

# **Dossier Projet**

**Titre de développeur web et web mobile**



**Stage réalisé du 01/03/2021 au 26/04/2021**  
**Au sein de la couveuse des innovateurs**  
**Avec Onlineformapro**  
**Formation Access Code School**

# SOMMAIRE

<b>0. Remerciement:</b>	<b>3</b>
<b>1. Liste des compétences du référentiel qui sont couvertes par le projet :</b>	<b>4</b>
A. Maquetter une application	4
B. Réaliser une interface utilisateur web statique et adaptable	4
C. Réaliser une interface utilisateur avec une solution de gestion de contenu	4
D. Développer une interface utilisateur web dynamique	5
E. Créer une base de données :	5
F. Développer les composants d'accès aux données et élaborer :	6
<b>2. Résumé du projet en français :</b>	<b>7</b>
<b>3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet :</b>	<b>8</b>
<b>4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile :</b>	<b>10</b>
Étape 1 : créer le dossier de notre thème	14
Étape 2 : créer le fichier style.css	14
Étape 3 : créer le fichier index.php	15
La hiérarchie des templates WordPress	15
Créer les fichiers du header et du footer	16
Créer le fichier functions.php	17
<b>5. Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile</b>	<b>18</b>
<b>6. Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)</b>	<b>26</b>
<b>7. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité</b>	<b>29</b>
<b>8. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone</b>	<b>30</b>
<b>9. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes)</b>	<b>31</b>

## 0. Remerciement:

Tout d'abord, je voudrais remercier mon maître de stage, *M. Christophe Dollet ainsi que Mme Valérie Barthelet, Directrice de l'association, sans qui le stage n'aurait pas pu être possible*. Ils ont su me faire confiance lors de cette aventure dans le monde professionnel et ont partagé leurs connaissances de manière très pédagogique. Je les remercie aussi pour leur disponibilité et la qualité de leur encadrement au sein de l'association.

Je souhaite ensuite adresser mes remerciements au corps formateur et administratif d'Onlineformapro, pour la qualité de l'enseignement offert et de l'équipe administrative.

Je remercie également *M. Alain Merucci, formateur au sein de l'Access Code School*, pour ses conseils sur la rédaction de mon rapport de stage et ses enseignements lors de ma formation au titre de développeur Web & Web Mobile.

# 1. Liste des compétences du référentiel qui sont couvertes par le projet :

Front-end :

## A. Maquetter une application

Concernant la partie front-end j'ai choisi de vous présenter le site internet de la Couveuse des innovateurs. Avec l'aide du cahier des charges, j'ai pris en compte le logo de la société et les différents sites dont mon tuteur de stage appréciait le design pour réaliser le maquettage. Mon tuteur m'a demandé de créer un site simple et facile d'utilisation. La maquette a été soumise et validée par mon tuteur de stage avant l'intégration et développement. Pour réaliser celle-ci, j'ai utilisé Figma, un éditeur de graphiques vectoriels basé sur le web.

## B. Réaliser une interface utilisateur web statique et adaptable

J'ai réalisé l'interface à l'aide de Visual Studio Code ainsi qu'en fonction de la maquette, des besoins des autres pages avec du HTML/CSS, Javascript ainsi que Bootstrap.

L'interface a été faite sur la règle du mobile first tout ça grâce à la grille Bootstrap que j'ai utilisé au maximum ainsi que des media queries lorsque cela était nécessaire.

## C. Réaliser une interface utilisateur avec une solution de gestion de contenu

**Qu'est-ce qu'un gestionnaire de contenu ?**

Le gestionnaire de contenu vous permet de créer facilement et rapidement du contenu dynamique et éditorial (texte, images, vidéos, etc.) pour votre site Web. Le gestionnaire de contenu existe à toutes les étapes de la création de contenu Web, de la vérification des pages ou des articles à la gestion des règles de publication.

Pour être défini comme CMS, le logiciel doit être capable de :

- Utilisé par plusieurs personnes en même temps
- Être un canal de distribution de contenu recommandés
- Permettre une gestion séparée des tables et du contenu

J'ai développé le site : [www.couv.giannidubief.fr](http://www.couv.giannidubief.fr) avec le CMS Wordpress. Wordpress est le CMS le plus présent actuellement sur le WEB, principalement par le fait qu'il est gratuit ainsi que administrable assez facilement et open source.

Au niveau du référencement j'ai utilisé le plug-in Yoast, il m'a permis d'améliorer le référencement via des balises ainsi que des mots clés qui s'ajouteront aux métadonnées.

## **D. Développer une interface utilisateur web dynamique**

Toutes les pages sont créées dynamiquement pour permettre aux utilisateurs de naviguer sur le site Web à partir de n'importe quel type d'appareil. J'ai également pris en compte la taille de la police, en fonction de la taille de l'écran du support utilisé pour accéder au site. Afin d'obtenir un affichage dynamique de ces pages, j'ai utilisé Bootstrap, qui permet une adaptation visuelle à la taille de l'écran. Mais dans certains cas particuliers, j'utilise également des medias queries.

## **Back-end :**

### **E. Créer une base de données :**

Lors de l'installation de Wordpress, il est nécessaire de créer une base de données au préalable. Celle-ci se fera sous Phpmyadmin ou autres gestionnaires de bases de données. Ainsi, lors de son installation Wordpress viendra créer les tables de manière automatique déjà présentes dans le CMS.

Le fichier wp-config.php permettra de modifier la racine de connexion du site. Nous pourrions donc changer au besoin ces points suivant pour la mise en ligne du site :

- DB\_NAME : Nom de la base de données.
- DB\_USER : Nom de l'utilisateur ayant accès à cette base de données.
- DB\_PASSWORD : Mot de passe permettant l'accès à cette dernière.
- DB\_HOST : Elle gèrera le nom de l'hôte.

Ce fichier est important, c'est celui qui permet la connexion du projet avec la base de données.

Pour l'importation d'une base de données, ceci se fera assez simplement, il suffira de se connecter sur Phpmyadmin et de mettre exporter sur la base de données qui nous intéresse et la télécharger.

Il faudra donc aussi penser à bien modifier les chemins et certaines requêtes SQL.

## F. Développer les composants d'accès aux données et élaborer :

Pour la création d'un thème parent sous Wordpress, nous devons ajouter des fonctionnalités dans le fichier **functions.php**, cela permettra la création d'une zone de widget, créer des menus, définir la pagination, créer un bouton, intégrer Bootstrap ou bien modifier des images.

