TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN



PRJ – REPORT  
CHUYÊN ĐỀ CMS

Giảng Viên: Bùi Thị Phương Thảo

**Nhóm 1 học sáng thứ 3**

1.Nguyễn Phương Tân - 18211TT2680

2.Trần Triều Thuận - 18211TT0530

Ngày 22/11/2020



**MỤC LỤC**

[**I. CẤU TRÚC THEME, TẠO HEADER, FOOTER CƠ BẢN** 5](#_Toc58069175)

[**1. Hướng dẫn cài đặt cấu trúc theme** 5](#_Toc58069176)

[**2. Chỉnh sửa file function.php** 18](#_Toc58069177)

[**3. Tạo Header** 23](#_Toc58069178)

[**3.1 Cách thêm link css, js vào header** 23](#_Toc58069179)

[**3.2 Xử lý menu xuất ra trình duyệt** 25](#_Toc58069180)

[**4. Chỉnh sửa logo website** 34](#_Toc58069181)

[**5. Tạo footer cho website** 37](#_Toc58069182)

[**6. Đăng ký Widget, cách tạo widget cho footer** 39](#_Toc58069183)

[**6.1 Đăng ký một widget sidebar** 39](#_Toc58069184)

[**6.2 Tạo 4 widget tùy chỉnh cho 4 cột trong footer** 40](#_Toc58069185)

[**II. CHỈNH SỬA HEADER, FOOTER – PHẦN MỞ RỘNG** 50](#_Toc58069186)

[**1. Thêm popup login, register trong header** 50](#_Toc58069187)

[**2. Thêm class vào thẻ <ul> trong widget footer** 53](#_Toc58069188)

[**III. PLUGIN WOOCOMMERCE VÀ CART ĐƠN GIẢN** 57](#_Toc58069189)

[**1. Cài đặt plugin Woocommerce vào theme** 57](#_Toc58069190)

[**2. Tạo sản phẩm, danh mục sản phẩm** 64](#_Toc58069191)

[**IV. TẠO BLOCK EMAIL CHO POST** 79](#_Toc58069192)

**MỤC LỤC HÌNH**

[Hình 1: Tạo thư mục theme trong wordpress 5](#_Toc58021162)

[Hình 2. Truy cập vào theme vừa tạo 6](#_Toc58021163)

[Hình 3. Giao diện theme vừa được tạo 7](#_Toc58021164)

[Hình 4: Truy cập vào phpmyadmin. 9](#_Toc58021165)

[Hình 5. Login vào phpmyadmin. 10](#_Toc58021166)

[Hình 6. Nơi chứa file database 11](#_Toc58021167)

[Hình 7. Import database. 11](#_Toc58021168)

[Hình 8. Import database. 11](#_Toc58021169)

[Hình 9. Kiểm tra database đã chọn. 12](#_Toc58021170)

[Hình 10. Thực diện import. 13](#_Toc58021171)

[Hình 11. Thông báo thành công. 13](#_Toc58021172)

[Hình 12. Danh sách table có trong database. 14](#_Toc58021173)

[Hình 13. Đăng nhập Admin Wordpress. 15](#_Toc58021174)

[Hình 14. Danh Mục Appearance. 16](#_Toc58021175)

[Hình 15. Danh sách themes. 16](#_Toc58021176)

[Hình 16. Kích hoạt theme. 17](#_Toc58021177)

[Hình 17. Thông báo kích hoạt. 17](#_Toc58021178)

[Hình 18. Giao diện demo theme cms-cookie. 18](#_Toc58021179)

[Hình 19: Cấu Trúc Thu Mục Assets 24](#_Toc58021180)

[Hình 20: Gọi hàm xuất menu 26](#_Toc58021181)

[Hình 21: Biến $slug xuất ra class là main-menu 26](#_Toc58021182)

[Hình 22: Vào edit menus 27](#_Toc58021183)

[Hình 23: Giao diện trang quản lý các menu 27](#_Toc58021184)

[Hình 24: Thêm danh mục vào menu hiện có 28](#_Toc58021185)

[Hình 25: Kéo thả để tùy chỉnh các cấp cho menu 29](#_Toc58021186)

[Hình 26: Thêm tùy chọn cài đặt cho menu 29](#_Toc58021187)

[Hình 27: Một số tùy chọn thêm cho mỗi danh mục trong menu 30](#_Toc58021188)

[Hình 28: Manage locations menu 31](#_Toc58021189)

[Hình 29: Lựa chọn menu mặc định cho header 31](#_Toc58021190)

[Hình 30: Gọi hàm get\_header trong file index 33](#_Toc58021191)

[Hình 31: Kết quả thanh menu trên trình duyệt 33](#_Toc58021192)

[Hình 32: Ảnh minh họa danh mục đa cấp về làm bánh 33](#_Toc58021193)

[Hình 33: Ảnh minh họa danh mục đa cấp về các loại bánh 34](#_Toc58021194)

[Hình 34: Ảnh minh họa danh mục đa cấp về bánh tráng 34](#_Toc58021195)

[Hình 35: Chuyển sang tùy chỉnh trên giao diện quản trị 35](#_Toc58021196)

[Hình 36: Nơi để thay đổi logo website 36](#_Toc58021197)

[Hình 37: Thêm hàm the\_custom\_logo() vào file header.php 37](#_Toc58021198)

[Hình 38: Kết quả thanh header đạt được 37](#_Toc58021199)

[Hình 39: Nội dung xuất ra của file footer.php 38](#_Toc58021200)

[Hình 40: Add widgets maim sidebar và thêm các element custom 40](#_Toc58021201)

[Hình 41: Bốn widgets vừa được tạo 41](#_Toc58021202)

[Hình 42: Bốn widgets được chia bốn cột được kích hoạt bởi sidebar 48](#_Toc58021203)

[Hình 43: Copyright ở dưới cùng footer 49](#_Toc58021204)

[Hình 44: Giao diện sau khi code xong header, footer cơ bản 49](#_Toc58021205)

[Hình 45: các tùy chỉnh login và sign up 52](#_Toc58021206)

[Hình 46: Popup của sign up 52](#_Toc58021207)

[Hình 47: Popup của login 52](#_Toc58021208)

[Hình 48: Xuất bộ đệm có thêm class cho thẻ ul trong widget footer 55](#_Toc58021209)

[Hình 49: Add new plugin. 57](#_Toc58021210)

[Hình 50: Giao diện plugins. 57](#_Toc58021211)

[Hình 51: Cài đặt plugin. 58](#_Toc58021212)

[Hình 52: Quá trình cài đặt. 58](#_Toc58021213)

[Hình 53: Cài đặt thông tin cho plugin. 59](#_Toc58021214)

[Hình 54: Thông báo nhận về thông tin về sửa lỗi và phát triển. 60](#_Toc58021215)

[Hình 55: Chọn đủ để giỏ hàng. 60](#_Toc58021216)

[Hình 56: Lựa chọn phiên bản plugin. 61](#_Toc58021217)

[Hình 57: Lựa chọn nơi bán khác 62](#_Toc58021218)

[Hình 58: Chọn theme cần kích hoạt 63](#_Toc58021219)

[Hình 59: Các trang dành cho một shop cơ bản 64](#_Toc58021220)

[Hình 60: Vào đường dẫn Products -> Categories 65](#_Toc58021221)

[Hình 61: Các bước thêm một danh mục cho sản phẩm 66](#_Toc58021222)

[Hình 62: Ví dụ một số danh mục cho web của hàng bánh 67](#_Toc58021223)

[Hình 63: Tạo menu theo các danh mục sản phẩm 68](#_Toc58021224)

[Hình 64: Vào đường dẫn Products -> Add New 69](#_Toc58021225)

[Hình 65: Thêm tên và nội dung cho sản phẩm 69](#_Toc58021226)

[Hình 66: Thêm category và ảnh đại diện 70](#_Toc58021227)

[Hình 67: Thêm giá và mô tả khác cho sản phẩm 71](#_Toc58021228)

[Hình 68: Nhấn public để đăng sản phẩm 71](#_Toc58021229)

[Hình 69: Sản phảm mới thêm hiện trên danh sách các sản phẩm đã tạo 72](#_Toc58021230)

[Hình 70: Tạo file cart.php 72](#_Toc58021231)

[Hình 71: Gắn header, footer, giao diện giỏ hàng 73](#_Toc58021232)

[Hình 72: Định nghĩa hằng GO\_TO\_HOME 73](#_Toc58021233)

[Hình 73: Thêm hằng vào nút HOME 74](#_Toc58021234)

[Hình 74: Bảng hiện ra các sản phẩm đã được thêm vào giỏ hàng 75](#_Toc58021235)

[Hình 75: Thêm form nhập coupon giảm giá 76](#_Toc58021236)

[Hình 76: Khung thông báo chi tiết tổng kết giỏ hàng 77](#_Toc58021237)

[Hình 77: Kết quả giao diện trang giỏ hàng của website 78](#_Toc58021238)

[Hình 78: Cây thư mục plugins 79](#_Toc58021239)

[Hình 79: Plugins 82](#_Toc58021240)

[Hình 80: Trang plugins 82](#_Toc58021241)

[Hình 81: Posts 83](#_Toc58021242)

[Hình 82: Open block 83](#_Toc58021243)

[Hình 83: Danh sách blocks 84](#_Toc58021244)

[Hình 84: Block thông báo sai email 85](#_Toc58021245)

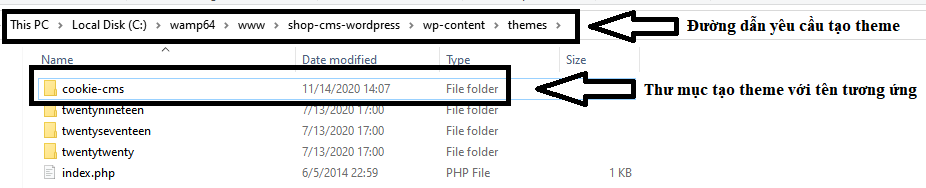
[Hình 85: Block thông báo đúng email 85](#_Toc58021246)

# **I. CẤU TRÚC THEME, TẠO HEADER, FOOTER CƠ BẢN**

## **1. Hướng dẫn cài đặt cấu trúc theme**

**Bước 1:** Di chuyển vào thư mục Wordpress tạo thư mục mang tên theme bên trong đường dẫn như sau:

**C:\wamp64\www\ shop-cms-wordpress\wp-content\themes**



Hình 1: Tạo thư mục theme trong wordpress

**Bước 2:** Tạo theme cơ bản với 2 file chính: **index.php** và **styles.css,** hai files này được tạo trong thư mục **cookie-cms**.

**Giải thích:**

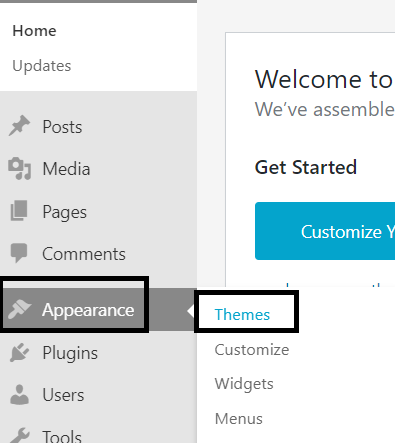
* **style.css** là file bắt buộc phải có trong wordpress cũng là nơi để viết css cho theme.
* **index.php** là trang chính của wordpress nơi mà toàn bộ html css javascript sẽ được trình duyệt render thành ứng dụng web.

Ngoài ra còn có thể định background đại diện cho theme với mục đích phân biệt với những theme khác trong tương lai thông qua 1 hình ảnh bất kì và chỉnh sửa tên ảnh theo định dạng của wordpress: **screenshot.png.**

**Bước 3:** Quay lại trang admin với đường dẫn:

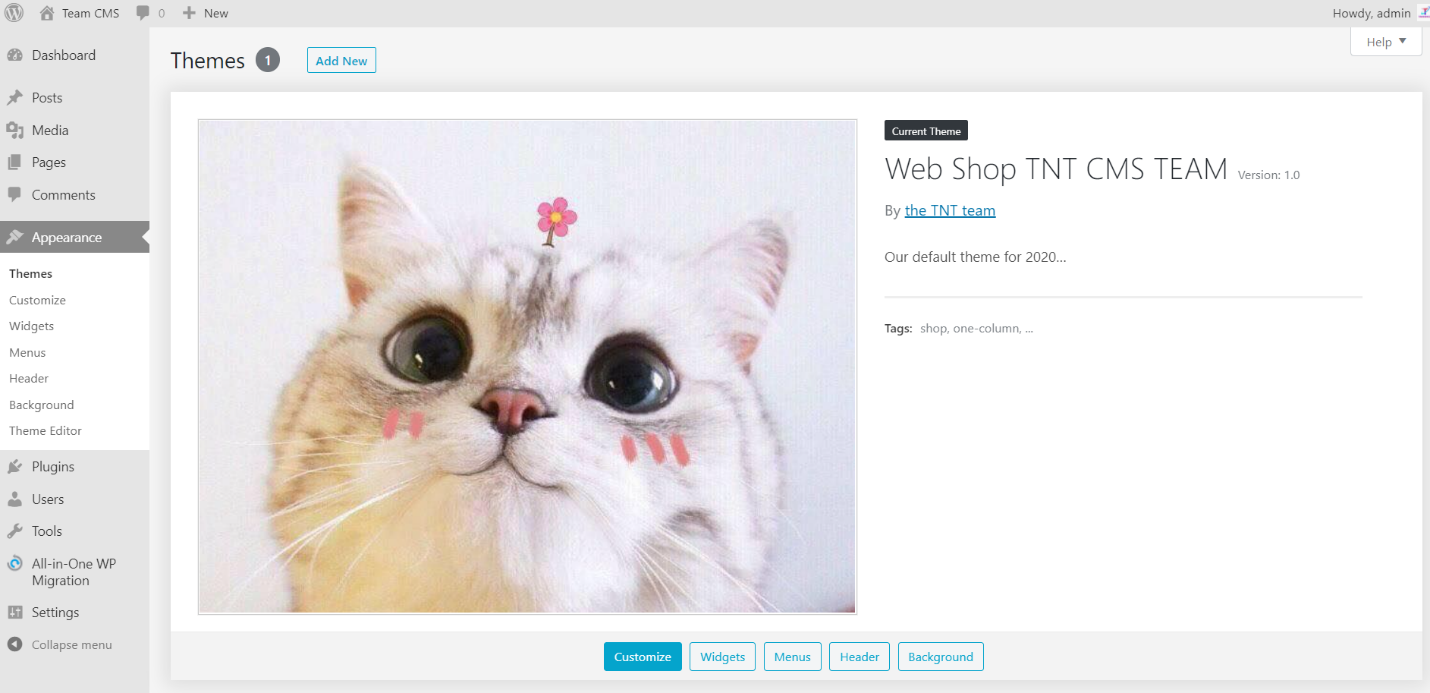
**localhost/shop-cms-wordpress/wp-admin/index.php** (Yêu cầu phải được đăng nhập với quyền admin từ trước)

chọn **Appearance** -> **theme** theo hình dưới đây:



Hình 2. Truy cập vào theme vừa tạo

Sau khi truy cập vào theme vừa tạo bạn sẽ được giao diện theme vừa tạo như hình:



Hình 3. Giao diện theme vừa được tạo

Để được kết quả hoàn chỉnh như hình trên sẽ cần thêm thông tin từ file **style.css** với nội dung sau:

/\* Theme Name: Web Shop TNT CMS TEAM

Author: the TNT team

Theme URI: https://tanhongit.com

Author URI: https://tanhongit.com/author/tanhongit/

Version: 1.0

Requires at least: 4.7

License: GNU General Public License v2 or later

License URI: http://www.gnu.org/licenses/gpl-2.0.html

Description: Our default theme for 2020...

Tags: shop, one-column, ...

Requires PHP: 7.3

License: GNU General Public License v2 or later

Text Domain: TNT Team

This theme, like WordPress, is licensed under the GPL.

Use it to make something cool, have fun, and share what you've

learned with others. \*/

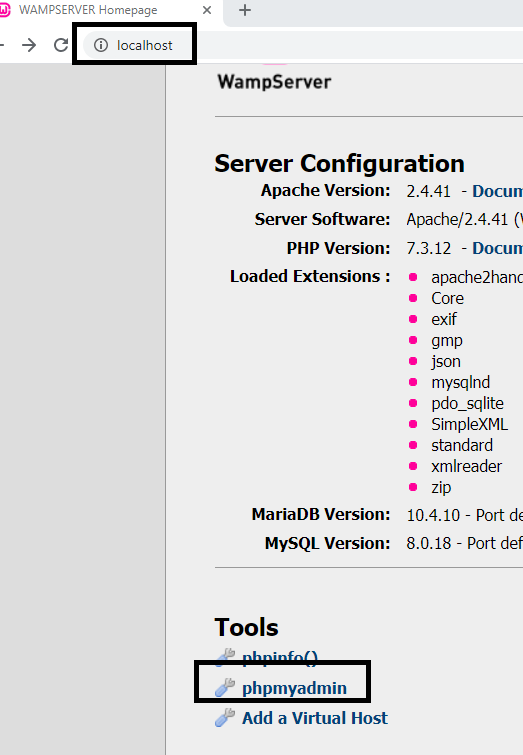
Thông tin trên sẽ được cập nhật trong file **style.css** với mục đích hiển thị tên theme, version và một vài thông tin khác của theme nếu muốn hiển thị.

**Lưu ý:** Phần nội dung trên bắt buộc phải được đặt ở đầu file **style.css**

**Link Download Project:** [**https://github.com/TanHongIT/shop-cms-wordpress**](https://github.com/TanHongIT/shop-cms-wordpress)

**Nếu như bạn không thể truy cập vào được trang web thì hãy tiến hành làm các công việc sau:**

Để có thể giúp cho theme hoạt động một cách ổ định và không bị lỗi mất trắng dữ liệu, ta cần phải truyền database có sẵn vào database bên trên **localhost** **->** **phpMyAdmin**.



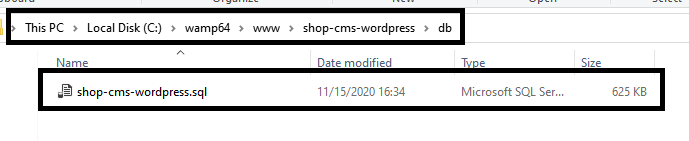
Hình 4: Truy cập vào phpmyadmin.

Từ giao diện đăng nhập phpMyAdmin cần phải có username và password. Mặc định username là **root** và **password rỗng**.



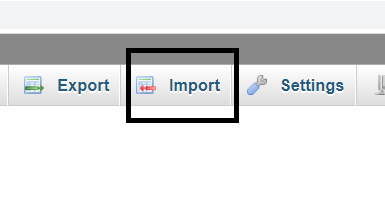
Hình 5. Login vào phpmyadmin.

