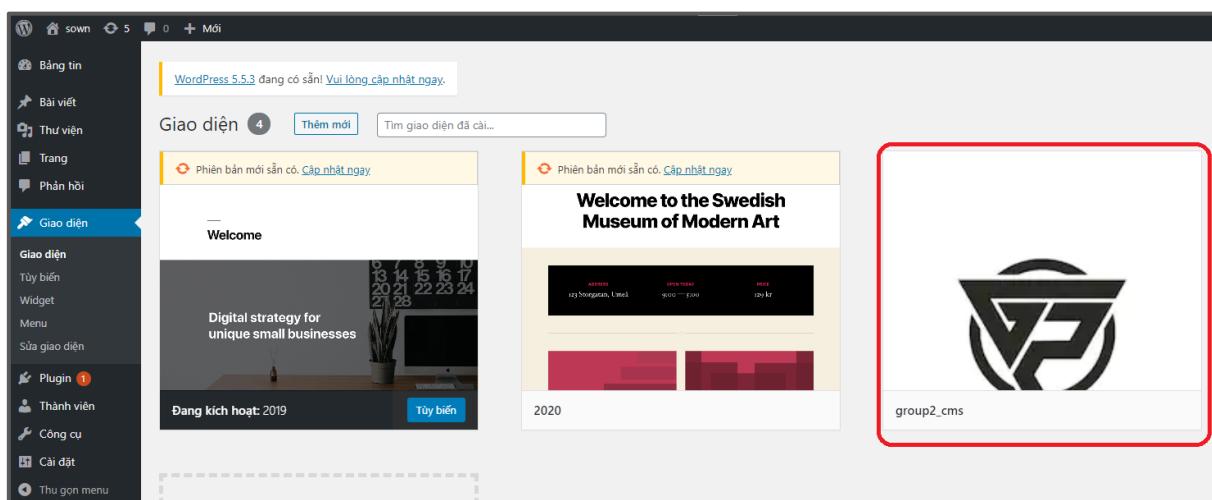
- Đây là giao diện theme khi chưa thêm hình vào.

- Vì vậy, cần thêm 1 hình ảnh (khuyến khích đuôi .png) trong thư mục **group2_cms** để làm hình ảnh đại diện cho Theme mà chúng ta đang tạo.



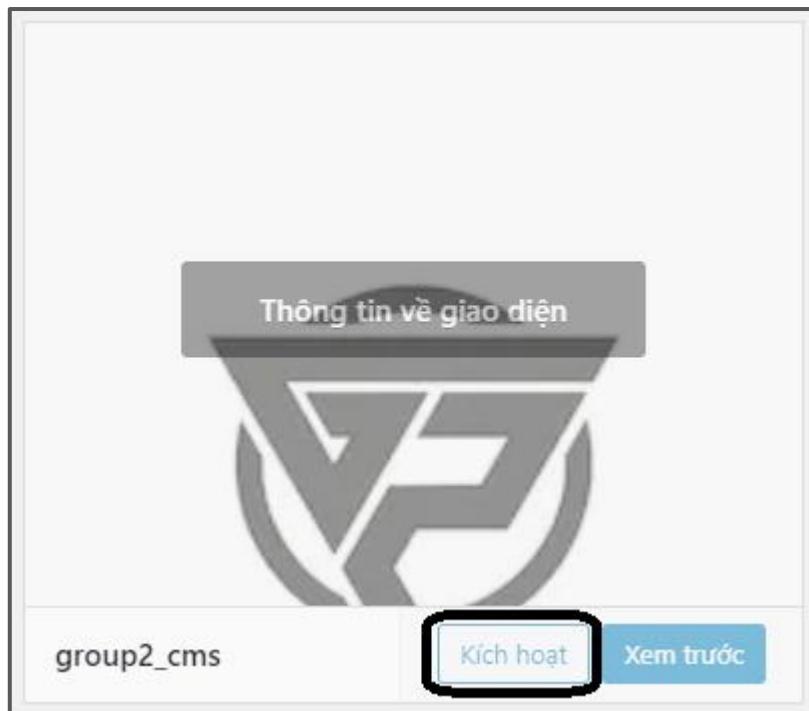
Hình 52. Thêm logo vào theme đã tạo

Kết quả sau khi bạn đã thêm hình vào file **group2_cms**.

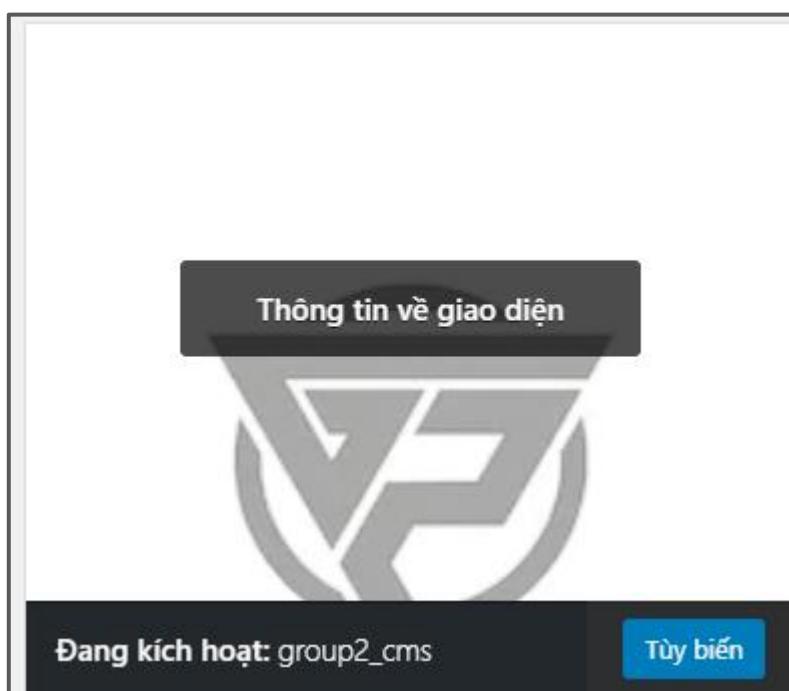


Hình 53. Kết quả sau khi thêm

-Truy cập trang web wp-admin để kích hoạt giao diện chúng ta vừa tạo

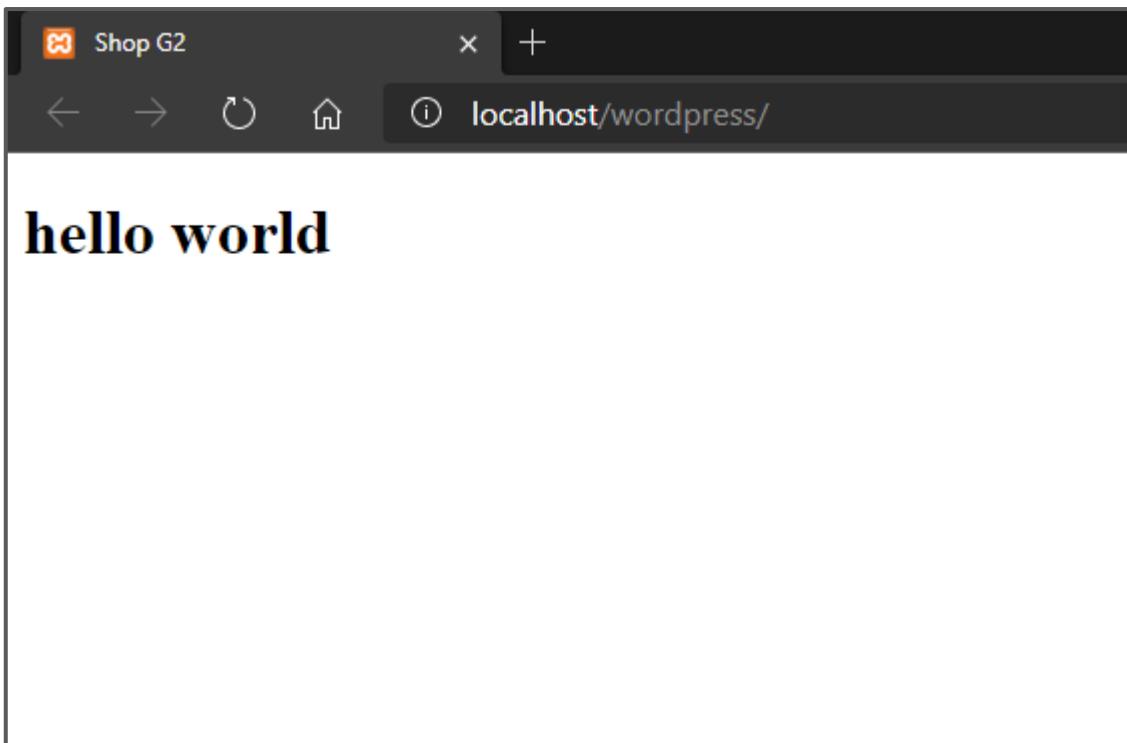


Hình 54. Kích hoạt theme



Hình 55. Sau khi kích hoạt hành công

-Tiếp theo, truy cập vào giao diện chính của theme vừa kích hoạt.



Hình 56. Show kết quả sau khi kích hoạt

Bây giờ, chúng ta viết **Code** cho phần **Header** và **Footer**.

Luồng đi của các file php sẽ chạy từ trên xuống dưới trong file **index.php**

```
index.php X
wp-content > themes > group2 cms > index.php > ...
1  <?php
2  // gọi file header.php
3  get_header();
4  ?>
5
6  <!-- content project -->
7  <h1>hello wordpress</h1>
8  <!-- end content -->
9
10 <?php
11 // gọi file footer.php
12 get_footer();
13 ?>
```

The image shows a code editor with the file 'index.php' open. The code is written in PHP and defines the structure of a single-page template. It includes calls to 'header.php' and 'footer.php' files. The code is numbered from 1 to 13. The code editor interface shows the file path 'wp-content > themes > group2 cms > index.php > ...' at the top.

Hình 57. Code bô cục của một trang

Và file header sẽ chạy đầu tiên nên chứa thẻ mở **<body>** và **</html>**

Nên thẻ đóng `</body>` và `<html>` bắt buộc phải được viết ở trong **footer.php** (footer chạy cuối cùng) để ngăn không bị ‘lỗi logic’ trong quá trình khởi chạy project.

Sau khi đã tạo được theme và kích hoạt nó. Bây giờ chúng ta đi sâu hơn vào các mục và thành phần khác.

CHƯƠNG VI: BÁO CÁO ĐỒ ÁN

1. Giới thiệu đồ án

Yêu cầu	Nội dung
Hình thức	Trang web bán hàng trực tuyến.
Tên đồ án	G2 SHOP
Mục đích	Bán hàng online (Quần áo, giày dép).
Đối tượng hướng tới	Các bạn trẻ đam mê mua sắm hoặc các khách hàng không có thời gian để tới shop lựa chọn.
Nghĩa	<p>- Mang lại doanh thu và tiết kiệm thời gian cho cả người bán và người mua.</p> <p>- Góp phần thúc đẩy kinh tế trong và ngoài nước.</p>

Đồ án bán hàng, sử dụng Wordpress và plugin hỗ trợ là Woocommerce cho việc xử lý sản phẩm.

