



Dossier Projet

< ACCESS CODE >

S C H 0 0 L

SOMMAIRE

SOMMAIRE	2
Remerciements	3
Introduction	4
Liste des compétences du référentiel qui sont couvertes par le projet	5
Front-end :	5
Back-end :	5
Projet 1 : Front-end	6
Résumé du projet	6
Cahier des charges	6
Spécifications techniques du projet	7
CHOIX DES TECHNOLOGIES	7
Réalisations du candidat	9
CRÉATION DU THÈME	9
CRÉATION DES PAGES	10
AJOUT DES CHAMPS D'ÉDITION	19
FORMULAIRE DE CONTACT	23
Projet 2 : Back-end	25
Résumé du projet	25
Cahier des charges	25
Spécifications techniques du projet	26
CHOIX DES TECHNOLOGIES	26
Réalisations du candidat	27
RÉCUPÉRATION DES INFORMATIONS DES PRODUITS	27
IMPORTATION PRODUIT SIMPLE	31
IMPORTATION PRODUIT VARIABLE	33
Présentation du jeu d'essai	36
Descriptions	38
Veille	38
Situation de travail ayant nécessité une recherche en anglais	39
Conclusion	40
Annexes	41

Remerciements

Durant mon stage, j'ai eu la chance de pouvoir bénéficier des conseils de **Igor KATMAKOV** et **Xavier LAMY**, qui étaient, lors de mon stage, employés pour l'un et en alternance pour l'autre. Je voudrais les remercier pour le temps qu'ils m'ont accordé, pour leurs réponses à mes questions et leur aide précieuse. J'ai beaucoup progressé et c'est à eux que je dois cette progression.

J'aimerais ensuite remercier **Marian DENYS**, pour m'avoir donné la chance de découvrir le côté professionnel d'un milieu qui me passionne. Mais également pour m'avoir fourni des projets variés et d'avoir été là pour m'aiguiller lorsque j'en avais besoin.

Pour terminer, il apparaît évident de remercier **OnlineFormaPro** pour la chance qu'ils m'ont laissée de pouvoir me réorienter professionnellement en intégrant l'Access Code School afin d'apprendre un métier qui me plaît.

Et comment parler d'OnlineFormaPro sans parler d'**Alain MERUCCI**, notre formateur, celui qui nous a appris la majorité de ce que nous savons du métier, celui qui nous a appris à apprendre mais qui nous a également poussé vers l'avant, motivé et qui m'a permis d'en être là où j'en suis actuellement.

Merci pour tout !

Je vous souhaite une très bonne lecture.

Léo

Introduction

J'ai effectué mon stage dans l'agence web DMWeb à Lons-le-Saunier dans le cadre de la formation Développeur Web & Web Mobile. Ce stage a duré du 14 novembre 2022 au 13 janvier 2023.

DMWeb est une agence web qui accompagne ses clients de la conception à la maintenance de leur outil de communication.

L'agence propose une prestation clé en main, le client peut ainsi modifier le contenu de son site grâce au CMS¹ WordPress.

Durant ce stage, j'ai été amené à travailler sur divers projets, 3 d'entre eux étaient des intégrations en HTML / CSS (ou SCSS) / JavaScript sous WordPress. Il m'est arrivé d'utiliser des librairies² pour différentes actions à réaliser.

J'ai également travaillé avec une API (European Sourcing) afin d'importer des produits dans le plugin WordPress (WooCommerce).

Pour finir, j'ai réalisé différents petits projets comme la création d'un formulaire de contact visant à être adaptable sur plusieurs sites différents, ou encore la création d'une page de scores de matchs de football.

Tant de tâches variées qui m'ont fait découvrir différents aspects professionnels d'un développeur web.

1 : Content Management System -> Système de gestion de contenu.

2 : Une librairie est un ensemble de fonctions déjà codées qui permet au développeur de gagner du temps en ne développant pas une fonction déjà existante.

Liste des compétences du référentiel qui sont couvertes par le projet

Front-end :

- Maquetter une application
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique

Back-end :

- Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité
- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Elaborer et mettre en oeuvre des composants dans une application de gestion de contenu ou e-commerce

Projet 1: Front-end

Résumé du projet

Une entreprise industrielle qui travaille l'acier nous a contacté dans le but de réaliser un site vitrine¹ qu'ils pourraient ensuite éditer avec le contenu qu'ils souhaitent. Le projet avait été confié à 2 personnes : Tiana, également stagiaire chez DM Web et moi.

1 : Site qui présente en ligne les produits ou les services d'une organisation dans le but d'en faire la promotion.

Cahier des charges

Le projet qui m'a été confié consiste à créer un site vitrine contenant 8 pages dont un formulaire de contact. à partir d'une maquette qui m'a été donnée. Les pages doivent s'adapter à n'importe quel support (Téléphone, Tablette, Ordinateur, ...). Malheureusement aucune maquette pour une version tablette ou téléphone ne m'avait été fournie, j'ai donc dû adapter les éléments pour rendre la version téléphone du site agréable à visiter.

Tout le contenu des différentes pages du site (sauf le formulaire) doit pouvoir être modifiable facilement par le client.

Le projet est réalisé sur le CMS WordPress ce qui permet d'avoir une administration plus simple du site et une gestion de contenu avancée.

Aucun délai ne m'a été donné lors de la réalisation de ce projet.

Spécifications techniques du projet

CHOIX DES TECHNOLOGIES

Comme dit plus haut, l'entièreté du projet a été faite grâce au système de gestion de contenu WordPress en version 6.1.1.

WordPress permet une administration simplifiée de son site web, grâce à ce CMS, il est facilement possible d'ajouter des pages à son site en utilisant un Template¹ déjà existant et en y ajoutant le contenu de notre choix de différentes manières dont je reparlerais.

Il est actuellement le système de gestion de contenu le plus populaire et le plus utilisé car il est gratuit. Différentes fonctionnalités comme la création de boutiques en ligne, de blogs, etc.

Il est également possible d'ajouter des plugins dans WordPress, ces derniers sont des extensions à WordPress qui permettent plus de fonctionnalités, je présenterais certains de ceux que j'ai utilisés.

La création des templates a été faite en HTML5², PHP³, JavaScript et SCSS⁴.

L'entreprise DMWeb utilise beaucoup le SCSS plutôt que le CSS car il permet beaucoup de libertés telles que la création de variables réutilisables, des fonctions pour étendre le style d'un élément sur un autre élément sans avoir besoin de tout réécrire, ainsi qu'une écriture du code légèrement différente qui, si elle est bien maîtrisée, peut faire gagner énormément de temps.

1 : Un template est un modèle "de base" qui a été créé en amont et qui contient déjà une prédisposition des éléments.

2 : HyperText Markup Language, est un langage de balisage conçu pour les pages web.

3 : PHP : Hypertext Preprocessor, est un langage de programmation pour produire des pages web dynamiques

4 : Syntactically Awesome Style Sheets, est un langage compilé ou interprété par CSS qui permet des fonctionnalités qui ne sont pas disponibles sur du CSS classique (création de variables, indentation améliorée, etc).

Pour que le client puisse facilement changer les contenus des différentes pages, j'ai utilisé le plugin WordPress ACF.

ACF : Advanced Custom Fields est un plugin qui permet de remplir des champs depuis la page administration de WordPress pour pouvoir appeler leur contenu grâce à une fonction dans le code.

Afin de créer facilement un formulaire de contact, j'ai utilisé le plugin GravityForm. Il permet de créer un plugin avec des actions en glisser déposer, et d'être appelé dans le code. Le plugin permet également de gérer les réponses du formulaire directement depuis l'administration WordPress.

J'ai également utilisé la librairie AnimationOnScroll pour ajouter du dynamisme au site en ajoutant des animations sur les différents éléments.

Pour faire fonctionner PHP et donc WordPress en local, j'ai eu besoin de WampServer. Un environnement qui comprend trois serveurs (MySQL, Apache et MariaDB), un interpréteur de script ainsi que PHPMyAdmin qui me permet d'administrer les bases de données MySQL.

MySQL est un système de gestion de bases de données relationnelles. C'est grâce à MySQL que j'ai stocké toutes les informations de mon WordPress.

Afin de versionner (créer des versions) au mieux mon code, j'ai utilisé git, c'est un logiciel de gestion de versions. Il permet de créer différentes versions de son code comme des sauvegardes.

Pour ne pas avoir à gérer tous les versionnages avec les lignes de commande que propose initialement Git, j'ai utilisé SourceTree qui offre une interface graphique à Git et qui permet de gérer ses actions beaucoup plus simplement.

Enfin, toutes les versions créées précédemment sont stockées sur GitLab, un logiciel permettant des dépôts de code source et de gestion de leurs différentes versions.

Pour rédiger mon code, j'ai utilisé Visual Studio Code, un éditeur de code développé par Windows.