Sau khi vào được phpMyAdmin tiến hành kích hoạt chức năng “**Import**” database có sẵn trong thư mục như hình dưới đây:



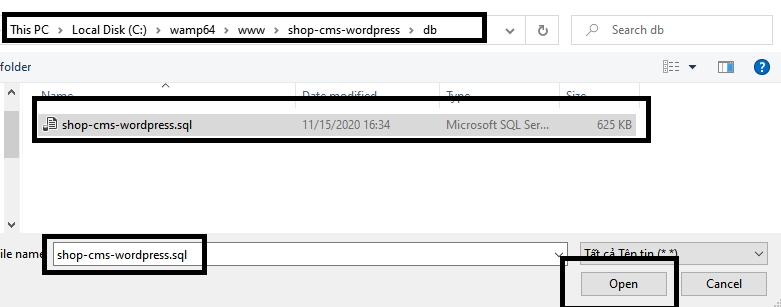
Hình 6. Nơi chứa file database

Hãy click vào nút Import trên thanh công cụ của phpmyadmin



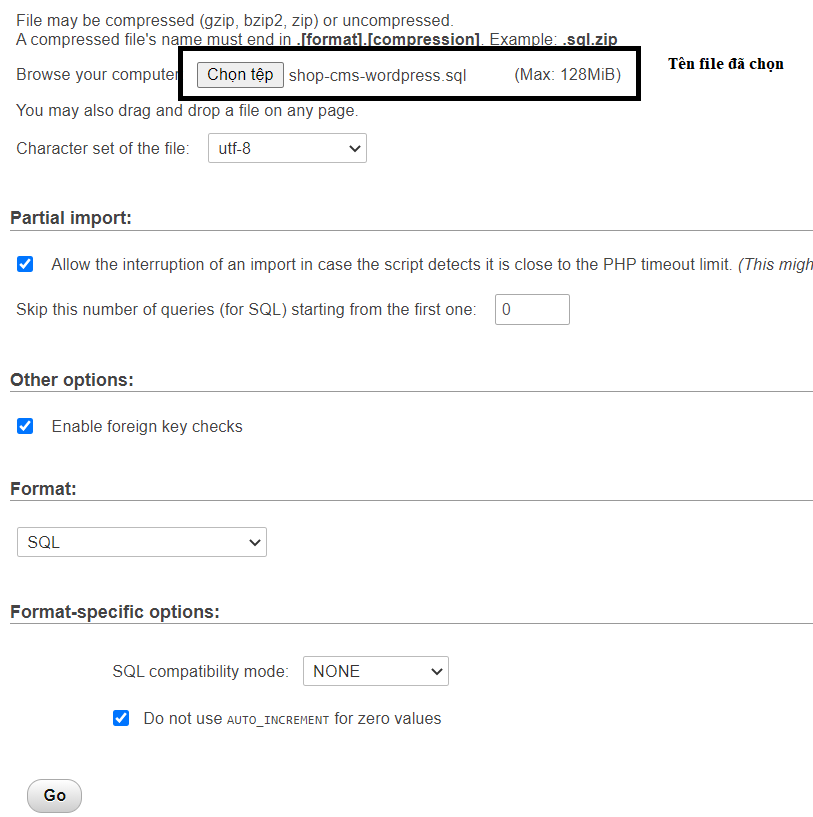
Hình 7. Import database.

Sau khi click vào nút “**Import**” một thư sẽ được hiển thị cho phép chọn file database import vào.



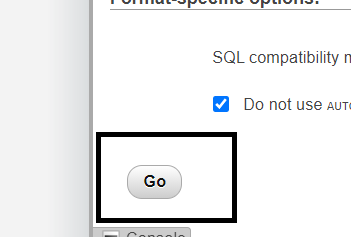
Hình 8. Import database.

Sau khi chọn đúng file database cần import tiếp tục ấn vào nút “**Open**”, phpMyAdmin sẽ trả về một giao diện như hình bên dưới:



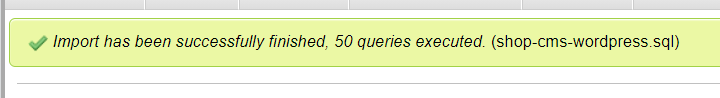
Hình 9. Kiểm tra database đã chọn.

Sau khi kiểm tra chính xác file database cần thêm tiến hành click vào nút “**Go**”



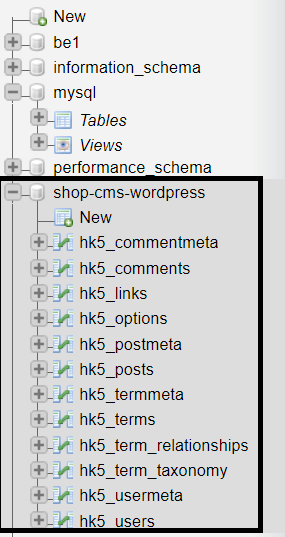
Hình 10. Thực diện import.

Sau khi thực thi việc add database vào phpmyadmin thì một thông báo được import has been successfully được hiện ra như hình dưới đây:



Hình 11. Thông báo thành công.

Tiến hành kiểm tra việc database đã được add thành công.

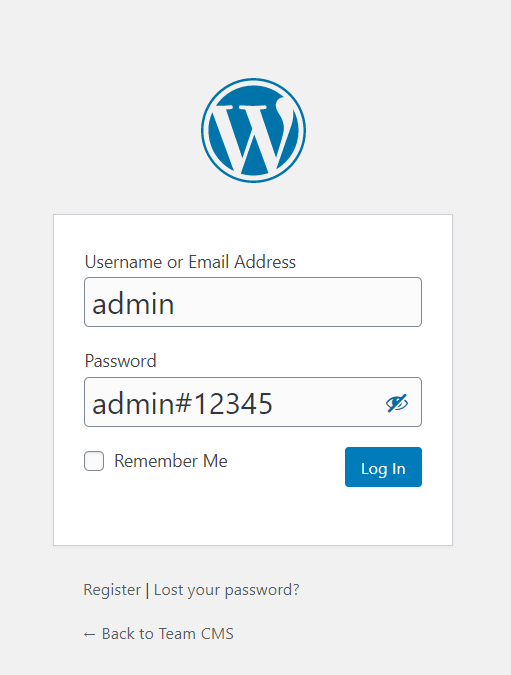


Hình 12. Danh sách table có trong database.

Sau khi đã tiến hành add database thành công việc tiếp theo là đăng nhập và active theme **cookie-cms**.

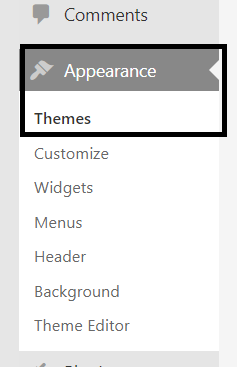
username mặc định là: **admin**

Password là: **admin#12345**

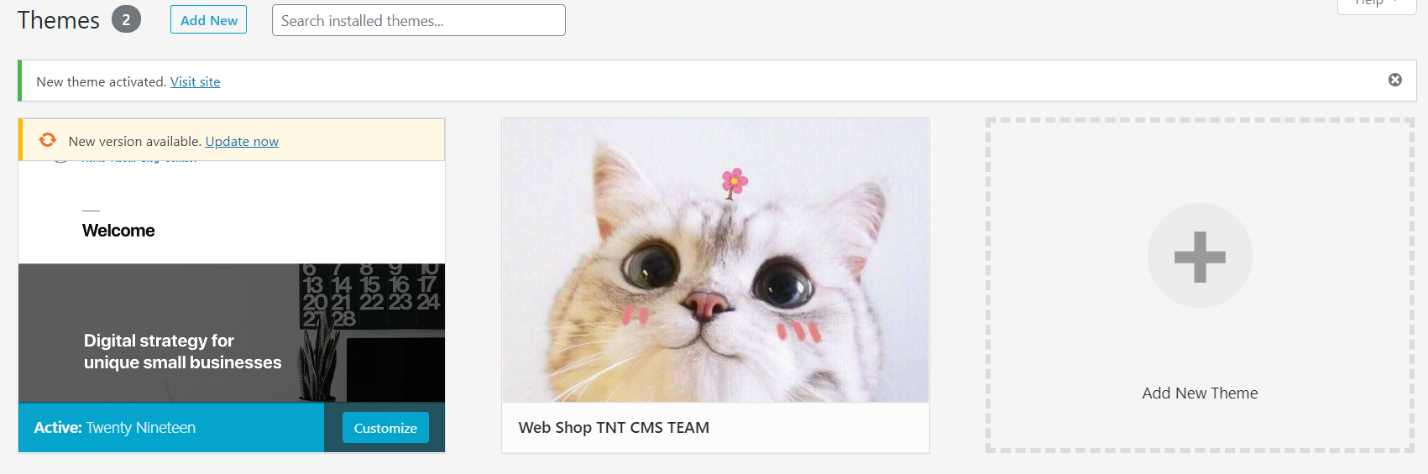


Hình 13. Đăng nhập Admin Wordpress.

Sau khi login thành công sẽ nhận được giao diện trang admin và để active được theme cms-cookie tiến hành click vào “**Appearance -> Themes**” sẽ nhận được giao diện với nhiều theme như hình:

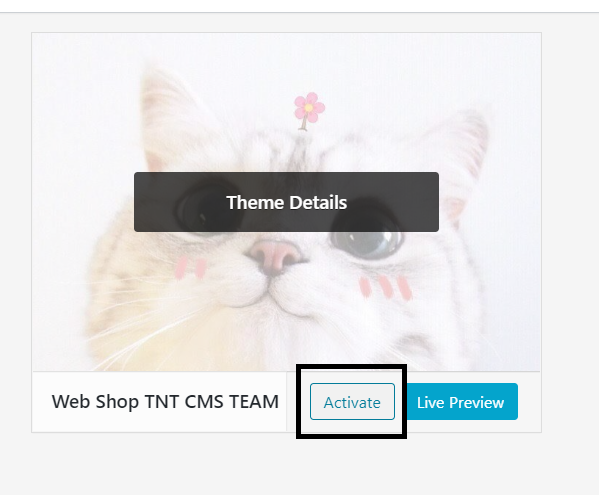


Hình 14. Danh Mục Appearance.



Hình 15. Danh sách themes.

Để kích hoạt được theme tiến hành di chuột vào theme muốn kích hoạt và chọn ”**Active**”



Hình 16. Kích hoạt theme.

Sau khi active ngay lập tức trang giao diện theme sẽ hiển thị dòng thông báo:

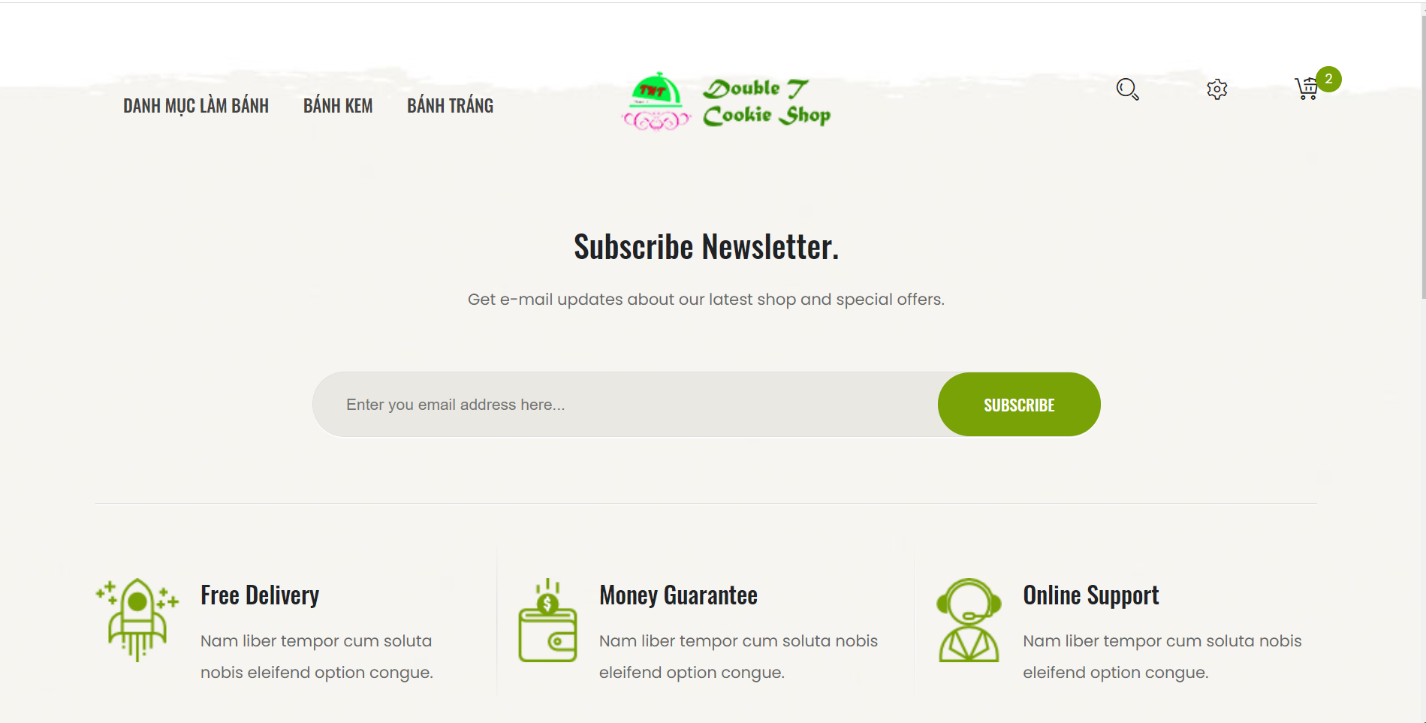


Hình 17. Thông báo kích hoạt.

Sau khi hoàn tất việc active việc cần làm là truy cập vào địa chỉ:

**localhost/shop-cms-wordpress/**

giao diện của theme **cms-cookie** sẽ được hiện ra như hình dưới đây:



Hình 18. Giao diện demo theme cms-cookie.

## **2. Chỉnh sửa file function.php**

Trước tiên chúng ta cần phải cài đặt một số hàm, hằng quan trọng cần có để có thể sử dụng lại nhiều lần sau này trong quá trình hoàn thiện theme.

Đầu tiên ta sẽ tạo hằng **THEME\_URL** chứa đường dẫn tới thư mục theme. Còn hằng **CORE** sẽ dẫn đến các file quan trọng sẽ code trong quá trình hoàn thiện theme sau này.

/\*

    @ Thiết lập các hằng dữ liệu quan trọng

    @ THEME\_URL = get\_stylesheet\_directory() - đường dẫn tới thư mục theme

    @ CORE = thư mục /core của theme, chứa các file nguồn quan trọng.

\*/

define('THEME\_URL', get\_stylesheet\_directory());

define('CORE', THEME\_URL . '/core');

Thiết lập hàm hiển thị menu:

/\*

@ Thiết lập hàm hiển thị menu

@ tnt\_menu( $slug )

\*/

if (!function\_exists('tnt\_menu')) {

    function tnt\_menu($slug)

    {

        $menu = array(

            'theme\_location' => $slug,

            'container' => 'nav',

            'container\_class' => $slug,

        );

        wp\_nav\_menu($menu);

    }

}

Hàm này sẽ giúp ta hienr thị toàn bộ thôn gtin menu ra màn hình. Nó sẽ rất cần thiết cho ta khi thực hiện làm thanh header.

Tiếp theo là hàm giúp tự động truyền các file **css, js** vào theme:

/\*

@ Chèn CSS và Javascript vào theme

@ sử dụng hook wp\_enqueue\_scripts() để hiển thị nó ra ngoài front-end

\*/

function tnt\_styles()

{

    /\*

     \* Hàm get\_stylesheet\_uri() sẽ trả về giá trị dẫn đến file style.css của theme

     \* Nếu sử dụng child theme, thì file style.css này vẫn load ra từ theme mẹ

     \*/

    wp\_register\_style('main-style', get\_template\_directory\_uri() . '/style.css', 'all');

    wp\_enqueue\_style('main-style');

}

add\_action('wp\_enqueue\_scripts', 'tnt\_styles');

Mặc định file **style.css** ở cấp thư mục cha sẽ là file style chính cho toàn trang web. Như vậy, ta sẽ tự động truyền nó vào thẻ <head> thông qua hàm trên.

**Và bây giờ ta sẽ thêm các hàm chức năng được theme hỗ trợ.**

        /\*

        \* Thêm chức năng post thumbnail

        \*/

        add\_theme\_support('post-thumbnails');

        /\*

        \* thêm chức năng title-tag để tự thêm <title>

        \*/

        add\_theme\_support('title-tag');

        /\*

        \* thêm chức năng custom logo

        \*/

        add\_theme\_support('custom-logo');

        /\*

        \* thêm chức năng custom header

        \*/

        add\_theme\_support('custom-header');

        /\*

        \* Thêm chức năng post format: tùy biến việc hiển thị post theo các định dạng

        \*/

        add\_theme\_support(

            'post-formats',

            array(

                'image',

                'video',

                'gallery',

                'quote',

                'link'

            )

        );

        /\*

        \* Tạo sidebar cho theme

        \*/

        $sidebar = array(

            'name' => \_\_('Main Sidebar', 'tnt\_team'),

            'id' => 'main-sidebar',

            'description' => 'Main sidebar for TNT TEAM theme',

            'class' => 'main-sidebar',

            'before\_title' => '<h3 class="widgettitle">',

            'after\_title' => '</h3>'

        );

        register\_sidebar($sidebar);

        /\*

        \* Thêm chức năng custom background: đổi lại màu nền hoặc thêm ảnh nền cho website

        \*/

        $default\_background = array(

            'default-color' => '#e8e8e8',

        );

        add\_theme\_support('custom-background', $default\_background);

        /\*

        \* Tạo menu cho theme

        \*/

        register\_nav\_menu('main-menu', \_\_('Main Menu', 'tnt\_team'));

Với các hàm được theme hỗ trợ sẵn được liệt kê ở trên trên thì chúng ta có thể đưa hết vào một hàm setup theme chính để tiện theo dõi và thay đổi sau này hơn.

Vậy, file **functions.php** sẽ có full code đầy đủ như sau:

<?php

/\*

    @ Thiết lập các hằng dữ liệu quan trọng

    @ THEME\_URL = get\_stylesheet\_directory() - đường dẫn tới thư mục theme

    @ CORE = thư mục /core của theme, chứa các file nguồn quan trọng.

\*/

define('THEME\_URL', get\_stylesheet\_directory());

define('CORE', THEME\_URL . '/core');

require\_once(CORE . '/init.php');

// @ Thiết lập $content\_width để khai báo kích thước chiều rộng của nội dung

if (!isset($content\_width)) {

    /\*

     \* Nếu biến $content\_width chưa có dữ liệu thì gán giá trị cho nó

     \*/

    $content\_width = 620;

}

/\*

    @ Thiết lập các chức năng sẽ được theme hỗ trợ

\*/

if (!function\_exists('tnt\_theme\_setup')) {

    /\*

     \* Nếu chưa có hàm tnt\_theme\_setup() thì sẽ tạo mới hàm đó

     \*/

    function tnt\_theme\_setup()

    {

        /\*

        \* Thiết lập theme có thể dịch được

        \*/

        $language\_folder = THEME\_URL . '/languages';

        load\_theme\_textdomain('tnt\_team', $language\_folder); // tên nhận diện các chuỗi mà chúng ta sẽ cho phép dịch trong theme

        /\*

        \* Thêm chức năng post thumbnail

        \*/

        add\_theme\_support('post-thumbnails');

        /\*

        hêm chức năng title-tag để tự thêm <title>

        \*/

        add\_theme\_support('title-tag');

        /\*

        \* Thêm chức năng post format: tùy biến việc hiển thị post theo các định dạng

        \*/

        add\_theme\_support(

            'post-formats',

            array(

                'image',

                'video',

                'gallery',

                'quote',

                'link'

            )

        );

        /\*

        \* Thêm chức năng custom background: đổi lại màu nền hoặc thêm ảnh nền cho website

        \*/

        $default\_background = array(

            'default-color' => '#e8e8e8',

        );

        add\_theme\_support('custom-background', $default\_background);

        /\*

        \* Tạo menu cho theme

        \*/

        register\_nav\_menu('primary-menu', \_\_('Main Menu', 'tnt\_team')); //textdomain dùn để dịch

          /\*

        \* thêm chức năng custom logo

        \*/

        add\_theme\_support('custom-logo');

        /\*

        \* thêm chức năng custom header

        \*/

        add\_theme\_support('custom-header');

        /\*

        \* Tạo sidebar cho theme

        \*/

        $sidebar = array(

            'name' => \_\_('Main Sidebar', 'tnt\_team'),

            'id' => 'main-sidebar',

            'description' => 'Main sidebar for TNT TEAM theme',

            'class' => 'main-sidebar',

            'before\_title' => '<h3 class="widgettitle">',

            'after\_title' => '</h3>'

        );

        register\_sidebar($sidebar);

    }

    add\_action('init', 'tnt\_theme\_setup');

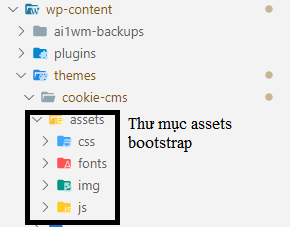
}

## **3. Tạo Header**

### **3.1 Cách thêm link css, js vào header**

Trước tiên ta sẽ tiến hành tải về **bootstrap** và **font-awesome.**

Tiếp theo ta sẽ truy cập vào thư mục **cookie-cms**, tạo thư mục **assets** và copy hết bootstrap, font vào. Ta sẽ được cấu trúc thư mục như hình dưới đây:



Hình 19: Cấu Trúc Thu Mục Assets

Tiếp theo, tại thư mục theme, ta tạo thêm một file **header.php** cùng cấp với file **index.php** và bắt đầu code.