Sản phẩm chủ yếu là các loại quần, áo, giày, nón. Tên website là G2 Shop.

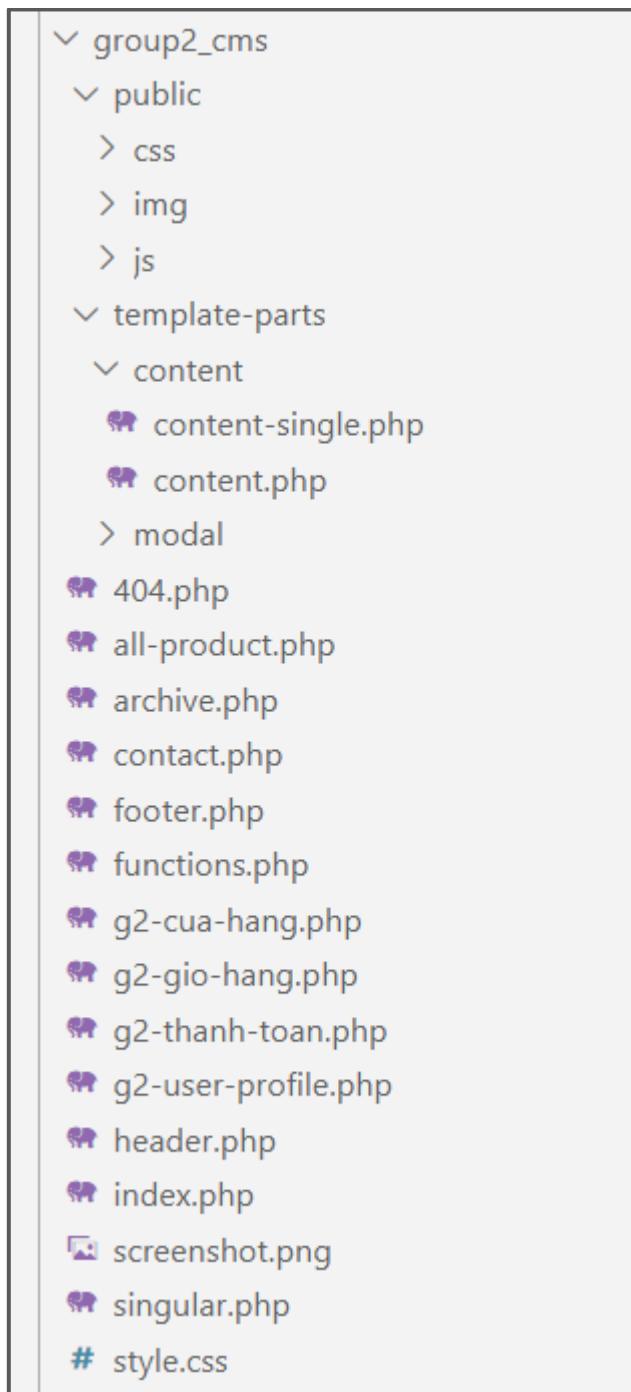
Trang web được lấy cảm hứng từ nhiều nơi mang phong cách đơn giản nhưng đầy sang trọng.

Vì sử dụng Wordpress nên trang web gồm 3 phần chính:

- ✧ Header
- ✧ Content
- ✧ Footer

Header và Footer là 2 thành phần không thể thiếu trong các page của trang web. Content là phần nội dung chính, nó thay đổi tùy thuộc vào người dùng (trang chủ, danh mục, ...)

Cấu trúc thư mục để tạo nên trang web đã hoàn chỉnh như sau:



Hình 58. Cấu trúc thư mục hoàn chỉnh

Thông nhất lại, trang web sẽ thực hiện khá đầy đủ các chức năng cần có cho việc bán hàng. Về chi tiết từng phần sẽ được mô tả bên dưới.

2. Header và Footer

1. Header

Giao diện của trang web sẽ hiển thị như sau



Hình 59. Header của trang web

Để tạo ra giao diện trên, trong file header.php ta tạo đoạn code như bên dưới.

```

101 <body>
102   <header>
103     <div class="header">
104       <div class="header__top">
105         <a href=<?php echo esc_url(home_url()) ?>>
106           <?php
107             if (has_custom_logo()) {
108               echo '/public/img/logo.png" alt="logo">
112             }
113           ?
114         </a>
115         <ul class="nav__link">
116           <li class="nav__link--item">
117             |   <a href=<?php echo esc_url(home_url()) ?>>Home</a>
118           </li>
119           <li class="nav__link--item">
120             |   <a href=<?php echo esc_url(get_permalink(get_page_by_path('tat-ca-san-pham'))) ?>>sản phẩm</a>
121           </li>
122           <li class="nav__link--item">
123             |   <a href=<?php echo esc_url(get_permalink(get_page_by_path('contact'))) ?>>liên hệ</a>
124           </li>
125           <li class="nav__link--item">
126             |   <?php global $woocommerce; ?>
127             |   <a class="cart-contents" style="color:#FEC144"
128               href=<?php echo $woocommerce->cart->get_cart_url(); ?>
129               title=<?php _e('View your shopping cart', 'woothemes'); ?>>
130               |   <?php echo sprintf(_n('%d item', '%d <i class="fas fa-cart-plus"></i>', $woocommerce->cart->cart_contents_count, 'woothemes'), $woocommerce->cart->cart_contents_count); ?>
131               |   <!-- <?php echo $woocommerce->cart->get_cart_total(); ?> -->
132             </a>
133           </li>
134         </ul>
135     </div>

```

Hình 60. Tạo giao diện header trong header.php (part 1 - header-top)

Từ dòng 105 đến 115, là phần tạo logo (G2 Shop) trong phần giao diện, nó có thể chỉnh sửa trong wp-admin (website).

Nếu không chỉnh sửa thì mặc định sẽ là logo tĩnh.

Đối với các dòng 118, 121, 124. Đường dẫn trong thẻ a sẽ là lấy link của trang (page) theo slug của trang đó.

Còn đối dòng 126 trở đi, đó là phần lấy ra số lượng sản phẩm khi người dùng thêm sản phẩm vào giỏ hàng.

```

137      <?php
138      if (isset($_COOKIE['userName'])) {
139          $user = new User();
140          $user->setUser($_COOKIE['userName']);
141      }
142      <div class="dropdown">
143          <button>
144              <img onclick="myFunction()" alt="Avatar"
145                  src=<?php echo $user->getFilter("avatar")?>" class="avatar dropbtn img-fluid" style="box-shadow: 1px 1px 4px 0px rgba(0, 0, 0, 0.75);">
146          </button>
147          <div id="myDropdown" class="dropdown-content">
148              <a href=<?php echo esc_url(get_permalink(get_page_by_path('profile-user')))?>">Profile</a>
149              <form action="" method="post">
150                  <input type="hidden" name="logout" value="true">
151                  <input type="submit" value="Logout" class="form-control">
152              </form>
153              <!-- <a href="#about">Logout</a> -->
154          </div>
155      </div>
156      <?php
157      } else {
158          echo '';
159      }
160      <li class="nav__link--item"><a data-toggle="modal" href="javascript:void(0)" onclick="openLoginModal();">Đăng Nhập</a></li>;
161  }
162  ?>
163  </ul>
164
165  <?php
166  get_template_part('template-parts/modal/modal', 'login')
167  ?>
168

```

Hình 61. Tạo giao diện header trong header.php (part 2 - header-top)

3. Footer

Về giao diện hiển thị như sau



Hình 62. Giao diện footer

Để tạo được giao diện trên, ta làm như sau trong file footer.php

Đầu tiên là phần logo, tương tự như ở phần header.

```

1  <footer id="footer">
2    <div class="footer-contact" style="background: url('<?php echo bloginfo('template_directory') ?>/public/img/footer.png');">
3      <div class="container">
4          <div class="row">
5              <div class="col-xs-12 col-sm-12 col-md-4 col-lg-4 footer-logo">
6                  <a href=<?php echo esc_url(home_url()) ?>">
7                      <?php
8                          if (has_custom_logo()) {
9                              echo ' ';
10                         } else {?
11                             <img src=<?php echo bloginfo('template_directory') ?>/public/img/logo_transparent.png" alt="logoFooter" class="img-fluid">
12                         ?>
13                     </a>
14                 </div>

```

Hình 63. Hiển thị giao diện footer trong footer.php (part 1 - logo)

Các thành phần còn lại

```

15     <div class="col-xs-12 col-sm-12 col-md-4 col-lg-8 contact">
16         <div class="row">
17             <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4 ">
18                 <h6 class="footer-title">HỖ TRỢ KHÁCH HÀNG</h6>
19                 <li><a href=<?php echo esc_url(home_url()) ?>">Trang chủ</a></li>
20                 <li><a href=<?php echo esc_url( get_permalink( get_page_by_path( 'tat-ca-san-pham' ) ) )?>">Sản phẩm</a></li>
21                 <li><a href=<?php echo esc_url( get_permalink( get_page_by_path( 'contact' ) ) )?>">liên hệ</a></li>
22             </div>
23             <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
24                 <h6>THÔNG TIN</h6>
25                 <li><a href="#" onclick="updatingLinkPage();">Hướng dẫn mua hàng</a></li>
26                 <li><a href="#" onclick="updatingLinkPage();">Hình thức thanh toán</a></li>
27                 <li><a href="#" onclick="updatingLinkPage();">Chính sách bảo mật thông tin</a></li>
28             </div>
29             <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
30                 <h6>KẾT NỐI VỚI CHUNG TÔI</h6>
31                 <a href="#" onclick="updatingLinkPage();"><i class="fab fa-facebook-f"></i></a>
32                 <img src=<?php echo bloginfo('template_directory') ?>/public/img/payment.png" alt="payment" class="img-fluid payment-logo">
33             </div>
34         </div>
35     </div>
36     <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
37         <div class="footer-author">
38             <span>Copyright © 2020 G2 SHOP </span>
39         </div>
40     </div>
41 </div>
42 </footer>

```

Hình 64. Hiển thị giao diện footer trong footer.php (part 2)

Thế là xong phần Header và Footer của website, tiếp đến là phần content được trình bày bên dưới.

4. Trang chủ

Về phần trang chủ, chỉ hiển thị 2 thành phần sau:

- Danh sách sản phẩm mới nhất bằng slide
- Danh sách (6 sản phẩm) của danh mục

Đầu tiên là phần sản phẩm mới nhất về slide



Hình 65. Hiển thị sản phẩm mới nhất bằng slide (10 sản phẩm)

Về sản phẩm, nó hiển thị ảnh, tên, các button để di chuyển qua lại và nó cũng tự động di chuyển bằng bootstrap.