Réalisations du candidat

CRÉATION DU THÈME

Le thème WordPress est très important, c'est lui qui gère le style général du site. WordPress propose des thèmes préfabriqués mais ils sont surtout là pour présenter les nouveautés ou pour montrer aux utilisateurs qu'il est possible de créer un thème. Peu de personnes utilisent les thèmes par défaut du CMS. Ils préfèreront alors choisir une des 2 solutions suivantes :

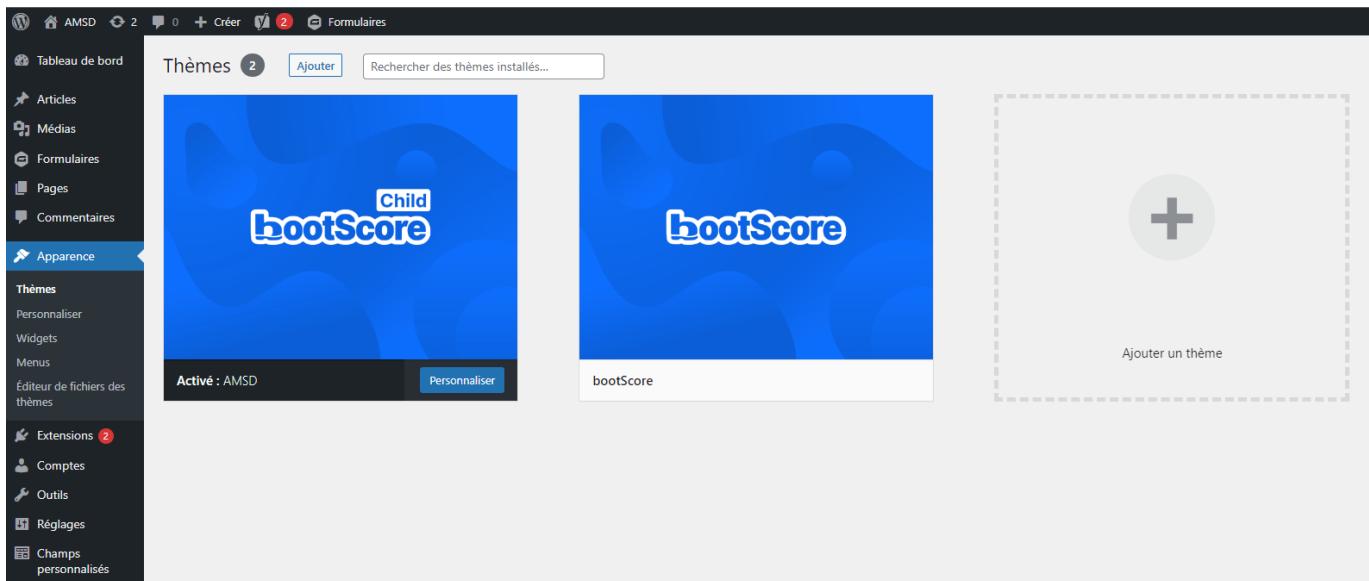
- Utiliser un thème déjà existant qui leur correspond, qu'ils soient gratuits ou payants, il existe une énorme quantité de thèmes déjà créés qui n'attendent qu'à être utilisés. Il est très facile de faire ses recherches depuis l'administration WordPress ou sur internet pour ensuite importer le thème.
- Créer leur propre thème WordPress de zéro, ils devront alors créer les templates de leurs différentes pages et leur appliquer du style. Créer son propre thème est un peu plus long mais il garantit le résultat que le client attend.

Pour ce projet, je suis parti d'un thème existant BootScore. C'est un thème intégrant directement Bootstrap et de nombreux templates déjà existants pour des pages, des posts, des menus, etc.

J'ai pris la version enfant du thème (BootScore Child). Cette version rend la modification du thème encore plus facile, elle prend toutes les caractéristiques du thème de base, mais remplace les parties que l'on souhaite modifier. Si je souhaite modifier la page "404.php" qui correspond à la page "Page non trouvée" du site, il me suffit de créer une page "404.php" dans les dossiers du thème enfant pour qu'elle remplace la page parente du même nom.

Pour installer le thème, il m'a suffit de me rendre sur la page du thème BootScore, de télécharger la version parente (BootScore) et la version que je souhaite, ici, la version BootScore Child.

Je glisse ensuite les différents dossiers dans mon dossier WordPress > WP-Content > themes et il pourra désormais apparaître dans l'administration WordPress, et c'est là bas que je devrais l'activer.



Ici, le thème BootScore a été renommé avec le nom du projet. Il est également possible de personnaliser l'image du thème.

Une fois le thème importé, je peux commencer à créer mes pages, en commençant par la page d'accueil.

CRÉATION DES PAGES

Avant de m'atteler à la création des pages et de leur contenu, il faut créer les blocs qui seront communs à toutes les pages, le header (haut de page) et le footer (bas de page).

Création du header

Je crée le header : header.php dans le dossier du thème, le footer est créé par une stagiaire qui travaillait avec moi au début du projet. Ces fichiers seront appelés en début de page (header.php) et en fin de page (footer.php).

Dans le cas de ce projet, le header contient l'entête HTML et la navbar (barre de navigation / menu).

```

<!doctype html>
<html <?php language_attributes(); ?>>

<head>
    <meta charset=<?php bloginfo('charset'); ?>>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="profile" href="https://gmpg.org/xfn/11">

    <!-- Favicon -->
    <link rel="apple-touch-icon" sizes="180x180" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/favicon-16x16.png">
    <link rel="manifest" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/site.webmanifest">
    <link rel="mask-icon" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/safari-pinned-tab.svg" color="#ff6a26">
    <link rel="shortcut icon" href=<?php echo get_stylesheet_directory_uri(); ?>/favicons/favicon.ico">
    <meta name="msapplication-TileColor" content="#ff6a26">
    <meta name="msapplication-config" content=<?php echo get_stylesheet_directory_uri(); ?>/favicons/browserconfig.xml">
    <meta name="theme-color" content="#ff6a26">

    <?php wp_head(); ?>
</head>

<body <?php body_class(); ?>>

```

Ici, je remplis l'entête de mon HTML qui sera répété sur toutes les pages, j'utilise certaines fonctions de WordPress pour rendre les paramètres de mon site dynamiques.

`language_attributes()` permet d'afficher la langue utilisée.

`bloginfo()` donne des informations relatives au site. Charset correspond au type d'encodage de la page.

`get_stylesheet_directory_uri()` renvoie l'url du répertoire contenant le style du thème actuel.

Les Favicon sont les logos qui sont vus dans l'onglet du navigateur par exemple.

`wp_head()` permet d'appeler les fonctions d'ajout des feuilles de style.

`body_class()` affiche les class pour body, il est possible d'en ajouter une personnalisée depuis cette fonction.

D'autres fonctions sont propres au footer mais je n'ai pas travaillé sur ce dernier.

Le site étant sur plusieurs pages, la création d'un menu était obligatoire. WordPress embarque une création de menus, j'ai donc utilisé cette dernière pour faire le menu du site.

Dans un premier temps, il faut se rendre dans l'administration de WordPress afin de créer le menu et de définir les différentes pages vers lesquelles il renvoie.

Structure du menu

Nom du menu Principal 

Glissez chaque élément pour les placer dans l'ordre que vous préférez. Cliquez sur la flèche à droite de l'élément pour révéler d'autres options de configuration.

Sélection groupée

Industries	Page ▾
Équipements industriels	Page ▾
Agro-alimentaire	Page ▾
Collectivités urbaines	Page ▾
Service après-vente	Page ▾
Parc machines	Page ▾

Sélection groupée [Supprimer les éléments sélectionnés](#)

Réglages du menu

Ajoutez automatiquement des pages Ajouter automatiquement les pages de premier niveau à ce menu

Afficher l'emplacement Main menu
 Footer menu (Actuellement réglé sur : Principal bas de page)
 Footer Secondary Menu (Actuellement réglé sur : Secondaire bas de page)

[Supprimer le menu](#)

Mon menu s'appelle "Principal" il contient 6 pages. Le bloc du bas, "Réglages du menu" désigne l'emplacement où se situera le menu.

Pour appeler le menu en question dans le code de mon header.php, la marche à suivre est simple :

```
<?php wp_nav_menu( array(  
    'theme_location' => 'main-menu',  
) ); ?>
```

Le menu généré peut ensuite être mis en forme en css comme il serait possible de le faire à la main.

Pour accéder au menu, il faut cliquer sur un bouton qui l'affichera, il n'y a donc pas de second menu à créer pour le mobile mais simplement un réajustement pour l'adapter aux différents formats.

Création des templates

Une fois les “bases communes” créées, il a fallu créer le contenu des 8 pages du site.

WordPress a un fonctionnement qui lui est propre et dispose d'une hiérarchie particulière : WP-Hierarchy.

