

Dossier Projet

Développeur web et web mobile

Stage du 14 Novembre au 13 Janvier



Sommaire

Sommaire.....	2
Liste des compétences du référentiel couvertes par le projet.....	3
Le projet.....	4
1. Description du projet.....	4
2. Cahier des charges.....	4
3. Spécifications technique.....	7
Réalisations significatives.....	8
1. Maquettage du login et de la page administration WordPress.....	8
- Login et Pictogrammes.....	8
- Page d'administration	10
2. Intégration et développement du login.....	12
- Phase de recherche.....	12
-Intégration et développement.....	12
- Attaque Brut Force et Sécurisation.....	16
- mise en place de fonctionnalités WordPress.....	17
- Adaptation a l'environnement de travail.....	18
3. Intégration et développement de la page d'administration.....	18
-Intégration.....	18
-L'accessibilité numérique.....	21
-Responsive.....	22
4.Changement de direction.....	22
Jeu d'essai de la fonctionnalité la plus représentative.....	25
Description de la veille concernant la sécurité.....	27
Situation de travail ayant nécessité une recherche en anglais et extrait.....	29
En conclusion.....	31
Remerciements.....	32
Annexe.....	33

Liste des compétences du référentiel couvertes par le projet

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

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

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

- Développer les composant 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

Le projet

Description du projet

Yata!/Logitourisme est une agence web communication mais aussi un organisme de formation créée en 2008. Le propre de leur travail est basé sur le milieu du tourisme avec Logitourisme, mais grâce à la création de Yata! , ils ont pu s'ouvrir sur des domaines plus divers.

Mon stage était d'une durée de 2 mois, en 35h par semaine à l'agence de Lons-Le-Saunier.

À mon arrivée il m'a été demander de customiser la page de connexion des utilisateurs WordPress, un bloc contact support, un bloc pour les derniers sites crée par l'agence ainsi qu'un bloc de promotion produit Yata! Il m'a aussi été demandé de re-designer la page d'administration avec le contenu déjà présent et d'y intégrer les blocs que j'ai créés pour la page de connexion. J'ai dû maquetter mes blocs à l'aide du logiciel Figma. Pour ensuite intégrer, récupérer les données dont j'avais besoin grâce à la REST API de WordPress, rendre fonctionnel et automatiser les blocs que j'avais créés durant ma phase de maquettage.

Étant en présence sur l'agence, j'ai dû travailler en équipe avec le groupe de design ainsi que l'équipe de développement. J'étais seule en stage sur cette période. J'ai donc dû travailler en autonomie, sous la supervision de l'équipe en place ainsi que de mon maître de stage. La plus grosse partie de mon travail fut le « front-end » bien que j'aie dû faire tous de même un peu de « back-end », tout en m'adaptant à la demande des membres de l'agence.

Cahier des charges

Brief Technique R&D - Mise en place d'informations dynamiques sur la page de connexion et la page d'accueil des sites WordPress Yata!

Affichage des informations Yata! sur la page de connexion à l'administration WordPress ainsi que sur la page d'accueil de l'administration.

Objectif

Le but est de mettre en place via le système d'API de WordPress présent sur le site yata.fr pour faire remonter:

- Les dernières réalisations du groupe
- Bloc de promotion des produits Yata!
- Bloc d'informations « Nous contacter »

Consignes:

Template:

Il est demandé au stagiaire d'effectuer deux templates:

- Le template de la page de connexion
- Le template de la page d'accueil de l'administration (après connexion)

Ces templates devront respecter la charte graphique de Yata! Préalablement mise en place.

Une validation auprès de la direction artistique avant l'intégration est exigée.

Exigences pour l'intégration du bloc « Dernières réalisations »:

Les données de dernières réalisations devront être récupérées du site yata.fr via la REST API de WordPress, les informations sont récupérables, au format JSON, à cette adresse: <https://yata.fr/wp-json/wp/v2/realisations>

Nous devrons retrouver dans le bloc les informations suivantes pour chaque réalisation:

- Le titre de la réalisation
- L'image mise en avant
- L'URL vers le site internet yata.fr présentant la réalisation
- Une courte description présentant la réalisation

Exigences pour l'intégration du bloc « Promotion Produit Yata! »:

Le bloc des produits Yata devra être évolutif et ne pas modifier le design crée lors de l'ajout d'un produit, l'utilisation d'un slider est recommandée.

La liste des produits à intégrer est disponible sur la page d'accueil de connexion de Gutenberg.bac-a-sable.net.