Pour la majorité des fonctions Wordpress, elles sont déjà pré-enregistrées dans Wordpress, il suffira donc de les appeler et de modifier les paramètres.

Nous allons prendre l'exemple pour la création de la zone d'un menu.

Il suffit d'utiliser la fonction `register_nav_menus`, lui donner un nom ainsi qu'une position.

Exemple :

```
// on crée une zone pour le menu
register_nav_menu( 'menuheader', 'Menu du Header' );
```

Dans les fichiers templates, il est nécessaire d'appeler le contenu de chaque page avec une condition (if -> si la page existe, si la page possède du contenu alors on affiche quelque chose) et une boucle (while -> tant qu'il y a quelque chose alors affiche le). Voici la condition et la boucle présente sur chaque page :

```
<?php if (have_posts()) : ?>
  <?php while (have_posts()) : the_post(); ?>
```

Dans mon cas, chaque page ayant des besoins particuliers et une présentation spécifique, j'ai utilisé les Custom Post Type et Advanced Custom Field. Chaque élément est appelé grâce à une boucle par l'id du champ. On appelle alors le champ ACF (qui est lié à une page) au lieu d'appeler le contenu dans la page. Pour respecter au mieux les règles de Wordpress lors de la mise en place de ces fonctionnalités, j'ai suivi la documentation (en anglais) du CMS, le codex (disponible en français), mais aussi celle de Advanced Custom Fields (en anglais).

Voici les principales ressources que j'ai utilisé :

Wordpress :

<https://developer.wordpress.org>

<https://codex.wordpress.org>

## **2. Résumé du projet en français :**

J'ai effectué mon stage durant 6 semaines chez l'association La « Couveuse des Innovateurs ».

### **A1 : présentation de l'association**

La « Couveuse des Innovateurs » est une association de type loi 1901, sans but lucratif. Née en décembre 2019, elle s'inscrit clairement dans le prolongement du marathon d'innovation ouverte en santé « Hacking Health » de Besançon, pour assurer le suivi et l'accompagnement des projets qui en sont issus.

([www.hh-besancon.fr](http://www.hh-besancon.fr))

Elle n'a aucune vocation à se substituer aux dispositifs existants, en particuliers les incubateurs et autres spécialistes du soutien à la création d'entreprises. La Couveuse des Innovateurs accueille les porteurs qui n'ont pas l'ambition de créer une startup, mais souhaitent que leur projet se transforme en produit ou service nouveau au bénéfice du plus grand nombre. Les profils sont des professionnels de santé, patients ou aidants qui ont besoin de mûrir leur projet, le valider, le protéger avant d'envisager un développement avec des partenaires industriels établis ou un créateur d'entreprise qui croit en la pertinence de leur concept.

L'aide apportée par la Couveuse des Innovateurs prend des formes très variées : ateliers spécifiques, mise en relation avec des experts, échanges d'expériences grâce aux porteurs de projets des éditions précédentes, mais aussi portage de stages d'étudiants pour affiner un point précis, soutien à la réalisation de prototypes, etc.

La « Couveuse des Innovateurs » veille aussi à entretenir les liens entre l'ensemble des acteurs de l'écosystème bisontin dédié à l'innovation en santé, auquel elle entend apporter sa contribution singulière.

### **A2 Les objectifs du site :**

Il s'agit essentiellement d'un site vitrine, mais aussi d'un outil permettant aux porteurs de projets de garder le contact entre eux et avec les coaches.

### **A3 : Les cibles :**

Le site vise à apporter une information large à l'ensemble des publics intéressés par l'innovation en santé. Il assure également un lien permanent entre les porteurs de projets et la communauté du Hacking Health.

### **A4 : Les objectifs quantitatifs :**

Même s'il est accessible à tous, le site internet ne recevra pas un important trafic. Il ne s'agit pas d'un site très grand public, mais plutôt destiné à un public spécialisé. L'idéal serait évidemment qu'il puisse être bien référencé.

### 3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet :

#### Cahier des charges :

##### Indications générales :

Le contenu du site :

Le site internet devra remplir plusieurs fonctions :

1) Une partie ouverte à tous :

- Page d'accueil :
  - Elle mettra en avant une photo représentant “La couveuse des innovateurs”.
  - Il y aura un bloc pour présenter le but de l'association.
  - La page d'accueil aura pour objectif de mettre les nouveaux projets en avant à l'aide d'un slider de façon limitée en fonction des goûts de l'administrateur.
  - Il y aura une section dédiée aux événements de l'association pour les mettre en avant, tout en précisant la date et le lieu.
  - Elle contiendra aussi la mise en avant des sponsors.
- Page projets :
  - Être la vitrine officielle de l'association, expliquer ses missions, donner l'ensemble des contacts utiles.
  - Rendre visible les projets développés au sein de la Couveruse : les présenter et exprimer leurs besoins. Permettre la prise de contact direct avec les porteurs de projets.
  - Elles devront toujours être dans l'ordre chronologique du projet le plus récent au moins récent.
- Page agenda :
  - Proposer un agenda annonçant les ateliers et événements organisés par la Couveruse, mais aussi les rendez-vous susceptibles d'intéresser les porteurs de projets.
- Page partenaires :
  - Assurer une visibilité à nos sponsors et partenaires.



- Contiendra un bouton pour nous contacter.
  - Des collapses pour répondre aux questions liées à une demande de sponsoring ou partenariat.
- Page faq :
- Devra mettre en avant toutes les questions les plus récurrentes mise dans des collapses, pouvant être supprimées ou éditées sans limites.
- Page adhérents :
- Un annuaire de la « Communauté Hacking Health » présentant les lanceurs de projets, les experts, les coachs et autres personnes ressources, avec leur compétences, et un contact mail.

Le code devra être le plus simple et le plus commenté possible pour permettre une reprise simple si jamais il doit être retravaillé plus tard.

Le site doit être entièrement administrable sauf éléments définis au départ.

Il est nécessaire d'avoir un minimum de référencement grâce à :

- Référencement naturel
- Yoast

Les informations liées à Yoast pourront être modifiées ou ajoutées par l'administrateur du site et ainsi améliorer le référencement de ce dernier.