Je me suis servi de cette hiérarchie pour créer un template par page. Chaque template appelle le header et le footer grâce à des fonctions de wordpress.



Ce schéma explique comment WordPress appelle les différentes pages en fonction des différentes actions.

Ici, WP-Hierarchy me dit que pour créer la page d'accueil, il faut que je crée la page "front-page.php". Je me rends donc dans le dossier WordPress, et j'entre dans le fichier du thème que j'ai importé précédemment pour créer ma page : "front-page.php".

404.php	266	Fichier sou...	01/12/2022 15:22:36	0644	2137 2144
footer.php	1313	Fichier sou...	30/11/2022 14:48:48	0644	2137 2144
front-page.php	7430	Fichier sou...	05/12/2022 08:44:21	0644	2137 2144
functions.php	2659	Fichier sou...	02/12/2022 10:32:17	0644	2137 2144

On voit que footer.php y est également, c'est le bas de page créé en amont.

La page d'accueil "front-page.php" appelle les 2 parties récurrentes du site

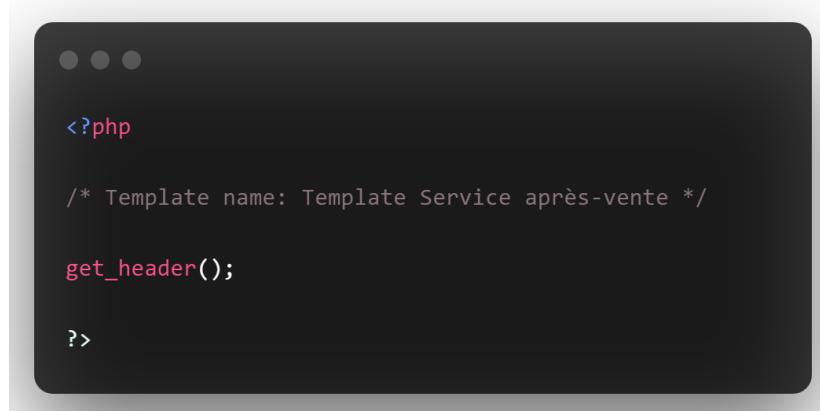


```
<?php  
get_header();  
?>
```

```
<?php  
get_footer();  
?>
```

A gauche, la fonction permettant d'appeler le fichier "header.php" et à droite, celle qui permet d'appeler le fichier "footer.php".

Pour la page principale (front-page), WordPress considère directement que ce sera la page principale du site. Cependant pour les autres pages je vais devoir lui informer qu'il s'agit d'un template.



```
<?php  
  
/* Template name: Template Service après-vente */  
  
get_header();  
  
?>
```

Grâce à ce commentaire, WordPress comprend que c'est un template (ici le "Template Service après-vente") et pourra l'ajouter à la liste des templates ainsi, je pourrais le sélectionner à la création de mes pages.

C'est entre les 2 fonctions précédemment citées que doit se situer le contenu de la page, ici de la page principale.

Pour réaliser le contenu des sites, je procède généralement par blocs :

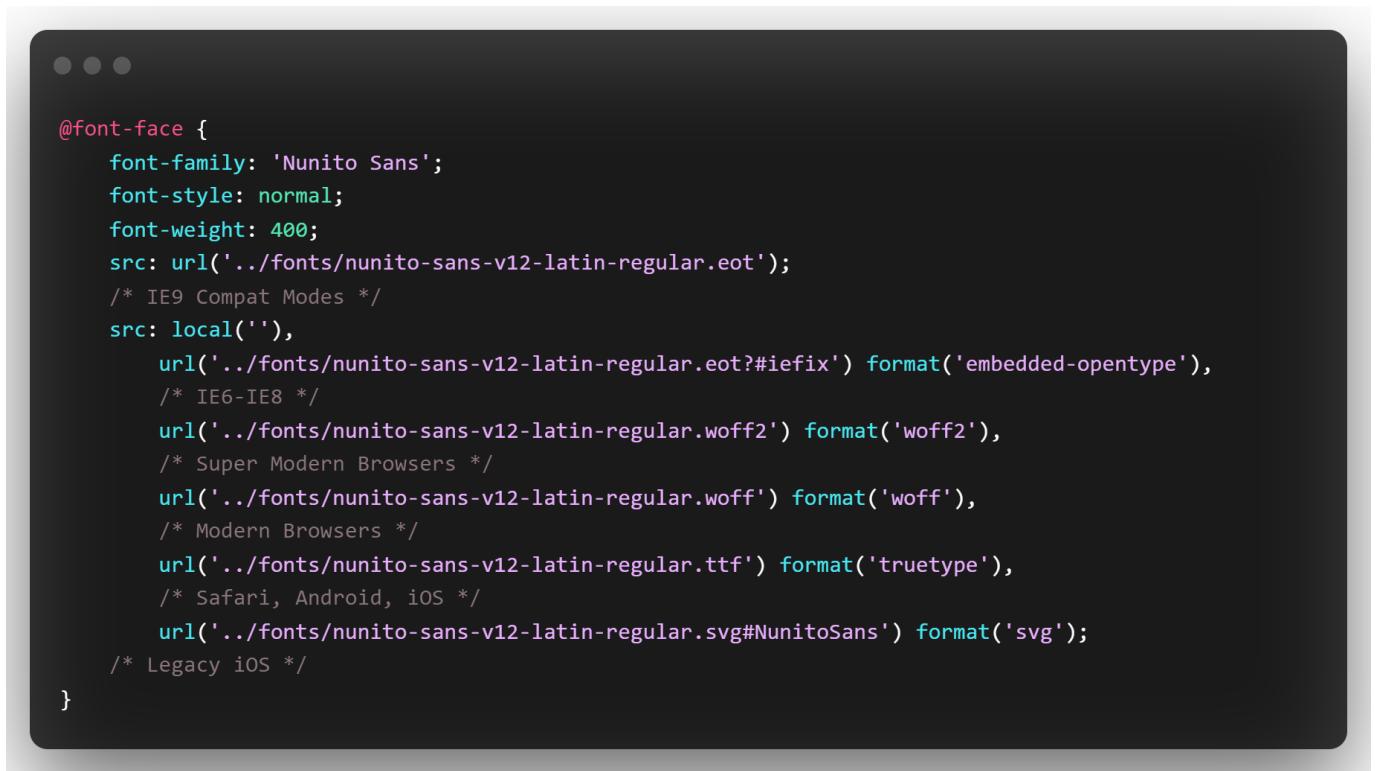
- Je crée la structure HTML du bloc
- Je met en forme le bloc
- Je travail le responsive du bloc

Grâce à ces 3 étapes, j'avance bloc après bloc et je m'assure d'avoir terminé le bloc précédent avant d'en commencer un nouveau.

La police utilisée pour le projet est la “Nunito Sans”, pour pouvoir utiliser cette police, j'ai dû l'importer, il ne s'agissait que d'une police mais en réalité, j'ai dû importer toutes ses variantes, celles compatibles avec les différents navigateurs et celles pour les différentes graisseurs de police (font-weight).

Heureusement, un site nommé Google Font Helper permet de sélectionner la police en question, ses différentes graisseurs de police et permet ensuite de générer le code et de télécharger directement toutes les polices.

Ainsi, je crée un dossier “fonts” pour stocker toutes les polices téléchargées et je peux mettre les liens d'import dans mon dossier SCSS.



```
@font-face {
    font-family: 'Nunito Sans';
    font-style: normal;
    font-weight: 400;
    src: url('../fonts/nunito-sans-v12-latin-regular.eot');
    /* IE9 Compat Modes */
    src: local(''),
        url('../fonts/nunito-sans-v12-latin-regular.eot?#iefix') format('embedded-opentype'),
        /* IE6-IE8 */
        url('../fonts/nunito-sans-v12-latin-regular.woff2') format('woff2'),
        /* Super Modern Browsers */
        url('../fonts/nunito-sans-v12-latin-regular.woff') format('woff'),
        /* Modern Browsers */
        url('../fonts/nunito-sans-v12-latin-regular.ttf') format('truetype'),
        /* Safari, Android, iOS */
        url('../fonts/nunito-sans-v12-latin-regular.svg#NunitoSans') format('svg');
    /* Legacy iOS */
}
```

Voici à quoi ressemble une des variantes de la police, ici le font-size par défaut.

Ce code peut être inséré directement dans le dossier CSS ou SCSS mais le SCSS permet une autre manière de procéder plus ordonnée.

Dans mon dossier CSS, là où mon style se trouve, j'ai créé un fichier “theme” qui contient le fichier “_fonts.scss”.