(Demander les informations de connexions au maître de stage)

Exigences pour l'intégration du bloc « Nous contacter »:

Il doit y figurer les informations suivantes:

- Numéro de téléphone Yata: 03 84 24 42 35
- Adresse mail support Yata: support@yata.fr
- Une mention invitant les utilisateurs à nous contacter en cas de souci.

Exigences globales:

Le développement devra perdurer dans le temps et au fil des mises à jour du noyau de WordPress, ainsi une utilisation correcte des Hooks WordPress est obligatoire.

Mise en Oeuvre:

Dans un premier temps:

Création des deux templates.

Dans un second temps:

Récupération des données via l'API de WordPress pour le bloc « Dernières réalisations » (Une fonction, réutilisable, faisant appel aux réalisations, sur la page de connexion et la page d'accueil de l'administration).

En dernier temps:

Intégration des templates en HTML et CSS (LESS si l'utilisation est plus pratique, à la préférence du stagiaire).

1- Charte Graphique: Colorimétrie

Jaune:  #F9DE4B

Blanc:  #FFFFFF

Noir:  #131313

Noir Opaque:  #000000 55%

2. Charte Graphique: Polices

Open Sans Bold - Titres:

The quick brown fox jumps over the lazy dog 1234567890

Open Sans Light - texte sous texte:

The Quick Brown fox jumps over the lazy dog 1234567890

Open Sans Regular - texte:

The Quick Brown fox jumps over the lazy dog 1234567890

3. Charte Graphique: Logo



Spécifications techniques

Le projet était déjà en place, je n'ai fait que rejoindre un existant, une base fonctionnelle. Je n'ai donc pas choisi les technologies à utiliser sur le projet. J'ai dû m'adapter à la façon de travailler déjà en place. Les outils utilisés chez Yata! Sont ceux vus pendant la période de formation, je n'ai de ce fait pas eu à m'adapter à ce niveau.

Pour réaliser mes maquettes pour le projet, travaillant sur mon propre matériel, je n'avais pas accès à la suite Adobe, j'ai donc pu utiliser Figma, le logiciel que nous avions appris à utiliser pendant la période de formation

Réalisations significatives

1- Maquettage du login et de la page administration WordPress

Login et Pictogrammes

Mon maître de stage m'a laissé la liberté de m'organiser comme je le souhaitais. Donc après avoir pris connaissance du cahier des charges qu'il m'a été fourni, j'ai pris la décision de commencer par le maquettage de tout le projet. Sur toute cette partie du projet, j'ai travaillé sous la supervision de la directrice artistique de l'agence. Chaque maquette, chaque élément créé, on dut être révisé et valider par elle.

J'ai commencé par le maquettage du login qui m'était demandé. J'avais un existant :



Identifiant ou adresse e-mail

Mot de passe
 Mot de passe oublié ?

Se souvenir de moi

[Mot de passe oublié ?](#)

[← Aller sur Yata](#)

[Politique de confidentialité](#)

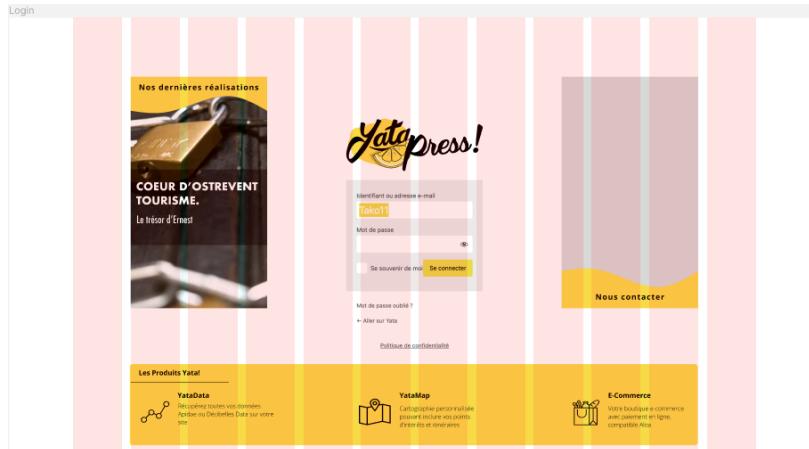
J'ai dû composer avec ce qui était déjà présent. On m'a demandé d'afficher :

- Un bloc de dernières réalisations
- Un bloc des produits Yata!
- Un bloc de contact support.