Le demandeur fournira l'ensemble des textes et photos devant figurer sur le site, et particulièrement :

- La présentation des missions de l'association
- La présentation des projets de la Couveuse
- L'ensemble des contacts des coachs
- Les logos des sponsors et partenaires
- Les photographies des personnes et d'illustration

### **Charte graphique :**

1- Couleur : #e0272b (rouge) #303030 (gris) #ffff (blanc)

2- Logo : 

## **4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile :**

### **A. Méthodologie de gestion de projet :**

#### **Contexte :**

Le projet de ce stage était de développer un site vitrine pour une association tout en ayant une administration simple et efficace.

Durant ce stage j'étais le seul développeur sous la tutelle de M. Dollet, en cas de problèmes techniques ou de blocage, il y avait notre formateur M. Merucci.

#### **Choix de l'outil de gestion de projet :**

Pour la gestion de projet, nous avons utilisé la plateforme collaborative GitHub, qui permet de publier son code.

C'est la technologie de versionnage de code dans laquelle nous avons été formé ainsi que la plus utilisée.

Les raisons principale pour laquelle nous allons utiliser cet outil sont :

- Se positionner sur une version antérieure de son code
- Un outil sécurisant permettant de faire des expériences ou certaines erreurs sans devoir tout recommencer depuis le début grâce au système de branche.

### **B. Technologies utilisées pour le développement :**

Après identifications des besoins de l'association et des porteurs du projet, nous avons conclu qu'il fallait un CMS simple et administrable même sans aucune connaissance du code.

J'ai donc décidé de travailler principalement sur Wordpress en PHP ainsi qu'avec ACF-Repeater.

## Le CMS Wordpress :

J'ai choisi d'organiser mon code en fonction de Wordpress.

C'est un CMS sur lequel nous avons été formés, ou nous avons pu voir la création d'un thème de base, d'un plug-in et aussi d'un site E-commerce via l'extension Woocommerce.

## LES LANGAGES DE WORDPRESS :

Pour fonctionner, ce CMS est basé sur plusieurs langages informatiques :

- PHP (de la version minimale 5.2.4 à la dernière version PHP 7) : c'est ce qui fait fonctionner WordPress ;
- SQL : une base de pour stocker vos contenus (ainsi que toutes vos options et vos utilisateurs) ;
- HTML : pour générer le rendu des contenus. C'est ce que l'utilisateur voit dans son navigateur ;
- CSS : pour mettre en forme les contenus ;
- JavaScript: pour enrichir les fonctionnalités du site.

WordPress se base ainsi sur PHP et toutes les fonctions de ce langage pour fonctionner. Il y ajoute ensuite ses propres fonctions (dont nous allons parler plus loin).

Pour la partie JavaScript, sachez que WordPress s'oriente de plus en plus vers ce langage. Les développeurs WordPress devront donc de plus en plus se former à ce dernier afin de pouvoir réellement maîtriser tous les aspects du CMS. Par exemple, les équipes du cœur de l'outil développent notamment :

- **L'API REST**, une api permettant d'envoyer et de recevoir des données de WordPress au format JSON ;
- **Gutenberg**, l'éditeur de WordPress en live (pour pouvoir éditer un contenu sans devoir utiliser l'administration du site) ;

## La sécurité dans Wordpress :

### 1- Le compte Administrateur :

Tout d'abord, quelle que soit la méthode d'installation que vous choisissiez, toujours créer un nouveau compte ADMIN avec un nom de connexion et un mot de passe sécurisé. Ne pas utiliser le nom "Admin", son nom ou son prénom.

## 2- Le mot de Passe :

Utilisez toujours des mots passe complexes contenant des lettres, des symboles et des chiffres. Vous feriez mieux d'utiliser un générateur de chaînes aléatoires avec plus de 8 caractères. Cela rendra la connexion plus sécurisée.

## 3- Masquer la version de Wordpress :

Votre version donne des informations aux hackers pour trouver d'éventuelles failles de sécurité. Dans le fichier function.php de votre thème, ajoutez ce bout de code :

```
// Masquer la version Wordpress  
remove_action("wp_head", "wp_generator");
```

## 4- Faire des sauvegardes :

Les backups du système sont à effectuer au moins toutes les semaines pour prévenir un piratage ou un crash disque.

## 5- Faires toutes les mises à jour :

Toujours avoir la dernière version, c'est essentiel :

- Mettre à jour WordPress, ça permet d'avoir tous les derniers correctifs des failles de sécurité.
- Mettre à jour nos plugins afin d'avoir toujours la version la plus récente donc secure et éviter tout risque d'incompatibilité.
- Mettre à jour son thème.

## 6- Protégez les fichiers :

Penser à bloquer la navigation dans nos dossiers Wordpress, par défaut n'importe qui peut accéder à nos dossiers WordPress (wp-content).

Pour protéger le fichier, on peut le faire via notre .htaccess et ajouter :

```
<Files wp-config.php>  
order allow,deny  
deny from all  
</Files>
```

## 7- Changez le préfixe "Wp"

Par défaut c'est le préfixe "Wp\_" qui est utilisé. Ce préfixe est connu de tous les utilisateurs Wordpress et peut donc être vulnérable aux injections.

## 8- Un hébergement spécialisé Wordpress

Laissez votre hébergeur se préoccuper de la sécurité. Avec un hébergeur WordPress, votre site est entre de bonnes mains, certes plus cher que le mutualisé, ce type d'hébergement maintient, protège, répare et optimise votre site.

### **Github :**

J'ai utilisé Github uniquement pour le versionning du thème.

Voici le lien de mon repository : <https://github.com/Giaduub/stage>

### **Maquettage de l'application mobile :**

#### **Figma :**

J'ai réalisé la maquette à partir de Figma, j'ai utilisé Figma, un éditeur de graphiques vectoriels basé sur le web.. La charte graphique a été conçue à partir du logo "La couveuse des innovateurs" ainsi qu'à partir du site du hacking health. La typographie utilisée est Ubuntu, elle diffère de la typographie du logo, elle a été choisie dans le but de créer un éloignement tout en soulignant la proximité avec le hacking health.

La couleur principale sera le rouge #e0272b du logo pour rappeler le symbole de l'urgence, de l'attention, qu'elle est associée à un contexte humanitaire.

La couleur secondaire est le blanc #ffffff, elle l'est pour rassurer, elle calme, c'est la couleur de l'apaisement, de la pureté.

La couleur tertiaire est l'anthracite #303030, elle est là pour apporter de la sobriété, de l'élégance en étant raffinée.

Pour la page d'accueil nous verrons que les containers sont collés, que l'espace se fait rare dans le but de donner un effet magazine, donner de l'impact ainsi que du contenu à la page.