Để làm được điều này, ta tạo một cấu trúc thư mục template-parts như sau



Hình 66. Cấu trúc thư mục template-parts

Thư mục này tạo ra để tiện quản lý các thành phần hơn, tất nhiên nó sẽ được gọi vào file index.php để hiển thị

7	<!-- slide show -->
8	<div class="slide">
9	<?php
10	get_template_part('template-parts/content/content')
11	?>
12	</div>

Hình 67. Gọi file trong template-parts vào index.php

Ở trong file content.php, ta cấu hình nó như sau

```

1 <div id="myCarousel" class="carousel slide" data-ride="carousel" style="margin-top:-20px">
2   <!-- Indicators -->
3   <ol class="carousel-indicators">
4     <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
5     <li data-target="#myCarousel" data-slide-to="1"></li>
6     <li data-target="#myCarousel" data-slide-to="2"></li>
7   </ol>
8   <!-- Wrapper for slides -->
9   <div class="carousel-inner" style="width:30%;margin:auto">
10    <div class="item active">
11      
12      <div class="carousel-caption" style="color:#000;top:-8%">
13        <h3>G2 Shop</h3>
14        <p>Welcome!</p>
15      </div>
16    </div>
17    <?php
18    $products = getProduct();
19    foreach ($products as $product) {
20      ?
21      <div class="item" title="php echo $product-&gt;name?&gt;"&gt;
22        &lt;img src="<?php echo get_the_post_thumbnail_url($product-&gt;id) ?&gt;" alt="<?php echo $product-&gt;name?&gt;" style="width:100%;"
23        &lt;div class="carousel-caption" style="color:#000;top:-8%"&gt;
24          &lt;h3&gt;<?php echo $product-&gt;name?&gt;&lt;/h3&gt;
25        &lt;/div&gt;
26      &lt;/div&gt;
27      &lt;?php } ?&gt;
28    &lt;/div&gt;
</pre

```

Hình 68. Khởi tạo slide hiển thị sản phẩm mới nhất trong content.php (part 1)

Có 2 phần chính được quan tâm:

Đầu tiên, dòng 10 hay nói ô kê màu đầu tiên. Nó sẽ hiển thị là logo của G2 Shop, hay nói cách khác, nó hiển thị đầu tiên khi website được mở bằng class active.

Ô kẽ màu thứ 2 nó sẽ hiển thị nhiều nhất 10 sản phẩm mới nhất. Dòng 18 là hàm để lấy các sản phẩm đấy ra (được tạo trong file functions.php)

```

88  /**
89   * Hàm lấy 10 sản phẩm
90   *
91   * Số lượng sản phẩm là
92   */
93  function getProduct()
94  {
95      $args = array(
96          'status' => 'publish',
97          'numberposts' => 10,
98      );
99      $products = wc_get_products($args);
100     return $products;
101 }

```

Hình 69. Hàm lấy 10 sản phẩm mới nhất trong functions.php

Tiếp đến là các button để di chuyển qua lại giữa các sản phẩm

```

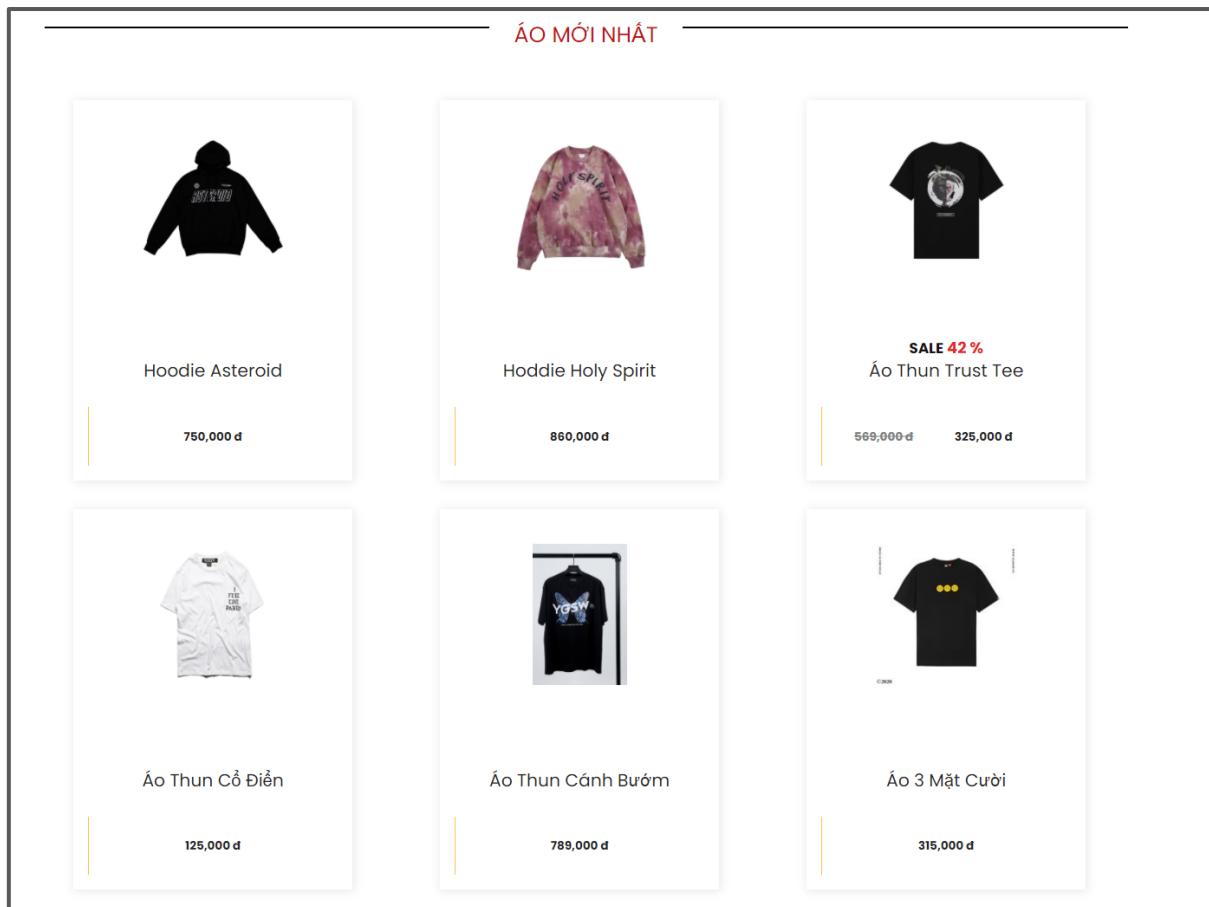
29  <!-- Left and right controls -->
30  <a class="left carousel-control" href="#myCarousel" data-slide="prev">
31      <span class="glyphicon glyphicon-chevron-left"></span>
32      <span class="sr-only">Previous</span>
33  </a>
34  <a class="right carousel-control" href="#myCarousel" data-slide="next">
35      <span class="glyphicon glyphicon-chevron-right"></span>
36      <span class="sr-only">Next</span>
37  </a>
38  </div>

```

Hình 70. Các button di chuyển trong content.php

Thέ là xong phần slide, tiếp đến là hiển thị các sản phẩm theo danh mục (6 sản phẩm)

Về giao diện nó sẽ như thế này



Hình 71. Giao diện phần hiển thị 6 sản phẩm theo danh mục

Để làm được điều này, trong file functions.php ta tạo một hàm để hiện thi nó ra

```

197 /**
198 * Xuất ra html của 6 sản phẩm theo danh mục
199 *
200 * Giá trị là tên của danh mục
201 */
202 function getHtmlProductByCategory($category)
203 {
204     echo '<div class="container page-wrapper">
205         <div class="page-inner">
206             <div class="heading_hotdeal">
207                 <h2 class="title-head" title="' . $category . '">
208                     <a href="">' . $category . ' Mới Nhất</a>
209                 </h2>
210             </div>

```

Hình 72. Hiển thị 6 sản phẩm theo danh mục trong functions.php (part 1)

Phần trên này, là phần hiển thị ra tên danh mục

```

212 $products = getProductByCategory($category);
213 foreach ($products as $product) {
214     echo '<a href="' . esc_url(get_post_permalink($product->id, false, false)) . '"'
215         . '<div class="el-wrapper" title="" . $product->name . ">'
216         . '<div class="box-up">
217             <img class="img" src="" . get_the_post_thumbnail_url($product->id) . "" alt="">
218             <div class="img-info">
219                 <div class="info-inner">';
220     if ($product->sale_price > 0) {
221         echo '<span class="p-sale"> <span class="sale">Sale</span> ' . getSale($product->regular_price, $product->sale_price) . ' %</span>';
222     }
223     echo '<span class="p-name">' . $product->name . '</span>
224         </div>
225         </div>
226         <div class="box-down">
227             <div class="h-bg">
228                 <div class="h-bg-inner"></div>
229             </div>';
230

```

Hình 73. Hiển thị 6 sản phẩm theo danh mục (part 2)

```

231     if ($product->sale_price > 0) {
232         echo '<a class="shop-cart" href="' . esc_url(get_post_permalink($product->id, false, false)) . '"'
233             . '<span class="price" style="left:25%"><s style="color:#818181;left:15%">' . number_format_i18n($product->regular_price) . ' đ</s>
234             </span>
235             <span class="price-sale" style="left:65%">' . number_format_i18n($product->sale_price) . ' đ</span>
236             <span class="add-to-shop-cart">
237                 <span class="txt">Thêm vào giỏ hàng</span>
238             </span>
239         </a>';
240     } else {
241         echo '<a class="shop-cart" href="' . do_shortcode('[add_to_cart_url id="' . $product->id . '"]') . '"'
242             . '<span class="price">' . number_format_i18n($product->price) . ' đ</span>
243             <span class="add-to-shop-cart">
244                 <span class="txt">Thêm vào giỏ hàng</span>
245             </span>
246         </a>';
247         echo '</div>
248             </div>
249         ';
250     }
251 }
252 echo '
253     </div>
254     </div>
255 </div>';
256

```

Hình 74. Hiển thị 6 sản phẩm theo danh mục (part 3)

Đối với dòng 240, có sử dụng hàm do_shortcode(). Shortcode nói đơn giản là một hàm để lấy ra dòng lệnh được cấu hình từ trước. Ở đây là add_to_cart_url, shortcode này là lấy sản phẩm để thêm vào giỏ hàng bằng cách truyền id sản phẩm đó vào. Tất nhiên đây là shortcode của plugin Woocommerce.

