**Cahier de spécifications fonctionnelles**

**PROJET:** CPT skills



**Table des matières**

[**Introduction et cadre du projet**](#_heading=h.yqos01k19swa) **3**

[Résumé du projet](#_heading=h.1fob9te) 3

[Contexte et intervenant](#_heading=h.3znysh7) 4

[Objectifs](#_heading=h.2et92p0) 4

[Enjeux](#_heading=h.tyjcwt) 4

[Livrables](#_heading=h.3dy6vkm) 4

[Critères de succès mesurables](#_heading=h.1t3h5sf) 4

[Planning](#_heading=h.4d34og8) 4

[Risque lié au projet](#_heading=h.2s8eyo1) 5

[**Fonctionnalités**](#_heading=h.sqh6k4x5fokn) **6**

[Description du projet](#_heading=h.3t3pmk3nkl23) 6

[Étapes du projet](#_heading=h.ld7brig53ui5) 8

[Lire les quelques ressources - 14.06](#_heading=h.z2uj7sx4txoa) 8

[Création du plugin (à faire ensemble) - 15.06](#_heading=h.v641h8nfcdmg) 8

[CPT - Custom Post Type - 15.06](#_heading=h.mdqv83fw4t47) 8

[Shortcode et Template - 16.06 au 24.06](#_heading=h.qkpmfjfrezup) 10

[Widget - 16.06 au 24.06](#_heading=h.duvegcuxues4) 13

[Wireframe - 16.06](#_heading=h.9uyhe31djzed) 15

[Shortcode Galerie](#_heading=h.omug9giykpv8) 15

[Shortcode Profil By ID](#_heading=h.omug9giykpv8) 15

[Page Profil unique](#_heading=h.omug9giykpv8) 15

[Shortcode 3 Last Profils](#_heading=h.omug9giykpv8) 15

[Shortcode Slider](#_heading=h.omug9giykpv8) 15

[Widget (bloc) Profil aléatoire](#_heading=h.omug9giykpv8) 15

[Widget (bloc) Recherche](#_heading=h.omug9giykpv8) 15

[Use case diagram - 16.06](#_heading=h.3cyle78u8yhj) 16

[CPT dans le dashboard](#_heading=h.gn9mzljq0z9c) 16

[Shortcode Galerie](#_heading=h.h2rmelm0zdm5) 16

[Shortcode 3 Last Profils](#_heading=h.vp9dmd6vhvnp) 16

[Shortcode Slider](#_heading=h.dly0e7uklfj2) 16

[Widget (bloc) Profil aléatoire](#_heading=h.363yakltkq1s) 16

[Widget (bloc) Recherche](#_heading=h.omug9giykpv8) 16

[**Limitations et contraintes**](#_heading=h.2xcytpi) **16**

[**Ressources**](#_heading=h.mjf1bke9kk63) **17**

[Les hooks](#_heading=h.27eb3frz07f3) 17

[Créer un plugin](#_heading=h.27eb3frz07f3) 17

[CPT](#_heading=h.jcftgtokft9m) 17

[Metaboxes](#_heading=h.njf0ocbvr8ay) 17

[Taxonomy](#_heading=h.xufmo7cu4z4b) 17

[Shortcode](#_heading=h.k7yts16pnkql) 18

[Template CPT](#_heading=h.omzpxjky8cj4) 18

[Widget](#_heading=h.43aaxshxhayt) 18

[Widget Zone et création de widget](#_heading=h.57g1cbr8tcgm) 18

[**Compétences et savoir-faire visé**](#_heading=h.r72q1gxr8rtv) **19**

**WORDPRESS**

**CONNECTION**

**Titre du site:** CPT-skills **/** CPTskills

**Nom du d’utilisateur:** gabsvn **/** gabs **/** *bavardages*

**Mot De Passe:** Jonxion1205 **/** qmbwDqJk4fHawz&mNk

**E-mail:** j.gabriel.fds.1@gmail.com

**Pour se connecter au WP :**

**Url:** http://localhost:8600/wp-admin/

**Host :** portainer.realise.ch (ou 192.168.150.11)

**Titre du site:** CPT-Skills

**User :** "diago"; "diogo" ; aniello ; "olivier" ; "**gabs**"

**Mdp :** dYHKXazY9SCkIPfnXj

**E-mail:** j.gabriel.fds.1@gmail.com

**Port :** 8600

**Pour se connecter au PhpMyAdmin :**

**Url:** http://portainer.realise.ch:19007/

**login** **:** CF

**mdp** **:** digital2022

**Pour se connecter au FTP :**

**Host :** portainer.realise.ch (ou 192.168.150.11)

**User :** CF

**Mdp :** digital2022

**Port :** 18007

# Introduction et cadre du projet

## Résumé du projet

Nos portfolios ont plu au client. Il voudrait proposer cette solution à certains de leurs client·e·s indépendant·e·s et aussi pour lui-même.