Pour la page projets, nous partirons sur l'effet inverse, plus d'espace, plus aéré pour faciliter la lecture ainsi que donner l'envie aux gens de lire les projets.

En version nous partirons sur un menu en position sticky( position:sticky ) avec le logo à gauche grâce à un float left et les menus en float right.

Le menu en format mobile deviendra un menu burger avec un collapse lors du clic et sera lui aussi en position sticky sur fond blanc pour que lors de la descente nous ne perdions pas le menu à disposition.

(**Voir la maquette page 35** )

## Création du thème

Les thèmes sont utilisés pour fournir un format cohérent pour chaque page ou article publié sur le site Web. Le contenu fourni par WordPress et sa communauté peut présenter certaines limitations dans la conception fournie, et vous pouvez souhaiter ou avoir besoin d'avoir votre propre thème pour personnaliser votre site Web en fonction de vos besoins.

En créant votre propre thème, vous pouvez rapidement améliorer l'apparence de votre site web: styliser et ajouter de nouveaux menus, modifier certaines balises ou ajouter des codes HTML et CSS, retoucher le design, personnaliser le footer ...

### Étape 1 : créer le dossier de notre thème

Pour créer un thème WordPress, vous devez d'abord créer un dossier dans le répertoire WordPress sur le serveur: wp-content / themes. Il est recommandé d'utiliser un nom court unique afin de pouvoir l'identifier facilement dans notre cas, mon thème s'appelle 'couveuse'. N'utilisez pas de caractères spéciaux, de marques d'accentuation ou d'espace. Si vous le nommez "Couveuse", le chemin est: wp-content / themes / couveuse.

Ce dossier devra contenir à minima les trois fichiers indispensables pour le bon fonctionnement de votre site :

1. le fichier style.css : pour la configuration et l'ajout des règles de style,
2. le fichier index.php : pour l'affichage du contenu de votre site,
3. le fichier functions.php : pour l'ajout de fonctionnalités WordPress dans votre thème.

### Étape 2 : créer le fichier style.css

Le premier fichier à créer est le fichier style.css. La déclaration et la configuration du thème sont obligatoires. Son existence est très importante pour deux raisons:

1. Il va vous permettre d'ajouter votre thème dans l'interface d'administration de WordPress pour qu'il puisse être reconnu par le CMS.
2. Vous pourrez ajouter des règles de style pour la mise en forme de vos contenus.

Dans ce fichier style.css, vous pouvez ajouter les lignes suivantes, qui permettront à WordPress de reconnaître votre nouveau thème, en lui fournissant des informations comme son nom, celui de l'auteur, une description et son numéro de version :

```
/*  
  
Theme Name: Theme Couveuse  
  
Theme URI:  
  
Author: DUBIEF Gianni  
  
Author URI:  
  
Description: 2021 thème la couveuse  
  
Requires at least: 4.9.6  
  
Requires PHP: 5.2.4  
  
Version: 1.0  
  
License:  
  
License URI:  
  
Tags:  
  
*/
```

## Étape 3 : créer le fichier index.php

### La hiérarchie des templates WordPress

Les thèmes WordPress utilisent un système de templates, qui correspondent à des fichiers .php, dont les noms permettent au CMS de savoir quel fichier charger selon

le contexte de la page à afficher. C'est ce qui s'appelle la hiérarchie des templates. Afin d'afficher un contenu, quel qu'il soit, WordPress aura toujours comme dernier recours l'utilisation du fichier `index.php`. C'est la raison pour laquelle il est indispensable et que vous devez à minima le créer.

Voici quelques exemples de pages et le chemin emprunté par WordPress pour afficher un modèle de page en fonction de celle demandée :

- Affichage de la page d'accueil : fichier `home.php` ou `front-page.php`, ou `index.php` (par défaut),
- Affichage d'une page d'article : `single-{post-type}-{slug}.php` (le *post-type* correspondant au type de publication), ou `single-{post-type}.php`, ou `single.php`, ou `singular.php`, ou `index.php` (par défaut),
- Affichage d'une page : un modèle de page choisi lors de la rédaction de la page parmi le menu « modèle de page » (si l'option est utilisée), ou `page-{slug}.php`, ou `page-{id}.php`, ou `page.php`, ou `singular.php` ou `index.php` (par défaut),
- Affichage d'une catégorie : `category-{slug}.php` (ex : `category-news.php` si le slug est `news`), ou `category-{id}.php`, ou `category.php`, ou `archive.php`, ou `index.php` (par défaut),
- Affichage d'un auteur : `author.php`, ou `archive.php`, ou `index.php` (par défaut),
- Affichage d'une page datée : `date.php`, ou `archive.php`, ou `index.php` (par défaut),
- Affichage d'une page de résultats de recherche : `search.php`, ou `index.php` (par défaut),
- Affichage d'une page 404 : `404.php`, ou `index.php` (par défaut).

## Créer les fichiers du header et du footer

Afin de factoriser – c'est-à-dire ne pas dupliquer du code inutilement – et de faciliter la lecture de votre code, il est recommandé de suivre les bonnes pratiques en créant les deux fichiers suivants :

- le fichier `header.php` : il doit contenir la fonction `wp_head()`,

```
<!doctype html>
<html lang="fr">
```



```

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="<?php echo get_stylesheet_uri();?>">

<title><?php bloginfo('name');?></title>

```

Les métadonnées sont récupérées grâce à la fonction `wp_head()`. La feuille de style est appelée grâce à la fonction native `get_stylesheet_uri()`. Pour la balise de métadonnées title (titre de la page, affiché dans l'onglet du navigateur), doit contenir le titre de la page, appelé avec `wp_title()`; et le nom du site La couveuse des innovateurs appelé lui avec le marqueur de modèle `bloginfo('name')`. Ainsi, ce titre changera automatiquement en fonction de la page où l'on se trouve.

- le fichier `footer.php` : il doit contenir la fonction `wp_footer()`.

```

<script src="<?=
site_url()."/wp-content/themes/".get_template()."/javascript/slider.js"; ?>">

</script>

<script src="<?=
site_url()."/wp-content/themes/".get_template()."/javascript/menu.js"; ?>">

</script>

<script src="<?=
site_url()."/wp-content/themes/".get_template()."/javascript/collapse.js";
?>">

</script>

```

`site_url()` et `get_template()` permettent de trouver la racine des fichiers plus facilement grâce aux fonctions de Wordpress déjà implanté et ainsi de permettre une modification plus simple des scripts Javascript.