Bây giờ bạn hãy mở file **header.php** ra, không cần viết thẻ mở PHP mà chúng ta sẽ khai báo bằng HTML có lồng PHP vào như sau:

<!DOCTYPE html>

<!--[if IE 8]> <html <?php language\_attributes(); ?> class="ie8"> <![endif]-->

<!--[if !IE]> <html <?php language\_attributes(); ?>> <![endif]-->

<head>

    <meta charset="<?php bloginfo('charset'); ?>" />

    <link rel="profile" href="http://gmgp.org/xfn/11" />

    <link rel="pingback" href="<?php bloginfo('pingback\_url'); ?>" />

    <!--hook này để giúp WordPress hiểu được đây là khu vực thẻ <head>-->

    <?php wp\_head(); ?>

</head>

<!--Thêm class tượng trưng cho mỗi trang lên <body> để tùy biến-->

<body <?php body\_class(); ?>>

Tiếp theo chúng ta cần phải truyền thêm các file css, js nằm trong thư mục assets vào header.

Ta sẽ tạo một hằng mới chứa chuỗi là đường dẫn truy xuất URI thư mục mẫu cho theme hiện tại.

Ta sẽ tạo thêm hằng '**GET\_TEMP\_URL**' trong file **functions.php**

define('GET\_TEMP\_URL', get\_template\_directory\_uri());

Sau đó, ta sẽ qua file **header.php** và thêm đường dẫn đến các file css như sau:

<head>

    <meta charset="<?php bloginfo('charset'); ?>" />

    <link rel="profile" href="http://gmgp.org/xfn/11" />

    <link rel="pingback" href="<?php bloginfo('pingback\_url'); ?>" />

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/bootstrap.min.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/font-awesome.min.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/ie7.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/plugins.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/style.css">

    <script src="<?= GET\_TEMP\_URL ?>/assets/js/vendor/modernizr-3.5.0.min.js"></script>

    <!--hook này để giúp WordPress hiểu được đây là khu vực thẻ <head>-->

    <?php wp\_head(); ?>

</head>

Như vậy là ta đã gọi được toàn bộ css, js mà ta code phục vụ cho giao diện của theme này.

Tiếp theo, phần menu là phần quan trọng và nằm trong top khó xử lý nhất đối với một website. Chúng ta sẽ tiến hành tạo một menu và cho nó hiện ra màn hình.

### **3.2 Xử lý menu xuất ra trình duyệt**

Trong file **functions.php** nhóm đã đăng ký sẵn một menu với hàm **tnt\_menu($slug)**

/\*

@ Thiết lập hàm hiển thị menu

@ tnt\_menu( $slug )

\*/

if (!function\_exists('tnt\_menu')) {

    function tnt\_menu($slug)

    {

        $menu = array(

            'theme\_location' => $slug,

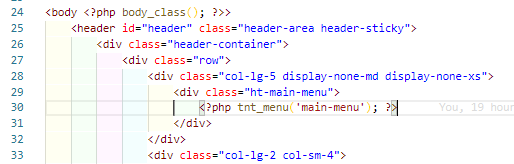
            'container' => 'nav',

            'container\_class' => $slug,

        );

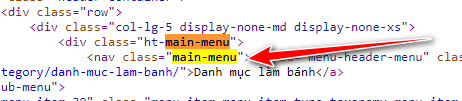
        wp\_nav\_menu($menu);

**Giải thích:** Biến **$slug** chính là tên slug của menu mà bạn cần gọi ra và nó sẽ tự áp dụng tên slug đó vào làm class cho menu hiển thị.



Hình 20: Gọi hàm xuất menu

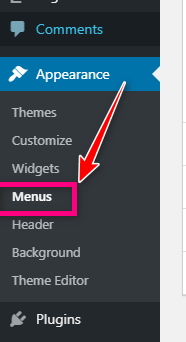
Trong dòng số 30 ở hình trên, mình gọi hàm **tnt\_menu** với slug truyền vào là **main-menu**. Như vậy, khi xuất ra trình duyệt, class được đặt cho thanh menu này sẽ có tên là **main-menu**.



Hình 21: Biến $slug xuất ra class là main-menu

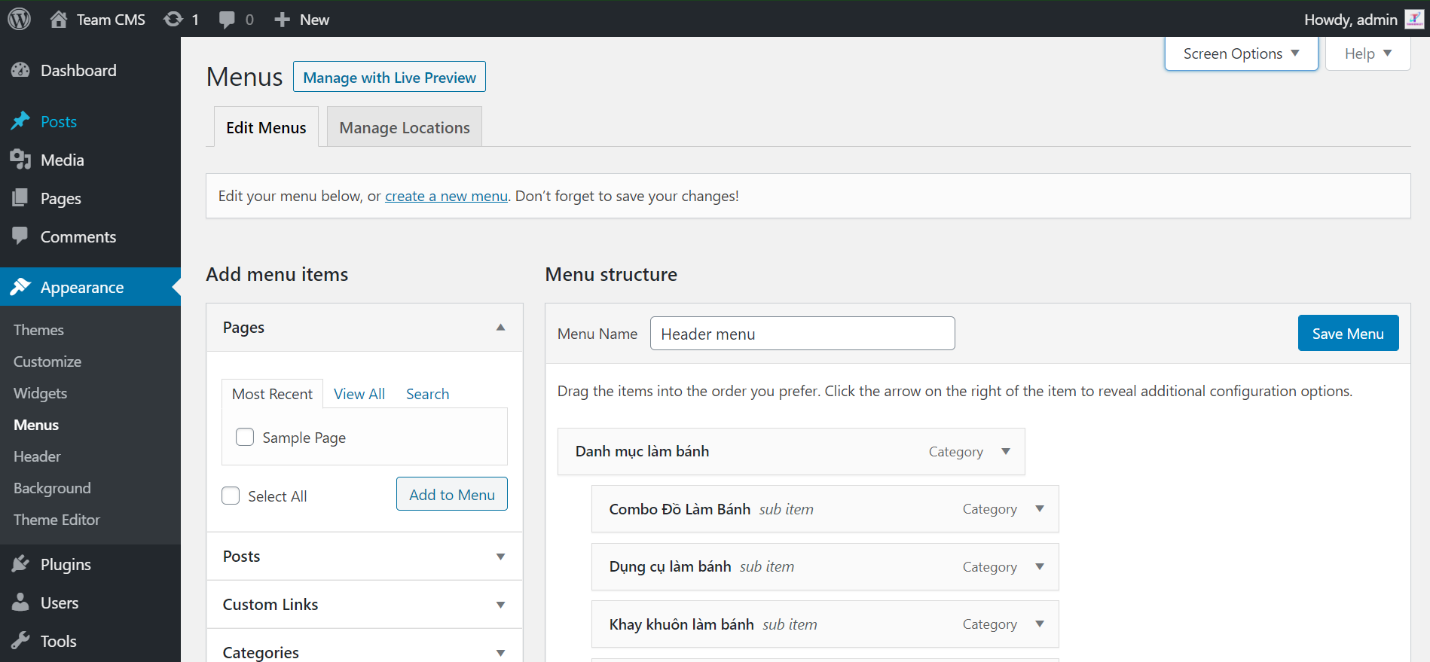
Sau khi bạn sử dụng hàm trên để đăng ký một menu, lập tức tại trang quản trị admin sẽ có phần quản lý menu cho các bạn có thể tùy ý thêm, xóa hoặc sửa, thay đổi các menu.

Bạn vào **Appearance** và chọn **Menus**.



Hình 22: Vào edit menus

Sau khi vào bạn sẽ thấy thấy menu hiển thị lên màn hình:



Hình 23: Giao diện trang quản lý các menu

Tiếp theo các bạn hãy thêm một số danh mục khác vào thanh menu thông qua bảng danh sách bên trái màn hình.

Hãy chọn những danh mục bạn muốn thêm và sau đó thì nhấn **Add to Menu**.

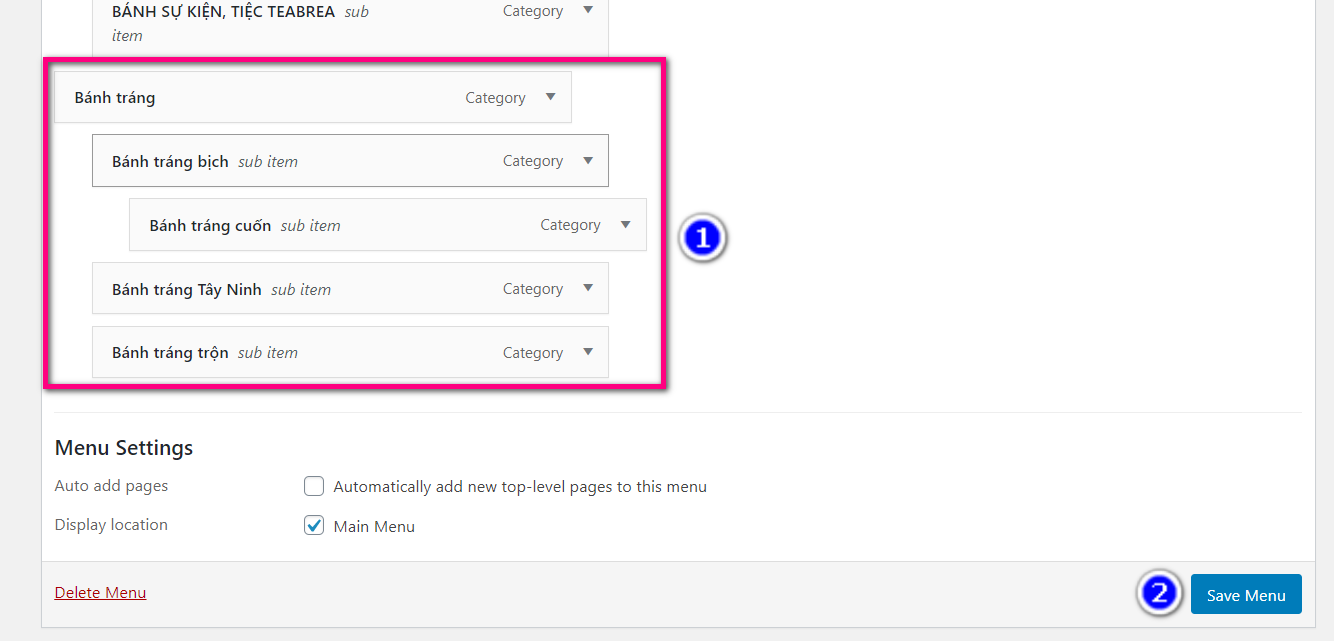


Hình 24: Thêm danh mục vào menu hiện có

Các bạn có thể tùy chinh lại cấp danh mục được hiển thị trên menu thành nhiều cấp theo ý muốn.

**Ví dụ:**

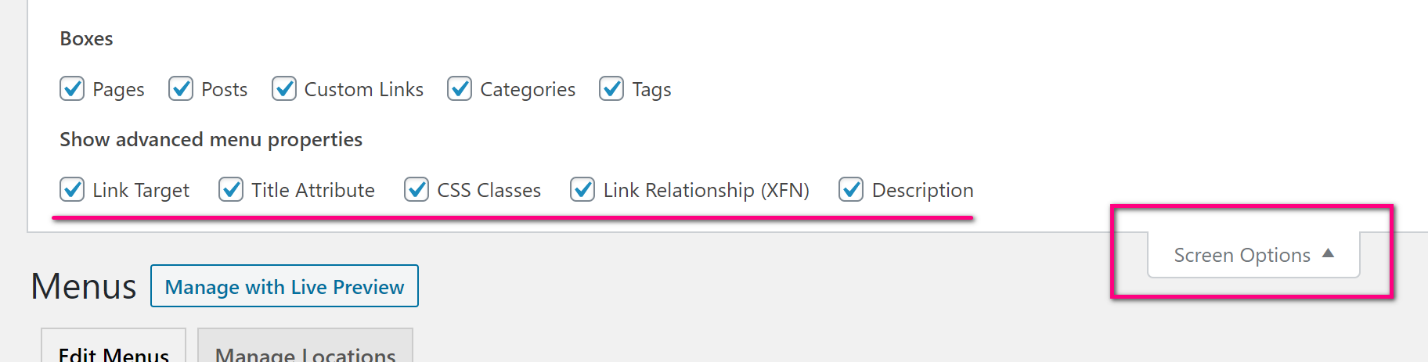
Ở hình dướng mình tùy chỉnh thành 3 cấp. Trong đó, danh mục “**Bánh tráng cuốn**” nằm ở cấp sâu nhất là cấp thứ 3.



Hình 25: Kéo thả để tùy chỉnh các cấp cho menu

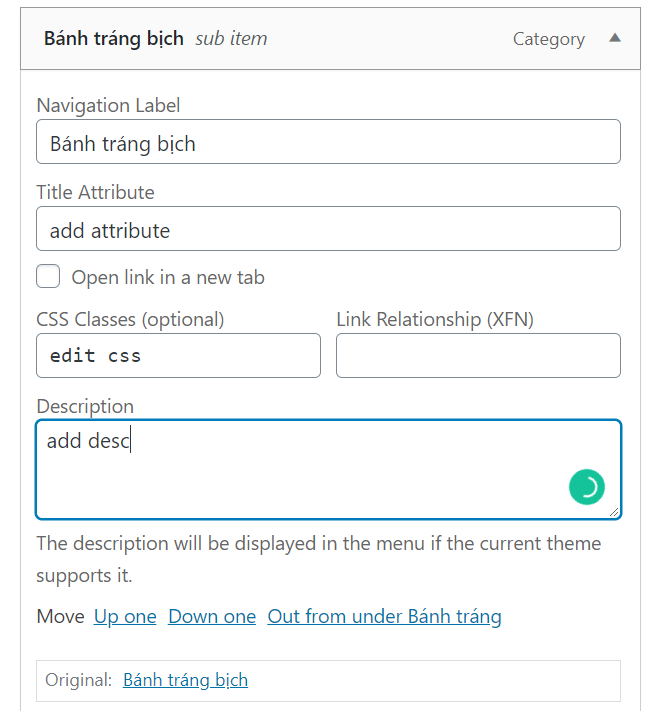
Ngoài ra bạn có thể thêm vào menu một số tùy chỉnh khác như Tag, Post, Page,…Cuối cùng bạn phải nhấn **Save Menu** để lưu lại những thay đổi này nhé!

Để định dạng lại css hoặc description,…cho từng danh mục hoặc link trong menu thì bạn hãy kéo lên đầu trang, nhấn chọn chuột vào Screen Options và click vào những tùy chọn bạn muốn.



Hình 26: Thêm tùy chọn cài đặt cho menu

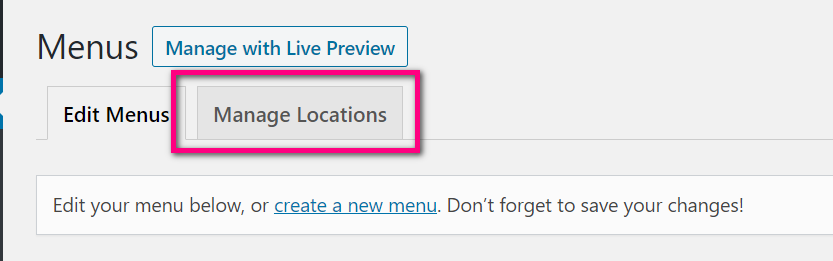
Kết quả bạn sẽ có thêm một số ô nhập liệu khác trong từng link, danh mục của menu:



Hình 27: Một số tùy chọn thêm cho mỗi danh mục trong menu

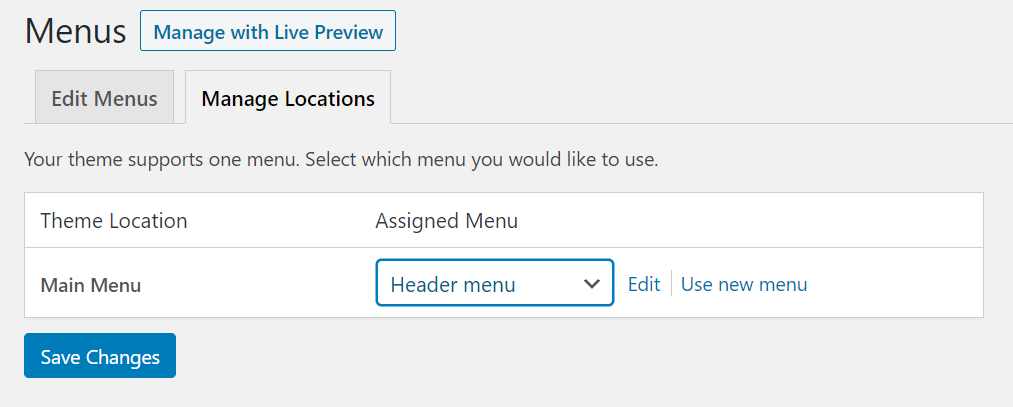
Sau khi mọi hành động đã thao tác xong thì bạn nhớ nhấn nhấn **Save Menu** để lưu lại những thay đổi này nhé!

Tiếp theo, kéo lên đầu trang và chọn vào **Manage Locations** để cài đặt thanh menu nào sẽ được hiển thị trên **header**.



Hình 28: Manage locations menu

Sau khi đã chọn được menu ưng ý để cho phép hiện trên header rồi thì nhấn **Save Changes**.



Hình 29: Lựa chọn menu mặc định cho header

Bây giờ để xuất menu ra được, ta sẽ code lại file **header.php** như sau:

<!DOCTYPE html>

<!--[if IE 8]> <html <?php language\_attributes(); ?> class="ie8"> <![endif]-->

<!--[if !IE]> <html <?php language\_attributes(); ?>> <![endif]-->

<head>

    <meta charset="<?php bloginfo('charset'); ?>" />

    <link rel="profile" href="http://gmgp.org/xfn/11" />

    <link rel="pingback" href="<?php bloginfo('pingback\_url'); ?>" />

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/bootstrap.min.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/font-awesome.min.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/ie7.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/plugins.css">

    <link rel="stylesheet" href="<?= GET\_TEMP\_URL ?>/assets/css/style.css">

    <script src="<?= GET\_TEMP\_URL ?>/assets/js/vendor/modernizr-3.5.0.min.js"></script>

    <!--hook này để giúp WordPress hiểu được đây là khu vực thẻ <head>-->

    <?php wp\_head(); ?>

</head>

<!--Thêm class tượng trưng cho mỗi trang lên <body> để tùy biến-->

<body <?php body\_class(); ?>>

    <header id="header" class="header-area header-sticky">

        <div class="header-container">

            <div class="row">

                <div class="col-lg-5 display-none-md display-none-xs">

                    <div class="ht-main-menu">

                        <?php tnt\_menu('main-menu'); ?>

                    </div>

                </div>

                <div class="col-lg-2 col-sm-4">

                    <div class="logo text-center">

                        <?php the\_custom\_logo(); ?>

                    </div>

                </div>

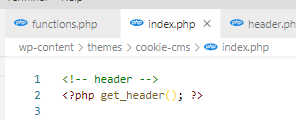
            </div>

        </div>

    </header>

Ở đoạn lệnh trên, ta sử dụng **tnt\_menu('main-menu')** để xuất menu đã tạo khi nãy, ngoài ra ta còn dùng thêm **the\_custom\_logo()** nhằm hiển thị logo trên thanh header. (Cả 2 hàm này đã được xây dựng trong file **functions.php**).

