WeDevs Technical Test.

Objetivo del plugin

Crear un plugin que muestre un formulario a traves de un shortcode (xxxxx) para que el usuario complete y envie una aplicacion (de trabajo?). El formulario debe guardar los datos en una base de datos llamada XXXXXX previamente creada cuando se activa el plugin por primera vez. A su vez debe enviar un mail de confirmacion cuando el usuario envie el formulario de forma correcta.

En el backend el admin podra entrar a ver todas las aplicaciones desde una opcion en el menu, podra ordenarlas por orden ascendente o descendente (elegi solo dos lugares como determinantes para el orden, el apellido y la hora de subida), podra buscar en cualquier campo y borrar tanto individual como grupalmente las aplicaciones.

Por ultimo el administrador tendra acceso a un widget en el dashboard que mostrara las ultimas 5 aplicaciones enviadas.

This plugin only has two files, ap\_wedevs.php and uninstall.php

**ap\_wedevs.php -** Tiene todas las funciones para el funcionamiento de todo el plugin.

**uninstall.php -** Se ejecutara cuando se borre el plugin, el archivo se encargara de borrar la tabla generada.

Funciones creadas para cada objetivo del plugin

**Al activar el plugin:**

Al activar pro primera vez el plugin, a traves del register\_activation\_hook se ejecutara la funcion ap\_wedevs\_create\_db.

A su vez la linea de codigo con el hook add\_action('admin\_menu', 'ap\_wedevs\_add\_menu'); agregara la opcion de tener acceso a una opcion en el menu del dashboard con acceso a la pagina donde veremos todas las aplicaciones.

Por ultimo usamos dos require\_once, tanto para la clase Wp List Table como para file.php. El primero lo utilizaremos para el manejo y la impresion en pantalla de todas las aplicaciones, el segundo sera necesario para que en el formulario el usuario pueda subir su archivo.

Para poder utilizar luego el shortcode [applicant\_form], utilice el hook add\_shortcode, en la linea add\_shortcode('applicant\_form', 'register\_ap\_wedevs\_shortcode');. Este llamara a la funcion function register\_ap\_wedevs\_shortcode() cada vez que encuentre en una publicacion el shortcode requerido..

Por ultimo el hook para ejecutar y tener el widget en el dashboard es el add\_action('wp\_dashboard\_setup', 'wedevs\_add\_widget');. Este hook llama a la funcion wedevs\_add\_widget que sera la encargada del widget per se.

**Creacion de la BD**

**function ap\_wedevs\_create\_db**

La funcion ap\_wedevs\_create\_db creara una tabla dentro de la base de datos, de nombre applicant\_submissions, sin prefijo y con 9 columnas, estas son:

id, mediumint (9) con auto incremento.

time, datetime DEFAULT '0000-00-00 00:00:00'

firstname, varchar(25)

lastname, varchar(25)

presentaddress, varchar(100)

emailaddress, varchar(100)

mobile, varchar(20)

postname, varchar(55)

cv, varchar(255)

Cada columna guardara los registros, uno por fila, de cada aplicacion submitted. Elegi este diseño para guardar los datos en la DB ya que era el mas practico en funcion del tiempo que tenia para hacer el test. Otra forma optima era iterando en un array con solo dos columnas por applicant donde en la columna mas importante estaria guardandotodos los datos en un array en una simple celda.

**Funciones encargadas del funcionamiento del frontend**

Aqui estan todas las funciones utilizadas para el manejo de la impresion del formulario en pantalla y el funcionamiento del mismo tanto en el front, como en el backend.

**function print\_app\_form\_wedevs**

The "app\_form\_wedevs" function is the one that shows the form that needs to be completed on the page where the shortcode is placed.

En la misma empezamos definiendo las variables de cada campo del formulario. En la linea 74 agregamos un if ($completed) que su mision es limpiar el formulario, todos los campos en blanco, en caso de que se haya subido uno previamente de forma correcta. A su vez imprimira un mensaje donde nos notificara que la aplicacion fue enviada y si el mail se envio correctamente, tambien imprimira un mensaje que el mail fue enviado correctamente. Caso contrario tambien lo especificara con un mensaje.

Desde la linea 103 a la 141 le damos el estilo al formulario.

En la linea 144 tenemos un if (is\_wp\_error($wp\_error)) donde estamos pidiendo que en el caso de encontrar un error lo imprima en pantalla.

A partir de la linea 155 tenemos todo lo referido al codigo para imprimir el formulario. No hay nada muy notorio ni complicado en esto. Si debo aclarar que todos los requisitos o limites de cada formulario los controlo tanto en el frontend como en el backend. Por ejemplo, en el frontend utlilizo la palabra “required” en el codigo para decir que un campo es requerido, pero ademas, mas adelante en la funcion explicada a continuacion muestro como pido esos mismos requisitos en el backend.

**function registration\_validation**

//This function check if all the form fields are ok and continue with the submission

Esta funcion es la que controla que todos los limites del formulario, que fueron establecidos previamente, se cumplan. Ellos son si el campo requerido esta completo (esto lo hacemos utilizando el “empty”), si la direccion de email ingresada es correcta y si los campos estan dentro del limited de caracteres asignados. Ante la libertad en la tarea, todo esto lo decidi por mi parte.