## Créer le fichier `functions.php`

Autre étape importante, le fichier `functions.php` permet d'ajouter des fonctionnalités au sein de votre thème WordPress. Parmi les fonctions principales, vous aurez besoin de quelques lignes de code pour créer le menu de votre site.

## 5. Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant, y compris pour la sécurité et le web mobile

### Définition du custom post type :

Un custom post type (CPT) correspond à un type de contenu personnalisé.

Par défaut, WordPress permet de créer des articles ainsi que des pages. Cela dit, en fonction du site Internet souhaité, il peut être intéressant d'y ajouter des types de contenus supplémentaires.

### Ajouter un custom post-type :

Pour les besoins de l'association, j'ai dû créer un custom post-type pour les projets. Ce custom post-type avait pour avantage de pouvoir rentrer chaque projet d'une seule et unique façon tout en gardant le même comportement pour tous. Elle sera créée dans le dossier **functions.php**.

Comme on peut le voir, on va venir créer une fonction ici appelée "wpm\_custom\_post\_type", elle peut porter le nom que l'on désire.

```
function wpm_custom_post_type() {
```

Ensuite, on va créer une variable "\$labels" qui deviendra le tableau et permettra d'ajouter ce que l'on désire.

```
    // On rentre les différentes dénominations de notre custom post type qui
    // seront affichées dans l'administration

    $labels = array(
```

On aperçoit que le code est séparé en deux, sur la gauche ce sont des paramètres déjà présentes dans le tableau au sein même de Wordpress, à droite les valeurs que l'on veut afficher dans notre custom post type dans l'administration.

```

    // Le nom au pluriel

    'name'                => _x( 'Projets', 'Post Type General Name'),

    // Le nom au singulier

    'singular_name'       => _x( 'Projet', 'Post Type Singular Name'),

    // Le libellé affiché dans le menu

    'menu_name'           => __( 'Projet'),

    // Les différents libellés de l'administration

    'all_items'            => __( 'Tous les projets'),

    'view_item'            => __( 'Voir les projets'),

    'add_new_item'         => __( 'Ajouter un nouveau projet'),

    'add_new'              => __( 'Ajouter un projet'),

    'edit_item'            => __( 'Editer le projet'),

    'update_item'          => __( 'Modifier le projet'),

    'search_items'         => __( 'Rechercher un projet'),

    'not_found'            => __( 'Non trouvée'),

    'not_found_in_trash'   => __( 'Non trouvée dans la corbeille'),

    'featured_image'        => __( 'Featured Image', 'text_domain' ),

    'set_featured_image'    => __( 'Set featured image', 'text_domain' ),

    'remove_featured_image' => __( 'Remove featured image', 'text_domain'
),

    'use_featured_image'    => __( 'Use as featured image', 'text_domain'
),

    'insert_into_item'      => __( 'Insert into item', 'text_domain' ),

    'uploaded_to_this_item' => __( 'Uploaded to this item', 'text_domain'
),

    'items_list'           => __( 'Items list', 'text_domain' ),

```

```

        'items_list_navigation' => __( 'Items list navigation', 'text_domain'
    ),

        'filter_items_list'      => __( 'Filter items list', 'text_domain' ),

    );

```

On peut prendre l'exemple d'add new qui donnera la valeur 'Ajouter un projet' au paramètre.

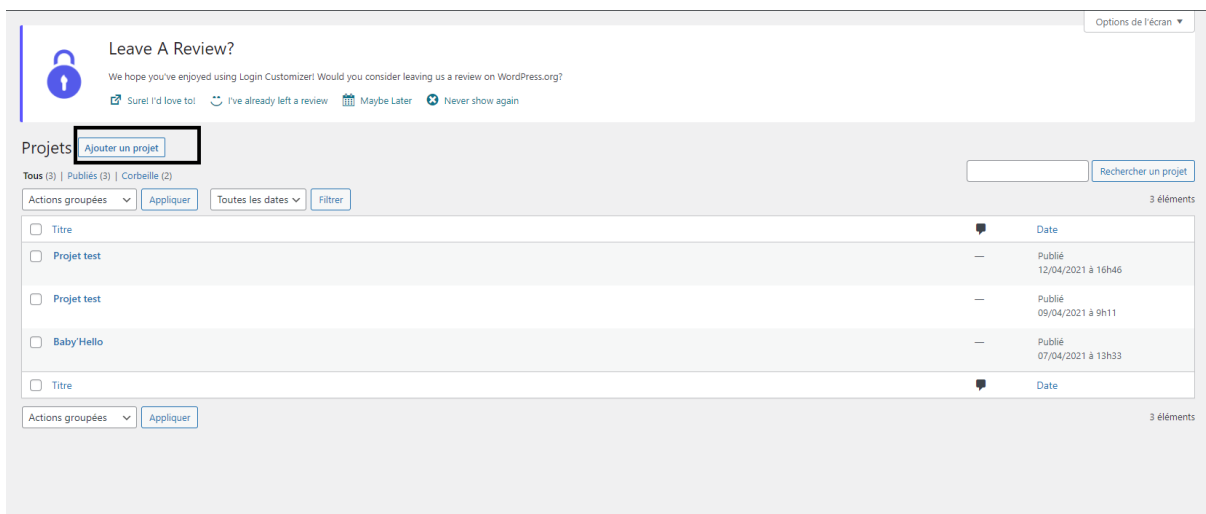
Exemple :

```

'add_new'          => __( 'Ajouter un projet'),

```

ça donne ceci :



Ensuite on peut définir d'autres options pour notre custom post type en créant à nouveau une variable mais qui elle, sera différente pour ajouter un "label" et une description.

```

$args = array(
    'label'          => __( 'Projets' ),
    'description'    => __( 'Tous sur projets' ),
    'labels'         => $labels,

```

On définit les options disponibles dans l'éditeur de notre custom post type

```

'supports'         => array( 'title', 'editor', 'slug', 'comments',
'thumbnail', 'revisions', 'custom-fields', ),

```

On ajoutera aussi d'autres options supplémentaires.

```
'show_in_rest'      => true,
    'hierarchical'   => false,
    'public'         => true,
    'has_archive'     => true,
    'rewrite'        => array( 'slug' => 'projet'),
```

On retrouvera tous les paramètres dans sur ce lien :

[https://developer.wordpress.org/reference/functions/register\\_post\\_type/](https://developer.wordpress.org/reference/functions/register_post_type/)

Notre custom post type est enfin créé, nous allons maintenant avoir ce résultat :



Pour l'affichage d'un custom post-type il existe deux type de page :

La page **single-{name}.php**, qui va être la façon dont doit se comporter un projet, c'est la base de l'affichage du custom post type.