Il trouverait pratique de pouvoir gérer les compétences via le menu WP pour chaque profil. Il voudrait aussi un système de navigation par profils sur une galerie permettant d’afficher le détail d’une personne et ses compétences.

Ce qu’il voudrait donc, ce serait d’avoir un plugin à commercialiser qui permet d’implémenter des portfolios sur n’importe quel site WP avec une gestion des compétences.

## Contexte et intervenant

Nous avons obtenu le partenariat avec la boîte de com qui a bien aimé nos portfolios et a une première idée de projet.

## Objectifs

Avoir un plugin fonctionnel en **stand-alone** et commercialisable. Utilisable avec les portfolios.

## Enjeux

Nous permettre de consolider notre partenariat et d’avoir un premier produit. Le plugin devrait aussi permettre la facilitation de la création de porfolios.

## Livrables

* Ce cahier des spécifications fonctionnelles rempli et complété
* Les fichiers du plugin sur GitHub commenté
* Un fichier readme sur GitHub

## Critères de succès mesurables

Le plugin est fonctionnel et permet d’ajouter un menu dans le dashboard permettant la gestion des compétences

Le client est satisfait et une première communication sur le plugin est lancé

## Planning

Projets à finir le **vendredi 24 juin** matin et à présenter devant le product owner le **25 juin**

* **Mardi 14 juin**, analyse du cahier des spécifications, lecture des liens (étape 1)
* **Mercredi 15 juin,** création du repo GitHub, de l’arborescence des fichiers, du plugin et du CPT (étape 2 et 3) (+ first commit)
* **Jeudi 16 juin,** création des wireframe et use case diagram (**étape** **4** et **5**)
* **Vendredi 17 juin,** création des squelettes HTML et des CSS avec des commentaires. Il faut aussi mettre des commentaires pour expliquer où mettre le code PHP de WordPress et ce code va faire retourner (ne pas oublier les commit) (**étape 4** et **5**)
* **À partir du lundi 20 juin,** codage PHP en WordPress des différents éléments (**étape** **4** et **5**)
* **Vendredi 24 juin après-midi,** finalisation du commentaire du code, merge, préparation de la présentation et écriture du fichier readme sur GitHub
* **Lundi 25 juin matin 9h,** présentation du projet au product owner

## Risque lié au projet

* Le CPT n’est pas fonctionnel et il n’est pas possible de l’utiliser et l’implémenter sur les sites portfolio des clients
* Retard sur les sites portfolio des clients
* Perte de confiance avec l’entreprise partenaire

# Fonctionnalités

## Description du projet

Via le dashboard, on va pouvoir créer le profil d’une personne avec un **prénom**, **nom**, une **phrase d’accroche**/**citation**, un **descriptif** et une **photo**.

Pour chaque profil, on doit pouvoir aussi sélectionner trois **types** de savoir :

* Savoir (**connaissance**)
* Savoir faire (**compétence**)
* Savoir être (**softskill** ou **compétence transverse**)

Il est possible d’en ajouter et d’en sélectionner plusieurs de chaque.

Pour chaque profil, on peut ajouter et sélectionner un ou plusieurs intitulé(s) et/ou sous-intitulé(s). **Par exemple**, son métier et sa spécialisation dans ce métier.

Et enfin, sur chaque profil, on peut ajouter et sélectionner un ou plusieurs **secteur**(s) d’activité et **sous-secteur**(s) d’activité.

Un **shortcode** permettra d’afficher *tous les* ***profils*** dans une galerie avec photo, **nom** et **prénom** et **son** **intitulé**. Lorsque l’on cliquera sur un profil, on aura accès aux détails de celui-ci. On pourra filtré cette galerie par **savoirs** et **secteur** d’activités.

On aura aussi un **shortcode** qu’on peut placer sur n’importe quelle page en spécifiant l’**id** du profil pour afficher celui-ci sur cette page.   
Il faudra aussi travailler sur la **page single-{“Nom du CPT”}.php** pour avoir un affichage correct d’un profil lorsqu’on navigue vers cette page. En clair, créer ***un template*** pour afficher un ***profil unique***.

Un autre **shortcode** aussi qui affichera les ***trois derniers profils* ajoutés.** Le **premier** aura une photo **en grand** et **les deux autres** seront en **petits** dessous. Lorsqu’on passe la souris dessus, sera affiché sur la photo : le **nom** et **prénom**, son **intitulé** et sa p**hrase d’accroche** ainsi qu’un **bouton “Voir les détails”**.