Cuối cùng, ta phải gọi hàm **get\_header()** trong file index.php để có thể sổ giao diện ra trình duyệt:

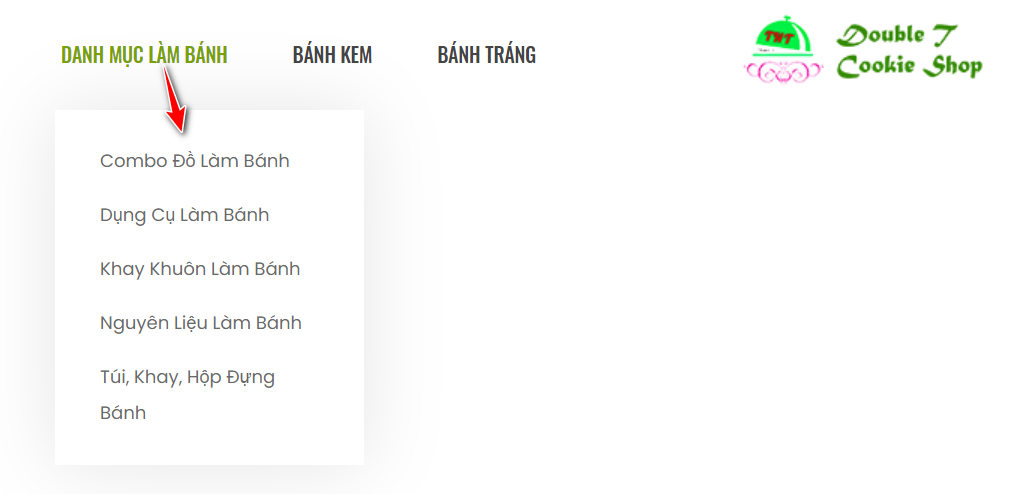


Hình 30: Gọi hàm get\_header trong file index

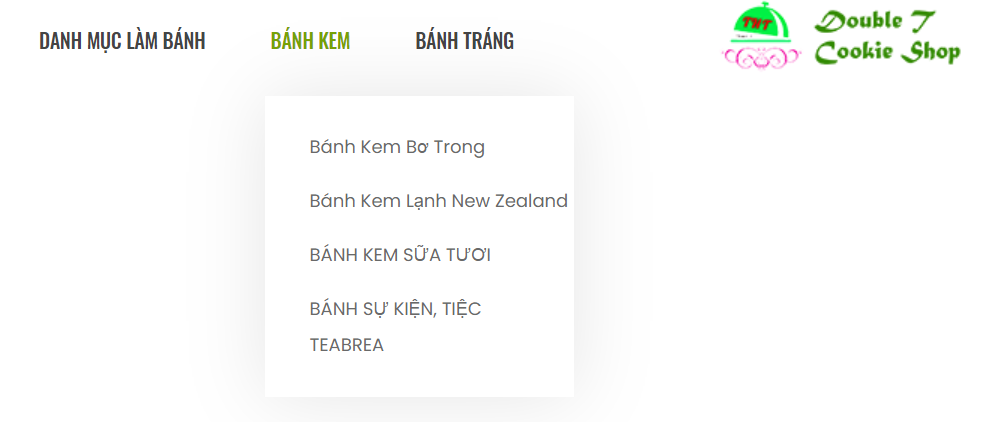
Kết quả xuất ra màn hình sau khi link thêm css vào theme:



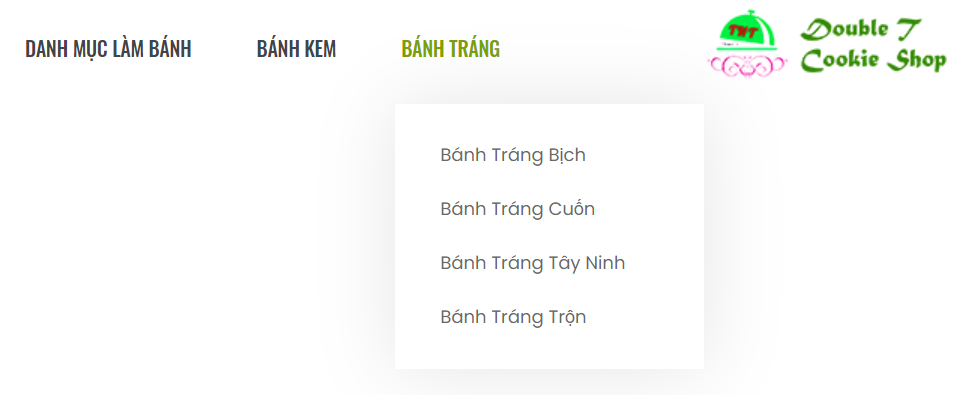
Hình 31: Kết quả thanh menu trên trình duyệt



Hình 32: Ảnh minh họa danh mục đa cấp về làm bánh



Hình 33: Ảnh minh họa danh mục đa cấp về các loại bánh



Hình 34: Ảnh minh họa danh mục đa cấp về bánh tráng

## **4. Chỉnh sửa logo website**

Để chỉnh sửa logo trong website, ta có thể chỉnh trực tiếp từ code, nhưng trên thực tế người dùng sử dụng giao diện quản trị admin và thực hiện custom trực tiếp lại chiếm đa số, cho nên ta cần phải hiện phần thay dổi logo này có sẵn trên giao diện quản trị để họ có thể trực tiếp thay đổi được mà không cần phải đụng đến dòng code nào.

Để có thể hiển thị phần chỉnh sửa logo trên customize giao diện quản trị, trong file **functions.php** ta cần phải có cấu hình sau:

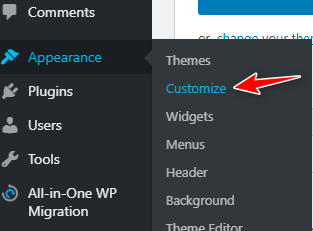
/\*

        \* thêm chức năng custom logo

        \*/

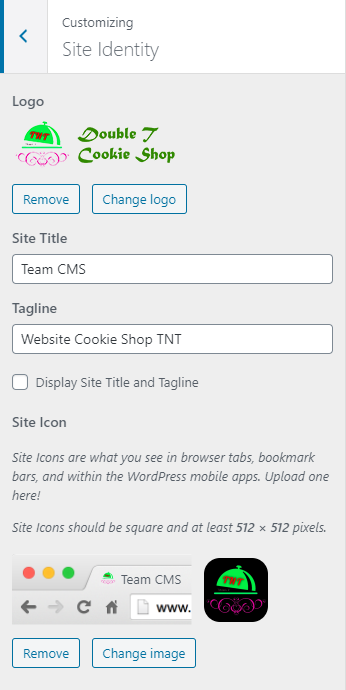
        add\_theme\_support('custom-logo');

Sau khi thêm xong, hãy vào trang quản trị admin, chọn **Customize** nằm bên trong **Appearance**.



Hình 35: Chuyển sang tùy chỉnh trên giao diện quản trị

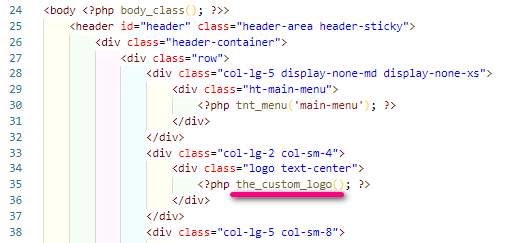
Hãy vào **Site Identily** và đổi lại logo theo ý muốn của bạn. Sau khi dổi thì nhớ nhấn vào nút **Public**.



Hình 36: Nơi để thay đổi logo website

Để hiển thị được logo trên thanh header, ta cần phải thêm hàm gọi ra logo trong file **header.php** như sau:

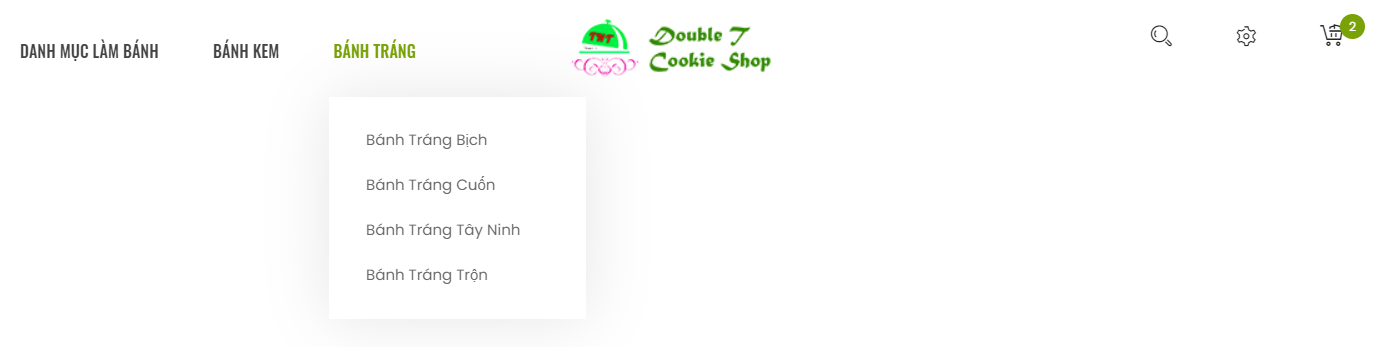
**the\_custom\_logo();**



Hình 37: Thêm hàm the\_custom\_logo() vào file header.php

**Kết luận:**

Như vậy sau hai phần tạo header và thêm logo, chỉnh logo cho website thì thanh header nhóm đã làm được có giao diện như hình sau:



Hình 38: Kết quả thanh header đạt được

## **5. Tạo footer cho website**

Sau khi tạo header xong rồi thì bây giờ ta sẽ đi đến phần tạo footer cho một trang web.

Trước tiên ta cần tạo một file tên **footer.php** nằm cùng cấp với file **index.php**. Sau đó, ở file **index.php**, ta thêm code vào như sau:

<!-- header -->

<?php get\_header(); ?>

<!-- body -->

<!-- footer -->

<?php get\_footer(); ?>

Điều kiện là hàm **get\_footer** phải luôn nằm dưới **get\_header** và các hàm content khác của trang.

Sau đó ta sẽ vào file **footer.php** và code như sau:

<!-- Footer Area Start -->

<footer class="footer-area">

<h1>Day la footer</h1>

</footer>

<!-- Footer Area End -->

<!-- All js here -->

<script src="<?= GET\_TEMP\_URL ?>/assets/js/vendor/jquery-3.2.1.min.js"></script>

<script src="<?= GET\_TEMP\_URL ?>/assets/js/popper.min.js"></script>

<script src="<?= GET\_TEMP\_URL ?>/assets/js/bootstrap.min.js"></script>

<script src="<?= GET\_TEMP\_URL ?>/assets/js/plugins.js"></script>

<script src="<?= GET\_TEMP\_URL ?>/assets/js/ajax-mail.js"></script>

<script src="<?= GET\_TEMP\_URL ?>/assets/js/main.js"></script>

</body>

</html>

Ở file này ta sẽ không mở thẻ html mà chỉ có đóng thẻ, bởi vì thẻ mở đã được gọi ở file **header.php**. Và cũng giống như header.php, ta sẽ gọi một số file js cần sử dụng trong quá trình phát triển theme thông qua hằng số **GET\_TEMP\_URL**

Kết quả của đoạn code trên sau khi hiện ra màn hình.



Hình 39: Nội dung xuất ra của file footer.php

Với một footer chưa cần chuyên nghiệp nhưng ta vẫn cần phải có đầy đủ các yếu tố như: linh hoạt chia nhỏ nhiều thành phần và có thể quản lý, thay đổi đa dạng thông qua trang quản trị admin.

Để làm được điều đó ta cần phải tạo ra các widget quản lý cho footer.

Đối với footer của website mà nhóm làm, mục tiêu sẽ **chia ra thành 4 cột khác nhau** và có những nội dung tùy chỉnh khác nhau.

Như vậy, từ yêu cầu trên ta sẽ cần **phải** **có 4 widget** cho **footer**. Làm sao để đăng ký một widget mới cho footer? Chúng ta sẽ đi vào phần tiếp theo.

## **6. Đăng ký Widget, cách tạo widget cho footer**

### **6.1 Đăng ký một widget sidebar**

Mở tệp **functions.php** từ trình Editor và tìm kiếm dòng lệnh sau: **register\_sidebar**

Nó sẽ đưa bạn đến khu vực mà tất cả các **widget sidebar** được đăng ký trong **theme** của bạn.

Nếu không tìm được chứng tỏ chúng ta chưa đăng ký bất kỳ một widget nào. Giờ chúng ta sẽ thực hiện điều đó.

Ở bên trong hàm **tnt\_theme\_setup(),** chúng ta sẽ tiến hành đăng ký một widget mới.

/\*

        \* Tạo sidebar cho theme

        \*/

        $sidebar = array(

            'name' => \_\_('Main Sidebar', 'tnt\_team'),

            'id' => 'main-sidebar',

            'description' => 'Main sidebar for TNT TEAM theme',

            'class' => 'main-sidebar',

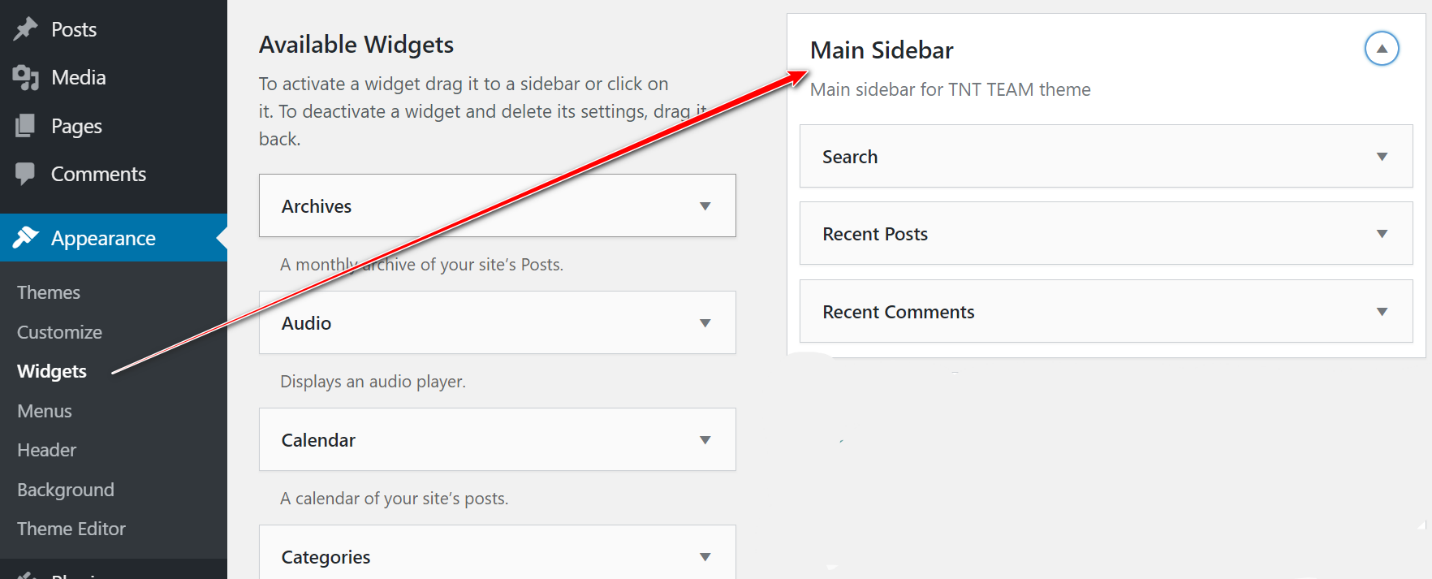
            'before\_title' => '<h3 class="widgettitle">',

            'after\_title' => '</h3>'

        );

        register\_sidebar($sidebar);

Sau khi hàm **register\_sidebar()** được thực thi, bạn sẽ thấy một Widget có tên là **Main Sidebar** nằm trong **Appearance -> Widgets**



Hình 40: Add widgets maim sidebar và thêm các element custom

Các bạn có thể tùy biến, thêm bất kỳ element widget nào ở phía bên trái kéo thả vào để thêm cho widget vừa tạo.

Với widget **Main Sidebar** này mình sẽ dùng nó để làm thanh sidebar cho trang chủ.

Còn hiện tại, chúng ta đang cần sử dụng 4 widget dành cho footer mà phần trước đã nêu ra. Như vậy, ta sẽ đi qua phần tiếp theo để tiến hành tạo 4 widget cho footer.

### **6.2 Tạo 4 widget tùy chỉnh cho 4 cột trong footer**

Chúng ta sẽ tiến hành tạo thêm 4 widget nữa trong file **functions.php**

register\_sidebar(array(

            'name' => 'Footer Sidebar 1',

            'id' => 'footer-sidebar-1',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 2',

            'id' => 'footer-sidebar-2',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 3',

            'id' => 'footer-sidebar-3',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 4',

            'id' => 'footer-sidebar-4',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

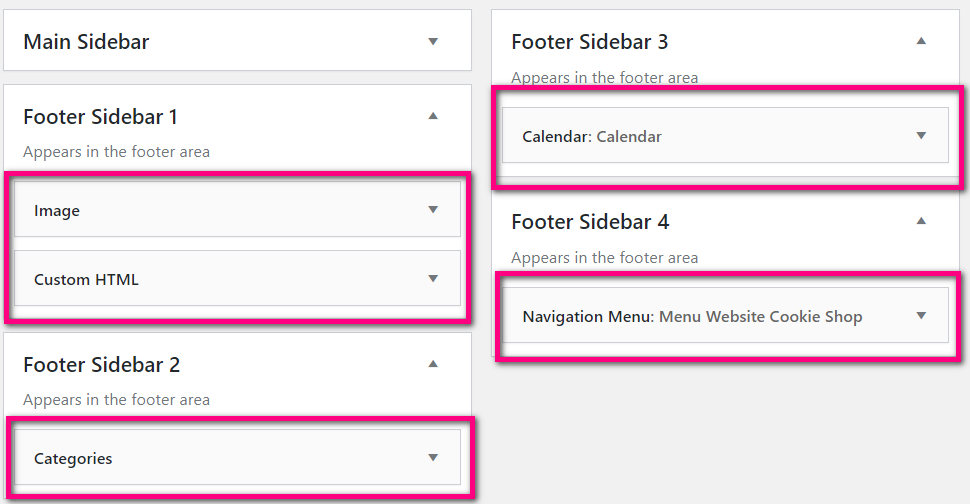
            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

Với đoạn code trên, ta đã tạo ra thêm 4 widget nữa, bạn có thể vào **Appearance -> Widgets** để xem chi tiết và thêm các element widget vào cho đầy đủ thêm. Ảnh minh họa phía dưới:



Hình 41: Bốn widgets vừa được tạo

Full code trong file **functions.php** khi này:

<?php

/\*

    @ Thiết lập các hằng dữ liệu quan trọng

    @ THEME\_URL = get\_stylesheet\_directory() - đường dẫn tới thư mục theme

    @ CORE = thư mục /core của theme, chứa các file nguồn quan trọng.