La page devra comporter au minimum :

Une condition if qui veut dire que "Si" on a un post dans notre custom post type, il devra alors faire une boucle avec "While".

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
```

On devra après ne pas oublier de fermer notre condition ainsi que notre boucle comme ceci.

```
<?php endwhile; ?>
<?php endif; ?>
```

Nous avons donc une page **single-{name}.php** qui est opérationnelle, maintenant il ne reste plus qu'à ajouter le HTML/CSS pour donner le style et comment elle doit se comporter.

Maintenant, on passe à une page importante de notre custom post-type, la page **archive-{name}.php**, cette page aura pour but de venir chercher tous les projets et ainsi de les afficher au sein d'une même page.

Ici, on viendra créer une requête avec la variable "\$loop" pour aller chercher dans le tableau les données qui nous intéressent, en l'occurrence ici, projet :

On voit qu'on a aussi mis une limite de posts par page de 15, ce qui veut dire que la boucle sur cette page s'arrêtera à 15.

```
<?php $loop = new WP_Query( array( 'post_type' => 'projet', 'posts_per_page' => '15' ) ); ?>
```

Grâce à la requête, nous avons donc récupéré les données de notre tableau qui nous intéresse, on va donc créer une boucle qui va se répéter pour chaque projet présent et ainsi les afficher.

```
<?php while ( $loop->have_posts() ) : $loop->the_post(); ?>
```

On va venir fermer la boucle avec endwhile et y ajouter la fonction wp\_reset\_query qui va venir détruire la requête précédente et configurer une nouvelle requête.

```
<?php endwhile; wp_reset_query(); ?>
```

Les ressources qui m'ont aidé pour le custom post type sont :

<https://wpmarmite.com/snippet/creer-custom-post-type/>

[https://developer.wordpress.org/reference/functions/register\\_post\\_type/](https://developer.wordpress.org/reference/functions/register_post_type/)

<https://www.wpbeginner.com/wp-tutorials/how-to-create-custom-post-types-in-wordpress/>

## Qu'est-ce que ACF ?

ACF ou Advanced Custom Fields, qu'on peut traduire en français par champs personnalisés avancés, est une extension très utilisée par les développeurs sur les sites WordPress de leurs clients, car elle leur permet de faire beaucoup de choses qui leur facilitent grandement la vie pour faire des sites aux fonctionnalités avancées.

## Ajouter ACF dans le custom post type Projet :

Pour rendre le custom post type plus administrable on va le coupler avec ACF, ce qui va permettre de créer une administration.

Tout d'abord on va créer des champs via ACF.

En 1er on va donner un titre à notre groupe de champs :

Modifier le groupe de champs [Ajouter](#)

ACF Projet

Ensuite, nous allons créer nos champs et leur donner des intitulés.

Comme ceci :

Ordre	Intitulé	Nom	Type
1	author	author	Texte
2	img-1	img-1	Image
3	text-1	text-1	Éditeur de contenu
4	search	search	Éditeur de contenu
5	tri	tri	Liste déroulante

[+ Ajouter un champ](#)

On va aussi établir des règles pour lui dire où il doit s'appliquer.

Emplacement

**Règles**  
Créez une série de règles pour déterminer les écrans sur lesquels ce groupe de champs sera utilisé

Montrer ce groupe quand

Type de publication  est égal à  Projet  [et](#)

ou

[Ajouter une règle](#)

On va maintenant appeler ces champs via les intitulés sur nos pages

**single-{name}.php** et **archive-{name}.php**.

On récupère nos champs ACF via la fonction **get\_field()** qui est intégrée à ACF, elle permet de récupérer la valeur de n'importe quel champ à partir de n'importe quel endroit.

```
<?php $author = get_field('author');  
$img1 = get_field('img-1');  
$img2 = get_field('img-2');  
$text = get_field('text-1');  
$search = get_field('search');  
$tri = get_field('tri'); ?>
```

Les ressources utilisées pour ACF :

<https://www.advancedcustomfields.com/resources/>

**Création d'un menu responsive :**

Pour le menu responsive, je suis parti sur un menu assez simple, un menu qui affiche tous les chemins sur la version Desktop et qui devient un menu burger en mobile.

La version desktop :



[Accueil](#) [Nos projets](#) [Nos événements](#) [Nos partenaires](#) [Vos questions](#) [Nos experts](#) [Contact](#) [Connexion](#)

La version mobile :



En allant plus en détail, nous allons voir que le menu est appelé via une fonction Wordpress qui est celle ci :

```
<?php
    wp_nav_menu( array(
        'container' =>false,
        'class' =>' ',
        'theme_location' => ' ',
        'menu_id' => ' ',
        'echo' => true,
        'before' => ' ',
        'after' => ' ',
        'link_before' => ' ',
        'link_after' => ' ',
        'depth' => 0,
        'items_wrap' => '%3$s',
        'walker' => ' ')
    );
?>
```

wp\_nav\_menu() est une fonction native de Wordpress, elle nous permet de récupérer des paramètres et de les changer si besoin.

On peut trouver les paramètres et leur utilité ici :

[https://developer.wordpress.org/reference/functions/wp\\_nav\\_menu/](https://developer.wordpress.org/reference/functions/wp_nav_menu/)

Maintenant, passons au responsive, pour le responsive j'ai utilisé du media queries grâce au CSS :



Pour le premier, je lui dis qu'en dessous de 1250 pixels, il doit être un display: inline-block, ce qui est une manière de mettre les éléments inline tout en préservant leurs capacités d'éléments block, tels que la possibilité de définir une largeur et une hauteur, des marges et paddings top et bottom, etc.

```
@media screen and (max-width: 1250px) {  
  .hamburger-custom {  
    display: inline-block;  
  }  
}
```

Pour le second cas, je gère un logo qui devra devenir un display block quand l'écran est inférieur à 1250 px, pour que le logo puisse apparaître lors de l'ouverture du menu burger et disparaître lorsque que le menu burger n'est plus utile.

```
@media screen and (max-width: 1250px) {  
  .logo-none{  
    padding-bottom:20px;  
    display:block;  
  }  
}
```

Dans ce cas le media queries va gérer le margin global de 8 pixels et la taille du texte dans les à présent dans **.nav-menu .menu-item**.