Un dernier **shortcode** permettra d’intégrer un ***slider*** avec une **photo des profils** et afficher dessus le **nom** et **prénom**, son **intitulé** et **sa phrase d’accroche** ainsi qu’un **bouton “Voir les détails”**. On devrait pouvoir choisir le nombre de slide et des filtres par intitulé et secteur d’activité.

Il faudrait encore pouvoir avoir sur **chaque page** dans une **barre latérale droite**, un **profil** **aléatoire** qui s’affiche **avec** sa **photo**, **nom** et **prénom**, son **intitulé** et sa **phrase** **d’accroche**. Ceci devrait être géré via **un widget (un bloc)**. Lorsque l’on cliquera sur le **profil**, on aura accès aux détails de celui-ci.

Et enfin, il faudra aussi un ***widget* (un bloc)** permettant d’effectuer une ***recherche de profils*** via plusieurs critères sélectionnés.

## Étapes du projet

### Lire les quelques ressources – **14.06**

* + - * 1. Pour tous : les CPT, Créer un plugin et les hooks en tout cas

### Création du plugin (à faire ensemble) – **15.06**

* + - * 1. Lecture des liens dans ressources
        2. Choix du nom, logo, etc, du plugin. Préparation de l'arborescence du projet

**Nom:** CPTplugin.php

**Logo:**

**Plugin:**

* + - * 1. Création des différents fichiers qui seront utilisés dans le projets

CPTplugin.php

Plugin

Templates

* + - * 1. Écriture de l’entête du plugin dans **nom-du-plugin.php**

***Plugin Name****: CPTplugin.php*

***Plugin URI****:* [*http://rachelmccollin.com*](http://rachelmccollin.com/)

***Description****: Plugin to register the Profile Custom post type*

***Version****: 0.1.*

***Author****: Gabriel\_FDS*

***Author*** *URI:*[*http://réalise.c*](http://rachelmccollin.com/)*h*

***Textdomain****: GABSVN*

***License****: GPLv2*

* + - * 1. Commentaires dans les fichiers pour définir où placer quel code

*/\*Définir notre fonction pour créer notre CPT\*/*

*/\*Définir les UI labels / les OPTIONS / les étiquettes pour notre type de message personnalisé\*/*

*/\*Définir les arguments pour notre type de message personnalisé\*/*

*/\*Enregistrer notre CPT\*/*

*/\**Accrochez-vous à l'action 'init' pour que la fonction - Contenant notre enregistrement de type de publication ne soit pas - inutilement exécutée*\*/*

* + - * 1. Création d’un repo git
        2. Premier commit du dossier plugin avec les sous-dossiers et les fichiers
        3. Lancement d’un docker-compose en local pour WP pour tester le plugin
        4. Activation du mode debug dans le fichier wp-config.php

### CPT - Custom Post Type - **15.06**

* + - 1. CPT  
         1. Réflexion et choix des différents termes à utiliser
         2. Création du code pour implémenter le **CPT** (back)

***function register\_cpt\_produit() {***

**$labels = array(**

'name' => \_x('Produits', 'produit'), 'singular\_name' => \_x('Produit', 'produit'), 'add\_new' => \_x('Ajouter', 'produit'), 'add\_new\_item' => \_x('Ajouter un produit', 'produit'),

'edit\_item' => \_x('Editer un produit', 'produit'),

'new\_item' => \_x('Nouveau produit', 'produit'), 'view\_item' => \_x('Voir le produit', 'produit'),

'search\_items' => \_x('Rechercher un produit', 'produit'),

'not\_found' => \_x('Aucun produit trouvé', 'produit'),

'not\_found\_in\_trash' => \_x('Aucun produit dans la corbeille', 'produit'),

'parent\_item\_colon' => \_x('Produit parent :', 'produit'),

'menu\_name' => \_x('Produits', 'produit'),

**);**

**$args = array(**

'labels' => $labels,

'hierarchical' => false,

'description' => 'Les produits de ma boutique.',

'supports' => array('title', 'editor', 'thumbnail', 'custom-fields', 'revisions'), 'taxonomies' => array('category', 'post\_tag'), 'public' => true,

'show\_ui' => true, 'show\_in\_menu' => true, 'menu\_position' => 5,

'show\_in\_nav\_menus' => true, 'publicly\_queryable' => true, 'exclude\_from\_search' => false,

'has\_archive' => true,

'query\_var' => true,

'can\_export' => true,

'rewrite' => true,

'capability\_type' => 'post'