Dòng 212 (hình 16), có hàm này để lấy sản phẩm theo danh mục và hàm getSale (dòng 221) để lấy phần trăm giảm giá sản phẩm, còn lại các phần khác chỉ là xuất các giá trị nó ra mà thôi.

```

129  /**
130  * Hàm lấy sản phẩm theo danh mục
131  *
132  * Danh mục ở đây là nhập vào để hiển thị ra giao diện chính (Home)
133  * Số lượng sản phẩm là 6
134  */
135 function getProductByCategory($category)
136 {
137     $args = array(
138         'post_type' => 'product',
139         'category' => array($category),
140         'orderby' => 'name',
141         'numberposts' => 6,
142     );
143     $products = wc_get_products($args);
144     return $products;
145 }

```

Hình 75. Hàm lấy sản phẩm theo danh mục trong functions.php

```

147 /**
148 * Hàm tính phần trăm giảm giá của sản phẩm
149 *
150 */
151 function getSale($price, $sale_price)
152 {
153     $sale = 100 - ((intval($sale_price) * 100) / intval($price));
154     return intval($sale);
155 }

```

Hình 76. Hàm tính phần trăm giảm giá sản phẩm trong functions.php

Cuối cùng để hiển thị ra giao diện, ta gọi hàm vào file index.php như sau

```

<!-- Lấy 6 sp theo danh mục -->
<div class="content" style="margin-top:20px;">
    <?php
        getHtmlProductByCategory( 'Áo' );
        getHtmlProductByCategory( 'Giày' );
    ?>
</div>

```

Hình 77. Gọi hàm lấy 6 sản phẩm theo danh mục vào index.php

Thế là xong phần Trang chủ website

5. Trang Sản Phẩm (Tất cả sản phẩm)

Từ giờ, mình sẽ sử dụng shortcode của Woocommerce để xuất các sản phẩm theo ý muốn.

Đầu tiên, giao diện sẽ hiển thị như sau

Tất cả sản phẩm:

Showing 1–8 of 18 results

Áo 3 Mặt Cười
315.000 ₫
[Add to cart](#)

Áo Thun Cánh Bướm
789.000 ₫
[Add to cart](#)

Áo Thun Cổ Điển
125.000 ₫
[Add to cart](#)

Áo Thun Trust Tee
569.000 ₫ - 325.000 ₫
[Add to cart](#)

Giày Carina Puma
29.000.000 ₫ - 27.000.000 ₫
[Add to cart](#)

Giày CloudFoam Pure
17.000.000 ₫
[Add to cart](#)

Giày Ecoskate Boarding
14.500.000 ₫
[Add to cart](#)

Giày Nike DownShifter
11.000.000 ₫
[Add to cart](#)

1 2 3 →

Hình 78. Giao diện tất cả sản phẩm (part 1)

Ở đây mình cho mặc định mỗi trang hiển thị 8 sản phẩm. Mình có 18 sản phẩm nên nó hiển thị 3 trang, cùng xem trang thứ 3 bên dưới.

Tất cả sản phẩm:

Showing 17–18 of 18 results

Quần Sweat Pan
185.000 ₫
Add to cart

Quần Thun Trẻ Trung
250.000 ₫
Add to cart

← 1 2 3 →

Hình 79. Giao diện tất cả sản phẩm (part 2)

Để làm được như vậy, ta tạo một file để hiện thị tất cả sản phẩm (ở đây mình tạo là all-product.php)

```

1  <?php
2  /* Template Name: G2-All-Product */
3  get_header();
4 ?>
5
6  <div class="all_product">
7      <div class="container">
8          <h2 class="products_title">
9              <?php echo get_the_title() . ':' ?>
10         </h2>
11         <div class="products_content">
12             <?php
13                 echo do_shortcode('[products columns=4 paginate=true order=ASC per_page=8]');
14             ?>
15         </div>
16     </div>
17 </div>
18
19 <?php
20 get_footer();
21

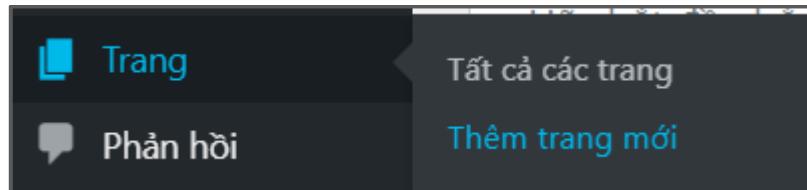
```

Hình 80. Khởi tạo trang all-product.php

Ở dòng 2, ta đặt tên cho trang vừa tạo là G2-All-Product (đặt tên gì cũng được cả). Dòng 9, để lấy tên của trang ta tạo là gì (tên cấu hình trong website wp-admin) và dòng 13 ta gọi shortcode của Woocommerce là products (tất cả sản phẩm) với columns (cột) là 4,

pagination (phân trang) bằng true, order là sắp xếp, per_page bằng 8 là mỗi trang có 8 sản phẩm.

Sau đó ta vào wp-admin (website), chọn Page (Trang), sau đó thêm trang mới



Hình 81. Thêm trang mới trong wp-admin

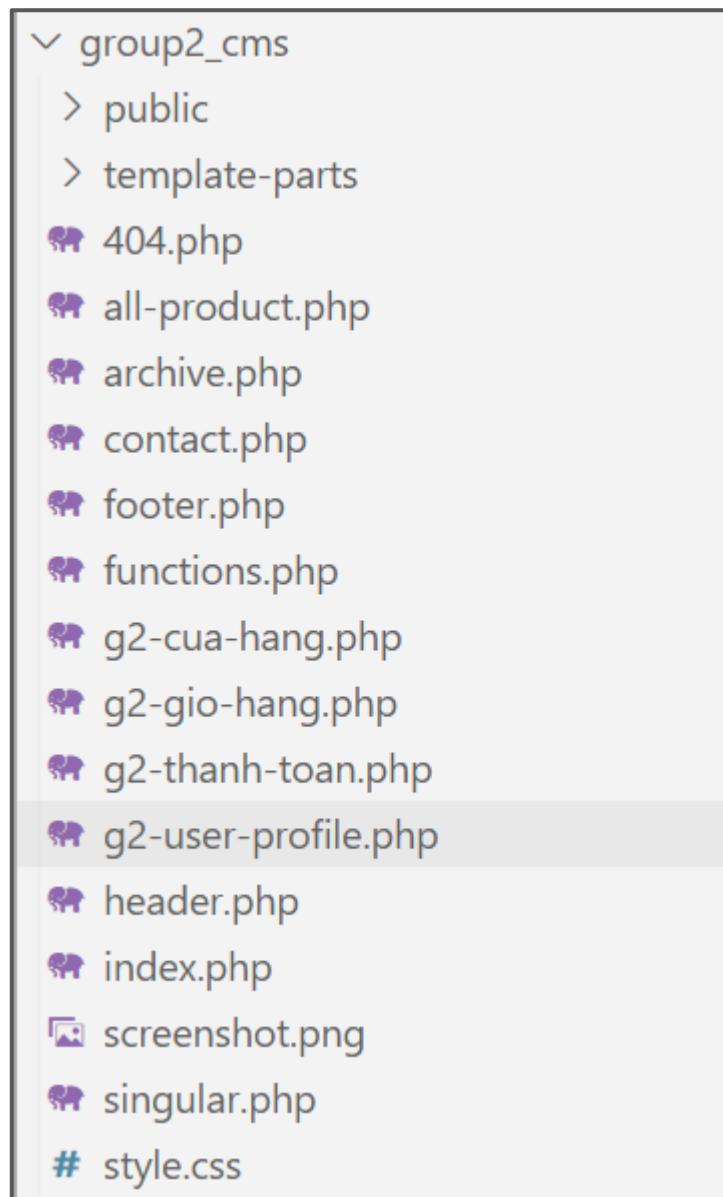
Hình 82. Tạo trang tất cả sản phẩm trong wp-admin

Ô kẽ màu đầu tiên, là đặt tên cho trang, còn ô thứ 2 bên dưới là chọn mẫu file giao diện (chọn G2-All-Product là tên của file all-product.php dòng thứ 2 lúc nãy ta vừa tạo). Nó sẽ có giao diện là file all-product.php

6. Trang Danh Mục (Danh mục sản phẩm)

Trang này ta cũng làm sử dụng ‘shortcode’ của **Woocommerce** như trang tất cả sản phẩm, nhưng nó sẽ hơi khác một chút.

Trong thư mục tạo 2 file, một file là singular.php và 1 file là **archive.php**



Hình 83. Cấu trúc thư mục đồ án

Cấu hình 2 file như sau:

```

wp-content > themes > group2_cms > & singular.php
1   <?php
2     get_header();
3
4     if (is_tax()) {
5       get_template_part('archive');
6     } else {
7       get_template_part('template-parts/content/content', 'single');
8     }
9     get_footer();
  
```

Hình 84. Cấu trúc file singular.php

Ở đây, dòng thứ 4 kiểm tra xem có phải đường dẫn truyền vào là một loại thuộc danh mục sản phẩm hay không. Nếu đúng thì sẽ gọi file archive.php, ngược lại nó sẽ trả về trang single.php (chi tiết sản phẩm) trong thư mục template-parts.

Trong file archive.php cấu hình của nó chịu ảnh hưởng hay nói cách khác nó được gọi qua file singular.php nên không cần gọi header và footer nữa.

```
wp-content > themes > group2_cms > 📄 archive.php > ...
1  <?php
2  $cat_slug = getCatSlugById(get_queried_object_id());
3  ?>
4  <div class="cate_product">
5   <div class="container">
6    <h2 class="products_title">
7     <?php the_archive_title() ?>
8    </h2>
9    <div class="products_content">
10   <?php
11   echo do_shortcode('[product_category category='.$cat_slug.' per_page=5
12   columns=4 orderby=date order=desc paginate=true]')
13   ?>
14  </div>
15 </div>
16 </div>
17
```

Hình 85. Cấu hình file archive.php (hiển thị sản phẩm danh mục)

Ở dòng 7, nó hiển thị trang này là trang này là trang gì. Còn dòng 11 và 12, sử dụng shortcode product_category để xuất sản phẩm theo danh mục bằng id với sản phẩm mỗi trang là 5, số cột là 4, sắp xếp theo ngày và tên.