Le SCSS, comme le PHP, permet d'importer des fichiers depuis d'autres fichiers,

depuis mon SCSS principal, j'ai importé le fichier contenant les imports de polices grâce à :

```
@import 'theme/_fonts.scss';
```

De la même manière, j'ai créé un dossier "partials" contenant le fichier "_base.scss" qui a pour objectif de généraliser le CSS, je défini une police par défaut, une taille de police par défaut, des marges etc ... Tout celà pourra être modifié par la suite dans mon fichier SCSS principal ou directement dans "_base.scss".

Pour l'intégration des blocs, j'ai pu remarquer que certains avaient un patronne qui se répète plusieurs fois sur les différentes pages du site.

Il m'a alors suffit de créer le bloc une fois et de le réutiliser autant de fois que nécessaire.

 INDUSTRIES

Cartérisation

AMSD met ses compétences à la disposition de ses clients pour protéger leurs moyens de fabrication. Elle conçoit des bâts de machines, des grilles de protection, des capots, des boîtiers, etc. Selon les besoins, elle utilise des matériaux de 10 à 40 mm d'épaisseur qu'elle recouvre d'une peinture epoxy liquide. Sa capacité de levage est de 2 tonnes.



Si je prends l'exemple de ce bloc, il est identique à celui d'une autre page que voici

Découpage/ équipements pour presse

En partenaire historique des industries du découpage, AMSD conçoit les équipements qui conviennent le mieux à leurs besoins : bennes à copeaux ou déchets, goulottes d'évacuation, convoyeurs à rouleaux, empileurs filaires et tubes, flasques et dévidoirs, cabines d'insonorisation.



Seul son contenu diffère, il est alors utile de ne pas refaire le même travail qui ne ferait qu'alourdir le site car je devrais ajouter du CSS supplémentaire.

En ce qui concerne le responsive, le site contenait des éléments graphiques qui n'auraient pas eu un effet très convainquant en version mobile.

Étant donné l'absence de maquette pour cette version. J'ai pris la décision de supprimer certains éléments graphiques non essentiels pour de meilleures performances du site et une meilleure expérience utilisateur (certains éléments étant trop peu visibles pour valoir la peine d'être affichés).

Ajout d'animations

Pour rendre son site plus dynamique et agréable à consulter, le client souhaite que son site comporte quelques animations sur les différents blocs composants le site.

Afin de ne pas avoir à créer des animations déjà existantes, j'ai décidé de m'aider d'une librairie d'animations. Pour ce projet, j'ai utilisé "AnimationOnScroll" ou "AOS".

Il m'a fallu importer un fichier CSS et un fichier JavaScript disponibles tous les 2 sur le site de la librairie.

J'ai ensuite dû initialiser la librairie en ajoutant AOS.init(); dans mon fichier JavaScript.

Ensuite, son utilisation est plutôt simple. Il existe plusieurs animations disponibles et toutes se déclenchent lorsque l'utilisateur scroll sur la page (navigue sur la page en utilisant sa molette).

Pour définir quelle animation je souhaite sur un élément en particulier, je dois ajouter data-aos:"*le nom de l'animation*". Toutes les informations de la librairie sont disponibles sur son site.



Par exemple, ici j'ai appelé l'animation fade-zoom-in. Elle possède l'effet ease-in-back, elle met 300ms à se réaliser. "Offset" désigne qu'il faut scroller davantage pour qu'elle apparaisse. Ici, il est par défaut : à 0. D'autres fonctionnalités sont disponibles pour cette librairie, il est possible de la personnaliser comme on le souhaite.

AJOUT DES CHAMPS D'ÉDITION

Comme dit plus haut, le contenu du site doit être intégralement éditabile par le client. Ce qui signifie que si ce dernier décide de changer les images, ou les textes, il devra être en capacité de le faire sans difficultés.

Pour résoudre cette problématique, j'ai utilisé le plugin WordPress Advanced Custom Fields (ACF).



Sous-titre deuxième bloc

SECTEURS D'ACTIVITÉ

Titre deuxième bloc

Des applications pour tous secteurs d'activité.

Description deuxième bloc

Grâce au savoir-faire accumulé par deux générations, AMSD s'adapte aux demandes des professionnels comme aux besoins ponctuels des particuliers.

Son utilisation par le client est plutôt facile, il devra se rendre sur la page qu'il souhaite modifier et sélectionner le champ correspondant à ce qu'il souhaite modifier.

<input type="checkbox"/> Titre	Description	Cle	Emplacement	Champs
<input type="checkbox"/> Agro-alimentaire		af group_6389c9be4ec06	Pages	10
<input type="checkbox"/> Collectivités urbaines		af group_6389c9db0e8b7	Pages	10
<input type="checkbox"/> Équipements industriels		af group_6389c98de63b5	Pages	15
<input type="checkbox"/> Industries		af group_6389c6ebd2484	Pages	15
<input type="checkbox"/> Page d'accueil		af group_6389b63d99803	Pages	21
<input type="checkbox"/> Parc machines		af group_6389ce8a82157	Pages	5
<input type="checkbox"/> Service après-vente		af group_6389cdc43c322	Pages	5
<input type="checkbox"/> Titre	Description	Cle	Emplacement	Champs

Pour mettre en place ces champs, il suffit d'aller dans l'administration du site, d'aller sur la page de gestion du plugin et de créer, dans un premier temps, les groupes de champs, j'en ai créé un par page pour mieux me repérer.

Champs			
N°	Libellé	Nom	Type
1	> Sous-titre premier bloc	subtitle_bloc1	Texte
2	> Titre premier bloc	title_bloc1	Zone de texte
3	> Description 1 premier bloc	content1_bloc1	Zone de texte
4	> Description 2 premier bloc	content2_bloc1	Zone de texte
5	> Image premier bloc	image_bloc1	Image
6	> Image premier bloc pour mobile	image_bloc1_mobile	Image

Ensuite, pour chaque groupe de champs, j'ai créé les champs nécessaires, ils varient en fonction du contenu de la page et des choses que nous voulons rendre éditables.

Différents types de champs sont disponibles, il est également possible de mettre des images.

ACF permet, grâce à un système de conditions, de gérer où chaque champ sera affiché.

The screenshot shows the ACF 'Règles' (Rules) section. It displays a condition: 'Afficher ce groupe de champs si' (Display this group of fields if). The condition is set to 'Page' is equal to 'Accueil'. There is also an 'ou' (or) option available. The interface includes tabs for 'Règles de localisation', 'Présentation', and 'Réglages du groupe'. A help icon and a 'et' (and) button are also visible.

Ici, j'affiche le groupe de champs que j'ai créé quand la page est "Accueil".

Une fois tous les champs créés, il faut appeler leur contenu dans le code HTML du template de la page.

```
<?php
$image_bloc1 = get_field('image_bloc1');
$image_bloc1_mobile = get_field('image_bloc1_mobile');
?>
<?php if($image_bloc1): ?>
<picture>
    <source media="(min-width: 770px)" srcset="<?php echo
esc_url($image_bloc1['url']); ?>">
    <source media="(max-width: 769px)"
srcset="<?php echo esc_url($image_bloc1_mobile['url']); ?>">

</picture>
<?php endif; ?>
```

Le cas de l'import de l'image est bien particulier car j'ai utilisé la balise picture qui permet de donner plusieurs images, par ordre de priorité, au navigateur qui choisira ensuite en fonction de ce qu'il est capable d'afficher (certains navigateurs ne supportant pas quelques formats d'image).

Dans un premier temps, je récupère les images à afficher, une version pour ordinateur et une autre pour mobile.

C'est la fonction `get_field('le nom de mon champ')`; qui me permet d'appeler le champ en question.

Je mets ensuite en place une condition pour que ma balise picture ne s'affiche que si une image pour ordinateur est sélectionnée, cette condition est clôturée par un `endif;` plus bas.

Grâce à WordPress, lorsque je sélectionne une image pour la mettre dans mon champ, il récupère directement quelques informations utiles pour notre image, notamment celles qui nous seront utiles ici : sa largeur (`width`), sa hauteur (`height`), son url, et sa description (`alt`).

Je définis ensuite les attributs de la balise picture avec les informations de l'image.

```
<source media="(min-width: 770px)" srcset="<?php echo esc_url($image_bloc1['url']); ?>">
```

Lorsque la taille de l'écran est supérieure à 770px, j'appelle l'image "classique" en récupérant son url dans le tableau contenant toutes les informations relatives à cette image : `"$image_bloc1"` que j'obtiens grâce à l'appel du champ ACF.

```
<source media="(max-width: 769px)" srcset="<?php echo esc_url($image_bloc1_mobile['url']); ?>">
```

Je fais la même chose lorsque la taille d'écran est inférieure à 770px en récupérant les informations de l'image mobile.

```
<img class="radius-left" src=<?php echo esc_url($image_bloc1['url']); ?>
    alt=<?php echo esc_attr($image_bloc1['alt']); ?>
    width=<?php echo esc_attr($image_bloc1['width']); ?>
    height=<?php echo esc_attr($image_bloc1['height']); ?>
    data-aos="fade-up"
    data-aos-duration="500"
    data-aos-offset="150">
```

Pour finir, je définis une image “par défaut” pour les navigateurs ne supportant pas la balise picture. Avec des informations supplémentaires, comme sa largeur et hauteur, sa description et, évidemment, sa source.