**);**

**register\_post\_type('produit', $args);**

***}***

**add\_action('init', 'register\_cpt\_produit', 10);**

* + - * 1. Création du code permettant l’affichage des champs du **CPT** dans le **dashboard** (~front) (**meta boxes**)
        2. Création du code permettant d’enregistrer les valeurs des champs (back)
        3. Création du code vérifiant que les bonnes valeurs ont été entrées dans les champs (front-back)
        4. Commentaires
        5. Un mot clef à comprendre : **register\_post\_type() ;**
      1. Taxonomy Savoir / Connaissances
         1. Réflexion et choix des différents termes à utiliser
         2. Choix du type de taxonomy (**hierarchical** or **not**)
         3. Création du code
         4. Commentaires
         5. Deux mot clef à comprendre : **register\_taxonomy()** ; **/ the\_terms()** ;
      2. Taxonomy Savoir faire / Skills
         1. Réflexion et choix des différents termes à utiliser
         2. Choix du type de taxonomy (**hierarchical** or **not**)
         3. Création du code
         4. Commentaires
         5. Deux mot clef à comprendre : **register\_taxonomy()** ; **/ the\_terms()** ;
      3. Taxonomy Savoir être / Soft Skills
         1. Réflexion et choix des différents termes à utiliser
         2. Choix du type de taxonomy (hierarchical or not)
         3. Création du code
         4. Commentaires
         5. Deux mot clef à comprendre : **register\_taxonomy()** ; **/ the\_terms()** ;

### Shortcode et Template - **16.06** au **24.06**

Quelques mots clefs à comprendre :

* **WP\_Query();** //Retourne une liste d’objet par rapport aux paramètres
* **have\_posts()**; //Méthode de classe pour vérifier la présence d’élément de type post
* **the\_post()**; //Permet de passer au post suivant.   
   Dans la loop WP, cela permet de parcourir tous les post un à un
* **get\_post\_meta()** ; // Retourne les informations du champ pour un post donné
* **get\_the\_ID()** ; //Retourne l’id du post en cours dans la loop
* **The loop** ; //- <https://codex.wordpress.org/The_Loop>
* **the\_title()**; //Retourne le titre du post en cours
* **the\_excerpt();** //Retourne le résumé du texte (content) du post en cours
* **wp\_reset\_postdata();** //à faire après un loop pour réinitialiser le curseur de la variable globale post
* **add\_shortcode( );** //Permet de définir le nom du shortcode et quelle fonction sera lancée quand ce nom sera détecté (quand le shortcode sera utilisé dans WP)  
    
    
  + - 1. Shortcode Galerie   
         1. Wireframe - **16.06**
         2. Squelette HTML - **17.06** / CSS - **17.06**
         3. Code shortcode en PHP - **20.06** au **24.06**
         4. Commentaires - **17.06** + **24.06**
      2. Shortcode Profil by ID   
         1. Wireframe - **16.06**
         2. Squelette HTML - **17.06** / CSS - **17.06**
         3. Code shortcode en PHP - **20.06** au **24.06**
         4. Commentaires - **17.06** + **24.06**

* + - 1. **Template d’affichage de profil unique**

Quelques mots clefs à comprendre :

* **single\_template** /Le hook à utiliser pour lancer notre fonction qui ira chercher notre template plutôt que la page par défaut
* Entête du Template :

/\*

\* **Template Name:** NomDuTemplate  
 \* **Template Post Type:** post, page, product

\*/

* **get\_header();** //Retourne l’entête HTML utilisée sur les pages
* **get\_footer();** //Retourne le pied de page HTML utilisée sur les pages
* Vous pouvez aussi ouvrir le fichier **single.php** de votre thème et vous en inspirer  
  + - * 1. Wireframe - **16.06**
        2. Créer un fichier avec le nom du template

*GABSVNSingleRandomTemplate.php*

* + - * 1. Faire l’appel de la fonction qui ira chercher le chemin vers ce fichier quand le hook **single\_template** sera rencontré

//DESCRIPTION :

add\_filter( 'single\_template' , [$pp\_event\_single\_template](http://hookr.io/filters/single_template/) , [$int](http://hookr.io/filters/single_template/) );

//PARAMETRES :

**$pp\_event\_single\_template**(chaîne)

=>'pp\_event\_single\_template'

//Le modèle unique d'événement pp.

**$int**(entier) =>16

//L'int.