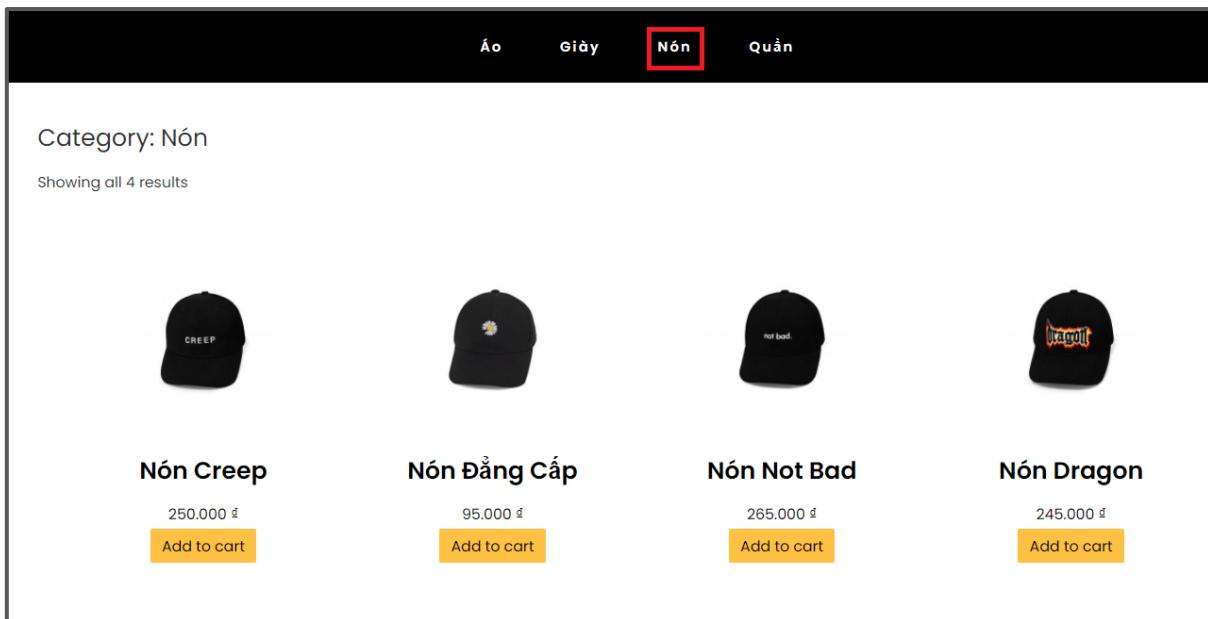
Vì ở đây, hàm shortcode này yêu cầu tham số danh mục là kiểu slug nên ở dòng 2, gọi hàm lấy danh mục theo id (được viết trong functions.php).

Hàm get_queried_object_id là để lấy id của trang hiện tại.

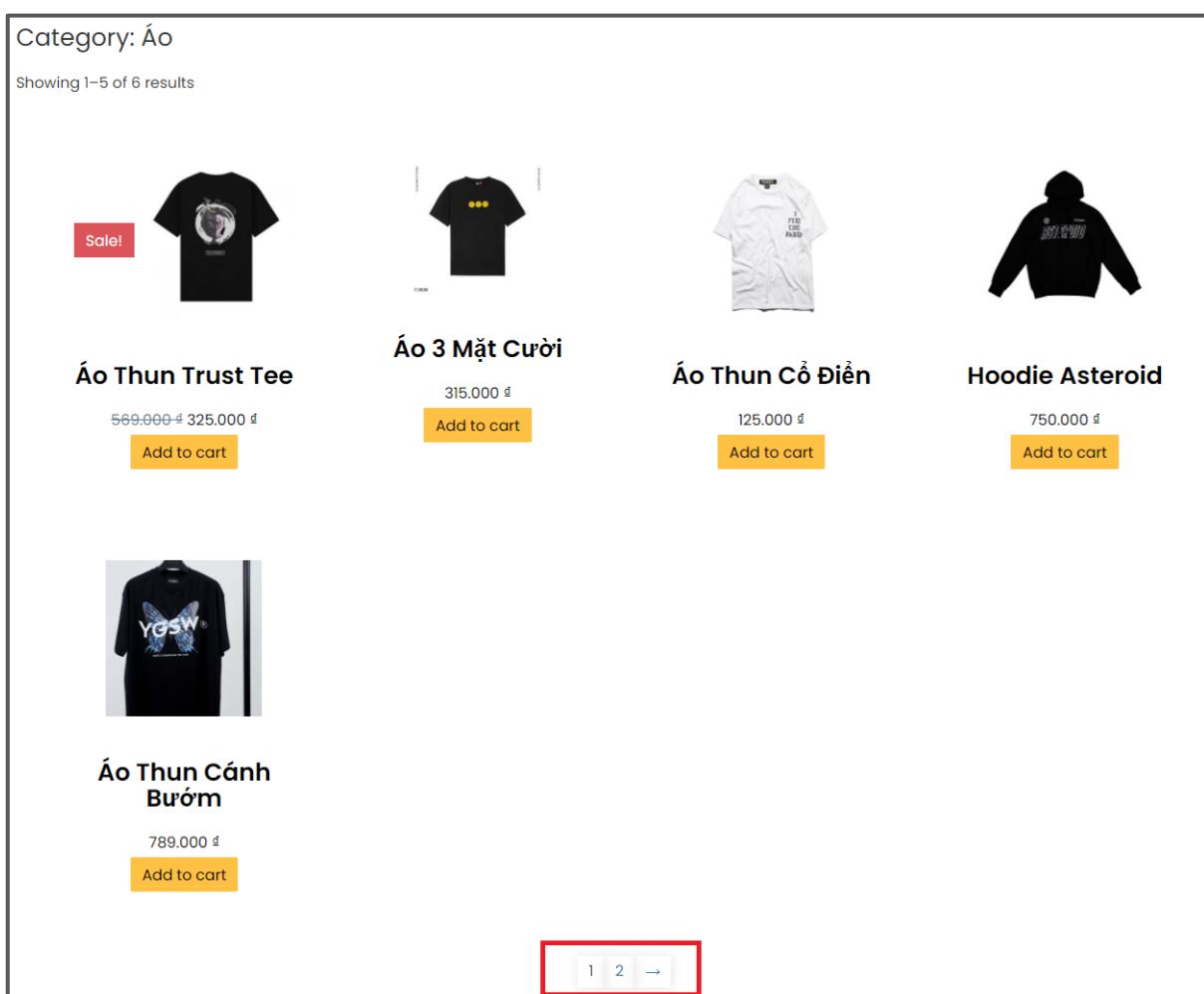
```
103 /**
104  * Hàm lấy slug danh mục theo id
105  *
106  */
107 function getCatSlugById($cat_id)
108 {
109  $term = get_term_by('id', $cat_id, 'product_cat', 'ARRAY_A');
110  return $term['slug'];
111 }
```

Hình 86. Hàm lấy slug danh mục theo id trong functions.php

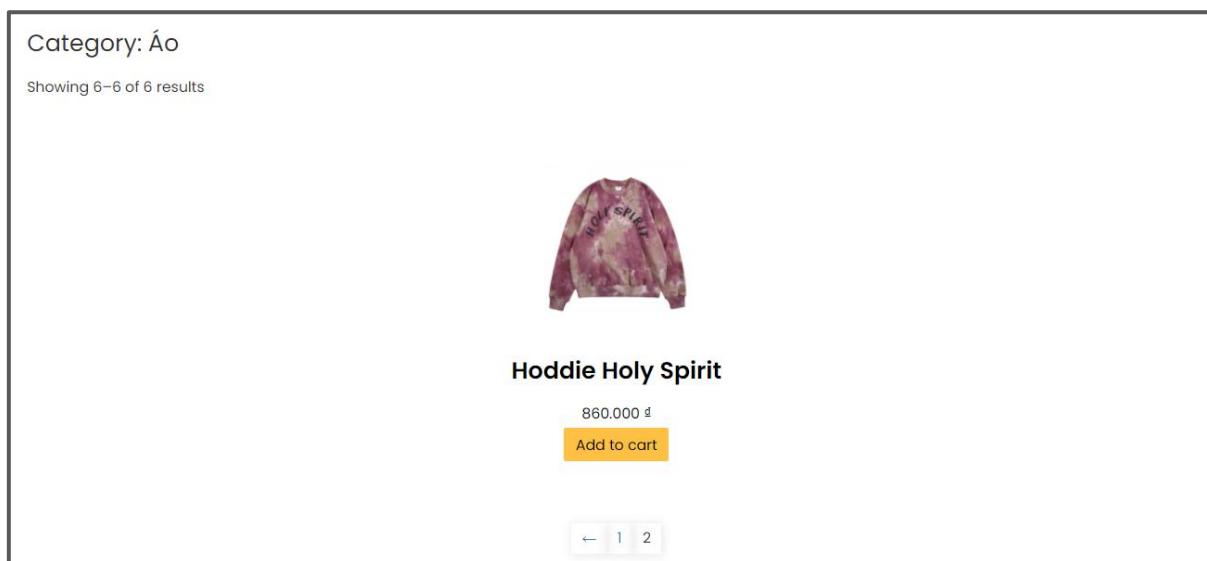
Kết quả, giao diện sẽ có được như thế này.



Hình 87. Giao diện sản phẩm khi nhấn vào danh mục (menu - part 1)



Hình 88. Giao diện sản phẩm khi nhấn vào danh mục (part 2 - phân trang)

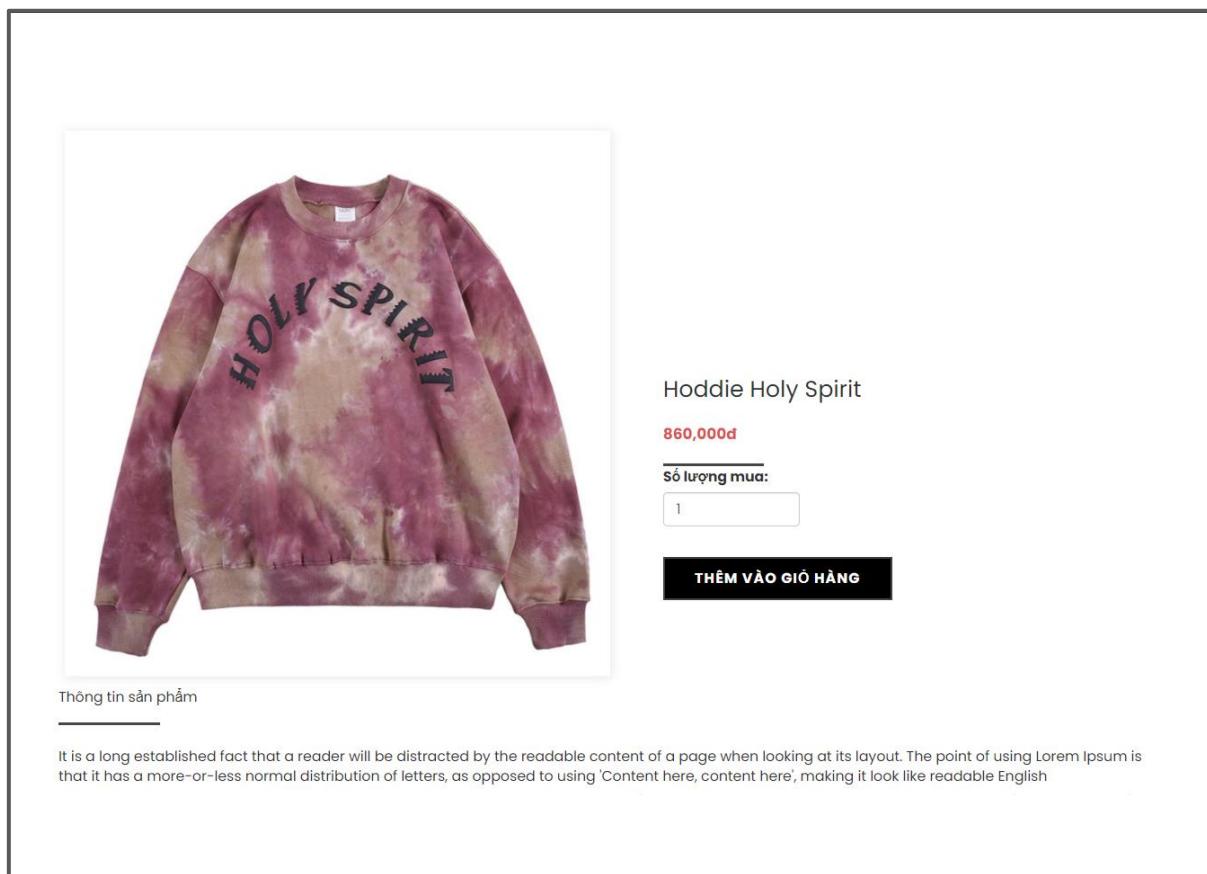


Hình 89. Giao diện sản phẩm khi nhấp vào danh mục (part 3 - phân trang)

7. Trang chi tiết sản phẩm

Trở lại với file singular.php, ta có điều kiện khi nó không phải là trang thuộc danh mục nên nó sẽ chuyển hướng gọi trang single.php trong template-parts.