\*/

define('THEME\_URL', get\_stylesheet\_directory());

define('CORE', THEME\_URL . '/core');

define('GET\_TEMP\_URL', get\_template\_directory\_uri());

require\_once(CORE . '/init.php');

// @ Thiết lập $content\_width để khai báo kích thước chiều rộng của nội dung

if (!isset($content\_width)) {

    /\*

     \* Nếu biến $content\_width chưa có dữ liệu thì gán giá trị cho nó

     \*/

    $content\_width = 620;

}

/\*

    @ Thiết lập các chức năng sẽ được theme hỗ trợ

\*/

if (!function\_exists('tnt\_theme\_setup')) {

    /\*

     \* Nếu chưa có hàm tnt\_theme\_setup() thì sẽ tạo mới hàm đó

     \*/

    function tnt\_theme\_setup()

    {

        /\*

        \* Thiết lập theme có thể dịch được

        \*/

        $language\_folder = THEME\_URL . '/languages';

        load\_theme\_textdomain('tnt\_team', $language\_folder); // tên nhận diện các chuỗi mà chúng ta sẽ cho phép dịch trong theme

        /\*

        \* Thêm chức năng post thumbnail

        \*/

        add\_theme\_support('post-thumbnails');

        /\*

        \* thêm chức năng title-tag để tự thêm <title>

        \*/

        add\_theme\_support('title-tag');

        /\*

        \* thêm chức năng custom logo

        \*/

        add\_theme\_support('custom-logo');

        /\*

        \* thêm chức năng custom header

        \*/

        add\_theme\_support('custom-header');

        /\*

        \* Thêm chức năng post format: tùy biến việc hiển thị post theo các định dạng

        \*/

        add\_theme\_support(

            'post-formats',

            array(

                'image',

                'video',

                'gallery',

                'quote',

                'link'

            )

        );

        /\*

        \* Thêm chức năng custom background: đổi lại màu nền hoặc thêm ảnh nền cho website

        \*/

        $default\_background = array(

            'default-color' => '#e8e8e8',

        );

        add\_theme\_support('custom-background', $default\_background);

        /\*

        \* Tạo menu cho theme

        \*/

        register\_nav\_menu('main-menu', \_\_('Main Menu', 'tnt\_team')); //textdomain dùn để dịch

        /\*

        \* Tạo sidebar cho theme

        \*/

        $sidebar = array(

            'name' => \_\_('Main Sidebar', 'tnt\_team'),

            'id' => 'main-sidebar',

            'description' => 'Main sidebar for TNT TEAM theme',

            'class' => 'main-sidebar',

            'before\_title' => '<h3 class="widgettitle">',

            'after\_title' => '</h3>'

        );

        register\_sidebar($sidebar);

        register\_sidebar(array(

            'name' => 'Footer Sidebar 1',

            'id' => 'footer-sidebar-1',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 2',

            'id' => 'footer-sidebar-2',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 3',

            'id' => 'footer-sidebar-3',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

        register\_sidebar(array(

            'name' => 'Footer Sidebar 4',

            'id' => 'footer-sidebar-4',

            'description' => 'Appears in the footer area',

            'before\_widget' => '<aside id="%1$s" class="widget %2$s">',

            'after\_widget' => '</aside>',

            'before\_title' => '<h3 class="widget-title">',

            'after\_title' => '</h3>',

        ));

    }

    add\_action('init', 'tnt\_theme\_setup');

}

// --------------------------------------------------------------------------------------------------

//  Header

// ==================================

/\*

@ Thiết lập hàm hiển thị logo

@ tnt\_logo()

\*/

if (!function\_exists('tnt\_logo')) {

    function tnt\_logo()

    { ?>

        <div class="logo text-center">

            <?php if (is\_home()) {

                printf(

                    '<h1><a href="%1$s" title="%2$s">%3$s</a></h1>',

                    get\_bloginfo('url'),

                    get\_bloginfo('description'),

                    get\_bloginfo('sitename')

                );

            } else {

                printf(

                    '<p><a href="%1$s" title="%2$s">%3$s</a></p>',

                    get\_bloginfo('url'),

                    get\_bloginfo('description'),

                    get\_bloginfo('sitename')

                );

            } // endif

            ?>

            <p class="site-description"><?php bloginfo('description'); ?></p>

        </div>

<?php }

}

// %1$s: get\_bloginfo( ‘url’ )

// %2$s: get\_bloginfo( ‘description’ )

// %3$s: get\_bloginfo( ‘sitename’ )

function themename\_custom\_logo\_setup()

{

    $defaults = array(

        'height'      => 100,

        'width'       => 400,

        'flex-height' => true,

        'flex-width'  => true,

        'header-text' => array('site-title', 'site-description'),

        'unlink-homepage-logo' => true,

    );

    add\_theme\_support('custom-logo', $defaults);

}

if (!function\_exists('the\_custom\_logo')) {

    the\_custom\_logo();

}

/\*

@ Thiết lập hàm hiển thị menu

@ tnt\_menu( $slug )

\*/

if (!function\_exists('tnt\_menu')) {

    function tnt\_menu($slug)

    {

        $menu = array(

            'theme\_location' => $slug,

            'container' => 'nav',

            'container\_class' => $slug,

        );

        wp\_nav\_menu($menu);

    }

}

/\*

@ Chèn CSS và Javascript vào theme

@ sử dụng hook wp\_enqueue\_scripts() để hiển thị nó ra ngoài front-end

\*/

function tnt\_styles()

{

    /\*

     \* Hàm get\_stylesheet\_uri() sẽ trả về giá trị dẫn đến file style.css của theme

     \* Nếu sử dụng child theme, thì file style.css này vẫn load ra từ theme mẹ

     \*/

    wp\_register\_style('main-style', get\_template\_directory\_uri() . '/style.css', 'all');

    wp\_enqueue\_style('main-style');

}

add\_action('wp\_enqueue\_scripts', 'tnt\_styles');

Như vậy chúng ta đã có 4 widget dành cho 4 cột trong footer. Nhưng làm thế nào đẻ có thể xuất các widget trên ra màn hình. Chúng ta sẽ vào file **footer.php** để code:

  <!-- Footer Widget Area Start -->

        <div class="footer-widget-area">

            <div class="container">

                <div class="row">

                    <div class="col-lg-4 col-md-6">

                        <div class="single-footer-widget">

                            <?php

                            if (is\_active\_sidebar('footer-sidebar-1')) {

                                dynamic\_sidebar('footer-sidebar-1');

                            }

                            ?>

                        </div>

                    </div>

                    <div class="col-lg-2 col-md-3">

                        <div class="single-footer-widget">

                            <?php

                            if (is\_active\_sidebar('footer-sidebar-2')) {

                                dynamic\_sidebar('footer-sidebar-2');

                            }

                            ?>

                        </div>

                    </div>

                    <div class="col-lg-2 col-md-3">

                        <div class="single-footer-widget">

                            <?php

                            if (is\_active\_sidebar('footer-sidebar-3')) {

                                dynamic\_sidebar('footer-sidebar-3');

                            }

                            ?>

                        </div>

                    </div>

                    <div class="col-lg-4">

                        <div class="single-footer-widget">

                            <?php

                            if (is\_active\_sidebar('footer-sidebar-4')) {

                                dynamic\_sidebar('footer-sidebar-4');

                            } ?>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <!-- Footer Widget Area End -->

Ở đoạn code trên mình đã gọi ra các widget thông qua hàm **dynamic\_sidebar** và trước khi gọi chúng ta mình sẽ kiểm tra xem chúng có hoạt động hay không thông qua hàm **is\_active\_sidebar.**

Trong đó, tham số bên trong chính là ID của widget sidebar mà bạn đã tạo ở bên file **functions.php**

Như vậy, kết quả xuất trên trình duyệt ta sẽ nhận được như sau:



Hình 42: Bốn widgets được chia bốn cột được kích hoạt bởi sidebar

Như vậy chúng ta đã tạo thành công một footer với 4 cột có thể tủy chỉnh đa dạng tạo trải nghiệm tốt cho người dùng hơn.

Tiếp theo ta cần tạo một đoạn bản quyền copyright cho **footer.php**

<!-- Footer Bottom Area Start -->

    <div class="footer-bottom-area pt-15 pb-30">

        <div class="container">

            <div class="row">

                <div class="col-lg-6 d-flex col-md-6">

                    <div class="footer-text-bottom">

                        <p>Copyright &copy; <a href="#">Team TNT</a>. All Rights Reserved</p>

                    </div>

                </div>

                <div class="col-lg-6 col-md-6">

                    <div class="payment-img d-flex justify-content-end">

                        <img src="<?= GET\_TEMP\_URL ?>/assets/img/payment.png" alt="">

                    </div>

                </div>

            </div>

        </div>

    </div>

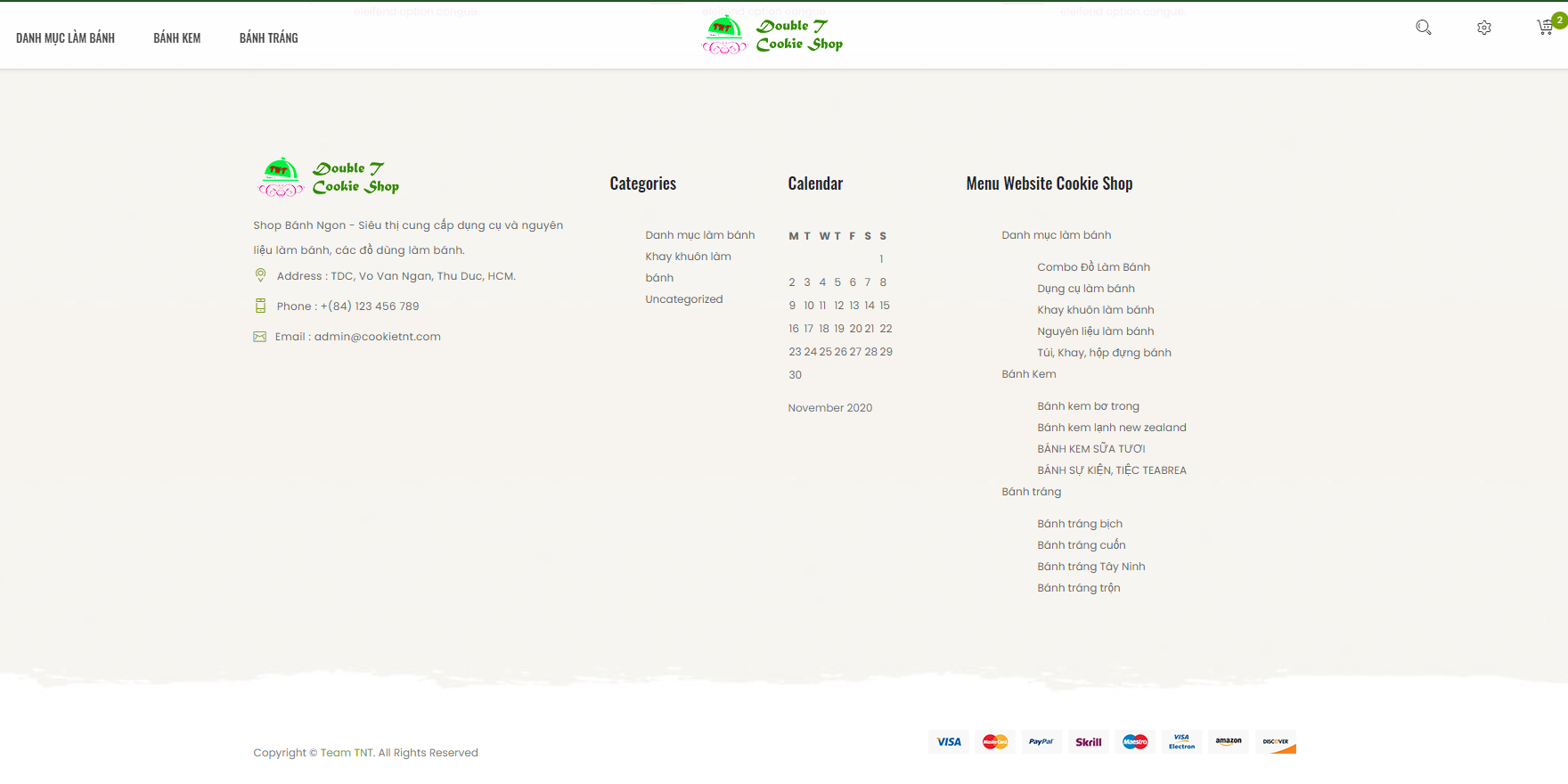
    <!-- Footer Bottom Area End -->

Kết quả xuất trên màn hình:



Hình 43: Copyright ở dưới cùng footer

Như vậy, thông qua 2 phần hướng dẫn trên, giao diện hiện tại chúng ta có được như sau:



Hình 44: Giao diện sau khi code xong header, footer cơ bản

# **II. CHỈNH SỬA HEADER, FOOTER – PHẦN MỞ RỘNG**

## **1. Thêm popup login, register trong header**

Mặc định, wordpress sẽ có trang riêng để người dùng có thể đăng nhập và đăng ký, nhưng vì một số vấn đề bảo mật nhằm tránh để lộ trang quản trị cho hacker ở một mức độ nhất định cũng như giao diện chưa được bắt mắt người dùng nên ta có thể code thêm một popup hiển thị trực tiếp trên trang chính cho người dùng đăng nhập.

Ở file header.php, thêm đoạn code sau và nhớ định css giúp cho giao diện thêm thân thiện hơn:

<!--Start of Login Form-->

<div class="modal fade" id="login\_box" tabindex="-1" role="dialog">

<div class="modal-dialog" role="document">

    <div class="modal-content">

        <div class="modal-header">

            <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span

                    aria-hidden="true"><i class="fa fa-close"></i></span></button>

        </div>

        <div class="modal-body">

            <div class="form-pop-up-content">

                <h2>Login to your account</h2>

                <form action="#" method="post">

                    <div class="form-box">

                      <input type="text" placeholder="User Name" name="username">

                      <input type="password" placeholder="Password" name="pass">

                    </div>

                    <div class="checkobx-link">

                        <div class="left-col">

                            <input type="checkbox" id="remember\_me"><label for="remember\_me">Remember Me</label>

                        </div>

                        <div class="right-col"><a href="#">Forget Password?</a></div>

                    </div>

                    <button type="submit">Sign In</button>

                </form>

            </div>

        </div>

    </div>

</div>

</div>

<!--End of Login Form-->

<!--Start of Register Form-->

<div class="modal fade" id="register\_box" tabindex="-1" role="dialog">

<div class="modal-dialog" role="document">

    <div class="modal-content">

        <div class="modal-header">

            <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span

                    aria-hidden="true"><i class="fa fa-close"></i></span></button>

        </div>

        <div class="modal-body">

            <div class="form-pop-up-content">

                <h2>Sign Up</h2>

                <form action="#" method="post">

                    <div class="form-box">

                      <input type="text" placeholder="Full Name" name="fullname">

                      <input type="text" placeholder="User Name" name="username">

                      <input type="email" placeholder="Email" name="email">

                     <input type="password" placeholder="Password" name="pass">

            <input type="password" placeholder="Confirm Password" name="re\_pass">

                    </div>

                    <div class="checkobx-link">

                        <div class="left-col">

                            <input type="checkbox" id="remember\_reg"><label for="remember\_reg">Remember

                                Me</label>

                        </div>

                    </div>

                    <button class="text-uppercase" type="submit">Register</button>

                </form>

            </div>

        </div>

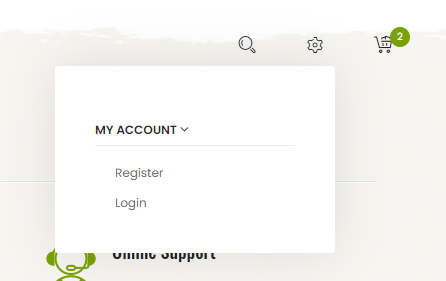
    </div>

</div>

</div>

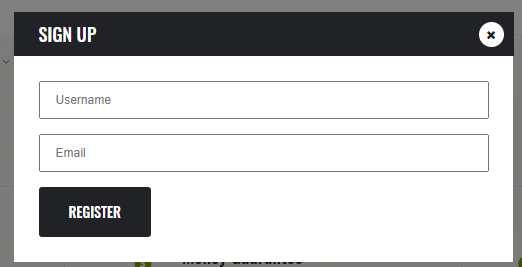
<!--End of Register Form-->

Khi này trên thanh header sẽ có một tùy chọn biểu tượng cài đặt có popup xổ xuống sau:



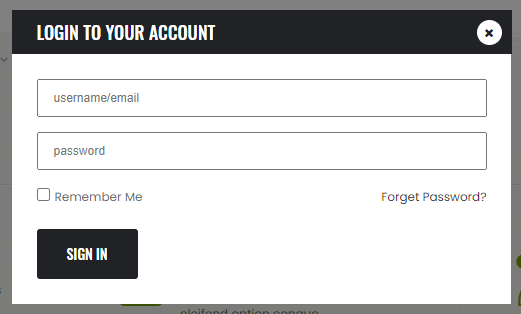
Hình 45: các tùy chỉnh login và sign up

Khi nhấn vào Register sẽ có một popup mới hiện ra:



Hình 46: Popup của sign up

Và nhấn vào Login thì có popup sau hiển thị:



Hình 47: Popup của login

Và bây giờ chúng ta có thể đăng ký và đăng nhập trực tiếp vào quản trị website từ các popup này.,.

## **2. Thêm class vào thẻ <ul> trong widget footer**

Vầ phần footer, hiện tại dang sử dụng các widget để quản lý các cột trong footer, vậy làm sao để thay đổi giao diện bằng css để giúp cho footer đẹp hơn.

**Giải thích:**

Mặc định khi bạn tạo một widget, các thẻ bên trong nó sẽ được cấu hình theo định dạng có sẵn của wordpress. Cụ thể là show ra các danh mục hoặc các text link.

Khi show theo danh mục, wordpress sẽ mặc định tạo ra thêm thẻ ul và các thẻ li bên trong sẽ là các danh mục đó.

Ví dụ:

Đây sẽ là những gì mà Wordpress sẽ tạo tự động cho bạn.

<h4>title</h4>

<ul>

    <li></li>

    <li></li>

    <li></li>

    <li></li>

    <li></li>

</ul>

</div>

Như đoạn code ở trên Wordpress sẽ không hề thêm bất kỳ 1 class nào vào thẻ ul hay li cho nên ta sẽ rất khó để thêm các css cho nó.

Như vậy làm sao để ta có thể thay đổi, hoặc thêm class vào các thẻ ul, li đó nhằm chỉnh thêm css cho giao diện footer đẹp hơn.

**Hướng giải quyết:**

Ở file **footer.php**, các dòng lệnh xuất ra các widget theo từng côt ta sẽ phải sửa lại như sau:

<!-- Footer Widget Area Start -->

<div class="footer-widget-area">

    <div class="container">

        <div class="row">

            <div class="col-lg-4 col-md-6">

                <div class="single-footer-widget">

                    <?php

                    if (is\_active\_sidebar('footer-sidebar-1')) {

                        ob\_start();

                        dynamic\_sidebar('footer-sidebar-1');

                        $sidebar = ob\_get\_contents();

                        ob\_end\_clean();

                        $sidebar\_corrected\_ul = str\_replace("<ul>", '<ul class="footer-widget-list">', $sidebar);

                        echo $sidebar\_corrected\_ul;

                    }

                    ?>

                </div>

            </div>

            <div class="col-lg-2 col-md-3">

                <div class="single-footer-widget">

                    <?php

                    if (is\_active\_sidebar('footer-sidebar-2')) {

                        ob\_start();

                        dynamic\_sidebar('footer-sidebar-2');

                        $sidebar = ob\_get\_contents();

                        ob\_end\_clean();

                        $sidebar\_corrected\_ul = str\_replace("<ul>", '<ul class="footer-widget-list">', $sidebar);

                        echo $sidebar\_corrected\_ul;

                    }

                    ?>

                </div>

            </div>

            <div class="col-lg-2 col-md-3">

                <div class="single-footer-widget">

                    <?php

                    if (is\_active\_sidebar('footer-sidebar-3')) {

                        ob\_start();

                        dynamic\_sidebar('footer-sidebar-3');

                        $sidebar = ob\_get\_contents();

                        ob\_end\_clean();

                        $sidebar\_corrected\_ul = str\_replace("<ul>", '<ul class="footer-widget-list">', $sidebar);

                        echo $sidebar\_corrected\_ul;

                    }

                    ?>

                </div>

            </div>

            <div class="col-lg-4">

                <div class="single-footer-widget">

                    <?php

                    if (is\_active\_sidebar('footer-sidebar-4')) {

                        ob\_start();

                        dynamic\_sidebar('footer-sidebar-4');

                        $sidebar = ob\_get\_contents();

                        ob\_end\_clean();

                        $sidebar\_corrected\_ul = str\_replace("<ul>", '<ul class="footer-widget-list">', $sidebar);

                        echo $sidebar\_corrected\_ul;

                    } ?>

                </div>

            </div>

        </div>

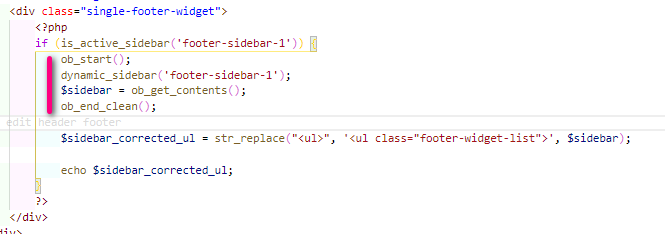
    </div>

</div>

<!-- Footer Widget Area End -->

Trong đoạn code trên, mỗi col mình đã chỉnh lại phần xuất ra widget.

Ta sẽ tạo ra một bộ đệm bằng hàm **ob\_start()** mà trong đó có chứa nội dung của widget mà Wordpress sẽ tự động tạo cho ta.



Hình 48: Xuất bộ đệm có thêm class cho thẻ ul trong widget footer

Tiếp theo sẽ sử dụng hàm **str\_relace** để thay đổi thẻ **ul** bên trong nhằm mục đích chèn thêm class vào.