//DEFINI :

[/inc/pp-events-templates.php](http://hookr.io/plugins/buddypress-simple-events/2.2.3/files/inc-pp-events-templates/)

add\_filter( 'single\_template', 'pp\_event\_single\_template', 16 );

*//Pour exécuter le crochet, copiez l'exemple ci-dessous.*

$pp\_event\_single\_template = apply\_filters( 'single\_template' , $pp\_event\_single\_template , $int );

if ( !empty( $pp\_event\_single\_template ) ) {

// tout a mené jusqu'à ce point...

}

*//L'exemple suivant concerne l'ajout d'un rappel de hook.*

// définit le callback single\_template

function filter\_single\_template ( $pp\_event\_single\_template , $int ) {

// faites en sorte que la magie du filtre se produise ici...

retourne $pp\_event\_single\_template ;

} ;

// ajoute le filtre

add\_filter( 'single\_template' , 'filter\_single\_template' , 10 , 2 );

Pour supprimer un rappel de hook, utilisez l'exemple ci-dessous.

// supprimer le filtre

remove\_filter( 'single\_template' , 'filter\_single\_template' , 10 , 2 );

* + - * 1. Squelette HTML - **17.06** / CSS – **17.06**

****

***HTML :***

**<!-- wp:template-part {"slug":"header","tagName":"header"} /-->**

<!-- wp:group {"tagName":"main"} -->

<main class="wp-block-group"><!-- wp:group {"layout":{"inherit":true}} -->

<div class="wp-block-group"><!-- wp:post-title {"level":1,"align":"wide","style":{"spacing":{"margin":{"bottom":"var(--wp--custom--spacing--medium, 6rem)"}}}} /-->

<!-- wp:post-featured-image {"align":"wide","style":{"spacing":{"margin":{"bottom":"var(--wp--custom--spacing--medium, 6rem)"}}}} /-->

<!-- wp:separator {"align":"wide","className":"is-style-wide"} -->

<hr class="wp-block-separator alignwide is-style-wide"/>

<!-- /wp:separator --></div>

<!-- /wp:group -->

<!-- wp:spacer {"height":32} -->

<div style="height:32px" aria-hidden="true" class="wp-block-spacer"></div>

<!-- /wp:spacer -->

<!-- wp:post-content {"layout":{"inherit":true}} /-->

<!-- wp:spacer {"height":32} -->

<div style="height:32px" aria-hidden="true" class="wp-block-spacer"></div>

<!-- /wp:spacer -->

<!-- wp:group {"layout":{"inherit":true}} -->

<div class="wp-block-group"><!-- wp:group {"layout":{"type":"flex"}} -->

<div class="wp-block-group"><!-- wp:post-date {"format":"F j, Y","style":{"typography":{"fontStyle":"italic","fontWeight":"400"}},"fontSize":"small"} /-->

<!-- wp:post-author {"showAvatar":false,"fontSize":"small"} /-->

<!-- wp:post-terms {"term":"category","fontSize":"small"} /-->

<!-- wp:post-terms {"term":"post\_tag","fontSize":"small"} /--></div>

<!-- /wp:group -->

<!-- wp:spacer {"height":32} -->

<div style="height:32px" aria-hidden="true" class="wp-block-spacer"></div>

<!-- /wp:spacer -->

<!-- wp:separator {"className":"is-style-wide"} -->

<hr class="wp-block-separator is-style-wide"/>

<!-- /wp:separator -->

<!-- wp:post-comments /--></div>

<!-- /wp:group --></main>

<!-- /wp:group -->

<!-- wp:template-part {"slug":"footer","tagName":"footer"} /-->

***CSS :***

{

"version": 2,

"title": "Swiss",

"settings": {

"color": {

"duotone": [

{

"colors": [

"#000000",

"#FFFFFF"

],

"slug": "default-filter",

"name": "Default filter"

}

],

"palette": [

{

"slug": "foreground",

"color": "#FFFFFF",

"name": "Foreground"

},

{

"slug": "background",

"color": "#1A1A1A",

"name": "Background"

},

{

"slug": "primary",

"color": "#FF7179",

"name": "Primary"

},

{

"slug": "secondary",

"color": "#F4F4F2",

"name": "Secondary"

},

{

"slug": "tertiary",

"color": "#0000000",

"name": "Tertiary"

}

]

},

"custom": {

"spacing": {

"small": "max(1.25rem, 5vw)",

"medium": "clamp(1.75rem, 6vw, calc(3 \* var(--wp--style--block-gap)))",

"large": "clamp(3.5rem, 8vw, 6rem)",

"outer": "var(--wp--custom--spacing--small, 1.25rem)"

},

"typography": {

"font-size": {

"colossal": "clamp(3.5rem, 8vw, 5rem)"