Giao diện hiển thị như sau.



Hình 90. Giao diện trang single.php (chi tiết sản phẩm)

Để tạo được giao diện trên, ta cấu hình file single.php như sau

```
wp-content > themes > group2 cms > template-parts > content > content-single.php > section#sec-9bc7.u-clearfix.u-section-2
1  <?php
2  $product = wc_get_product(get_the_ID());
3  ?>
4  <section class="u-clearfix u-section-2" id="sec-9bc7">
5      <div class="u-clearfix u-sheet u-valign-middle-lg u-sheet-1">
6          <div class="u-container-style u-expanded-width u-product u-product-1">
7              <div class="u-container-layout u-valign-top-md u-valign-top-sm u-valign-top-xs u-container-layout-1">
8                  <!--product_image-->
9                  
10                 <!--/product_image-->
11                 <div class="u-align-left u-container-style u-expanded-width-md u-expanded-width-sm u-expanded-width-xs u-group u-shape-rectangle u-group-1">
12                     <div class="u-container-layout u-valign-middle-lg u-valign-middle-xl u-container-layout-2">
13                         <!--product_title-->
14                         <h2 class="u-product-control u-text u-text-1">
15                             <div class="u-product-title-link" style="font-weight:450;text-transform: capitalize">
16                                 <?php echo $product->name ?>
17                             </div>
18                         </h2>
```

Hình 91. Giao diện single trong single.php (part 1 - ảnh và tên sản phẩm)

```
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
<!--product_price-->
<div class="u-product-control u-product-price u-product-price-1">
    <div class="u-price-wrapper u-spacing-10">
        <!--product_regular_price-->
        <?php
        if ($product->sale_price > 0) {
            ?
            <div class="u-price u-text-palette-2-base" style="font-size: 1.5rem; font-weight: 700;text-decoration: line-through;color:#8a8686!important">
                <!--product_regular_price_content--><?php echo number_format_i18n($product->regular_price) ?>
                <!--/product_regular_price_content-->
            </div>
            <div class="u-price u-text-palette-2-base" style="font-size: 1.5rem; font-weight: 700;">
                <!--product_regular_price_content--><?php echo number_format_i18n($product->sale_price) ?>
                <!--/product_regular_price_content-->
            </div>
        } else {
            ?
            <div class="u-price u-text-palette-2-base" style="font-size: 1.5rem; font-weight: 700;">
                <!--product_regular_price_content--><?php echo number_format_i18n($product->regular_price) ?>
                <!--/product_regular_price_content-->
            </div>
        }
        ?
        <!--/product_regular_price-->
    </div>
</div>
<!--/product_price-->
```

Hình 92. Giao diện single trong single.php (part 2 - Giá tiền sản phẩm)

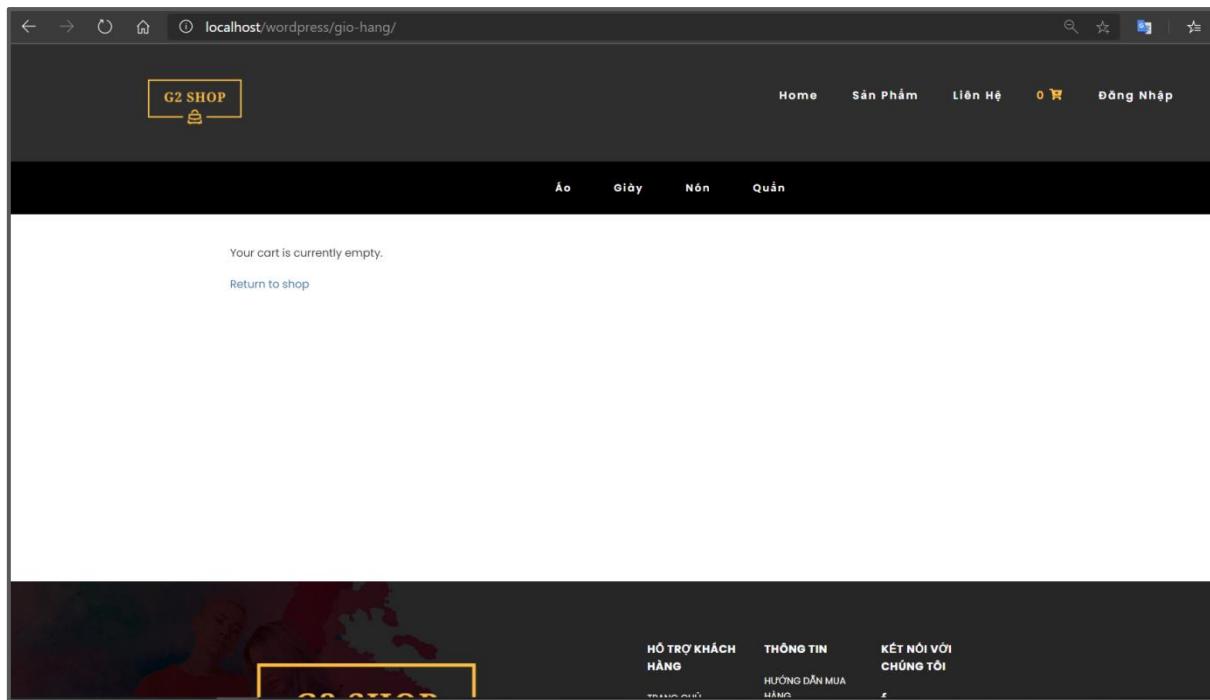
```
48
49 <div class="u-border-3 u-border-grey-dark-1 u-line u-line-horizontal u-line-1"></div>
50 <form action="" method="POST" id="formCart">
51     <label for="qty">Số lượng mua:</label>
52     <input type="hidden" name="productId" value="<?php echo $product->id ?>">
53     <input type="number" name="qty" id="qty" min="1" max="50" value="1" class="form-control" style="width:30%">
54 </form>
55     <a href="<?php echo do_shortcode('[add_to_cart_url id="' . $id . '"]') ?>" class="u-black u-border-2 u-border-grey-75 u-btn u-button-style u-hover-grey-75 u-product-control u-btn-1">Thêm vào giỏ hàng</a>
56     <!--/product_button-->
57 </div>
58 <div class="desc">
59     <h5>Thông tin sản phẩm</h5>
60     <div class="u-border-3 u-border-grey-dark-1 u-line u-line-horizontal u-line-1"></div>
61     <p>
62         <?php
63             echo $product->description ?>
64     </p>
65 </div>
66 </div>
67 <!--/product_item-->
68 <!--/product-->
69 </div>
70 </section>
```

Hình 93. Giao diện single trong single.php (part 3 - Thêm giỏ hàng và Thông tin)

Ở dòng 2 (hình 88), sử dụng hàm mặc định của Woocommerce để lấy thông tin 1 sản phẩm bằng id sản phẩm đó. Sau đó chỉ xuất các giá trị cần thiết ra thôi là được.

