

A collage of various colorful sneakers in different styles and colors (red, yellow, green, blue, white) arranged in a circular pattern around the central text area.

SneakersShop

2021

Universidad de Alicante

14 MAYO

Portal de comercio electrónico
Creado por: Pablo Guillén García
Correo: pgg98@gcloud.ua.es
DNI: 48794309Q



Descripción general

Este portal de comercio electrónico se llama SneakersShop. El producto que ofrecemos son en su mayoría zapatillas, aunque también se puede encontrar algún otro complemento en ciertos lapsos de tiempo.

Aquí puedes encontrar todo tipo de zapatillas en diferentes tallas y colores, además de exclusivas promociones que te dejarán con la boca abierta.

Trabajo realizado

PLUGINS

- Advanced Custom Fields -> creación de campos personalizados, utilizado previamente para crear los campos de las Promociones, pero se han eliminado al final por entrar en conflicto con los creados mediante código.
- Advanced Custom Fields: Theme Code -> permite ver el código PHP de los campos personalizados creados con el plugin anterior para poder trasladarlo fácilmente a un fichero.
- Child Theme Configurator -> utilizado para crear el tema hijo sobre el que se trabaja toda la práctica.
- Contact Form 7 -> creación de formularios de contacto, en este caso utilizado para crear el formulario del apartado sugerencias.
- Duplicator -> copias de seguridad de la página.
- Easy Accordion -> utilizado para crear el apartado Preguntas Frecuentes, permite realizar ese tipo de bloques apilados que se necesita para dicho apartado de forma sencilla y funcional.
- Members -> gestiona los diferentes tipos de usuarios de nuestra página web. Con el se ha creado el usuario Promotor y se le han dado permisos para que pueda gestionar el tipo Promociones.
- Mycountplugin -> plugin desarrollado por el creador de la página que se utiliza para contar las visitas que tiene cada Promoción y para mostrar la lista ordenada de Promociones.
- Restrict User Acces -> utilizado para restringir el acceso de ciertas páginas a un tipo de usuario, concretamente al apartado Sugerencias (al cual solo puedes acceder los usuarios registrados) y al apartado Añadir Promoción al cual solo pueden acceder los Promotores.
- Site Reviews -> permite introducir bloques de reseñas/comentarios/puntuaciones para una página. Se ha utilizado para el tipo promociones, donde solo los usuarios Promotores podrán ver estos bloques y escribir en ellos comentarios que al igual solo podrán ver ellos.
- Ultimate Posts Widget -> Tiene varias funciones sobre flexibilidad y facilidad de uso en WordPress. En este caso, se ha utilizado para mostrar el Custom Post Type Promociones en el menu de edición de WordPress.
- University of Alicante Wikidata Plugin -> permite realizar consultas a Wikidata. No se ha utilizado.

- User Registration -> permite crear un formulario de registro de usuarios. En este caso se a utilizado para crear un formulario de resgistro para usuarios standard.
- WooCommerce -> con este plugin se han implementado todas las funcionalidades de la tienda online como el diseño de los productos básicos de la tienda y su contenido, la pasarela de pago (sin activar), etc.

CÓDIGO

La mayoría del código utilizado en la práctica es de las plantillas y apuntes proporcionados por el profesorado de la asignatura. Aún así hay código añadido que procedo a explicar a continuación.

Empezando por el archivo functions.php , introduje una función para hacer login y logout llamada 'dcms_items_login_logout' :

```

20 //Login y Logout
21 add_filter( 'wp_nav_menu_items', 'dcms_items_login_logout', 10, 2);
22
23 function dcms_items_login_logout( $items, $args ) {
24
25     if ($args->theme_location == 'primary') {
26         if (is_user_logged_in())
27         {
28             $items .= '<li class="menu-item btn-menu btn-logout">
29                 <a href="'. wp_logout_url(get_permalink()) .' ">'. __("Log Out") .'</a>
30             </li>';
31         }
32         else
33         {
34             $items .= '<li class="menu-item btn-menu btn-login">
35                 <a href="'. wp_login_url(get_permalink()) .' ">'. __("Log In") .'</a>
36             </li>';
37         }
38     }
39
40     return $items;
41 }
42 //Fin login y logout

```

También tenemos el código para crear el custom post type promociones:

```

44 add_action('pre_get_posts', 'add_my_post_types_to_query');
45 function add_my_post_types_to_query($query){
46     if (is_home() && $query->is_main_query())
47     {
48         $query->set('post_type', array('post', 'promociones'));
49     }
50 }

```

```

64  ✓ function cptui_register_my_cpts_promo() {
65
66  ✓
67      /**
68      * Post Type: Promociones.
69      */
70  ✓
71      $labels = [
72          "name" => __( "Promociones", "custom-post-type-ui" ),
73          "singular_name" => __( "Promocion", "custom-post-type-ui" ),
74          "menu_name" => __( "Promociones", "custom-post-type-ui" ),
75          "all_items" => __( "Todas las promociones", "custom-post-type-ui" ),
76          "add_new" => __( "Añadir nuevo", "custom-post-type-ui" ),
77          "add_new_item" => __( "Añadir nueva promocion", "custom-post-type-ui" ),
78          "edit_item" => __( "Editar promocion", "custom-post-type-ui" ),
79          "new_item" => __( "Nueva promocion", "custom-post-type-ui" ),
80          "view_item" => __( "Ver promocion", "custom-post-type-ui" ),
81          "view_items" => __( "Ver promociones", "custom-post-type-ui" ),
82          "search_items" => __( "Buscar promociones", "custom-post-type-ui" ),
83      ];
84
85      $args = [
86          "label" => __( "Promociones", "custom-post-type-ui" ),
87          "labels" => $labels,
88          "description" => "",
89          "public" => true,
90          "publicly_queryable" => true,
91          "show_ui" => true,
92          "show_in_rest" => true,
93          "rest_base" => "",
94          "rest_controller_class" => "WP_REST_Posts_Controller",
95          "has_archive" => "promos",
96          "show_in_menu" => true,
97          "show_in_nav_menus" => true,
98          "delete_with_user" => false,
99          "exclude_from_search" => false,
100          "capability_type" => "promo",
101          "map_meta_cap" => true,
102          "hierarchical" => false,
103          "rewrite" => [ "slug" => "promo", "with_front" => true ],
104          "query_var" => true,
105          "supports" => [ "title", "editor", "thumbnail" ],
106          "show_in_graphql" => false,
107      ];
108
109      register_post_type( "promo", $args );
110
111      add_action( 'init', 'cptui_register_my_cpts_promo' );
112