**function complete\_registration**

//"complete\_registration" If everything is correct, this function add the records to the table

Luego de comprobar que no hay ningun error en los campos, se ejecutara esta funcion por orden la funcion register\_ap\_wedevs\_shortcode, luego de haber apretado el boton Submit. En la misma guardamos el archivo en el directorio “uploads/year/month” y guardamos los datos en la base de datos mediante un insert. Utilizamos “insert” y no “query” por varios motivos, el fundamental es evitar la inyeccion de codigo en la DB.

function deliver\_mail($emailaddress)

//IMPORTANT! this function doesnt work in localhost because wp\_mail doesnt work in localhost.

Esta es la funcion que se encarga de enviar el mail a traves de wp\_mail. Esta funcion no funciona en el localhost.

**function register\_ap\_wedevs\_shortcode**

//register the shortcode and check the form for errors and submission

Esta es la funcion principal del frontend y sus acciones. Cuando es llamada a traves del shortcode, llama a print\_app\_form\_wedevs. A su vez, cuando apretamos el boton de submit es la que llama a la funcion registration\_validation(), sanitiza la informacion de cada campo y llama a la funcion complete\_registration. Por ultimo, al comprobar que esta todo bien llama a la funcion deliver\_mail.

**Funciones y clase encargadas del funcionamiento del backend**

A continuacion la clase y todas las funciones para tener acceso a la pagina en el dashboard donde veremos todas las aplicaciones enviadas.

**class wedevsListTable extends WP\_List\_Table**

// Extending class

Para crear y mostrar los datos en el admin utilizamos y extendimos la clase WP\_List\_Table de Wordpress con varias funciones.

**function get\_table\_data**

// Get table data

Esta funcion nos mostrara todos los datos de las aplicaciones si el campo busqueda esta vacio. Si el campo busqueda tiene algun caracter, decidi hacer la busqueda de ese string en todos los campos de datos disponibles (firstname, lastname, etc), esto es facilmente modificable eliminando cualquiera de las lineas que van desde las 362 a la 368.

**public function get\_columns**

// Define table columns

Imprime los nombres de cada columna y establece cuales seran los datos que debera mostrar abajo de cada columna.

**public function delete\_entry**

// Delete a entry record.

Utilizamos esta funcion para borrar una entrada unica (desde la opcion del menu que aparecera abajo del nombre)

**function prepare\_items**

// Bind table with columns, data and all

Recupera desde la base de datos, toda la informacion que va a mostrar en la pagina de administracion.

**public function column\_cb**

//Fix the checkbox in every file

Esta funcion esta creada para poder tener un checkbox al principio de cada linea para poder seleccionar luego las bulk actions

**function get\_sortable\_columns**

// let me choose what items are selected for the sort

Decidi solo utilizar dos columnas para ordenar como ejemplo, la de time y la de lastname. Si deseamos tener mas parametros para ordenar las filas, podemos agregar aqui una linea extra por parametro. Por ejemplo si quisiera poder clickear en firstname y ordenar por nombre toda la tabla, deberia agregar una linea similar a 'firstname'=> array('firstname', false),

**function usort\_reorder**

// Sorting function

Establece un orden por defecto (en este caso decidi que fuera por time).

**function column\_firstname**

// Adding action links to column

Esta es la funcion que nos permite tener lo que se llama action links para cada campo. En este caso decidi que solo se pudiera tener un action link (Delete) y que solo apareciera debajo del First Name de cada fila. Si se toca en Delete, se borrara toda la fila de la tabla en la base de datos.

**public function get\_bulk\_actions**

// To show bulk action dropdown

Desplega las opciones que hayamos elegido, en nuestro caso unicamente Delete. Al clickear en los checkbox y luego seleccionar la opcion Delete, llamara a la funcion correspondiente para borrar todos los datos seleccionados de la Base de Datos.

**function process\_bulk\_action**

//As the name says, this function process the bulk action

**function process\_single\_action**

//As the name says, this function process the single action

**function ap\_wedevs\_admin\_page**

// Plugin menu callback function

Esta funcion es la que maneja la pagina en el administrador a traves de la clase creada previamente y de 3 funciones.

**Funciones encargadas del funcionamiento del menu**

**function ap\_wedevs\_add\_menu**

//Admin menu setting

Es la encargada de crear la opcion en el menu del dashboard y de llamar a la funcion previamente nombrada.

**Funciones encargadas del funcionamiento del widget**

**function wedevs\_add\_widget**

//add Widget

Agrega un widget vacio, con un titulo en el dashboard y llama a la funcion siguiente.

**function wedevs\_show\_widget**

Muestra la informacion del widget. Para mostrar los ultimos 5 resultados, hago una llamada a la bd a traves de la linea SELECT \* FROM $table\_name ORDER BY time DESC LIMIT 5.

Luego lo imprimo en la pantalla a traves del codigo que viene a continuacion en el echo.

Author: Ignacio Chitnisky

Email: ignacio @ chitnisky . com

Date: November 2022