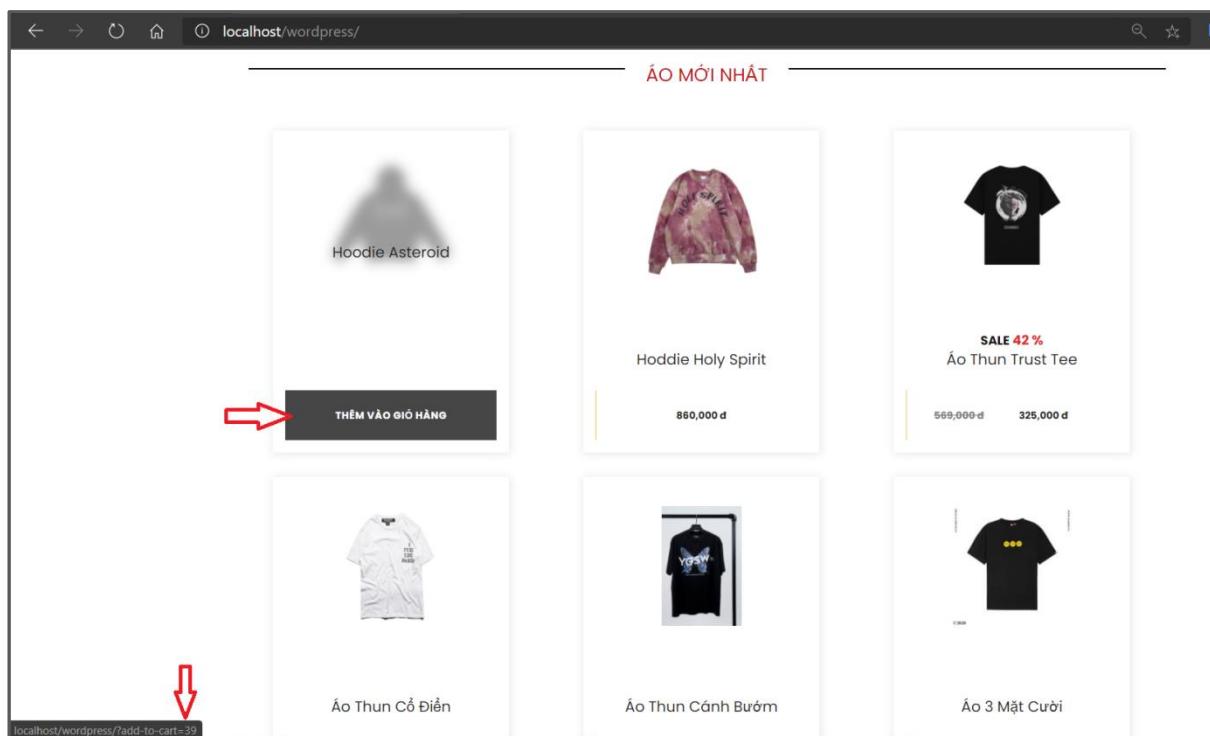
8. Trang Giỏ Hàng và Thanh Toán

Trang giỏ hàng có giao diện như sau



Hình 94. Giao diện trang giỏ hàng khi chưa có sản phẩm

Để thêm sản phẩm ta làm như sau



Hình 95. Hướng dẫn thêm sản phẩm vào giỏ hàng

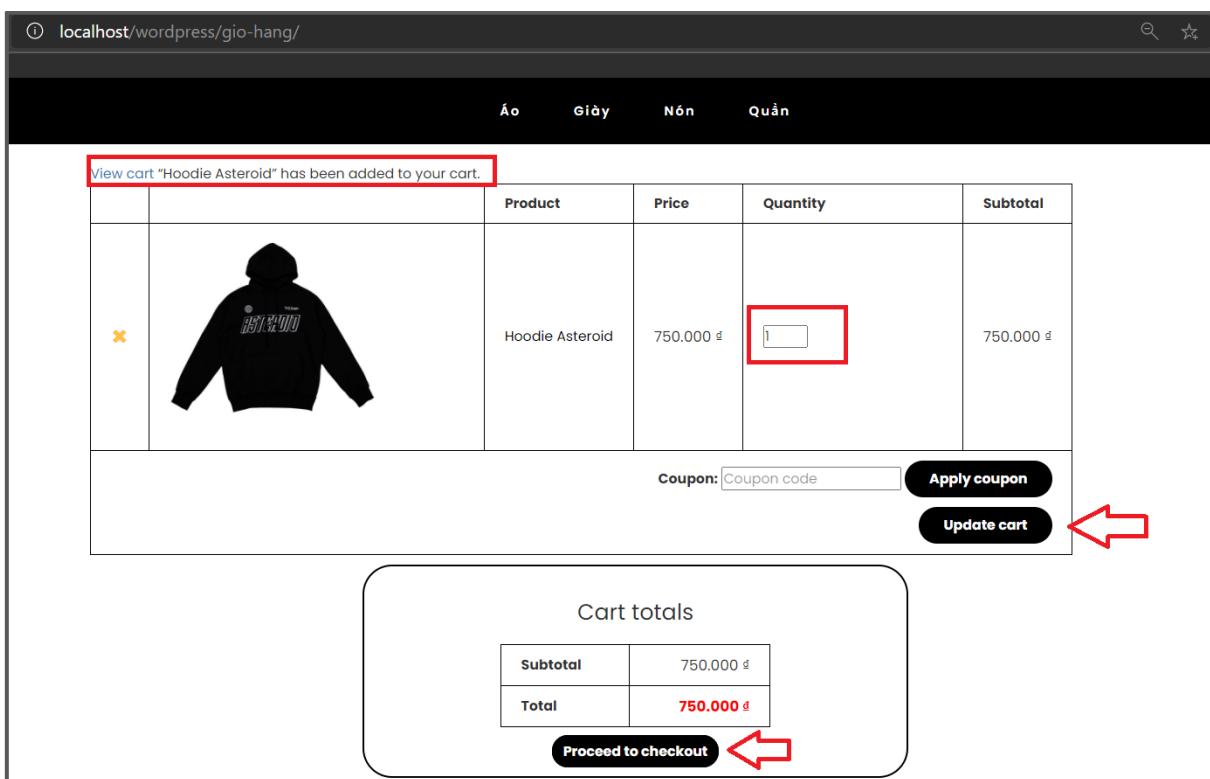
Như hình trên, ta sẽ thấy khi hover vào button Thêm vào giỏ hàng thì sẽ hiển thị đường link “.../?add-to-cart=39”. Số 39 ở đây là id của sản phẩm. Tiếp đó ta nhấn button thì sẽ hiển thị ra như sau.



Hình 96. Kết quả khi thêm vào giỏ hàng

Kết quả nhận được khi nhấn vào button là trên thanh menu (chỗ giỏ hàng), sẽ tăng thêm 1 đơn vị.

Ta lại nhấn vào đó để vào trang giỏ hàng.



Hình 97. Trang giỏ hàng khi mới thêm sản phẩm

Ở đây, nó sẽ hiển thị tên sản phẩm vừa được thêm vào giỏ hàng.

Tiếp đến, phần Quantity (số lượng sản phẩm), trong đó ô input có thể tăng hoặc giảm tùy ý. Sau đó nhấn button Update cart để cập nhật giỏ hàng. Còn button Proceed to checkout trong Cart totals là để tiến hành thanh toán.

Ví dụ, ta tăng sản phẩm lên 3 thì sẽ có thay đổi sau.

View cart "Hoodie Asteroid" has been added to your cart.

		Product	Price	Quantity	Subtotal
		Hoodie Asteroid	750.000 ₫	<input type="button" value="3"/>	750.000 ₫

Coupon:

Cart totals

Subtotal	750.000 ₫
Total	750.000 ₫

Hình 98. Tiến hành cập nhật giỏ hàng (tăng số lượng sản phẩm)

G2 SHOP

Home Sản Phẩm Liên Hệ Đăng Nhập

Áo Giày Nón Quần

Cart updated.

		Product	Price	Quantity	Subtotal
		Hoodie Asteroid	750.000 ₫	<input type="button" value="3"/>	2.250.000 ₫

Coupon:

Cart totals

Subtotal	2.250.000 ₫
Total	2.250.000 ₫

Hình 99. Kết quả update giỏ hàng thành công

Sau khi update cart, có 4 thứ thay đổi. Đầu tiên, giỏ hàng trên menu sẽ tăng hoặc giảm số lượng tùy theo hành động của người dùng (ở đây là tăng lên). Tiếp theo là sản phẩm bao gồm Quantity (số lượng), Subtotal (tổng tiền của sản phẩm). Cuối cùng là total (tổng tiền giỏ hàng) trong Cart totals.

Để tạo nên giao diện trang giỏ hàng, ta cũng sử dụng cách làm tương tự trang tất cả sản phẩm.

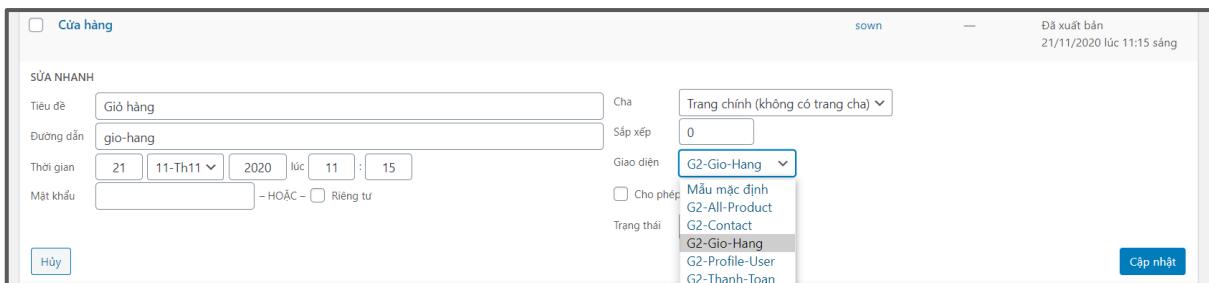
Đầu tiên, tạo 1 file trong theme (tên ở đây là g2-gio-hang.php), với tên Trang là G2-Gio-Hang.

```
wp-content > themes > group2 cms > g2-gio-hang.php > ...
1  <?php
2  /* Template Name: G2-Gio-Hang */
3  get_header();
4  ?>
5  <div class="shopping-cart">
6      <div class="container">
7          |     <?php echo do_shortcode(' [woocommerce_cart] ') ?>
8      </div>
9  </div>
10 <?php
11 get_footer();
```

Hình 100. Cấu trúc file g2-gio-hang.php

Sử dụng shortcode “woocommerce_cart” để gọi html mặc định giỏ hàng của Woocommerce.

Cũng giống như tạo tất cả giỏ hàng, ta lên wp-admin (website) để tạo 1 trang là Giỏ Hàng (hoặc sử dụng trang có sẵn của Woocommerce), sau đó chọn phần Mẫu là G2-Gio-Hang.



Hình 101. Tạo Trang Giỏ Hàng (sử dụng trang có sẵn của Woocommerce)

Sau đó, click button cập nhật là thành công trong việc tạo trang Giỏ Hàng.

Để tiến hành thanh toán ta click vào button Proceed to checkout, sau đó sẽ chuyển qua giao diện trang Thanh toán.

The screenshot shows a checkout form titled "Billing details". At the top, there is a link "Have a coupon? Click here to enter your code". Below the title, there are several input fields:

- First name.***: An empty text input field.
- Last name.***: An empty text input field.
- Company name (optional)**: An empty text input field.
- Country / Region.***: A dropdown menu set to "Vietnam".
- Street address.***: An empty text input field.
- Apartment, suite, unit, etc. (optional) (optional)**: An empty text input field.
- Postcode / ZIP (optional)**: An empty text input field.
- Town / City.***: An empty text input field.
- State / County (optional)**: An empty text input field.
- Phone.***: An empty text input field.
- Email address.***: An empty text input field.

At the bottom of the form, there is a section labeled "Additional information" which is currently empty.

Hình 102. Giao diện thanh toán mặc định (part 1)

Additional information

Order notes (optional)

Notes about your order, e.g. special notes for delivery.

Your order

Product	Subtotal
Hoodie Asteroid × 3	2.250.000 ₫
Subtotal	2.250.000 ₫
Total	2.250.000 ₫

Chuyển khoản ngân hàng

Thực hiện thanh toán vào ngay tài khoản ngân hàng của chúng tôi. Vui lòng sử dụng Mã đơn hàng của bạn trong phần Nội dung thanh toán. Đơn hàng sẽ được giao sau khi tiền đã chuyển.

Kiểm tra thanh toán

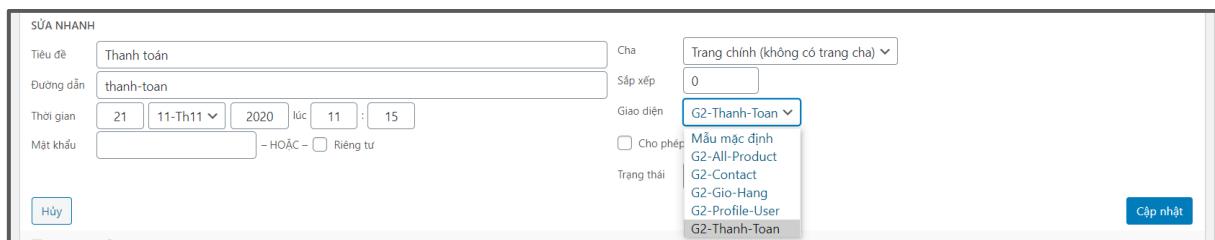
Trả tiền mặt khi nhận hàng

Thông tin cá nhân của bạn sẽ được sử dụng để xử lý đơn hàng, tăng trải nghiệm sử dụng website, và cho các mục đích cụ thể khác đã được mô tả trong [privacy policy](#).