FORMULAIRE DE CONTACT

La plupart des sites vitrine possèdent un formulaire de contact. Après avoir exposé tous ses biens et services, le propriétaire met à disposition un formulaire de contact afin de faciliter le contact avec son potentiel futur client ou pour tous retours.

Le plugin “GravityForm” permet de créer des formulaires sur mesure sans avoir à coder.

Pour créer le formulaire, il suffit de se rendre sur l’administration WordPress, de créer le formulaire et d’y ajouter les champs que l’on souhaite.

Il est ensuite possible d’ajouter des conditions (apparition de champs supplémentaires en cas de réponses particulières), de rendre certains champs obligatoires (le formulaire ne s’envadera pas si ce champ n’est pas complété)

Société *	Civilité
<input type="text"/>	<input type="text" value="Monsieur"/>
Nom *	Prénom *
<input type="text"/>	<input type="text"/>
Email *	Téléphone
<input type="text"/>	<input type="text"/>

C'est ici que je peux ajouter des champs en "glisser déposer" et gérer les réglages de ces derniers.

Le plugin embarque le système de protection anti-spam ReCaptcha. Il différencie un humain d'un robot lors de l'envoi du formulaire.

Pour récupérer les réponses du formulaire, le client peut aller sur l'administration et y voir toutes les réponses, il peut aussi choisir de les recevoir par mail.

Projet 2 : Back-end

Résumé du projet

Je n'ai pas eu de contexte précis pour ce projet mais je sais qu'un client possède un site sur lequel il répertorie tout ce qu'il vend. Il souhaite ajouter une liste de produits de différents fournisseurs à son site.

Cahier des charges

A l'aide de l'API application programming interface ou « interface de programmation d'application », je devais récupérer les différents fournisseurs et chacun de leurs produits.

Je devais ensuite insérer chaque produit sur WooCommerce, un plugin de WordPress destiné aux e-commerce.

Il fallait également vérifier les catégories et les ajouter aux produits correspondants et ajouter des attributs pour les produits variables.

Je n'ai pas eu d'intégration à faire pour ce projet. Je devais créer le script de récupération des données depuis l'API European Sourcing et les intégrer à WooCommerce en prenant en compte les catégories déjà existantes et en les ajoutant aux produits correspondants.

Spécifications techniques du projet

CHOIX DES TECHNOLOGIES

Comme le projet précédent, tout le site a été fait sur WordPress.

Toutes les informations ont été récupérées depuis l'API European Sourcing.

Une API ou Application Programming Interface qui se traduit par interface de programmation d'application est un moyen de communication entre un utilisateur et un fournisseur. Grâce à une API, il est possible d'accéder à des informations spécifiques qui ont été rendues publiques par le fournisseur (celui qui a créé l'API).

Pour faire la connexion à l'API et l'import de toutes les informations et des produits, j'ai utilisé WordPress. L'import se fait vers le plugin WooCommerce.

WooCommerce est un plugin permettant la création d'une boutique en ligne (e-commerce). Il permet la création de produits et intègre plusieurs systèmes de paiement sécurisé. c'est un excellent moyen de se créer une boutique en ligne facilement.

J'ai utilisé les mêmes logiciels pour le versionnage ou l'édition de code que pour le projet précédent.

Réalisations du candidat

RÉCUPÉRATION DES INFORMATIONS DES PRODUITS

Toutes les informations dont j'ai besoin se trouvent sur l'API European Sourcing.

La plupart des API nécessitent une clé afin d'être utilisées, elle est généralement obtenue à l'achat d'un abonnement, si l'API est payante, ou à la création du compte, si elle est gratuite. Pour ma part, la clé API m'a été fournie par mon patron.

Pour savoir comment récupérer les informations dont j'ai besoin, j'ai eu besoin de savoir comment fonctionne l'API. J'ai donc consulté la documentation, elle me donne toutes les fonctions que j'ai besoin d'utiliser pour faire une requête à l'API.

La méthode utilisée était un cURL (client URL) cette dernière doit contenir le détail des informations que je souhaite recevoir, où la requête doit être envoyée, et plusieurs paramètres garantissant la sécurisation de la requête.

Le cURL est une commande qui permet de communiquer avec des serveurs. Il peut être utilisé pour envoyer des requêtes à des serveurs et recevoir des réponses en retour. Par exemple, pour envoyer une requête à un site web et recevoir le code HTML de la page en réponse. Dans mon cas, j'envoie des requêtes à l'API de European Sourcing et j'attends les informations de tous les produits en retour.

```

$inputJson = '{
    "lang": "fr",
    "include_fields": [],
    "exclude_fields": [],
    "limit": '.$limit.',
    "search_handlers": [
        {
            "supplier_id": {
                "include": [
                    "'.$idSupplier.'"
                ]
            }
        }
    ]
}';

$ch = curl_init();
$certificate_location = '../cacert.pem';
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, $certificate_location);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, $certificate_location);
curl_setopt($ch, CURLOPT_URL, 'https://product- api.europeansourcing.com/api/v1.1/search');

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/ld+json',
    'Accept: application/ld+json',
    'X-AUTH-TOKEN: ' . $this->token
));
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $inputJson);
$response = curl_exec($ch);

if (false === $response) {
    echo 'Curl error: ' . curl_error($ch);
    die();
}

curl_close($ch);
$response = json_decode($response, true);

```

Voici un exemple de cURL : \$inputJson contient le détail de ce que je veux comme réponse, ici une réponse en français, avec une limite fixée par une variable et provenant des “suppliers” fournisseurs en français qui sont définis également par une variable.

Je commence ensuite par initialiser le cURL, et lui fournir son certificat SSL ainsi que l’URL qu’il devra utiliser pour sa requête.

Un certificat SSL (Secure Sockets Layer) est un fichier de données électroniques qui permet de sécuriser les communications sur internet. Il est principalement utilisé pour sécuriser les connexions au site web d'une entreprise et est généralement installé sur le serveur web de cette entreprise.

Ici, travaillant en local, il est installé dans les dossiers de mon site. Je lui fournis le chemin pour qu'il puisse y accéder.

Je n'ai pas le détail de chaque "setopt", mais elles permettent ensemble de créer une requête propre et sécurisée.

\$this->token représente la clé API que je définis au début de mon document. Elle peut être susceptible de changer, je la mets donc en variable pour pouvoir la modifier plus facilement.

Une fois toutes les options configurées, j'exécute la requête, si cette dernière est correctement interprétée et contient des résultats, je stocke le résultat dans une variable. La réponse est envoyée au format JSON (JavaScript Object Notation), c'est un langage facile à lire par l'homme mais il faudra cependant le décoder (parser) pour que PHP le comprenne et puisse le traiter.

Pour pouvoir s'entraîner à envoyer de bonnes requêtes à l'API, European Sourcing met en place une documentation ainsi qu'une SandBox, un bac à sable sur lequel on peut créer et envoyer facilement des requêtes et obtenir des retours afin d'affiner nos requêtes.

```

array (size=23)
  'has_marking' => boolean false
  'last_indexed_at' => string '2023-01-08T15:06:47.758894' (Length=26)
  'project_key' => string 'es' (Length=2)
  'country_of_origin' => null
  'project' =>
    array (size=2)
      'id' => int 695609
      'key' => string 'es' (Length=2)
  'created_at' => string '2022-05-13T09:41:50' (Length=19)
  'variants' =>
    array (size=1)
      0 =>
        array (size=30)
          'last_price_update' => null
          'keywords' =>
            array (size=0)
              ...
          'supplier_profiles' =>
            array (size=1)
              ...
          'variant_sizes' =>
            array (size=2)
              ...
          'last_image_update' => string '2022-07-12T14:07:32' (length=19)
          'description' => string 'Mini lapin Lindt HELLO en chocolat au lait de qualité supérieure dans un'
          'marking_additional_information' => null
          'supplier_reference' => string '131' (Length=3)
          'net_weight' => float 10
          'variant_packaging' =>
            array (size=7)
              ...
          'main_variant_image_id' => int 27340315
          'variant_minimum_quantities' =>
            array (size=1)
              ...
          'project_id' => int 5029795
          'variant_prices' =>
            array (size=0)
              ...
          'id' => int 5029795
          'stock' => null
          'gross_weight' => null
          'slug' => string 'mini-lapin-lindt-hello-dans-un-carton-personnalisable-oeuf-de-paques' (Length=68)
          'variant_images' =>
            array (size=1)
              ...
          'raw_description' => string 'Mini lapin Lindt HELLO en chocolat au lait de qualité supérieure dans'
          'european_article_numbering' => null
          '...': ...
        ...
      ...
    ...
  ...