Cuối cùng ta xuất lại bộ đệm đó ra thì class mà ta thêm ở hàm **str\_relace** vào thẻ **ul** sẽ được thực thi.

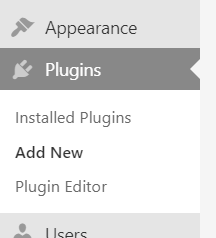
Khi này bạn chỉ cần thêm các **css** cho **class** này để cho giao diện của widget dưới footer thêm đẹp hơn!

# **III. PLUGIN WOOCOMMERCE VÀ CART ĐƠN GIẢN**

## **1. Cài đặt plugin Woocommerce vào theme**

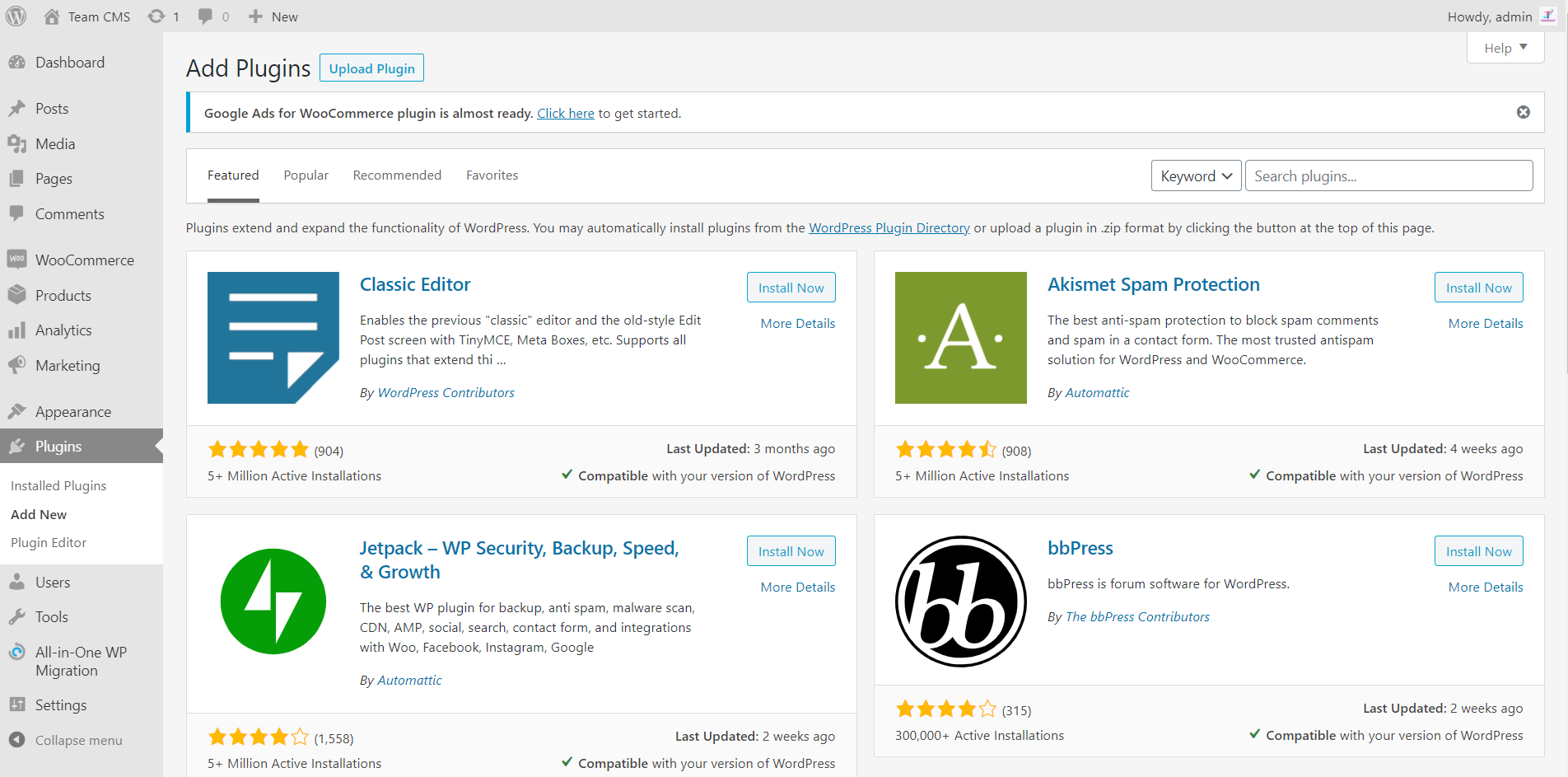
Để sử dụng plugin **WooCommerce** làm giỏ hàng cho ứng dụng web ta tiến hành cài đặt ta làm như sau:

**Plugins** cập vào **Plugins** -> **Add New**:



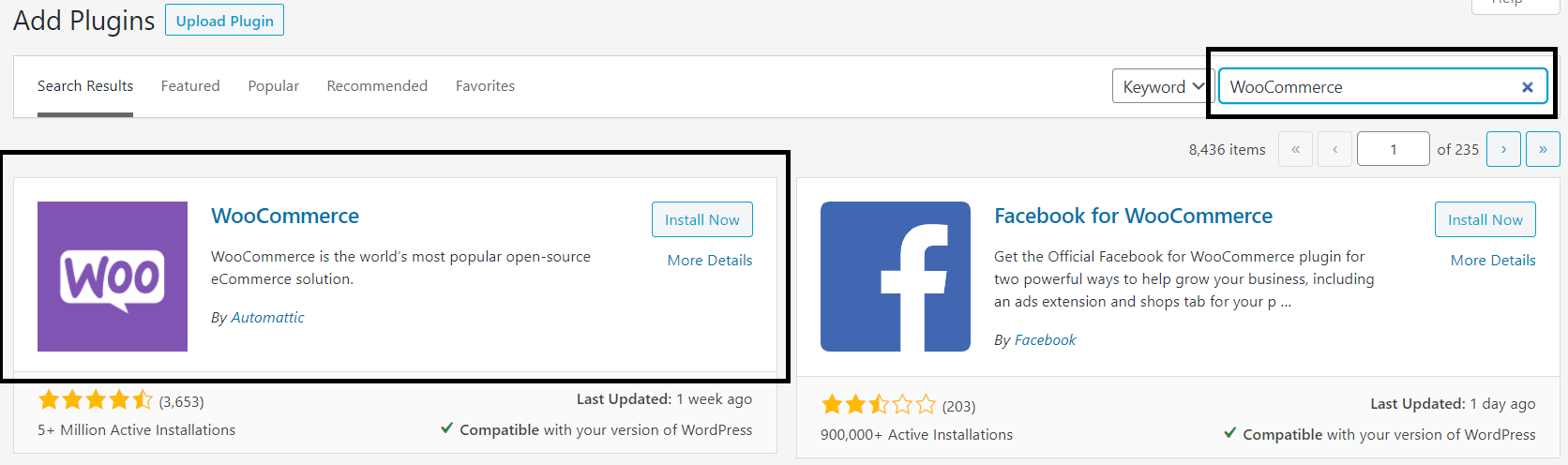
Hình 49: Add new plugin.

Sau khi truy cập vào trang để add plugin sẽ được giao diện như sau:



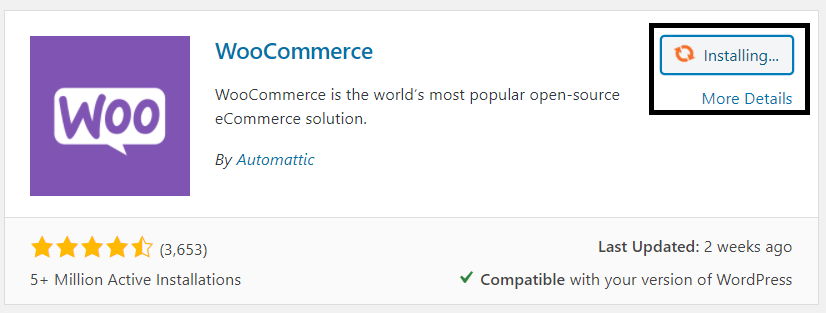
Hình 50: Giao diện plugins.

Tiến hành tìm kiếm plugin với từ khóa “WooCommerece” tại ô tìm kiếm:



Hình 51: Cài đặt plugin.

Tiến hành **Install** plugin. Điều này sẽ mất 1 khoảng thời gian tùy vào tốc độ mạng đang sử dụng.

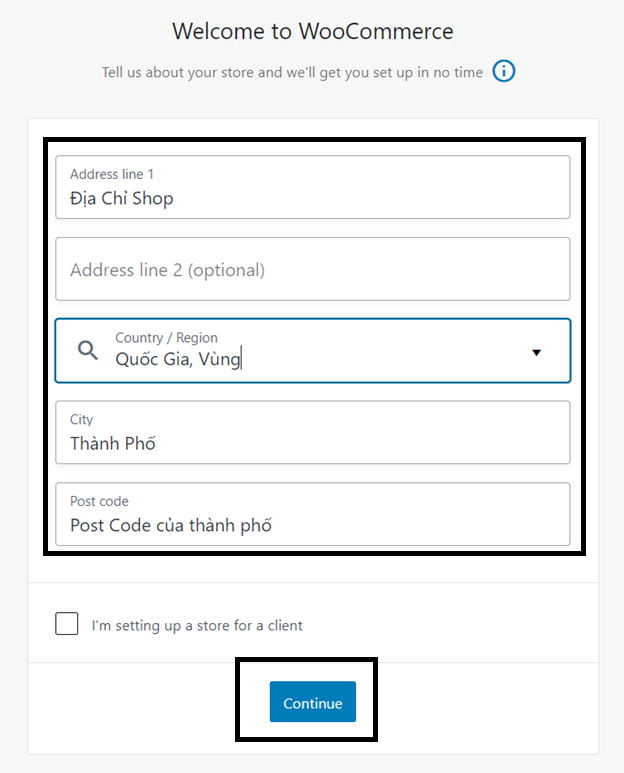


Hình 52: Quá trình cài đặt.

Sau khi tài đặt tiến hành **Active** plugin: theo 5 bước yêu cầu của plugin:

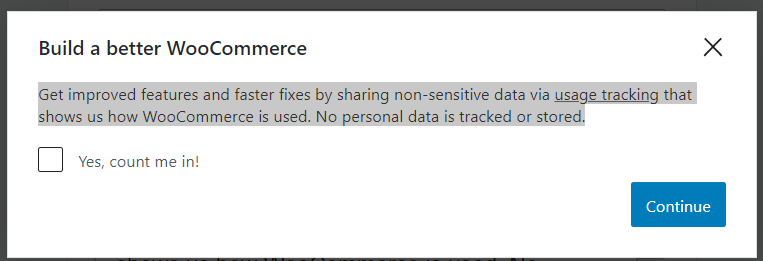
**Bước 1:**

Tiến hành cài đặt địa chỉ, quốc gia, thành phố và post code:



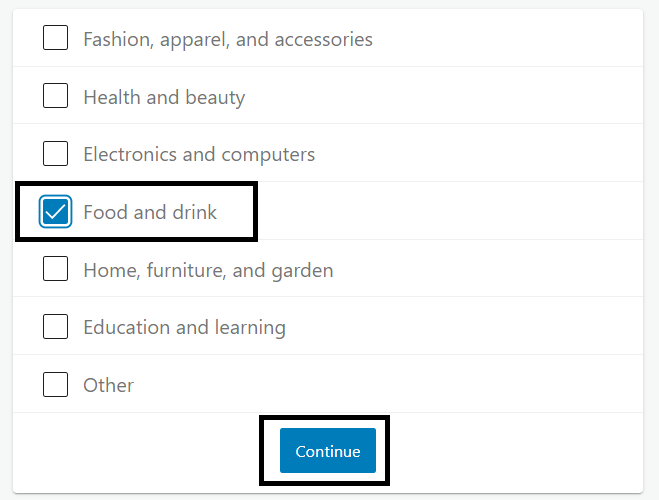
Hình 53: Cài đặt thông tin cho plugin.

Sau khi ấn vào nút Continue để tiếp tục, lúc này plugin sẽ thông báo muốn nhận thông báo về các tính năng sửa lỗi nhanh hơn bằng cách chia sẻ cho cộng đồng như hình dưới đây, bỏ qua nếu ko muốn nhận.

****

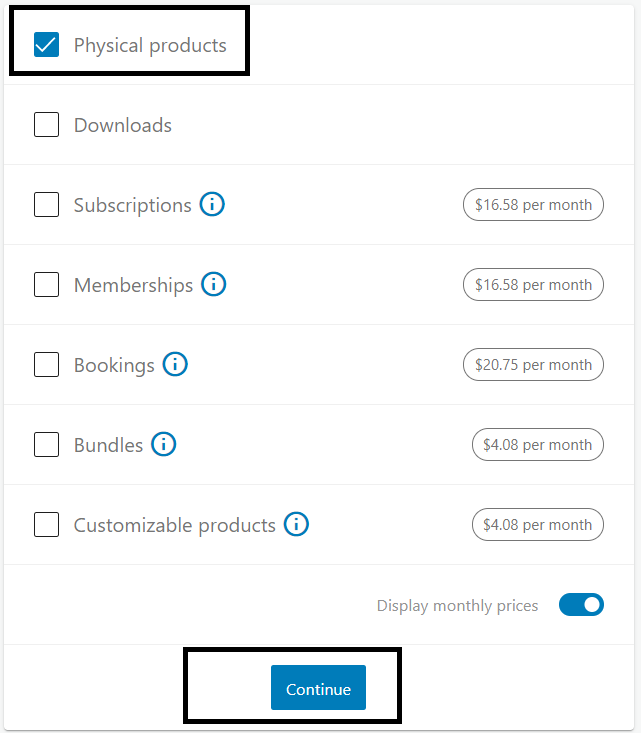
Hình 54: Thông báo nhận về thông tin về sửa lỗi và phát triển.

**Bước 2:** Chọn lĩnh vực, mục tiêu của shop, ở đây chủ đề shop là bán các loại bánh vậy nên sẽ tick vào ô **Food & Drink** như hình dưới đây:



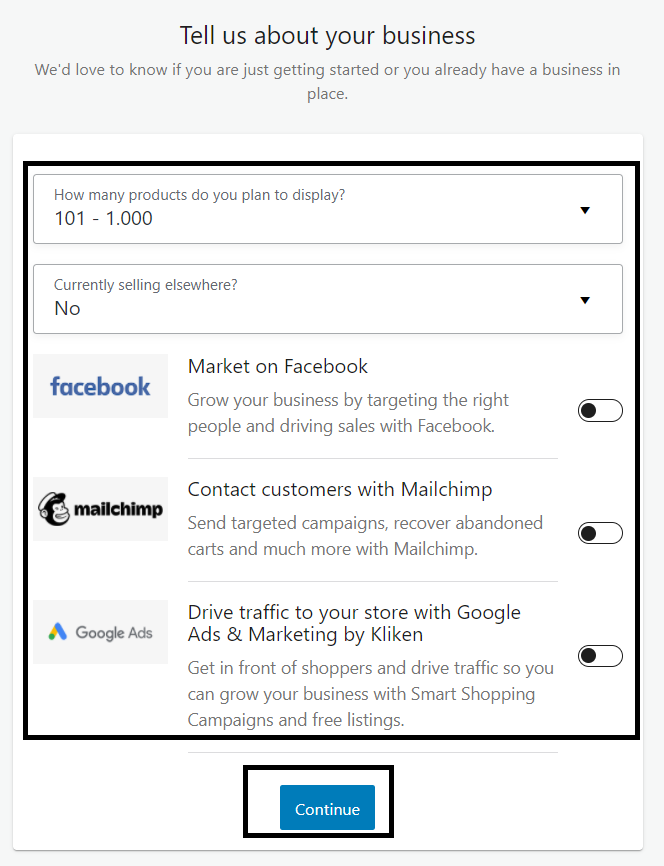
Hình 55: Chọn đủ để giỏ hàng.

**Bước 3:** chọn phiên bản dùng plugin WooCommerce ở đây chỉ dùng ở mức cơ bản nên sẽ chọn phiên bản miễn phí như hình:



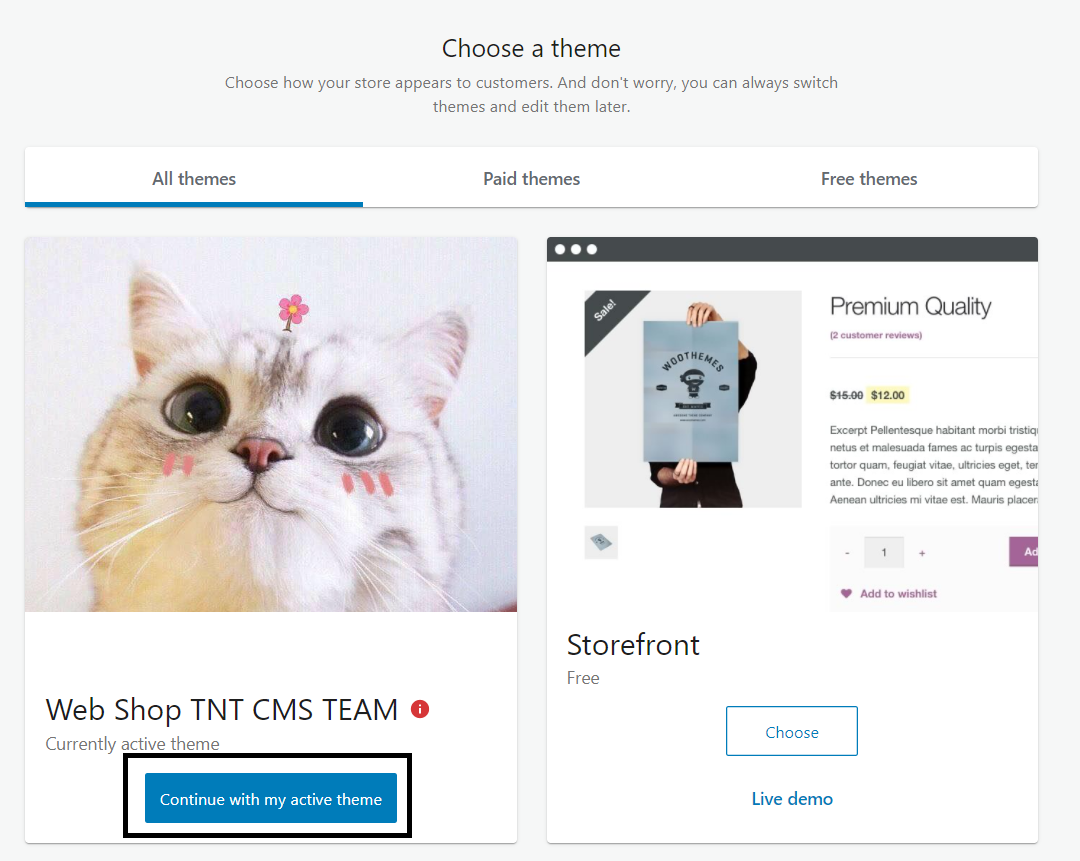
Hình 56: Lựa chọn phiên bản plugin.