Place order

Hình 103. Giao diện thanh toán mặc định (part 2)

Để tạo giao diện trên, ta làm tương tự như phần Giỏ hàng (mình vẫn sử dụng trang có sẵn của Woocommerce).



Hình 104. Tạo Trang Thanh Toán (sử dụng trang có sẵn của Woocommerce)

Về phần trang G2-Thanh-Toán, ta khởi tạo như giỏ hàng. Mình tạo 1 file tên g2-thanh-toan.php và đặt theme name là G2-Thanh-Toan.

Mình vẫn sử dụng shortcode để hiển thị ra giao diện của phần thanh toán.

```
wp-content > themes > group2 cms > g2-thanh-toan.php > ...
1  <?php
2  /* Template Name: G2-Thanh-Toan */
3  get_header();
4  ?>
5  <div class="cart_checkout">
6      <div class="container">
7          |     <?php echo do_shortcode(' [woocommerce_checkout] ') ?>
8      </div>
9  </div>
10 <?php
11 get_footer();
```

Hình 105. Cấu trúc file g2-thanh-toan.php

Thέ là xong phần giao diện, tiếp đến ta sẽ tới phần hướng dẫn xử lý. Ví dụ như trong form, ta để trống và nhấn button Place order thì sẽ xuất hiện thông báo điền các trường bắt buộc như sau



Hình 106. Thông báo lỗi khi để trống form trong Thanh Toán

Ví dụ, mình sẽ điền đầy đủ thông tin như bên dưới.

The screenshot shows a form with the following fields filled in:

- First name ***: Nguyen Van
- Last name ***: Son
- Company name (optional)**: Công Ty Xây Dựng A
- Country / Region ***: Vietnam
- Street address ***: Thủ Đức
- Apartment, suite, unit, etc. (optional) (optional)**: Apartment, suite, unit, etc. (optional)
- Postcode / ZIP (optional)**: 12455

Hình 107. Điền thông tin thanh toán (part 1)

The screenshot shows a form with the following fields filled in:

- Town / City ***: Hồ Chí Minh
- State / County (optional)**: (empty)
- Phone ***: 0965426000
- Email address ***: shino10112000@gmail.com
- Additional information**: (empty)
- Order notes (optional)**: Mong bạn gửi hàng sớm !!!
Cảm ơn rất nhiều.

Hình 108. Điền thông tin thanh toán (part 2)

Your order

Product	Subtotal
Hoodie Asteroid × 3	2.250.000 ₫
Subtotal	2.250.000 ₫
Total	2.250.000 ₫

Chuyển khoản ngân hàng

Thực hiện thanh toán vào ngay tài khoản ngân hàng của chúng tôi. Vui lòng sử dụng Mã đơn hàng của bạn trong phần Nội dung thanh toán. Đơn hàng sẽ được giao sau khi tiền đã chuyển.

Kiểm tra thanh toán

Trả tiền mặt khi nhận hàng

Thông tin cá nhân của bạn sẽ được sử dụng để xử lý đơn hàng, tăng trải nghiệm sử dụng website, và cho các mục đích cụ thể khác đã được mô tả trong [privacy policy](#).

Place order

Hình 109. Điền thông tin thanh toán (part 3)

Ở đây, mình chọn trả tiền khi nhận hàng thay vì chuyển khoản ngân hàng hay phương thức khác vì mình không chưa xử lý bên phía ngân hàng.

Sau đó nhấn vào button Place order và đợi nó xử lý thông tin.

Khi xử lý xong, nó sẽ gửi thông tin về admin của chờ xét duyệt. Tất nhiên, nó sẽ trả về một trang đã gửi yêu cầu thanh toán thành công như bên dưới

Cảm ơn bạn. Đơn hàng của bạn đã được nhận.

Mã đơn hàng:
Ngày:
Email:
Tổng cộng:
Phương thức thanh toán:

Trả tiền mặt khi giao hàng

Chi tiết đơn hàng

Sản phẩm	Tổng
Hoodie Asteroid x 3	2,250,000 ₫

Cảm ơn bạn đã mua sản phẩm

Tổng số phụ:	2,250,000 ₫
Phương thức thanh toán:	Trả tiền mặt khi nhận hàng
Tổng cộng:	2,250,000 ₫
Lưu ý:	Mong bạn giao hàng sớm. Cảm ơn rất nhiều!

Địa chỉ thanh toán

Son Nguyen Van
Công Ty Cổ Phần A
Thủ Đức
Hồ Chí Minh

0965426000
shino0112000@gmail.com

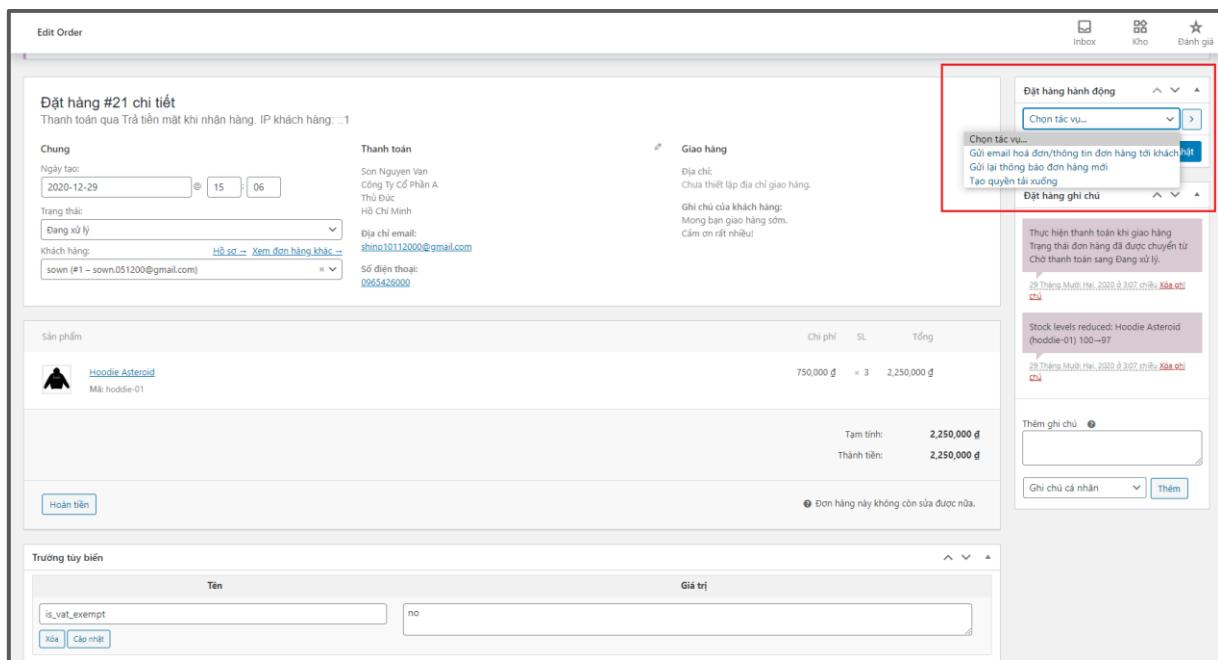
Hình 110. Gửi thông tin thanh toán thành công.

Sau đó, truy cập vào trang web wp-admin để check đơn hàng vừa đăng ký có được hay không. Nếu xuất hiện kết quả như hình 53 thì thành công.

Đơn hàng	Ngày	Tình trạng	Tổng
#21 Son Nguyen Van	1 phút trước	Đang xử lý	2,250,000 ₫
Đơn hàng	Ngày	Tình trạng	Tổng

Hình 111. Thông tin thanh toán của khách hàng (website - wp-admin)

Chỉ cần click vào đơn hàng để control theo ý muốn, ví dụ như huỷ đơn hàng, cháp thuận đơn hàng...



Hình 112. Control thông tin thanh toán khách hàng

Vậy là mình đã hướng dẫn và giới thiệu hết các chức năng cũng như công nghệ mà mình sử dụng trong đồ án.

TỔNG KẾT

1. Thuận lợi

1	Có thời gian phát triển đồ án
2	Nguồn tài liệu phong phú và được cộng đồng đồng đảo hưởng ứng
3	Có nhiều extension hỗ trợ
4	Mã nguồn mở miễn phí
7	Các thầy cô hỗ trợ nhiệt tình
8	Các thành viên đều có ý thức làm đồ án

2. Khó khăn

1	Hiệu năng đồ án thường yêu với các trang web có database lớn
2	Bảo mật kém
3	Dễ xung đột khi cài extension
4	Các extension ít được cập nhật dễ phát sinh lỗi khi update
5	Bất đồng quan điểm

3. Hướng khắc phục

1	Cập nhật liên tục các plugin
2	Bất đồng quan điểm thường sẽ biểu quyết theo số đông
3	Nhờ sự hướng dẫn của người khác
4	Xem các tài liệu trên internet và fix bug

4. Đánh giá thành viên

STT	HỌ TÊN	NỘI DUNG THỰC HIỆN	HOÀN THÀNH	CHỨC VỤ	CHỮ KÝ
1	Cao Thành Đạt	Chương I Menu, DB, CSS	100%	Thành viên	
2	Ngô Ngọc Điện	Chương II,III Category, DB, CSS	100%	Thành viên	
3	Hồ Khắc Lĩnh	Chương V Banner, Show Products	100%	Thành viên	
4	Nguyễn Văn Sơn	Chương VI Cart, Payment	100%	Nhóm Trưởng	
5	Nguyễn Văn Thái	Chương IV Detail, CSS	100%	Thành viên	

KẾT LUẬN

Trải qua quá trình học, thiết kế và thi công đồ án. Nhóm chúng em cũng đã hoàn thành nhu ý muốn. Với mong muốn hoàn thiện và phát triển thêm các chức năng mới, nhóm hy vọng các bạn đọc cũng có thể làm được và bổ sung, góp ý cho nhóm để cải thiện lại project này được đầy đủ hơn. Một lần nữa, Nhóm em xin cảm ơn Giảng viên hướng dẫn đã hỗ trợ và tạo điều kiện.

Group 2 xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

1. <https://wordpress.org/>
2. <https://thachpham.com/category/wordpress>
3. <https://codex.wordpress.org/>
4. <http://www.wpbeginner.com/>
5. <https://code.tutsplus.com/categories/wordpress/>