}

}

},

"typography": {

"fontFamilies": [

{

"fontFamily": "\"Inter\", sans-serif",

"name": "Inter",

"slug": "inter",

"fontFace": [

{

"fontFamily": "Inter",

"fontWeight": "200 900",

"fontStretch": "normal",

"src": [ "file:./assets/fonts/inter/Inter.ttf" ]

}

]

}

]

}

},

"styles": {

"blocks": {

"core/cover": {

"filter": {

"duotone": "var(--wp--preset--duotone--default-filter)"

}

},

"core/image": {

"filter": {

"duotone": "var(--wp--preset--duotone--default-filter)"

}

},

"core/post-author": {

"filter": {

"duotone": "var(--wp--preset--duotone--default-filter)"

}

},

"core/post-featured-image": {

"filter": {

"duotone": "var(--wp--preset--duotone--default-filter)"

}

},

"core/post-title": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"core/query-title": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)"

}

},

"core/site-logo": {

"filter": {

"duotone": "var(--wp--preset--duotone--default-filter)"

}

},

"core/site-title": {

"typography": {

"fontWeight": "700",

"fontStyle":"normal"

}

}

},

"elements": {

"h1": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"h2": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"h3": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"h4": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"h5": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"h6": {

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)",

"fontWeight": "700"

}

},

"link": {

"color": {

"text": "var(--wp--preset--color--primary)"

}

}

},

"typography": {

"fontFamily": "var(--wp--preset--font-family--inter)"

}

}

}

* + - * 1. Insertion du code PHP de WP pour contruire la page (ex: **get\_header**) et aussi qui permet de récupérer les données pour remplir la page (données du profil) 

<html>

<body>

**<?php**include("**GABSVN-header.php**"); ?>

<h1>Page d’accueil </h1>

<p>Bonjour c’est la page d’accueil de Portfolios by GABSVN merci pour la visite</p>

*// Quitte si accès direct*

if ( ! defined( 'ABSPATH' ) ) exit;

*// charge un modèle d'événement unique à partir d'un thème ou d'un plugin*

function pp\_event\_single\_template( $single\_template ) {

global $post ;

if ( $post->post\_type == 'événement' ) {

$theme\_template = locate\_template( 'single-event.php' );

if( file\_exists( $theme\_template ) )

$single\_template = $theme\_template ;

autre

$single\_template = PP\_EVENTS\_DIR . '/templates/single-event.php';

}

retourne $single\_template ;

}

add\_filter( 'single\_template', 'pp\_event\_single\_template', 16 );

*// charge le modèle de boucle d'événement à partir du thème ou du plugin*

function pp\_events\_template\_include( $template ) {

globale $wp\_query ;

si( ! bp\_is\_user() ) {

if( isset( $wp\_query->post->post\_title ) ) {

$page\_title = $wp\_query->post->post\_title ;

if ( $page\_title == \_\_( 'Events', 'bp-simple-events' ) ) {

$theme\_template = locate\_template( 'events-loop.php' );

if( file\_exists( $theme\_template ) )

$template = $theme\_template ;

autre

$template = PP\_EVENTS\_DIR . '/templates/events-loop.php';

}

}

}

retourner $modèle ;

}

add\_action( 'template\_include', 'pp\_events\_template\_include' );

*// modèles de profil*

fonction pp\_events\_register\_template\_location() {

renvoie PP\_EVENTS\_DIR . '/modèles/' ;

}

function pp\_events\_template\_start() {

if( function\_exists( 'bp\_register\_template\_stack' ) )

bp\_register\_template\_stack( 'pp\_events\_register\_template\_location' );

add\_action( 'bp\_init', 'pp\_events\_template\_start' );

**<?php** include("**GABSVN-footer.php**"); ?>

</body>

</html>

---------------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------

<html>

<body>

**<?php** include("**GABSVN-header.php**"); ?>

<h1>Page d’accueil </h1>

<p>Bonjour c’est la page d’accueil de Portfolios by GABSVN merci pour la visite</p>

**<?php** include("**GABSVN-footer.php**"); ?>

</body>

</html>

* + - * 1. Commentaires - **17.06** + **24.06** 

*/\**

*\* Template Name: GABSVN*

*\* Template Post Type: post, page, product*

*\*/*

*/\* Start the Loop \*/*

*// Parent post navigation.*

*/\* translators: %s: Parent post link. \*/*

*// If comments are open or there is at least one comment, load up the comment template.*

*// Previous/next post navigation.*

*// End of the loop.*

* + - 1. Shortcode 3 Last Profils  
         1. Wireframe - **16.06**
         2. Squelette HTML - **17.06** / CSS - **17.06**
         3. Code shortcode en PHP - **20.06** au **24.06**
         4. Commentaires - **17.06** + **24.06**
      2. Shortcode Slider  
         1. Wireframe - **16.06**
         2. Squelette HTML - **17.06** / CSS - **17.06**
         3. Code shortcode en PHP - **20.06** au **24.06**
         4. Commentaires - **17.06** + **24.06**