```

Voici à quoi ressemble une partie de la réponse d'un produit. Je dispose de beaucoup d'informations sur le produit, cependant, bon nombre d'entre elles ne me seront pas utiles dans mon cas.

J'ai été confronté à 2 types de produits durant ce projet :

- Les produits simples, qui ne possèdent pas de variantes (ou plutôt une seule), il existe un produit et aucune manière de changer ses attributs comme sa couleur, sa taille, etc.
- Les produits variables, ils possèdent au minimum 2 variantes, l'un des attributs du produit est modifiable. Le client pourra choisir celui qu'il souhaite.

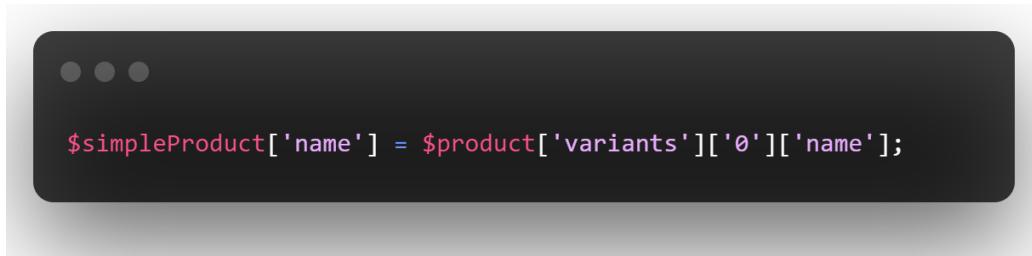
Il est important de bien séparer les produits variables des produits simples car la manière de les importer n'est pas la même.

Le produit ci-dessus ne comporte qu'une variante, cela signifie que c'est un produit simple.

A partir de là, il n'est pas dur de mettre une condition pour repérer de quel type de produit il s'agit et assigner des fonctions en conséquence.

Pour sélectionner l'élément qui m'intéresse, je dois me déplacer comme je le ferais dans un tableau en PHP (il s'agit d'un gros tableau comportant d'autres tableaux).

Si \$produit correspond au tableau vu au dessus, alors pour obtenir le nombre de variants, il me suffit de faire count(\$produit['variants']); count sert à connaître le nombre d'éléments dans le tableau.



Ici, \$product correspond au tableau entier, je me déplace ensuite dans la partie 'variants', je prends l'index 0 car un produit simple aura forcément une seule variante. Je récupère ensuite 'name' pour avoir le nom du produit.

IMPORTATION PRODUIT SIMPLE

Maintenant que j'ai accès aux informations et que je sais comment les récupérer précisément, il faut que j'importe un produit dans WooCommerce.

Je commence par les produits simples, ils nécessitent moins d'informations pour être créés.

```
//INSERT A SIMPLE PRODUCT ON WOOCOMMERCE
public function insertSimpleProduct($dataProduct){
    //Import Image in WordPress
    $productImageID = $this->importUrl($dataProduct['url']);
    $categories = [];

    $product = new WC_Product_Simple();
    $product->set_name($dataProduct['name']);
    $product->set_sku($dataProduct['id']);
    $product->set_image_id($productImageID);
    $product->set_description($dataProduct['desc']);

    $categories = $this->checkCategories($dataProduct['categories']);
    $product->set_category_ids($categories);
    $product->save();
}
```

Voilà toutes les informations nécessaires à la création d'un produit dans mon cas :

- Un nom (name)
- Un ID (ou SKU / UGS en français, Unité de Gestion des Stocks), unique au produit
- Une image
- Une description

J'ajoute également les catégories qui seront abordées plus en détail plus tard.

Pour importer un produit en PHP, WooCommerce met à disposition un objet. Il est alors plutôt facile de créer le produit.

J'ai également dû importer les images à WordPress, je mettrai la fonction que j'ai utilisé en Annexe.

La fonction `insertSimpleProduct()` prend comme variable toutes les informations dont j'ai besoin pour la création d'un produit, j'appelle la fonction autant de fois qu'il y a de produits simples à ajouter.

IMPORTATION PRODUIT VARIABLE

Il a été beaucoup plus difficile pour moi de créer un produit variable à cause des attributs. Il faut créer envoyer les informations du produit “de base” puis ajouter une liste d’attributs dans WooCommerce pour ensuite les assigner au produit correspondant puis créer des variations correspondantes aux attributs.

Tout cela m’a causé beaucoup de problèmes et c’est ce qui m’a pris le plus de temps.

Mais avant de voir comment importer un produit variable, il faut voir comment récupérer toutes les informations de chaque variante, et également les attributs.

```
for ($i=0; $i < count($product['variants']) ; $i++) {
    $variableProduct['variantId'][$i] = $product['variants'][$i]['id'];
    $variableProduct['desc'] = $product['variants'][$i]['raw_description'];

    $variableProduct['name'] = $product['variants'][$i]['name'];
    $variableProduct['url'][$i] = $product['variants'][$i]['variant_images'][0]
        ['url'];
    for ($j=0; $j < count($product['variants'][$i]['attributes']) ; $j++) {

        $attributes['variants'][$i]['attributes'][$j] = ['name' =>
            $product['variants'][$i]['attributes'][$j]['attribute_group']['name'],
            'value' =>
            $product['variants'][$i]['attributes'][$j]['value']];
    }
}
$variableProduct['categories'] = array_reverse($product['categories']);
$this->insertVariableProduct($variableProduct, $attributes);
```

Il m'a fallu faire beaucoup de boucles for() qui contenaient elles-mêmes des boucles for(), il faut alors faire attention pour bien se retrouver dans les boucles et savoir quel index appeler à quel moment.

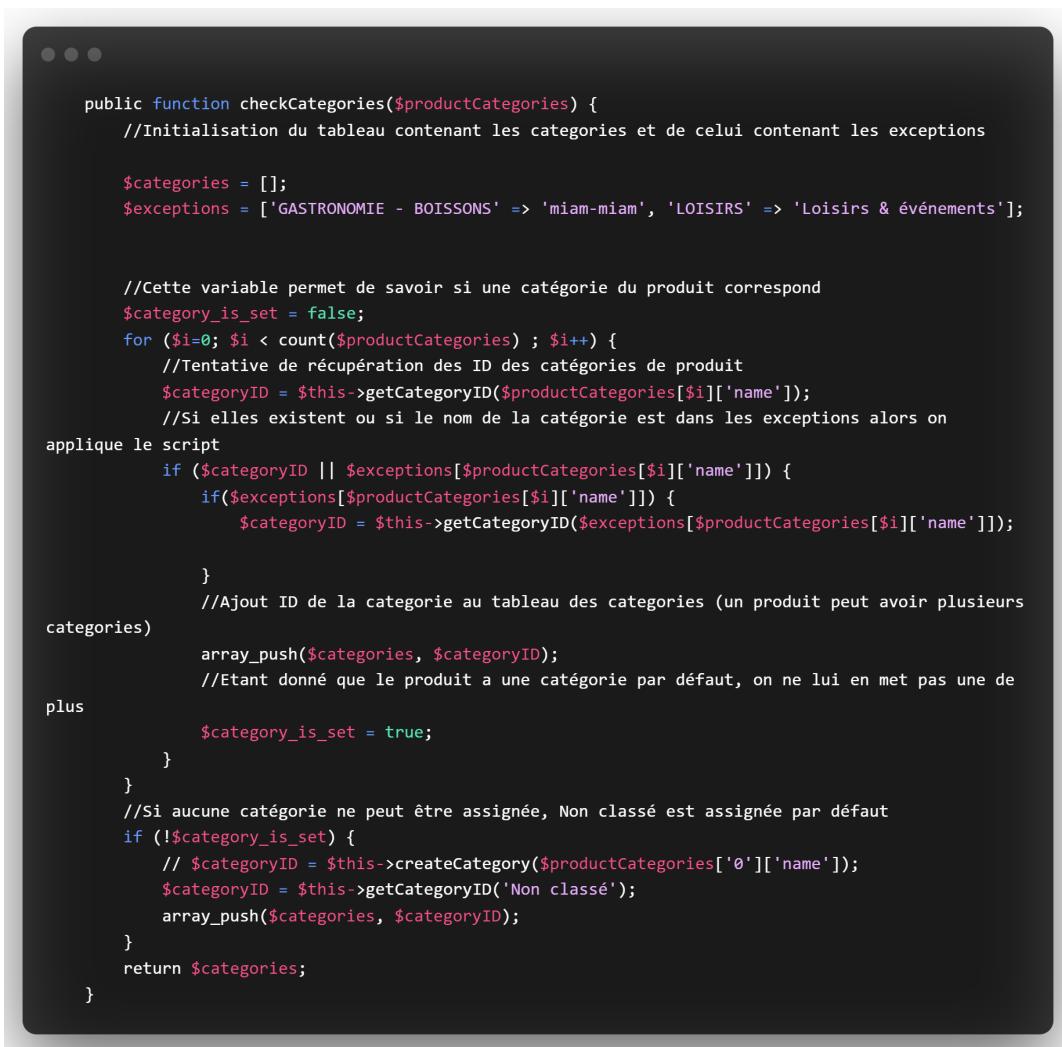
Pour commencer, je mets une boucle sur les variantes et je récupère l'id, la description, le nom et la première image de chaque variante.