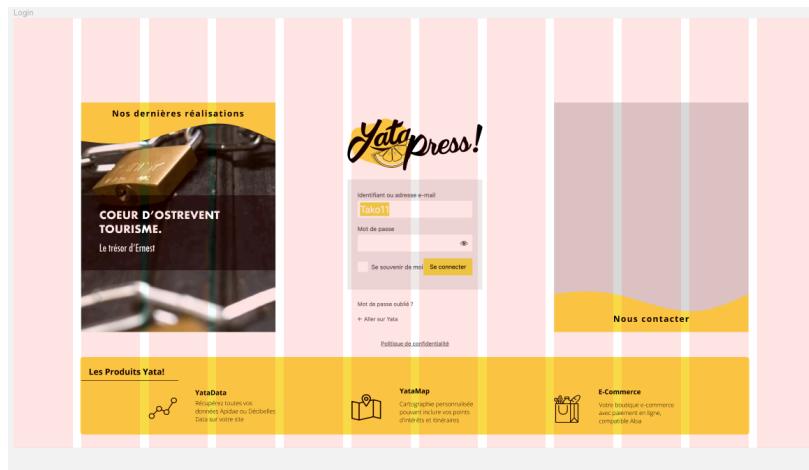
Pour les blocs, dernières réalisations et les produits, j'ai décidé de réaliser des carrousels, pour une question d'espaces et aussi pour une expérience client agréable, fluide et dynamique.

Après 3 différentes versions, mon travail a été validé par la directrice artistique

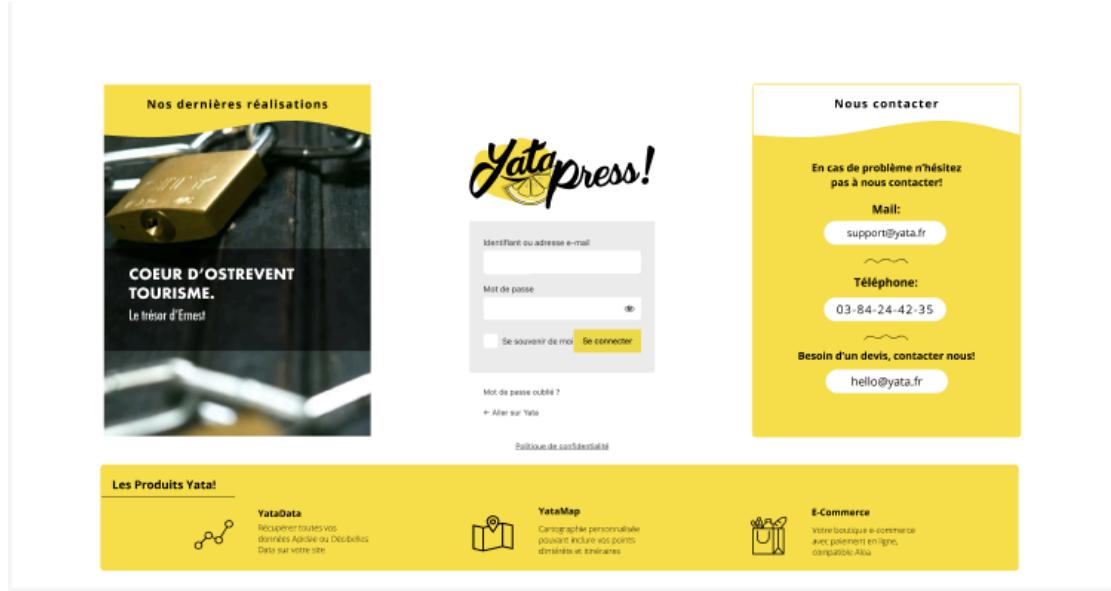
V1



V2



Version final



Sur cette maquette, j'ai pu apprendre à travailler avec le système de colonage, (on peut les voir apparaitre sur la version 1 et la version 2), ce système que je ne connaissais pas m'aidera par la suite pour le responsive durant mon temps

d'intégration. Chaque colonne représente les break-points (point de rupture pour le responsive ou l'adaptabilité de mon intégration selon les différentes tailles d'écrans).

En parallèle, il m'a été demander de choisir des pictogrammes pour illustrer les « produits yata », des illustrations importantes, mais simples dans leurs réalisations pour qu'il reste lisible même avec des tailles réduites.



Une fois validés j'ai pu les intégrer à mon bloc « produits ». On retrouvera ces pictogrammes sur la page d'administration aussi.