```
@media screen and (max-width: 1250px) {  
  .nav-menu .menu-item a {  
    font-size: 20px;  
    margin: 8px;  
  }  
}
```

Ensuite pour que cela marche parfaitement et que l'on puisse ouvrir le menu j'ai intégré du Javascript.

Dans un premier temps, je demande à Javascript de chercher dans le document et de sélectionner le premier élément comportant l'id : menu-toggler-custom et d'y ajouter un écouteur qui sera au clic, on ajoute (e) pour faire référence à la variable Événement.

Ensuite on lui dira d'ajouter une class au body qui est menu-active-custom.

Le toggle permet de pouvoir activer et désactiver sans avoir à recharger la page.

```
document.querySelector('#menu-toggler-custom').addEventListener('click', (e)
=> {
    toggleBodyClass('menu-active-custom');
});

function toggleBodyClass(className) {
    document.body.classList.toggle(className);
}
```

## **6. Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative (données en entrée, données attendues, données obtenues)**

Le jeu d'essai le plus représentatif va donc être nos champs ACF, voir si tout cela fonctionne et le résultat qu'il va y avoir.

En premier temps nous allons sur notre custom post type Projet voir si tous les champs ACF y figure, nous voyons que c'est bien le cas :

[illegible]

Nous allons donc tester si tout ça fonctionne, en remplissant chaque champ et voir comment cela se comporte.

Maintenant que nous avons validé tout ça nous pouvons voir que le slug `Projet du custom` est présent

```
'rewrite' => array( 'slug' => 'projet'),
```

et que la page a bel et bien récupéré les informations que nous lui avons données.

On voit aussi que sur la page **archive-{name}.php** tout y est bien récupéré, tout ça grâce à la boucle que nous avons mis dans la ces pages.

Accueil Nos projets Nos événements Nos partenaires Vos questions Nos experts Contact Connexion

Publié le 27 avril 2021

## Projet test

Gianni

« Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. »

6

### PROTOTYPAGE DU SYSTEME

Test en environnement réel, performances proches de l'objectif

A la recherche de

« Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. »

Publié le 12 avril 2021

On voit aussi que dans la base de données tout y est présent, que nos données y sont intégrées.

us	ping_status	post_password	post_name	to_ping	pinged	post_modified	post_modified_gmt	post_content_filtered	post_parent	guid	menu_order	post_type	post_mime_type	comment_count
			confidentielle							http://localhost/couveuse/wordpress/?page_id=3				
			2-revision-v1			2021-03-15 10:23:09	2021-03-15 09:23:09		2	http://localhost/couveuse/wordpress/?p=5	0	revision		0
			97-revision-v1			2021-04-15 14:41:13	2021-04-15 12:41:13		97	http://localhost/couveuse/wordpress/?p=349	0	revision		0
			97-revision-v1			2021-04-15 14:41:16	2021-04-15 12:41:16		97	http://localhost/couveuse/wordpress/?p=350	0	revision		0
			field_606c21f66998			2021-04-12 16:07:55	2021-04-12 14:07:55		211	http://localhost/couveuse/wordpress/?post_type=acf...	0	acf-field		0
			nos-projets			2021-04-08 10:01:11	2021-04-08 08:01:11		0	http://localhost/couveuse/wordpress/?page_id=14	0	page		0
			14-revision-v1			2021-03-15 10:36:13	2021-03-15 09:36:13		14	http://localhost/couveuse/wordpress/?p=15	0	revision		0
			nos-experts			2021-04-14 14:32:05	2021-04-14 12:32:05		0	http://localhost/couveuse/wordpress/?page_id=168	0	page		0
			test-projet			2021-03-15 10:47:20	2021-03-15 09:47:20		0	http://localhost/couveuse/wordpress/?post_type=pro...	0	projets		0
			18-revision-v1			2021-03-15 10:47:20	2021-03-15 09:47:20		18	http://localhost/couveuse/wordpress/?p=19	0	revision		0
			salut			2021-03-15 11:15:18	2021-03-15 10:15:18		0	http://localhost/couveuse/wordpress/?post_type=por...	0	portfolio		0
			20-revision-v1			2021-03-15 10:58:15	2021-03-15 09:58:15		20	http://localhost/couveuse/wordpress/?p=21	0	revision		0
			field_604f32ec74807			2021-03-15 11:12:52	2021-03-15 10:12:52		22	http://localhost/couveuse/wordpress/?post_type=acf...	0	acf-field		0
			field_604f32fb74808			2021-03-15 11:12:52	2021-03-15 10:12:52		22	http://localhost/couveuse/wordpress/?post_type=acf...	1	acf-field		0
			field_604f330c74809			2021-03-15 11:12:52	2021-03-15 10:12:52		22	http://localhost/couveuse/wordpress/?post_type=acf...	2	acf-field		0
			157-revision-v1			2021-04-16 16:10:24	2021-04-16 14:10:24		157	http://localhost/couveuse/wordpress/?p=355	0	revision		0
			id4fbf4079-362e-4592-a58f-55d42x			2021-03-15 11:15:01	2021-03-15 10:15:01		20	http://localhost/couveuse/wordpress/wp-content/upl...	0	attachment	image/png	0
			20-revision-v1			2021-03-15 11:15:18	2021-03-15 10:15:18		20	http://localhost/couveuse/wordpress/?p=29	0	revision		0
			group_606c21e632413			2021-04-12 16:07:55	2021-04-12 14:07:55		0	http://localhost/couveuse/wordpress/?post_type=acf...	0	acf-field-group		0
			logo			2021-03-16 10:26:01	2021-03-16 09:26:01		0	http://localhost/couveuse/wordpress/wp-content/upl...	0	attachment	image/png	0
			157-revision-v1			2021-04-16 16:33:38	2021-04-16 14:33:38		157	http://localhost/couveuse/wordpress/?p=363	0	revision		0
						2021-04-16 16:31:11	2021-04-16 14:31:11		157	http://localhost/couveuse/wordpress/?	0	revision		0

Après nos tests, nous voyons donc que tout marche correctement ici.

## **7. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité**

Durant mon projet j'ai effectué une veille technologique sur la sécurité de Wordpress et comment l'améliorer le plus possible. Je suis donc allé sur google en cherchant comment je pouvais améliorer la sécurité du site Web.

Je suis donc tombé sur plusieurs sites qui m'ont permis de partir plus loin dans la sécurité de Wordpress et d'approfondir le sujet sur la sécurité au sein même de Wordpress.

Ce qui en ressort après ces recherches et cette veille sont ces points ce qui m'a permis d'aller plus loin dans la sécurisation Wordpress :

- Passer votre WordPress en HTTPS, c'est la meilleure façon de protéger les données de vos utilisateurs et de vous défendre contre l'usurpation d'identité. Les sites sécurisés via SSL bénéficient en outre d'un avantage pour leur référencement dans les pages de résultats de recherche.
- Un moyen détourné pour se protéger est de camoufler son WordPress, pour se faire il existe le plugin HideMyWp. Ce dernier déplace certains répertoires et masque le fait que vous utilisiez un WP. Attention toutefois sa configuration peut être périlleuse...
- Enfin, l'antivirus que je préfère : WordFence. C'est un plugin freemium, c'est-à-dire que les options de base sont gratuites et certaines réservées aux utilisateurs premium. Très efficace et performant, je vous invite à lire le tutoriel complet de cet antivirus pour WordPress.

## **8. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone**

Durant mon projet, j'ai dû réapprendre les bases d'ACF pour pouvoir l'utiliser correctement, il m'a fallu consulter alors la documentation complète d'ACF qui est disponible de façon gratuite et entière.

Je me suis retrouvé à devoir réapprendre l'utilisation du plugin, c'est-à-dire comment l'installer, appeler et pouvoir récupérer les valeurs dans les champs et ainsi les récupérer dans mon code.

Mon premier problème rencontré a surtout été de retrouver comment récupérer des infos entrées dans les champs ACF et comment les faire apparaître dans ma page. La documentation étant bien faite, j'ai su trouver ce qu'il me fallait et voir l'emploi des méthodes `get_field()` et `the_field()` que j'ai détaillé et traduit plus bas.

La documentation pour ces problèmes a été trouvée surtout grâce au site : <https://www.advancedcustomfields.com> et à sa documentation. Et bien-sûr à l'aide de stackoverflow.

## 9. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes)

Documentation de l'extension Advanced Custom Fields :

<https://www.advancedcustomfields.com/resources/displaying-custom-field-values-in-your-theme/>

English :

### Displaying Fields

Displaying field values is ACF's party piece! Any field value can be returned as a PHP variable or output as HTML via the magical functions `get_field()` and `the_field()`. These functions (alongside many others) provide a developer friendly way to customize your WordPress theme without spending hours reading our docs!

Here is some example code to see how our intuitive API works!

