**Progetto promozione Turistica/Culturale su Venezia**

**Dellapasqua Lorenzo**

**IFTS 2022**

**Sunto**

Realizzazione di un progetto WordPress, a scopo di promozione culturale di Venezia, descrivendo informazioni relative a storia e cultura.

**Requisiti**

**Articoli Standard** devono contenere: titolo, contenuto, essere catalogati in categorie e avere tag.

**Custom Post** “monumento” devono contenere:

- Nome del monumento come titolo del post

- Breve testo come contenuto del Post

- Ubicazione o posizione su mappa (Opzionale tramite **custom field**)

- Anno realizzazione (Opzionale tramite **custom field**)

- Autore (Opzionale tramite **custom field**)

**Relazione**

**Installazione**

Ho scaricato **WordPress**(WP) nel mio computer e ho spuntato “scoraggiare i motori di ricerca dall’indicizzare il sito”. Infine ho creato un solo profilo di tipo **Administrator**.

**Soggetto e struttura**

Come soggetto ho scelto Venezia, ritenendola una candidata ideale per la prima traccia a scopo turistico

**Plugin**

Ho installato i plugin **PODS** e **WP Google Maps**, il primo per creare tipi di contenuto custom e il secondo per trovare poter inserire una mappa selezionando alcuni punti d’interesse. Infine ho eliminato i restanti plugin per ottimizzare il sistema e ho solo lasciato **Akismet Anti-Spam** in caso di modifiche future al progetto, le quali potrebbero includere commenti.

**Tema**

Come tema ho cercato qualcosa di visivo, provandone alcuni che ho preferito lasciare nella cartella temi per test futuri, nonostante non sia un “best practice”.

Ho usato il tema “**Mik**” ed ho creato subito una cartella figlio “**Mik-Child**” copiando i file **functions.php** e **style.css**.

In **functions.php** ho cancellato il codice e ho messo il testo qui riportato per connettere il tema figlio al padre

<?php

    //connetto theme -> theme-child

    add\_action('wp\_enqueue\_scripts', 'connectChildBlossomTravel');

    function connectChildBlossomTravel(){

    wp\_enqueue\_style('parent-style', get\_template\_directory\_uri().'/style.css');

    }

?>

In style.css ho cancellato il testo e ho inserito gli elementi descrittivi, in particolare quelli obbligatori: **Theme** **Name** e **Template**.

/\*!

//obbligatori Theme Name && Template

Theme Name: Mik child

Template: Mik

Theme URI:

Author: Lorenzo Dellapasqua

Author URI:

Description: custom Mik-Theme

Version: 2.1.1

Requires PHP: 5.6

Tested up to: 5.9

License: GNU General Public License v3 or later

License URI: http://www.gnu.org/licenses/gpl-3.0.html

Text Domain: mik

Tags: translation-ready, custom-header, custom-background, theme-options, custom-menu, threaded-comments, featured-images, footer-widgets, left-sidebar, right-sidebar, full-width-template, two-columns, three-columns, four-columns, grid-layout, custom-logo, blog, portfolio, photography

\*/

**Modifiche Back-End**

Ho copiato il file **single.php** e ho creato **single-monumento.php,** ho modificato il codice (**riga 34**) in modo che si potessero visualizzare i template di tipo custom “Monumento”

        while ( have\_posts() ) : the\_post();

//faccio visualizzare i template di tipo custom 'monumento'

            get\_template\_part( 'template-parts/content', 'monumento' );

Nella cartella **template-parts** ho modificato **content-single.php** creando il file **content-monumento.php** al fine di far visualizzare i meta-tag dei custom field.

            <?php

                the\_content();

                //Add sotto il content i custom fields -> output txt+get\_post\_meta?>

                <?php echo ("<br><b>Mappa: </b>".get\_post\_meta(get\_the\_ID(),'mappa', true));?> <!-- result field metatag -> id, str.k, >T.F.    -->

                <?php echo ("<br><b>Anno Realizzazione: </b>").(get\_post\_meta(get\_the\_ID(),'anno\_realizzazione', true));?> <!-- result field metatag    -->

                <?php echo ("<br><b>Periodo Storico: </b>").(get\_post\_meta(get\_the\_ID(),'periodo\_storico', true));?> <!-- result field metatag  -->

                <?php echo ("<br><b>Autore: </b>").(get\_post\_meta(get\_the\_ID(),'autore', true));?> <!-- result field metatag -->

Infine ho alterato il **Loop** copiando **Index.php** nel figlio, per far comparire nella Home i custom post.

<!-- Altero il loop per poter far comparire nella Home i custom Article tipo monumento

->

<?php

$args = array('posts\_per\_page' => 3,

    'post\_type' => 'monumento',

    'post\_status' => 'publish',

    'orderby' => 'rand' );

    $myquery = new WP\_Query($args);

    // if ($myquery->have\_posts()) {

    // while($myquery->have\_posts()) {

    // $myquery->the\_post();

    ?>

    <?php

                    if ( $myquery->have\_posts() ) :

                        /\* Start the Loop \*/

                        while ( $myquery->have\_posts()) : $myquery->the\_post('monumento');

                            get\_template\_part( 'template-parts/content', get\_post\_format('') );

                        endwhile; else :

                        get\_template\_part( 'template-parts/content', 'none' );

                    endif; ?>

</div><!-- #blog-posts-wrapper -->

**Contenuti**

Avendo scelto un tema molto visivo e tendente al minimale ho dovuto ricercare molte **immagini** ad alta risoluzione per evitare sgranature. In particolare ho creato un’immagine come background modificandone una presa da internet utilizzando un livello in trasparenza di bianco (tramite **Canvas**), al fine di ottimizzare la visualizzazione dei caratteri su sfondo bianco.

Ho inserito un **menu** al fine di agevolare la visibilità delle pagine all’utente.

All’apertura degli articoli normali e custom ho selezionato una **colonna laterale**, impostando delle immagini per selezionare gli **articoli** e ho modificato l’impaginazione delle **categorie**.

I **testi** sono stati prevalentemente scritti personalmente o lievemente modificati per motivi SEO nel riconoscimento di testi copiati di Google.

Inoltre ho provato ad inserire una **mappa**, ho optato per un plugin, ma non ho potuto selezionare GoogleMaps perché non sono riuscito ad ottenere gratuitamente l’uro delle loro API, ho quindi settato OpenLayers e inserito tre posizioni statiche su un’unica mappa.

A fine progetto ho **testato** la visualizzazione del sito tramite Mozzilla, Chrome, Edge.

Infine ho modificato la visualizzazione delle pagine, per modificare l’output evitando di far comparire alcune scritte nella home, ma ho poi **annullato tali modifiche** perché erano possibili solo attraverso il tema padre, e non sarebbe stata una pratica consigliabile modificarlo.

**Conclusioni**

Il progetto potrebbe ancora espandersi con contenuti, plugin e altri collegamenti, ma per il momento rispecchia le specifiche richieste. WordPress sembra essere un’ambiente di sviluppo molto utile per chi vuole creare velocemente prodotti complessi, accettando il compromesso di non poter facilmente modificare il codice ma lavorando prevalentemente sul front-end.

In particolare WordPress sembra essere impostato alla vendita di servizi con usufruire della versione PRO dei plugin utilizzati.