Page d'administration

Encore une fois je partais d'un existant, mais cette fois si j'ai eu plus de liberté, tous le contenu présent sur la page devait être présent dans ma version mais pour ce qui est de la mise en forme j'ai eu « carte blanche » sous réserve de validation de la directrice artistique toutes fois.

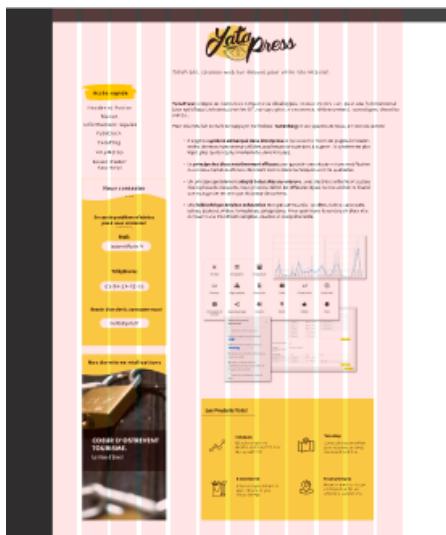
Il m'a encore fallu plusieurs essaie avant d'arriver à ma version finale. Cette page fut moins évidente que la précédente a créée. La directrice artistique ma donc beaucoup aiguiller. Elle m'a invité à chercher des exemples de tableau de bord pour faire ma

The screenshot shows the YataPress administration panel. On the left is a sidebar with a dark theme containing links like 'YataPress', 'YataMéto', 'YataClock', 'YataProg', 'YataMédo', 'Accès rapide', 'Header et Footer', 'Menus', 'Informations légales', 'Flamingo', 'Comptes', 'Contact', 'Apparence', 'Extensions', 'Components', 'Yata', 'Migrations', 'Outils', 'Champs personnalisés', 'Stark Matt', 'Duplicator', 'Products', 'Woocommerce', 'Statistiques', 'Marketing', and 'Réduire le menu'. The main content area has a header 'YataPress, solution web sur mesure pour votre site internet' and a sub-header 'YataPress intègre de nombreux composants développés, chacun d'entre eux, pour une fonctionnalité bien spécifique : édition, données BT, campagne, e-commerce, référencement, statistiques, données météo...'. It features a large 'YataPress!' logo with a yellow lemon icon. Below it is a 'Tableau de bord' section with several graphs and charts. Further down are sections for 'Les composants YataPress' (YataData, YataMap, E-Commerce, YataCommune, Flux domaines skiables) and 'YataPro' (YataPro, YataClock, YataGeo). A 'Boutique' section is also visible.

veille et de fonctionner par système de carte comme des post-it pour faciliter l'expérience utilisateur.

J'ai donc effectué ma veille sur le site pinterest.fr pour m'aider et m'inspirer. Après plusieurs discussions avec la directrice artistique j'ai compris mon erreur et ce qui m'empêchait d'avancer sur cette deuxième maquette, j'ai décidé par la suite de me séparer des composants que j'avais créés pour le login et recréé de nouveau composants pour cette page d'administration. Après trois essais différents, j'ai fini par avoir une version vraiment satisfaisante.

V1



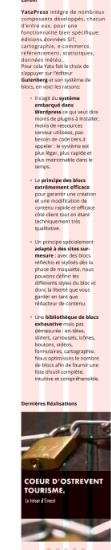
V2



V3



V3 Responsive téléphone



Cette étape de maquettage m'a aidé à me rendre compte qu'une maquette bien réaliser aide grandement à une intégration et développer plus efficacement.

2- Intégration et développement du login

Phase de recherche

Après cette phase de maquettage qui aura duré deux semaines, j'ai décidé de faire une phase de recherche avant de me lancer dans le développement.

Pour mon intégration, il m'a été conseillée d'utiliser l'API de [swiperjs.com](#). c'est une API dédié à la mise en place de carrousel pour le web. Elle fonctionne en Javascript. Ce site étant en anglais j'ai dû faire quelque recherche pour être sur de bien comprendre toutes les fonctionnalités.

J'ai aussi fait quelque recherche concernant les hooks (crochet) de WordPress, car j'ai vite réalisé que mon intégration du login allait être plus compliqué qu'une intégration « classique », WordPress ne m'autorisant pas à intégrer directement le contenu comme je le voulais dans le fichier login (je l'expliquerai plus en détail dans la partie « Intégration et développement »).

Intégration et développement

J'ai commencé mon intégration du login en délimitant mon espace de travail avec des blocs de la taille de chaque élément de ma page. Cela m'a permis de voir si tous correspondaient à la maquette que j'avais conçue et évitent les problèmes de placement.