Je récupère également toutes les catégories qui me serviront plus tard.

Ensuite, je crée une boucle dans la boucle précédente pour récupérer tous les attributs de toutes les variantes, par exemple, si il y a 2 attributs par variantes et 3 variantes, alors j'ai récupéré 6 attributs.

Je stocke tous ces attributs dans une variable pour pouvoir m'en servir dans ma fonction d'import plus tard. Fonction que j'appelle à la dernière ligne en lui fournissant les informations relatives au produit et ses variantes et les informations des attributs que je vais devoir créer et attribuer.

Le site étant déjà existant, je me sers de la liste des catégories déjà créées, je ne peux pas en créer d'avantage. Il faut alors que je compare les catégories données par l'API pour le produit (elle en donne 3) avec celles déjà existantes, si les catégories correspondent, alors que je les assigne. Si aucune des 3 catégories ne convient, alors j'ajoute une catégorie "Non classé". J'ai également mis en place une liste d'exceptions pour le cas où la catégorie existe mais sous un nom légèrement différent.



```
public function checkCategories($productCategories) {
    //Initialisation du tableau contenant les categories et de celui contenant les exceptions

    $categories = [];
    $exceptions = ['GASTRONOMIE - BOISSONS' => 'miam-miam', 'LOISIRS' => 'Loisirs & événements'];

    //Cette variable permet de savoir si une catégorie du produit correspond
    $category_is_set = false;
    for ($i=0; $i < count($productCategories) ; $i++) {
        //Tentative de récupération des ID des catégories de produit
        $categoryID = $this->getCategoryID($productCategories[$i]['name']);
        //Si elles existent ou si le nom de la catégorie est dans les exceptions alors on
        applique le script
        if ($categoryID || $exceptions[$productCategories[$i]['name']]) {
            if($exceptions[$productCategories[$i]['name']]) {
                $categoryID = $this->getCategoryID($exceptions[$productCategories[$i]['name']]);
            }
            //Ajout ID de la categorie au tableau des categories (un produit peut avoir plusieurs
            categories)
            array_push($categories, $categoryID);
            //Etant donné que le produit a une catégorie par défaut, on ne lui en met pas une de
            plus
            $category_is_set = true;
        }
    }
    //Si aucune catégorie ne peut être assignée, Non classé est assignée par défaut
    if (!$category_is_set) {
        // $categoryID = $this->createCategory($productCategories['0']['name']);
        $categoryID = $this->getCategoryID('Non classé');
        array_push($categories, $categoryID);
    }
    return $categories;
}
```

Voici le code commenté de la création d'un produit variable avec ajout des variations et des attributs.

```

//INSERT A VARIABLE PRODUCT ON WOOCOMMERCE WITH ATTRIBUTE AND VARIATION
public function insertVariableProduct($dataProduct, $dataAttributes){

    //Importation image via la fonction en Annexe
    $productImageID = $this->importUrl($dataProduct['url'][0]);

    //Appel objet de création de produit variable
    $product = new WC_Product_Variable();

    //Je définis les informations principales du produit, son nom, son id, son image, sa
description
    $product->set_name($dataProduct['name']);
    $product->set_sku($dataProduct['id']);
    $product->set_image_id($productImageID);
    $product->set_description($dataProduct['desc']);

    //Récupère tous les attributs pour un produit
    $attributes = $this->getAttributesByProduct($dataAttributes);
    $attributeList = array();
    foreach ($attributes as $key => $attribute) {
        // Je définis la taxonomy (nom attribut)
        $taxonomy = $attribute[0];
        //Je détruis la variable attribute[0]
        unset($attribute[0]);
        //Je crée le tableau des termes
        $terms = [];
        //Je les ajoute un à un, via la fonction addTerm() disponible plus bas
        foreach ($attribute as $term) {
            $this->addTerm($taxonomy, $term);
            array_push($terms, $term);
        }
        //Appel objet de création d'attribut
        $attribute = new WC_Product_Attribute();
        $attribute->set_name($taxonomy);
        $attribute->set_options($terms);
        $attribute->set_position(0);
        $attribute->set_visible(true);
        //Je ne souhaite avoir que couleurs comme variations, je pourrais ajouter d'autres
variations par la suite
        if($this->slugify($attribute['name']) == 'couleurs'){
            $attribute->set_variation(true);
        }
        else {
            $attribute->set_variation(false);
        }
        $attributeList[] = $attribute;
    }
    //association de tous les attributs du produit à ce dernier
    $product->set_attributes($attributeList);
    //Je vérifie et récupère tous les ID des catégories qui correspondent
    $categories = $this->checkCategories($dataProduct['categories']);
    //et je les assigne au produit
    $product->set_category_ids($categories);
    //Je sauvegarde le produit dans WooCommerce
    $product->save();

    //Création des variants
    for ($i=0; $i < count($dataAttributes['variants']); $i++) {
        //Importation l'image de chaque variante dans WordPress
        $productImageID = $this->importUrl($dataProduct['url'][$i]);
        //Appel de l'objet de création de variantes
        $variation = new WC_Product_Variation();
        //Je définis quel est le produit à qui cette variante va être associée
        $variation->set_parent_id($product->get_id());
        //ajout ID unique de la variante ainsi que son image
        $variation->set_image_id($productImageID);
        $variation->set_sku($dataProduct['variantid'][$i]['id']);
        //Pour chaque attribut, j'ajoute une variation, ici j'ai 2 attributs donc 2 variations.

        foreach ($dataAttributes['variants'][$i]['attributes'] as $key => $value) {
            $variation->set_attributes(array($this->slugify($value['name']) => $value['value']));
        }
        //Je sauvegarde les variations dans WooCommerce
        $variation->save();
    }
}

```

Pour créer un produit variable, il est impératif de créer, dans un premier temps, un attribut dans WooCommerce, j'ai donc créé une fonction qui permet de récupérer tous les attributs d'un produit ainsi que tous ses "termes".

Les termes d'un attribut sont toutes les variantes possibles d'un attribut, si je reprends l'exemple de mon stylo :

Le stylo c'est le produit.

La couleur c'est l'attribut.

Bleu , noir, rouge, vert sont les termes de l'attribut.

Le stylo de couleur bleu est, quant à lui, une variante de mon stylo, le stylo de couleur rouge en est une autre.

	Nom	UGS	Stock	Prix	Catégories	Étiquettes	★	Date
<input type="checkbox"/>	 7 familles « Halloween »	702708	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Quiz « Nutrition »	702711	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Puzzle à plat 12 pièces magnétiques	702710	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Jeu de mémo & coloriage « de Noël » – 33 cartes	702709	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Tapis flocké – 40 x 60 cm (danube, prune, forest)	702712	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Tapis flocké – 40 x 60 cm (iris,pivoine,émeraude)	702713	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34
<input type="checkbox"/>	 Piste de dés – 31 cm	702714	En stock	–	Loisirs & événements	–		Publié 20/12/2022 à 11h34

Voici à quoi ressemblent les produits une fois importés dans WooCommerce, il aurait été bien plus long et fastidieux de les ajouter à la main.

Présentation du jeu d'essai

Je vais maintenant tester la demande d'import d'un produit d'un fournisseur.

Je démarre avec un WooCommerce vierge de tout produit. Je demande ensuite 1 produit du fournisseur "AREKA" et clic sur "Afficher" qui devrait m'importer le produit dans WooCommerce.