**Bước 4:** lúc này plugin sẽ hỏi sản phẩm sẽ hiển thị ở số lượng bao nhiêu thì tùy vào mục đích sẽ chọn số lượng phù hợp, một shop bán bánh thông thương sẽ chỉ dừng lại ở mức hiển thị 101 – 1000 sản phẩm, ngoài ra plugin sẽ hỏi nơi bán thứ hai ngoài trang chủ, nếu không có bán trên bất kỳ trang nào khác thì chọn **NO** như hình dưới đây và bỏ tick hết tất cả và ấn Continue như hình:



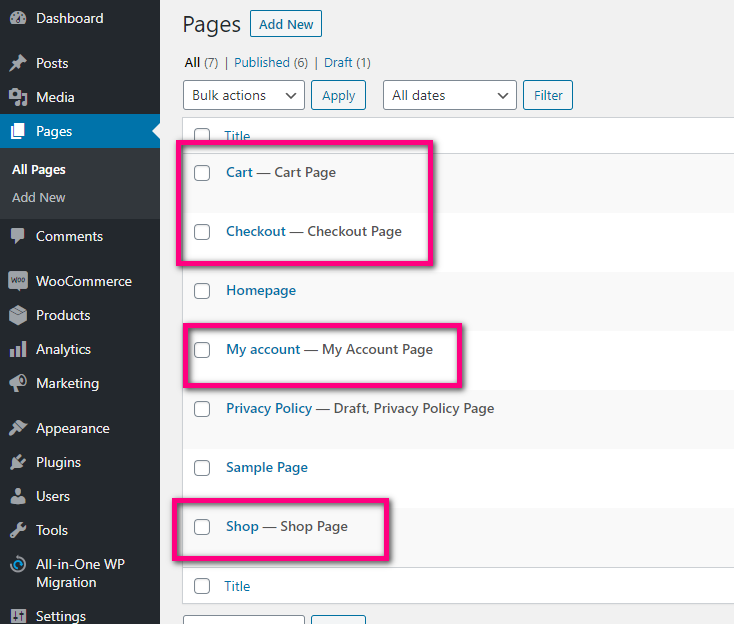
Hình 57: Lựa chọn nơi bán khác

**Bước 5:** Để hoàn tất việc cài đặt plugin lúc này sẽ phải chọn theme cần kích hoạt plugin WooCommerce như hình.



Hình 58: Chọn theme cần kích hoạt

Sau khi kích hoạt thành công, **plugin** sẽ tạo cho bạn thêm các trang tĩnh sau:



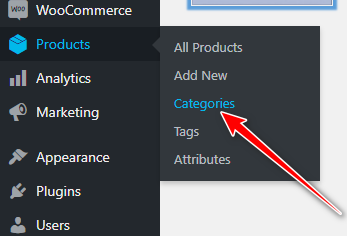
Hình 59: Các trang dành cho một shop cơ bản

## **2. Tạo sản phẩm, danh mục sản phẩm**

Khi tạo sản phẩm mới cho website, ta cần phải biết rằng sản phầm đó sẽ thuộc danh mục nào. Điều này giúp cho ta có thể hiện được các sản phẩm theo danh mục hoặc áp dụng cho chức năng tìm kiếm, phân trang sau này.

Chính vì thế việc đầu tiên sau khi cài đặt xong plugin là ta phải tạo các danh mục sản phẩm mới.

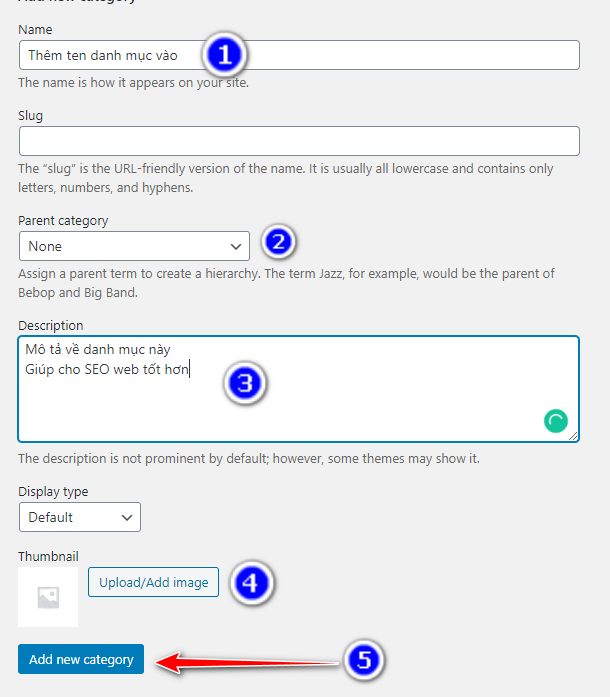
Vào trang quản trị danh mục theo đường dẫn ở hình dưới:



Hình 60: Vào đường dẫn Products -> Categories

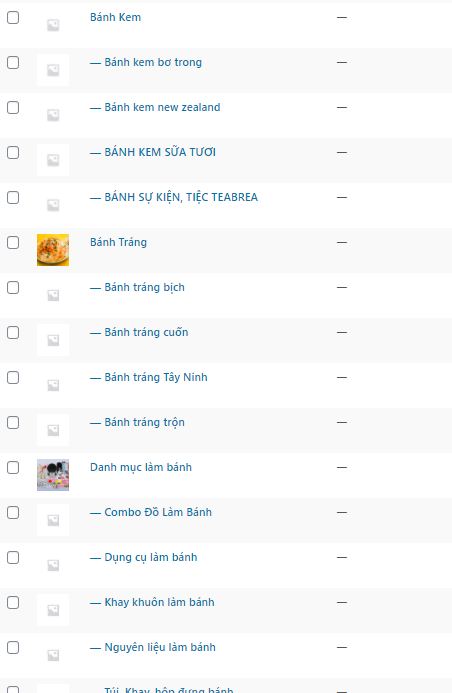
Sau đó bạn có thể thêm mới một danh mục theo các bước:

1. Ở ô nhập liệu đầu tiên hãy điển tên cho danh mục bạn muốn tạo vào.
2. Bỏ qua ô nhập slug đường dẫn, nó sẽ tự động tạo theo tên của danh mục. Bạn hãy chọn tiếp cấp danh mục nếu muốn.
3. Sau đó thêm nội dung mô tả nếu muốn (Hữu ích cho SEO web)
4. Có thể thêm hình ảnh đại diện cho danh mục này.
5. Cuối dùng là nhấn **Add new category.**



Hình 61: Các bước thêm một danh mục cho sản phẩm

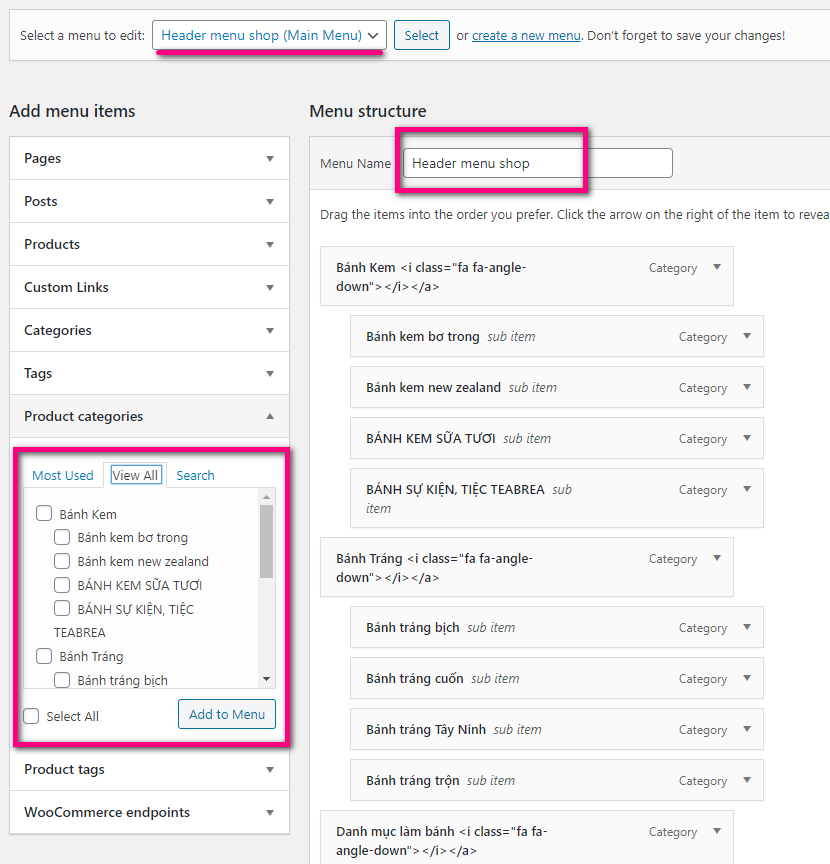
Đây là những danh mục mà nhóm đã tạo cho website này:



Hình 62: Ví dụ một số danh mục cho web của hàng bánh

Để có thể xuất đống danh mục này ra thanh menu trên header bạn hãy vào cài đặt menu , tạo một menu mới và add những danh mục sản phẩn này vào:

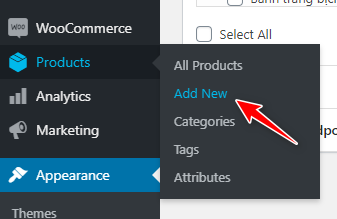
Nhóm đã tạo một menu có tên là Header menu shop và add những danh mcuj về **Product Categories** vào.



Hình 63: Tạo menu theo các danh mục sản phẩm

Tiếp theo, ta sẽ tạo một sản phẩm ví dụ như “**Bánh kem bơ trong”**:

**Bước 1:** Vào tạo product mới theo đường dẫn trong hình:



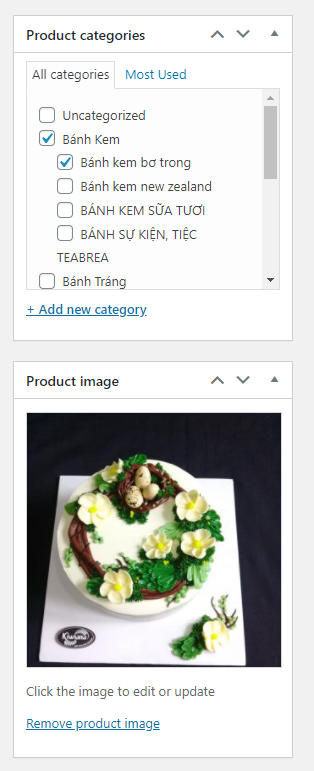
Hình 64: Vào đường dẫn Products -> Add New

**Bước 2:** Thêm tên và nội dung giới thiệu về sản phẩm.



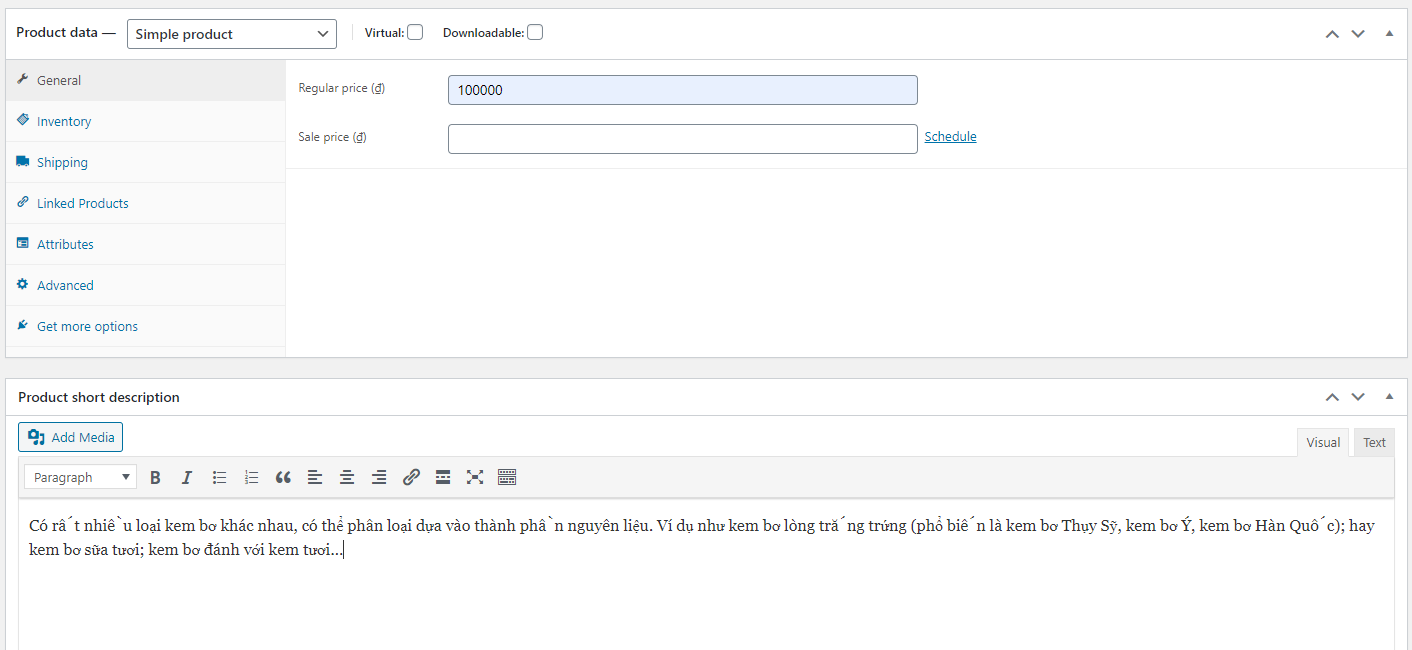
Hình 65: Thêm tên và nội dung cho sản phẩm

**Bước 3:** thêm danh mục cho sản phẩm và hình đại diện ở sidebar bên phải màn hình:



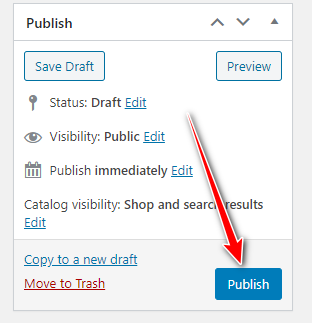
Hình 66: Thêm category và ảnh đại diện

Bước 4: Thêm giá và một só mô tả ngắn cho sản phầm.



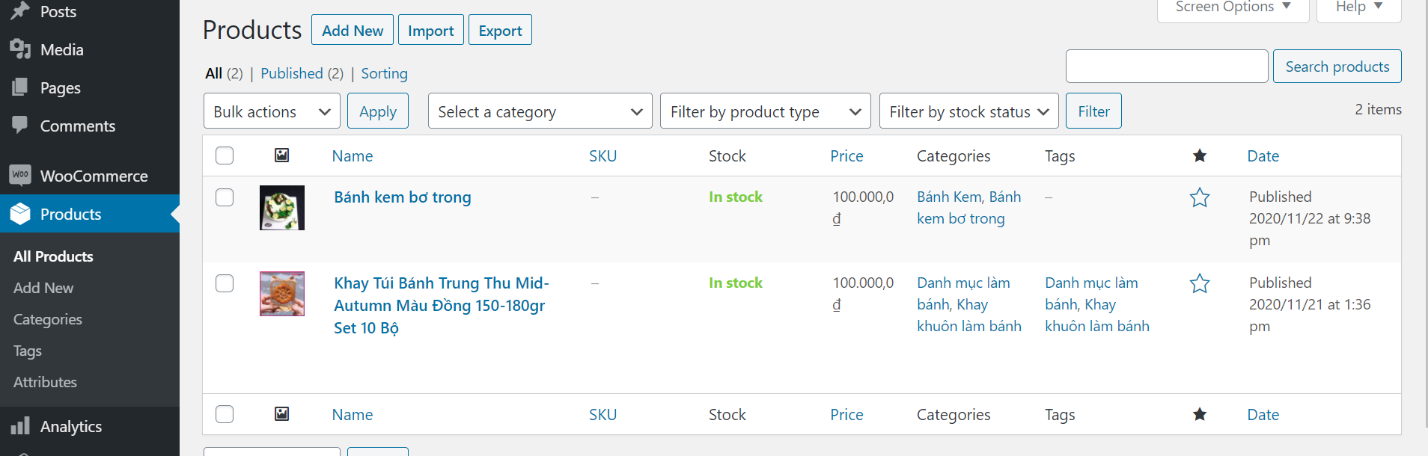
Hình 67: Thêm giá và mô tả khác cho sản phẩm

Bước 5 Sau khi đã thêm đầy đủ các nội dung cần thiết cho sản phầm thì nhấn Public. (nằm ở đầu thanh sidebar bên phải màn hình)



Hình 68: Nhấn public để đăng sản phẩm

Kết quả ta sẽ có thêm một sản phẩm như sau:



Hình 69: Sản phảm mới thêm hiện trên danh sách các sản phẩm đã tạo

**3. Tạo trang giao diện giỏ hàng**

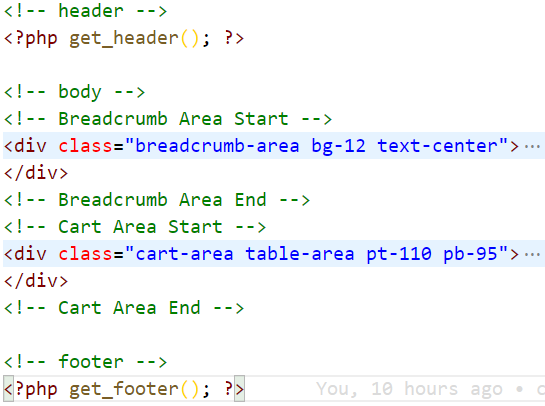
Load lại trang chủ của theme và tiến hành tạo file mới để chỉnh sửa lại giao diện giỏ hàng bằng cách như sau:

**Bước 1:** tạo một file **page-cart.php** như hình:



Hình 70: Tạo file cart.php

Sau đó tiến hành gắn header, footer và tạo giao diện (có sẵn từ template hoặc tự code):



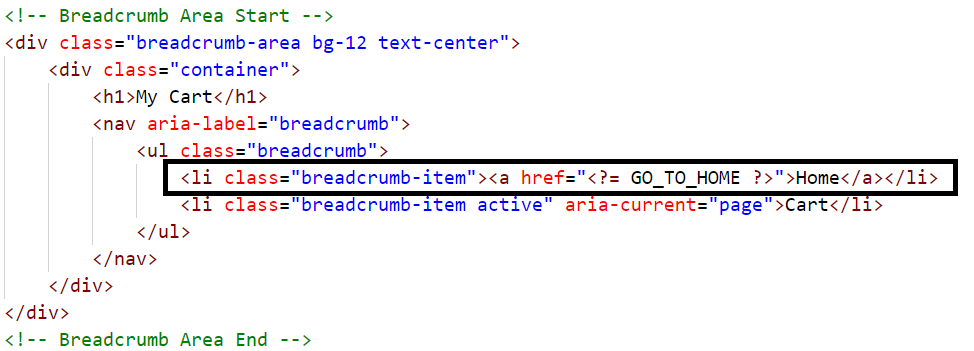
Hình 71: Gắn header, footer, giao diện giỏ hàng

Để có thể gắn nút trang chủ trong giỏ hàng tiến hành định nghĩa hằng **GO\_TO\_HOME** trong file **function.php** như sau:



Hình 72: Định nghĩa hằng GO\_TO\_HOME

Sau khi định nghĩa hằng tiến hành gọi hằng trong file **page-cart.php** để có thể chuyển về trang chủ thông qua thẻ **<a>** như hình dưới đây:



Hình 73: Thêm hằng vào nút HOME

Tiếp theo ta sẽ làm giao diện cho bảng hiện ra các sản phẩm đã được thêm vào giỏ hàng như sau:

<div class="table-responsive">

    <table class="table product-table text-center">

        <thead>

            <tr>

                <th class="table-remove">remove</th>

                <th class="table-image">image</th>

                <th class="table-p-name">product</th>

                <th class="table-p-price">price</th>

                <th class="table-p-qty">quantity</th>

                <th class="table-total">total</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td class="table-remove"><button><i class="fa fa-trash"></i></button></td>

                <td class="table-image"><a href="product-details.html"><img src="<?= GO\_TO\_HOME ?>/wp-content/uploads/2020/11/06\_de9a3fb951884589a7de023a8f500870\_grande-150x150.jpg" alt=""></a></td>

                <td class="table-p-name"><a href="product-details.html">Test product 1</a></td>

                <td class="table-p-price">

                    <p>$65.00</p>

                </td>

                <td class="table-p-qty"><input value="1" name="cart-qty" type="number"></td>

                <td class="table-total">

                    <p>$65.00</p>

                </td>

            </tr>

            <tr>

                <td class="table-remove"><button><i class="fa fa-trash"></i></button></td>

                <td class="table-image"><a href="product-details.html"><img src="<?= GO\_TO\_HOME ?>/wp-content/uploads/2020/11/4183c5944997aff7042e20d3ad7f4124-150x150.jpg" alt=""></a></td>