J'ai décidé d'utiliser les flex box, pour rendre mon responsive plus simple.

La contrainte pour cette page, était de ne pas avoir de scroll vertical ou horizontal. Tous devaient tenir dans la taille du view port (taille de l'écran).

J'ai commencé par le bloc que j'ai jugé le plus simple et rapide pour moi.

Le bloc des contacts de maintenance, il devait comporter le mail support et le téléphone ainsi que le mail pour le support commercial de l'entreprise.

On m'a demandé d'utiliser la fonctionnalité Mailto ainsi que tel pour rendre les mails et le téléphone plus simplement utilisable pour l'utilisateur.

```
<a class="mailto" href="mailto:hello@yata.  
fr">hello@yata.fr</a>
```

```
<a class="tel" href="tel:  
+33384244235">03-84-24-42-35</a>
```

Ensuite, j'ai décidé d'intégrer le bloc des nouvelles réalisations.

La contrainte ici était de rendre automatique la remonter des informations saisies dans le carrousel. Si de nouvelle réalisation était créée mon carrousel devais automatiquement se mettre à jour.

J'ai commencé par la structure de ce bloc avec le bandeau fixe.
Ensuite j'ai intégré le carrousel à l'aide de l'api swiper js.

J'ai dû lier mon projet à leurs CDN(Content Delivery Network) (ensemble d'ordinateurs reliés entre eux via internet, grâce à la mise en réseau, ces ordinateurs ont pour mission principale de rapprocher et diffuser rapidement le contenu de sites internet entre eux.)

Grâce à cela j'ai pu avoir accès à toutes les données de swiper js.

La documentation nous guide ensuite pour initialiser swiper en Javascript. Il nous suffit de créer un nouvelle objet swiper et de lui passer les paramètre que l'on souhaite.

```
const swiper = new Swiper('.swiper', {  
    // Optional parameters  
    direction: 'horizontal',  
    loop: true,  
  
    // If we need pagination  
    pagination: {  
        el: '.swiper-pagination',  
    },  
  
    // Navigation arrows  
    navigation: {  
        nextEl: '.swiper-button-next',  
        prevEl: '.swiper-button-prev',  
    },  
  
    // And if we need scrollbar  
    scrollbar: {  
        el: '.swiper-scrollbar',  
    },  
});
```

Ici j'ai choisi la direction horizontale pour l'orientation de mes slides. J'ai défini le mode loop pour que mon carrousel se comporte comme une boucle.

J'y ai rajouté une pagination ainsi que des flèches directionnelles pour simplifier l'utilisation.

C'est dans cet objet que je gère chaque aspect fonctionnel de mon carrousel. Je n'ai qu'à me référer à la documentation pour savoir ce qui est possible ou non et rajouter ce que je souhaite.

Une fois le carrousel fonctionnel, j'ai customisé les éléments pour coller à la maquette.

Pour travailler correctement j'ai commencé avec des images factices. Mais il m'était demandé de faire remonter les informations de ce carrousel grâce à la REST API de wordpress. Ce fut donc mon étape suivante.

Pour dialoguer avec cet api j'ai décidé d'utiliser PHP avec la fonction file_get_contents pour récupérer les 5 premiers noeuds du json

```
<?php  
$json = file_get_contents('https://yata.fr/wp-json/wp/v2/realisations?  
per_page=5');
```

J'ai ciblé ses 5 premiers noeuds en rajoutant à l'url le ?per_page=5 , comme cela je n'ai pas à utiliser les donnés inutiles et il est assez aisément de le modifier par la suite si nous souhaitons récupérer les 6 noeuds plutôt que 5 finalement.

```



```

d'automatisation et d'optimisation. Au lieu de répéter les slides à chaque fois ma boucle tourne et génère les slides tant qu'il y a des données du json à placer.

```

foreach ($datas as $key => $value) {
    $titre= $value['title']['rendered'];
    $link= $value['link'];
    $desc= strip_tags($value['acf']['designThinking']);
    $hrefImgJsons=$value['_links']['wp:featuredmedia'][0]
    ['href'];
    $jsonImg =file_get_contents($hrefImgJsons);
    $data= json_decode($jsonImg, true);
    $img= $data['media_details']['sizes']['1024W']
    ['source_url'];

    ?>
    <div class="swiper-slide" autoplay="3000">
        
        <div class="Cbandetexte">
            <div class="bandetexte">
                <a class="lienTitre" href="<?= $link ?
                ">"target=_blank">
                    |   <h3 class="titreSlide"><?= $titre; ?></h3>
                </a>
                <p class="descriptionSlide"><?= reduire($desc);?>
            </p>
        </div>
        </div>
    </div>
    <?php ?>
}