SHOW

AREKA

```
C:\wamp64\www\dmw-es\euopean-api-modules\class\API.php:173:
array (size=5)
  'id' => int 695609
  'name' => string 'MINI LAPIN LINDT HELLO dans un carton personnalisable "oeuf de Pâques"' (length=71)
  'desc' => string 'Mini lapin Lindt HELLO en chocolat au lait de qualité supérieure dans un carton personnalisé en'
  'url' => string 'https://uploads.european-sourcing-platform.com/products/products/680/695609/2-MINI-LAPIN-LINDT-HE'
  'categories' =>
    array (size=3)
      0 =>
        array (size=6)
          'full_hierarchy_name' => string 'GASTRONOMIE - BOISSONS' (length=22)
          'project_id' => int 1157
          'parent_id' => null
          'name' => string 'GASTRONOMIE - BOISSONS' (length=22)
          'id' => int 1157
          'slug' => string 'gastronomie-boissons' (length=20)
      1 =>
        array (size=6)
          'full_hierarchy_name' => string 'GASTRONOMIE - BOISSONS > épicerie sucrée' (length=42)
          'project_id' => int 1137
          'parent_id' => int 1157
          'name' => string 'épicerie sucrée' (length=17)
          'id' => int 1137
          'slug' => string 'epicerie-sucree' (length=15)
      2 =>
        array (size=6)
          'full_hierarchy_name' => string 'GASTRONOMIE - BOISSONS > épicerie sucrée > chocolat' (length=53)
          'project_id' => int 1122
          'parent_id' => int 1137
          'name' => string 'chocolat' (length=8)
          'id' => int 1122
          'slug' => string 'chocolat' (length=8)
```

Voici les données que j'ai récupérées de l'API.

	Nom	UGS	Stock	Prix	Catégories	Étiquettes	★	Date
<input type="checkbox"/>	 MINI LAPIN LINDT HELLO dans un carton personnalisable « oeuf de Pâques »	695609	En stock	—	Miam miam	—		Publié 10/01/2023 à 16h45
<input type="checkbox"/>	Nom	UGS	Stock	Prix	Catégories	Étiquettes		Date

Et ici le produit qui s'est importé dans WooCommerce, ici il s'agit d'un produit simple.

Tout fonctionne correctement, l'import fonctionne.

Descriptions

Veille

Lors de mon stage, j'ai réalisé un formulaire de contact. Il a pour but de permettre à celui qui navigue sur le site d'entrer en contact avec son propriétaire facilement sans avoir à passer par sa messagerie.

Cependant, sans protection, il n'est pas rare que ces formulaires soient sujets à des attaques de spam. Il s'agit de robots envoyant des messages répétés depuis le formulaire. Ces attaques ont pour effet de remplir inutilement votre boîte mail et donc, de vous faire rater des messages importants.

Il m'a fallu trouver une solution afin de pallier ce problème de spam. Étant donné que les formulaires n'étaient pas à destination de sites très visités, une protection simple à suffit.

Après vérification sur plusieurs sites, je me suis rendu compte que la technique la plus simple et efficace de bloquer les robots était d'utiliser un pot de miel (honeypot).

Cette technique consiste à insérer un champ supplémentaire dans le formulaire qui est dissimulé, ainsi, les visiteurs ne le voient pas mais le robot qui traite tous les champs du formulaire le remplit.

Après avoir créé tous les champs nécessaires du formulaire, j'ajoute un champ de type : "hidden", que seul le robot peut voir.

Dans le traitement de mon formulaire, je vérifie si ce champ est complété, dans le cas où il l'est, j'empêche l'envoie du mail.

Enfin, je vérifie si le traitement est exécuté après avoir visité la page du formulaire. Cette méthode évite les robots qui pourraient envoyer les données sans même avoir chargé le formulaire.

Voici le code permettant de vérifier d'où le formulaire est envoyé.

```
if(isset($_SERVER['HTTP_ORIGIN']) && $_SERVER['HTTP_ORIGIN'] == "url de mon formulaire")
```

Situation de travail ayant nécessité une recherche en anglais

Durant mon stage, j'ai eu à faire des recherches sur l'utilisation de WooCommerce. Notamment pour savoir comment importer les différents types de produits et les attributs.

Cependant, il se trouve que les résultats étaient en anglais, il m'a donc fallu les comprendre voici un extrait original puis traduit de cette réponse.

There are also methods that you could find useful:

- `set_featured()` – pass `true` if you want this product to be marked as featured.
- `set_gallery_image_ids()` – multiple image IDs can be passed as an array here.
- `set_menu_order()` – manual product order as an integer.
- `set_status()` – any post status here. Don't want the product to be published? Set `draft` here.
- `set_total_sales()` – product total sales can be passed here as an integer value.
- `set_catalog_visibility()` – visibility in the catalog can be configured with this method, `hidden`, `visible`, `search` and `catalog` values are accepted.
- `update_meta_data()` – any product meta data, pass a pair of meta key and meta value as first and second parameters appropriately.

Il y a également des méthodes qui peuvent être utiles:

- `set_featured()` – Passez à "True" si vous voulez que ce produit soit marqué comme présenté.
- `set_gallery_image_ids()` – plusieurs ID d'images peuvent être passés en tant que tableau ici.
- `set_menu_order()` – commande manuelle de produits sous forme d'entier.

- `set_status()` – n'importe quel statut de poste ici. Vous ne voulez pas que votre produit soit publié. Mettez `draft` ici.
- `set_total_sales()` – les ventes totales du produit peuvent être transmises ici en tant qu'entier.
- `set_catalog_visibility()` – la visibilité dans le catalogue peut être configurée avec cette méthode les valeurs '`hidden`', '`visible`', '`search`' et '`catalog`' sont acceptées.
- `update_meta_data()` – toutes les métadonnées de produit, transmettez une paire de clefs de méta et de valeurs méta en premier et deuxième paramètres de manière appropriée.

Conclusion

Durant ce stage, j'ai appris de nombreuses compétences et découvert le monde du travail au travers de réels projets avec de réelles contraintes.

J'ai pu mettre en pratique les connaissances acquises lors de ma formation Développeur Web & Web Mobile en les adaptant aux besoins réels de l'entreprise.

Entouré d'une équipe toujours prête à répondre à mes questions et à m'aider, j'ai pu bénéficier de leur expérience et apprendre les meilleures pratiques pour résoudre les défis que j'ai rencontrés.

En conclusion, ce stage a été une expérience très enrichissante pour moi et je suis reconnaissant d'avoir eu l'occasion de développer mes compétences chez DMWeb. Je suis convaincu que les compétences acquises au cours de ce stage me seront utiles pour ma carrière future en tant que développeur web.

Annexes

```
//IMPORT IMAGE ON WOOCOMMERCE BY URL AND RETURN ID
public function importUrl($imageUrl){
    require_once(ABSPATH . 'wp-admin/includes/file.php');

    // download to temp dir
    $temp_file = download_url( $imageUrl );

    if( is_wp_error( $temp_file ) ) {
        return false;
    }

    // move the temp file into the uploads directory
    $file = array(
        'name'      => basename( $imageUrl ),
        'type'      => mime_content_type( $temp_file ),
        'tmp_name'  => $temp_file,
        'size'      => filesize( $temp_file ),
    );
    $sideload = wp_handle_sideload(
        $file,
        array(
            'test_form'  => false // no needs to check 'action' parameter
        )
    );

    if( ! empty( $sideload[ 'error' ] ) ) {
        // you may return error message if you want
        return false;
    }

    // it is time to add our uploaded image into WordPress media library
    $attachment_id = wp_insert_attachment(
        array(
            'guid'          => $sideload[ 'url' ],
            'post_mime_type' => $sideload[ 'type' ],
            'post_title'    => basename( $sideload[ 'file' ] ),
            'post_content'  => '',
            'post_status'   => 'inherit',
        ),
        $sideload[ 'file' ]
    );

    if( is_wp_error( $attachment_id ) || ! $attachment_id ) {
        return false;
    }

    // update metadata, regenerate image sizes
    require_once(ABSPATH . 'wp-admin/includes/image.php');

    wp_update_attachment_metadata(
        $attachment_id,
        wp_generate_attachment_metadata( $attachment_id, $sideload[ 'file' ] )
    );
}

return $attachment_id;
}
```

AMSD

Applications Mécaniques | Soudures Diversifiées

PRÉSENTATION

AMSD,
l'acier sous
toutes ses
formes.

AMSD pilote à tout vos projets, tout vos besoins, toutes vos envies, acier, inox, aluminium, par pilage, découpage, roulage, cisailage... « On peut tout faire avec l'acier. Il se marie très bien avec d'autres matériaux, nobles comme le bois par exemple » fait remarquer Ronnald Giffard qui a repris l'entreprise créée par son père en 1995, à Thônes dans le Doubs.

On ne fait jamais les mêmes choses tant les applications sont variées »

AMSD

Applications Mécaniques | Soudures Diversifiées

ÉQUIPEMENTS DE L'INDUSTRIE

Toutes activités industrielles

AMSD réalise tous types d'équipements indispensables aux activités industrielles, à commencer par l'aménagement du poste de travail : établis, tables, moyens de stockage et de manutention (chariot), plates-formes, estrades... A chaque activité ses matériaux de prédilection : l'inox, par exemple, correspond aux exigences spécifiques à l'agro-alimentaire, l'acier, très résistant, à l'industrie...

AMSD

Applications Mécaniques | Soudures Diversifiées

ÉQUIPEMENTS DE L'INDUSTRIE

Paniers de traitement

AMSD sait répondre aux exigences élevées des