```

Y por último en este archivo también se añadió el código para crear el shortcode 'exclusive' que permite ocultar el contenido que se desee a cualquier usuario que no sea de tipo Promotor:

```

53  // Shortcode for exclusive content
54  add_shortcode( 'exclusive' , 'logged_in_content' );
55  function logged_in_content ( $atts , $content = null ) {
56      $user = wp_get_current_user();
57      if ( ! empty( $user ) && in_array( "promotor", (array) $user->roles ) ) {
58          return '<p>' . $content . '</p>';
59      }else return;
60
61  }

```

Para retocar la estética del botón login-logout que implementamos con la función anteriormente mencionada, en el archivo style.css hemos añadido un estilo básico según si el botón esta disponible para hacer un login o un logout:

```
14 .btn-menu a{
15     background: green;
16     padding:10px;
17     border-radius: 5px;
18     color:white;
19 }
20
21 .btn-menu a:hover,
22 .btn-menu a:focus{
23     text-decoration: none;
24     color:white;
25     background: gray;
26 }
27
28
29 .btn-menu.btn-login a{
30     background: green;
31 }
32
33 .btn-menu.btn-logout a{
34     background: darkred;
35 }
```

Para cada Promoción se ha creado el archivo single-promo.php donde hemos puesto en el código que si el ipo de post en es una Promoción, se muestre el titulo, la imagen, descripción y el resto de datos (campos personalizados) :

```
18 <?php
19 while ( have_posts() ) :
20     /*the_post();
21
22     do_action( 'storefront_single_post_before' );
23
24     get_template_part( 'content', 'single' );
25
26     do_action( 'storefront_single_post_after' );*/
27
28     the_post();
29
30     $type=get_post_type(get_the_ID());
31
32     if($type=='promo'){
33         the_title();
34
35         the_post_thumbnail();
36
37         the_content();
38         the_meta();
39     }
40
41 endwhile; // End of the loop.
42 ?>
```

Además, para la lista ordenada de promociones hemos creado un archivo llamado archive-promo.php a partir de archive.php, el cual hemos usado como plantilla paracrear la página de dicha lista. Aquí el cambio en el código ha sido sustituir el bucle por la llamada a la función mycp_popularity_list(), alojada en nuestro plugin mycountplugin.php que comentaré mas adelante.

```

26      <?php
27      //get_template_part( 'loop' );
28      mycp_popularity_list();

```

También tenemos un fichero llamado `page-insert-promo-restricted.php` que básicamente es donde se prepara el formulario para crear promociones y enviarlas a la base de datos para luego poder mostrarlas en nuestra página. Este archivo se utilizó de plantilla para la página de dicho formulario. Se declaran las variables para cada post y después se recogen en el formulario creado por html. No adjunto imagen de código porque es un archivo entero, pero está basado al 100% en código proporcionado por el profesorado de la asignatura.

Por último tenemos el archivo de nuestro plugin para contar visitas, `mycountplugin.php`, donde se ha modificado la función para mostrar una lista de promociones (ordenadas) pero que además no se muestren las promociones que hayan caducado:

```

58  /* Displays a list of posts ordered by post count */
59  function mycp_popularity_list($post_count = 10) {
60      $args = array(
61          "posts_per_page" => 10,
62          "post_type" => "promo",
63          "post_status" => "publish",
64          "meta_key" => "mycp_views",
65          "orderby" => "meta_value_num",
66          "order" => "DESC"
67      );
68
69      $mycp_list = new WP_Query($args);
70      if($mycp_list->have_posts()) { echo "<ul>"; }
71
72      //Fecha actual
73      //$fecha_actual = strtotime(date("d-m-Y H:i:00",time()));
74      $fecha_actual = date("Y")."-".date("m")."-".date("d");
75      $fecha_actual2 = date("Y").date("m").date("d");
76
77      while ( $mycp_list->have_posts() ) : $mycp_list->the_post();
78
79          // $key='fecha fin de oferta';
80          $fecha_promo=(get_post_meta(get_the_ID(), 'fecha_de_fin_de_oferta', true));
81
82          /*echo $fecha_promo;
83          echo "<br>";
84          echo $fecha_actual;*/
85
86          //Filtramos fecha
87          if($fecha_promo > $fecha_actual || $fecha_promo > $fecha_actual2){
88              echo '<li><a href="'.get_permalink($post->ID).'">'.the_title('','',false).'

```

Trabajo pendiente

En el tintero se ha quedado el plugin para hacer consultas en wikidata (punto 18), por falta de tiempo. El resto de puntos deberían estar implementados salvo algun fallito puntual que pueda surgir, en principio cumple todos los requisitos.