                <td class="table-p-name"><a href="product-details.html">Test product 2</a></td>

                <td class="table-p-price">

                    <p>$95.00</p>

                </td>

                <td class="table-p-qty"><input value="1" name="cart-qty" type="number"></td>

                <td class="table-total">

                    <p>$95.00</p>

                </td>

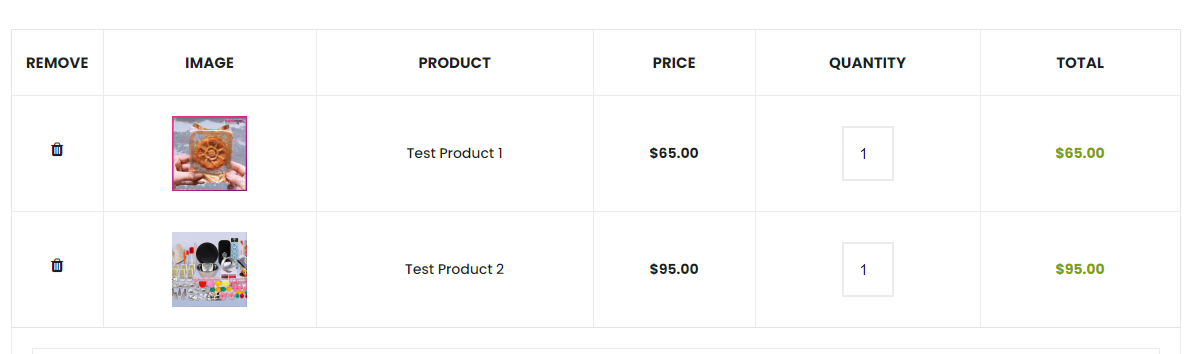
            </tr>

        </tbody>

    </table>

</div>

Ta sẽ có được giao diện như sau:



Hình 74: Bảng hiện ra các sản phẩm đã được thêm vào giỏ hàng

Ta có thể thêm một ô để cho người dùng được phép nhập coupon giảm giá như sau:

<div class="table-bottom-wrapper">

    <div class="table-coupon d-flex fix justify-content-start">

        <input type="text" placeholder="Coupon code">

        <button type="submit">Apply coupon</button>

    </div>

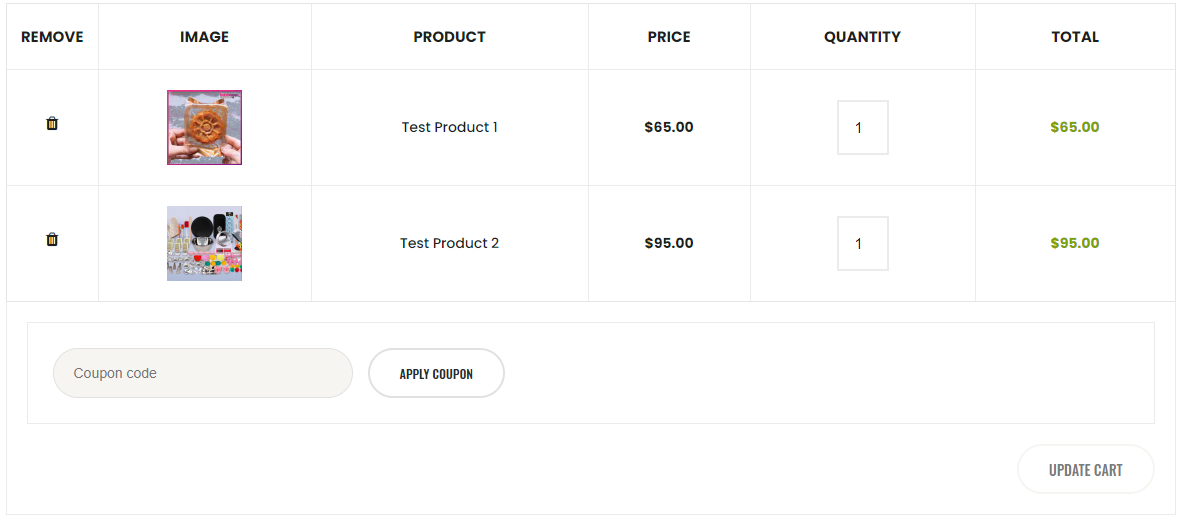
    <div class="table-update d-flex justify-content-end">

        <button type="button" disabled>Update cart</button>

    </div>

</div>

Ta được giao diện như sau:



Hình 75: Thêm form nhập coupon giảm giá

Cuối cùng ta sẽ thêm phần code để tạo ra khung thông báo chi tiết tổng kết giỏ hàng:

<div class="container">

<div class="table-total-wrapper d-flex justify-content-end pt-60">

    <div class="table-total-content">

        <h2>Cart totals</h2>

        <div class="table-total-amount">

            <div class="single-total-content d-flex justify-content-between">

                <span>Subtotal</span>

                <span class="c-total-price">$160.00</span>

            </div>

            <div class="single-total-content d-flex justify-content-between">

                <span>Shipping</span>

                <span class="c-total-price"><span>Flat Rate:</span> $5.00</span>

            </div>

            <div class="single-total-content d-flex justify-content-end">

                <a href="#">Calculate shipping</a>

            </div>

            <div class="single-total-content d-flex justify-content-between">

                <span>Total</span>

                <span class="c-total-price">$165.00</span>

            </div>

            <a href="checkout.html">Proceed to checkout</a>

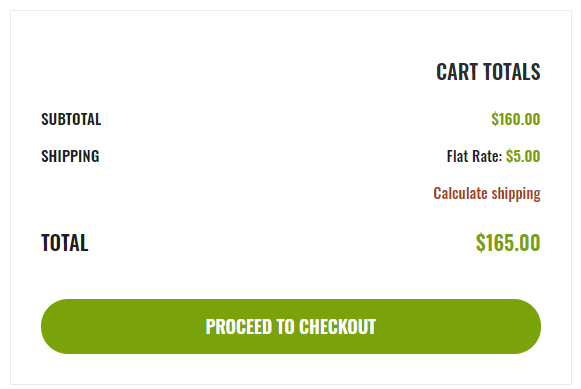
        </div>

    </div>

</div>

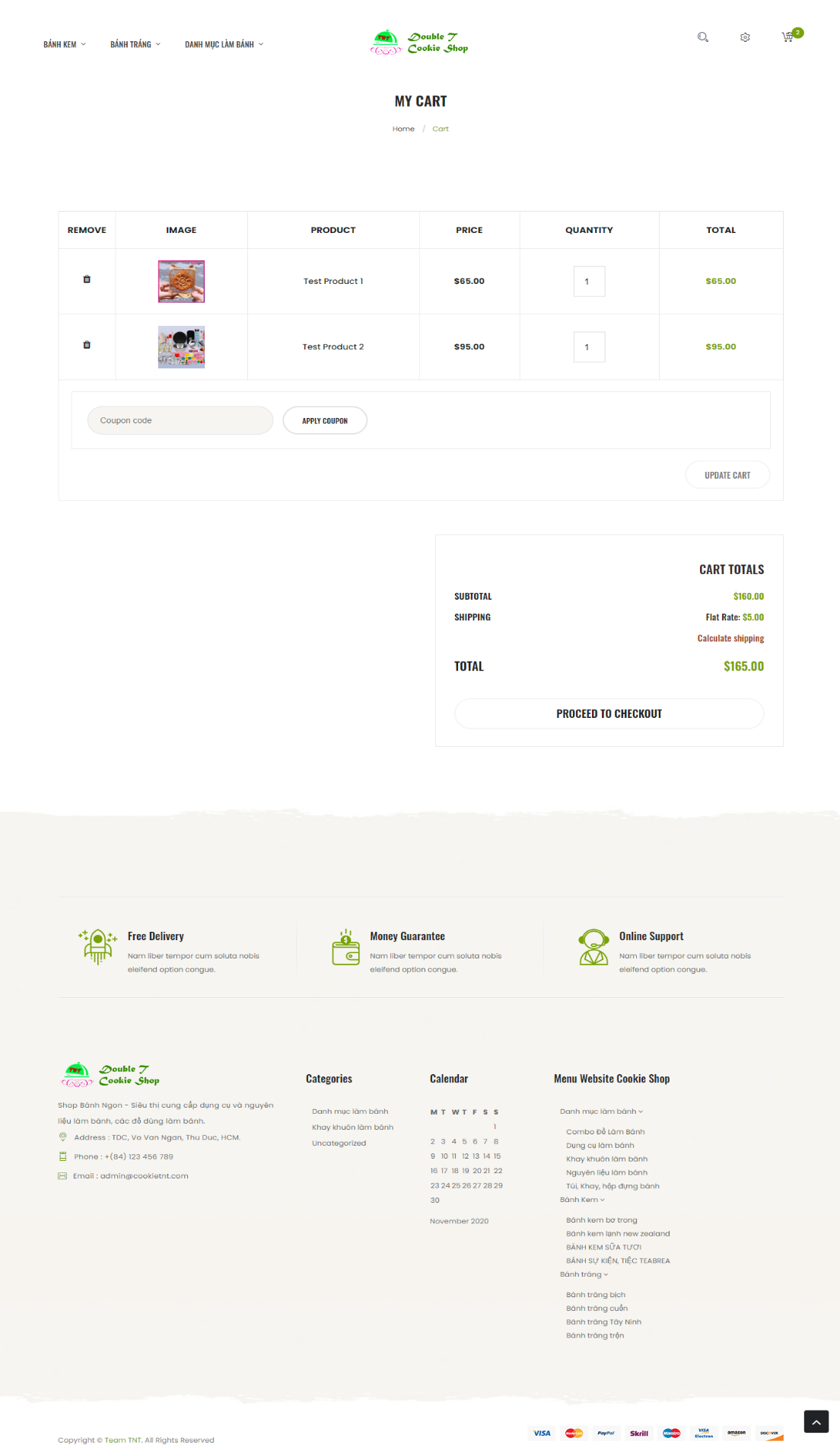
</div>

Ta được kết quả:



Hình 76: Khung thông báo chi tiết tổng kết giỏ hàng

Như vậy giao diện của trang giỏ hàng sau khi hoàn chỉnh các thay đổi cơ bản trên sẽ như sau:



Hình 77: Kết quả giao diện trang giỏ hàng của website

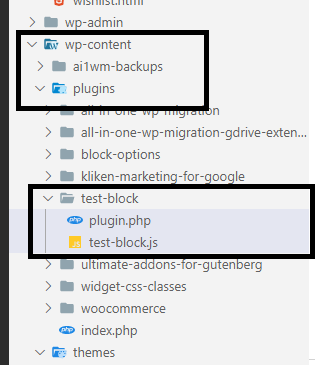
# **IV. TẠO BLOCK EMAIL CHO POST**

Để tiến hành tốt công việc phát triển website cá nhân thông qua wordpress thì việc tự custom một block là việc vô cùng cần thiết cho việc phát triển về mặt nội dung cũng như quản trị về giao diện.

Để tạo được một block cho người dùng nhập email, ta cần thực hiện các bước theo trình tự sau:

**Bước 1:**

Tạo một thư mục trong plugins thực hiện tạo 2 file **plugin.php** và **test-block.js** như hình dưới đây:



Hình 78: Cây thư mục plugins

**Bước 2:**

Thực hiện đăng ký block mới trong file **test-block.js** thông qua cú pháp:

wp.blocks.registerBlockType().

Code đăng ký block mới hoàn chỉnh:

/\* đăng ký một block mới\*/

wp.blocks.registerBlockType("brad/border-box", {

  title: " Email Box",

  icon: "smiley",

  category: "common",

  attributes: {

    content: { type: "string" },

  },

  /\* Cấu hình cáct thẻ html và style css \*/

  edit: function (props) {

    function updateContent(event) {

      props.setAttributes({ content: event.target.value });

      // biểu thức chính quy

      let reg = /^([A-Za-z0-9\_\-\.])+\@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

      var emailInput = event.target.value;

      var errorEmail = document.getElementById("email");

      if (reg.test(emailInput) != true) {

        errorEmail.style.border = "3px solid red";

      } else {

        errorEmail.style.border = "3px solid black";

      }

    }

    return React.createElement(

      "div",

      null,

      React.createElement("h3", null, "Simple Box"),

      React.createElement("input", {

        type: "text",

        value: props.attributes.content,

        onChange: updateContent,

      })

    );

  },

  save: function (props) {

    return wp.element.createElement(

      "h3",

      { style: { border: "3px solid " + props.attributes.color } },

      props.attributes.content

    );

  },

});

Để đăng ký một block mới nhóm mình đã dùng javascript kết hợp với React để tiến hành thiết lập. Đoạn mã trên thực hiện cấu hình block cơ bản và tinh chỉnh gồm một input type text và đường viền cho thẻ **<input>**.

**Bước 3:**

Thực hiện cấu hình ở phía php (nằm trong file **plugin.php**)

<?php

/\*\*

 \* Plugin Name: Test Block

 \* Author: Nhóm 1

 \* Version: 1.0.0

 \*/

function loadMyBlock()

{

    wp\_enqueue\_script(

        'my-new-block',

        plugin\_dir\_url(\_\_FILE\_\_) . 'test-block.js',

        array('wp-blocks', 'wp-editor'),

        true

    );

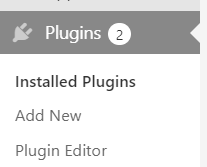
}

add\_action('enqueue\_block\_editor\_assets', 'loadMyBlock');

Đối với các hàm enqueue, cụ thể hơn là hàm **wp\_enqueue\_script()**, WordPress sẽ tự động đăng ký một **handle** mới trước khi đưa vào hàng đợi và xuất ra cho Website sử dụng. Lúc này sẽ không cần thiết phải đăng từ trước mới có thể dùng được.

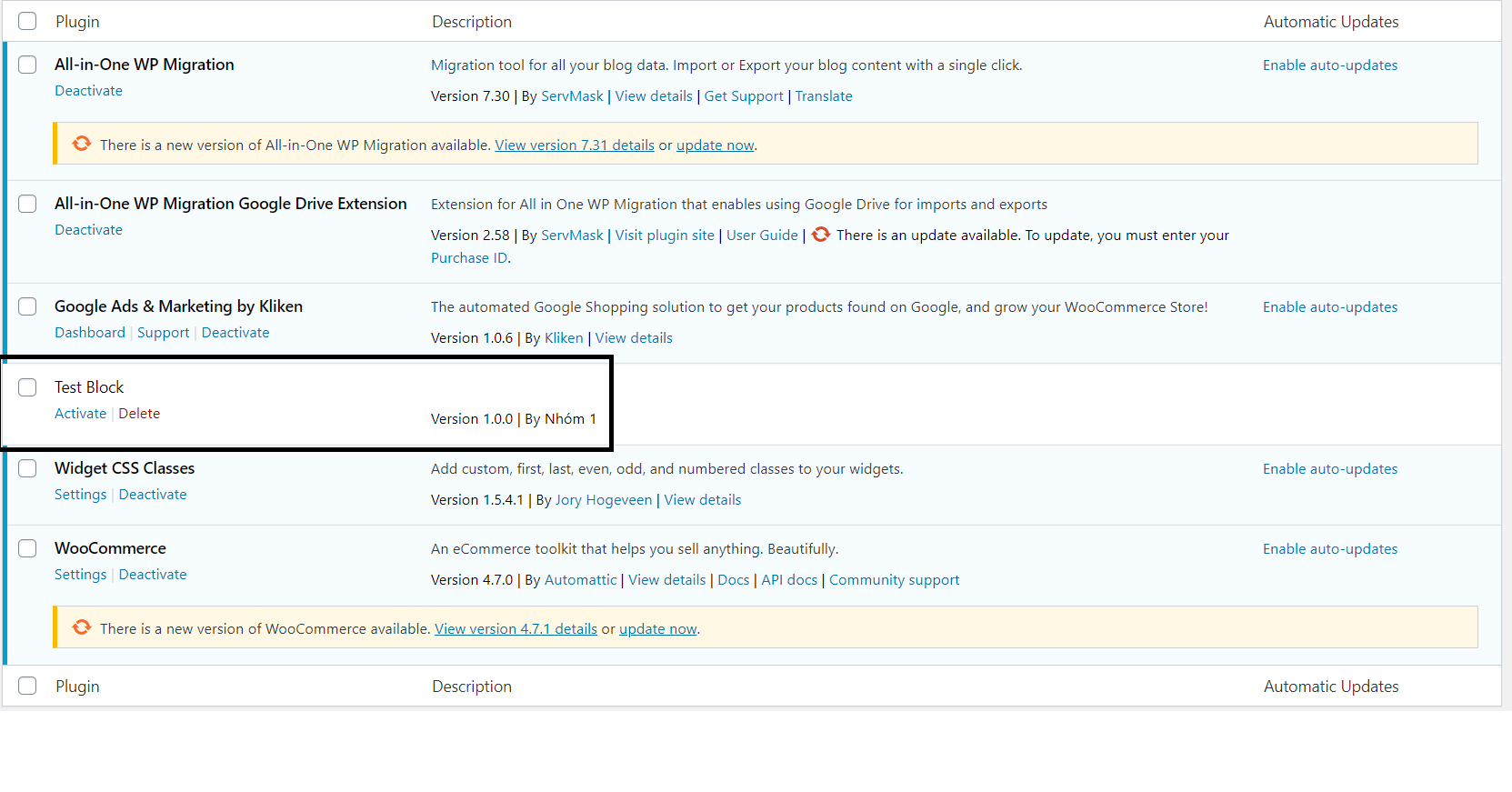
**Bước 4:**

Vào trang **admin** -> **Plugins** -> **Installed Plugins** như hình dưới đây:



Hình 79: Plugins

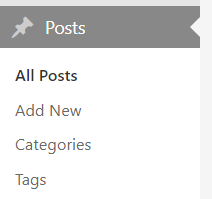
Sẽ thấy rằng Block vừa tạo là một dạng plugin và cần kích hoạt (nút **Activate**) để dùng như hình dưới đây:



Hình 80: Trang plugins

**Bước 5:**

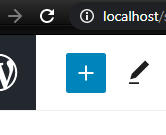
Sau khi kích hoạt thành công, từ trang admin truy cập vào **Posts -> All Posts**:



Hình 81: Posts

**Bước 6:**

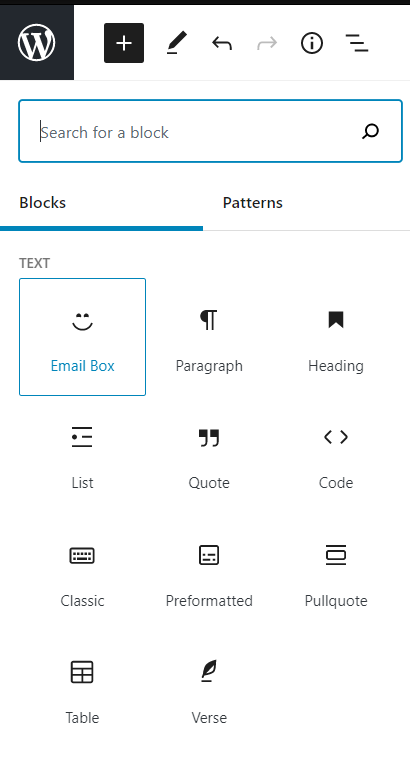
Sau khi vào post và tùy chọn edit hoặc thêm mới bất cứ bài post nào, bạn chọn nút **“+”** từ góc trái màn hình kích hoạt danh sách block như hình dưới đây:



Hình 82: Open block list

**Bước 7:**

Sau khi mở danh sách các block, lập tức sẽ thấy block vừa tạo được đẩy lên đầu tiên, và khi click vào sẽ tiến hành dùng block vừa tạo:



Hình 83: Block Email box vừa tạo nằm đầu tiên trong danh sách blocks

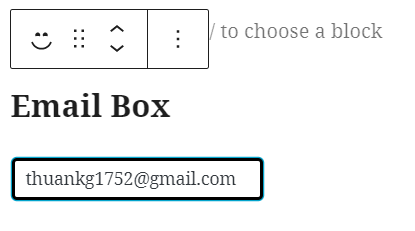
**Bước 8:**

Sau khi click dùng block **Email Box** sẽ lạp tức show thêm một block mới trong phần nội dung của bài post có kết quả như hình dưới với tiêu đề, đường viền được định nghĩa từ trong file **test-block.js**:



Hình 84: Block thông báo sai email

Nếu nhập vào email sai định dạng so với biểu thức chính quy được định nghĩa từ trước sẽ bị viền đỏ thông báo nhập lại email như hình trên, và sau khi người dùng nhập đúng định dạng thì Email Box sẽ trả về lại trạng thái bình thường như hình dưới đây:



Hình 85: Block thông báo đúng email