### Widget - **16.06** au **24.06**

Quelques mots clefs à comprendre :

* widgets\_init //Le hook d’initialisation des widgets
* register\_sidebar //Fonction qui prend un tableau en paramètre avec :  
  + **‘name’** //Nom de la zone widget dans le dashboard
  + **‘id’** //Identifiant unique de la zone widget qui permettra aussi d’appeler la nouvelle zone avec la fonction « dynamic\_sidebar »
  + **‘before\_widget’** //Ouverture d’une balise HTML avec une class
  + **‘after\_widget’** //Fermeture de la balise HTML ci-dessus
  + **‘before\_title’** //Ouverture d’une balise HTML de titre avec une class
  + **‘after\_title’** //Fermeture de la balise HTML de titre ci dessus
* **is\_active\_sidebar** //Fonction avec comme paramètre l’id de la zone widget et qui retourne vrai si un widget a été inséré cette nouvelle zone
* **dynamic\_sidebar** //Fonction qui appel via son id notre nouvelle zone de widget
* **add\_action( 'widgets\_init' , ‘nom-fonction-intialisation’)** ;  
   //Lorsque le hook d’initialisation des widgets va être rencontré, on va lui demander de lancer une fonction qui contiendra l’instruction **register\_widget** décrite ci-dessous
* **register\_widget** //Enregistre un widget dans les objets widget pour être utilisé
* **WP\_Widget** //La class widgets
* **extract** //Function PHP qui crée des variables pour chaque élément d’un tableau associatif.  
   Le nom de ces variables vont être les clefs du tableau
* **apply\_filters** //Permet d’affecter la valeur saisie par l’utilisateur
* **get\_posts** //Renvoie un tableau de tous les posts selon les paramètres)
* **setup\_postdata** //Attribue à des variables globales des valeurs selon le post
* **get\_permalink** //Renvoie le permalien du post en paramètre
* **strip\_tags** //Function PHP qui supprime les balises HTML et PHP d'une chaîne
* **esc\_attr** //Function qui permet d’enlever les attributs HTML
* **get\_field\_id** //Retourne l’id du champ du widget
* **get\_field\_name** //Retourne le nom du champ du widget

* + - * 1. Widget (bloc) Profil aléatoire - **16.06** au **24.06**

Wireframe - **16.06**

Squelette HTML - **17.06** / CSS - **17.06**

Code shortcode en PHP - **20.06** au **24.06**

Commentaires - **17.06** + **24.06**

* + - * 1. Widget (bloc) Recherche - **16.06** au **24.06**

Wireframe - **16.06**

Squelette HTML - **17.06** / CSS - **17.06**

Code shortcode en PHP - **20.06** au **24.06**

Commentaires - **17.06** + **24.06**

## Wireframe - **16.06**

### Shortcode Galerie

À remplir

### Shortcode Profil By ID

À remplir

### **Page Profil unique**

À remplir

### Shortcode 3 Last Profils

À remplir

### Shortcode Slider

À remplir

### Widget (bloc) Profil aléatoire

À remplir

### Widget (bloc) Recherche

À remplir

## Use case diagram - **16.06**

### CPT dans le dashboard

À remplir

### Shortcode Galerie

À remplir

### Shortcode 3 Last Profils

À remplir

### Shortcode Slider

À remplir

### Widget (bloc) Profil aléatoire

À remplir

### Widget (bloc) Recherche

À remplir

# 

# Limitations et contraintes

1. Choix technologique

* WordPress
* HTML
* Bootstrap ou SCSS
* PHP pour WP

1. Installation et outils

* Repo Git
* Lancement du docker-compose
* Installation en local de WP pour tester le plugin
* Développement en local du plugin avec des commit réguliers pour mettre à jour le plugin

# Ressources

## Les hooks

<https://developer.wordpress.org/plugins/hooks/>  
<https://www.wpbeginner.com/glossary/hooks/>  
<https://adambrown.info/p/wp_hooks/hook>  
<https://www.hostinger.com/tutorials/what-are-wordpress-hooks/>  
<https://kinsta.com/fr/blog/hooks-wordpress/>  
<https://www.whodunit.fr/petite-introduction-a-la-mecanique-des-hooks-sur-wordpress/>