```
<h1><?php the_title(); ?></h1>

<?php if( get_field('sub_heading') ): ?>
    <h2><?php the_field('sub_heading'); ?></h2>
<?php endif; ?>

<div class="hero">
    <?php $image = get_field('hero_image'); ?>
    "
/>
</div>

<?php if( have_rows('event') ): ?>
    <ul>
        <?php while( have_rows('event') ): the_row(); ?>
            <li>
                <a href="<?php the_sub_field('url'); ?>"><?php
the_sub_field('title'); ?></a>
```

```
</li>
<?php endwhile; ?>
</ul>

<?php endif; ?>
```

## Français :

### Afficher les champs

Afficher les valeurs des champs est une pièce maîtresse d'ACF. N'importe quelle valeur peut être retournée en tant que variable PHP ou sortie au format HTML via la fonction magique `get_field()` et `the_field()`. Ces fonctions (à côté de beaucoup d'autres) apportent une façon plus simple au développeur pour personnaliser votre thème WordPress sans passer des heures à lire notre docs !

Voici un exemple de code pour voir comment fonctionne notre API intuitive !

La source du texte est tirée d'ici :

<https://www.advancedcustomfields.com/resources/getting-started-with-acf/>



# Annexe

# Maquette version Web :



### Partenaires : ils nous accompagnent en 2020



## Nos projets

### Date



0

IDEE

Concept non prouvé, aucune expérience d'essai conduite

### Baby'Hello

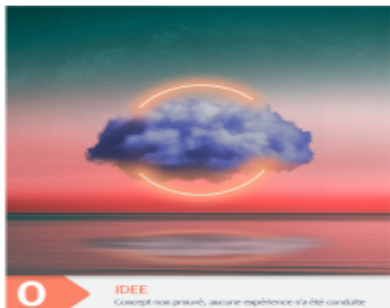
Emilie Raguin

Baby'Hello est un dispositif de diagnostic in vitro portable, peu invasif, rapide et fiable, permettant de diagnostiquer la jaunisse simplement chez le nourrisson né à terme ou prématurément. Ce dispositif médical s'adresse aux infirmières, et professionnels de santé, et leur permet de mesurer le taux de bilirubine, le pigment jaune responsable de la pathologie. Il a pour avantage d'être moins douloureux et plus fiable que les techniques actuelles, notamment chez le prématuré.

#### A la recherche de :

à la recherche de fonds pour financer la partie r&d du projet, ou de stagiaires souhaitant participer au développement d'un dispositif médical de diagnostic in vitro innovant, particulièrement sur la conception du système optique d'analyse sanguin et l'électronique du dispositif.

### Date



0

IDEE

Concept non prouvé, aucune expérience d'essai conduite

### Baby'Hello

Emilie Raguin

Baby'Hello est un dispositif de diagnostic in vitro portable, peu invasif, rapide et fiable, permettant de diagnostiquer la jaunisse simplement chez le nourrisson né à terme ou prématurément. Ce dispositif médical s'adresse aux infirmières, et professionnels de santé, et leur permet de mesurer le taux de bilirubine, le pigment jaune responsable de la pathologie. Il a pour avantage d'être moins douloureux et plus fiable que les techniques actuelles, notamment chez le prématuré.

#### A la recherche de :

à la recherche de fonds pour financer la partie r&d du projet, ou de stagiaires souhaitant participer au développement d'un dispositif médical de diagnostic in vitro innovant, particulièrement sur la conception du système optique d'analyse sanguin et l'électronique du dispositif.

## FAQ

lorem ipsum dolor sit amet ?



lorem ipsum dolor sit amet ?



## Nos partenaires

Partenaires : ils nous accompagnent en 2020



lorem ipsum dolor sit amet ?



lorem ipsum dolor sit amet ?



## Contact

Votre nom  Votre mail

Objet

Votre message

Envoyer →

## Nos experts

Nom coach	Compétences	Adresse email
Coach n°1	Coach n°1	<a href="mailto:coach@coach.fr">coach@coach.fr</a>

# Maquette mobile :