Avec un exemple pour ajouter bootstrap :  
<https://grafikart.fr/tutoriels/wordpress-actions-hook-1255>  
Bootstrap plus précisément :   
<https://www.greengeeks.com/blog/bootstrap-wordpress/>

## Créer un plugin

<https://wpformation.com/creer-plugin-wordpress/>  
<https://www.hostinger.fr/tutoriels/creer-un-plugin-wordpress>  
<https://www.undefined.fr/comment-creer-un-module-wordpress/>

[https://kinsta.com/blog/wordpress-custom-post-types/#register](https://kinsta.com/blog/wordpress-custom-post-types/" \l "register)

## CPT

<https://www.hostinger.com/tutorials/wordpress-custom-post-types>  
<https://wpchannel.com/wordpress/tutoriels-wordpress/creer-custom-post-types-wordpress/>

[https://kinsta.com/blog/wordpress-custom-post-types/#register](https://kinsta.com/blog/wordpress-custom-post-types/" \l "register)

## Metaboxes

<https://www.sitepoint.com/adding-meta-boxes-post-types-wordpress/>  
<https://developer.wordpress.org/plugins/metadata/custom-meta-boxes/>  
<https://blog.wplauncher.com/add-meta-box-wordpress-custom-post-type/>

## Taxonomy

[https://www.wpbeginner.com/wp-tutorials/create-custom-taxonomies-wordpress/#code](https://www.wpbeginner.com/wp-tutorials/create-custom-taxonomies-wordpress/" \l "code)  
<https://capitainewp.io/formations/developper-theme-wordpress/creer-taxonomies/>

## Shortcode

<https://karac.ch/blog/creer-ses-propres-shortcodes-wordpress>  
<https://jeremiechazelle.dev/creer-un-plugin-wordpress-de-shortcode-avec-son-bouton-dintegration-dans-tinymce/>  
<https://kinsta.com/fr/blog/shortcodes-wordpress/>

## Template CPT

<https://developer.wordpress.org/themes/template-files-section/custom-post-type-template-files/>  
<https://yogeshchauhan.com/custom-post-type-template-files-in-wordpress/>  
<https://squareonemedia.co.uk/community/tutorials/load-custom-post-type-template-plugin-wordpress/>  
<https://www.pradipdebnath.com/2018/05/31/display-custom-post-type-page-template-from-plugin/>  
Exemple de template :  
<https://torquemag.io/2018/06/create-custom-single-post-template/>

## Widget

<https://wpmarmite.com/widget-wordpress/>

## Widget Zone et création de widget

<https://karac.ch/blog/creer-ses-widgets-wordpress>  
<https://www.numelion.com/creer-un-widget-wordpress.html>

# Compétences et savoir-faire visé

Installer et configurer son CMS  
 Gérer et créer du contenu et des pages statiques et/ou dynamiques  
 Utiliser la taxonomie du CMS  
 Gérer et créer des utilisateurs et leurs droits  
 Gérer des templates  
 Gérer et configurer des extensions  
 Lier des outils d’analyse au CMS  
 Savoir utiliser des interfaces WYSIWYG  
 Adapter des templates  
 Création, découpe et respect de wireframe  
 Analyser les besoins  
 Comprendre les objectifs et enjeux  
 Rédiger des cahiers des charges et spécifications  
 Rédiger des use case diagram  
 Renseigner les variables d’environnement  
 Appliquer des mesures limitant les failles de sécurité  
 Découper, organiser, prioriser les tâches et définir les Sprint  
 Mettre en place des outils collaboratifs et de versioning  
 Mettre en place des conteneur

**CPTax:**

**TAXONOMIES**:

*ETIQUETTES*:

* **Savoir/Connaissances**: (étiquette)

ex: technos:

* Vue JS (étiquette)
* **Savoir Être/Soft Skills**: (étiquette)

ex: ponctuel, (étiquette)

*CATEGORIES*:

* **Savoir Faire/Skills**: (catégorie)

ex: page statique

* **Secteur D’activités**: (catégorie)

ex: bancaire (secteur)

finance

—------------------------------------------------------------

**Après midi:**

**21.06.22**

**14:15**

-Wireframe

-*Arborescence* **GIT**:

branch / **taĉhes** (qui fait quoi)

-Tableau **KANBAN**

—------------------------------------------------------------

28.06.2022 : **FIN DU PROJET – PRESENTATION**!!

**24.06.2022 -** 11:30:

Présenter TOUS les **PROTOTYPES**:

**Ex**: Galerie : voires ttoutes les photos, squelettes HTML, …

15h:

**PseudoCode**:

placement :

des fonctions

commentaires

...

—------------------------------------------------------------

06.07.2022 : **Finir** **PORTFOLIOS REALISE**