```

Pour le bloc des produit Yata! je n'avais pas accès à des données stocker j'ai donc décidé de créer un tableau d'objet avec toutes les données dont j'avais besoin. C'est-à-dire , le nom du produit les informations du produit et le pictogramme. C'était la meilleure solution pour moi, pour rendre l'utilisation future plus facile pour l'équipe de travail. Il ne suffit plus que de rajouter une entrer au tableau si de nouveau produits doivent être rajoutés.

Ensuite il m'a fallu décoder le json grâce à la fonction json_decode pour pouvoir exploiter les donnés.

Grâce à cela il m'a suffi de faire une boucle foreach qui entourait mon carrousel et de récupérer les bonnes données pour les placer aux bons endroits dans ma structure HTML.

La boucle qui entoure mon carrousel a pour but une idée

Donc dans mon cas comme on peut le voir. Je récupère les 5 titres, 5 liens, 5 descriptions. Pour la récupération d'images se fut un peu different car les image étais stocké dans un autre json présent dans le premier . J'ai donc dû refaire l'extraction de donner grâce à la même manipulation que plus haut mais cette fois si dans la boucle foreach.

Et pour exploiter les donner dans le deuxième carrousel de produit j'ai de nouveau utilisé une boucle foreach pour encore une fois un souci d'automatisation et d'optimisation.

Ici le but a été d'avoir un code, qui permettrait dans le futur, si de nouveau élément son rajouter que tous se face en ayant le moins de manipulation humaine à faire. Pour faire gagner du temps à l'équipe de travail.

Une fois tout cela mis en place j'ai été face à un souci. Les descriptions des nouvelles réalisations que j'avais récupérées étaient parfois trop grande par rapport à la place allouée pour elles dans mon intégration.

J'ai donc dû mettre en place une fonction de réduction de texte.

J'ai utilisé la fonction native mb_strlen qui réduit la taille d'une string en prenant en compte l'encodage(chaîne de caractère) .

Ma fonction se compose d'un if et d'un else. Si la chaîne de caractère récupéré est supérieure à 100 caractères alors il coupera et rajoutera '...' à la fin de la phrase . Sinon il laisse la chaîne de caractère tel quel.

Ici ma fonction est réutilisable si jamais nous en aurions besoin ailleurs dans la page. Au lieu de lui passer des variables spécifiques je lui passe la variable \$str comme cela la fonction s'adaptera à n'importe quelle string.

```
function reduire($str){  
  
    if(mb_strlen($str) > 100) {  
        return (mb_substr($str, 0 , 100)."...");  
    }else{  
        return $str;  
    }  
}
```

Comme je travaillais en local je n'ai pas encore eu à faire au contraire de WordPress. La mise en ligne fut donc mon étape suivante.

En vue de ma mise en place sur mon maquettage j'ai été mise face à une difficulté. Le fichier wp_login où je travaillais ne me permet pas l'intégration que j'ai mis en place WordPress définit un espace en haut et en bas du bloc de login mais pas sur les côtés comme je l'avais créé. J'ai donc dû penser à une solution. Ce qui m'a amener à l'idée suivante.

Attaque Brut Force et Sécurisation

Ici je vais expliquer les essais de sécurisation du fichier Wp-login de WordPress que j'ai pu faire. Je parlerais de ce qu'est une attaque brut force plus dans en détail dans la partie « description de la veille concernant la sécurité ».

Comme je l'ai expliqué plus haut ma maquette me posait quelque problèmes avec la compatibilité de WordPress, j'ai faits quelque recherche (en l'occurrence sur <https://wpmarmite.com/brute-force-wordpress/>) et je me suis dit que je pouvais régler ces deux soucis en même temps, facilité mon intégration et apporté plus de sécurité à l'administration.

C'est un sujet vraiment vaste. J'ai donc demandé ce qui avait été déjà mis en place dans l'agence les plugins utiliser par exemple. J'ai pris la décision de créer une nouvelle url pour la connexion. Grâce à cette manipulation les bots qui habituellement se dirige vers wp_login ou wp_login.php seront face à un mur. Cette tâche fut bien plus compliquée que prévu. Car l'agence travaille avec un plugin de sécurité qui demande le fichier wp_login pour fonctionner donc même si le fichier est supprimé il ira le rechercher (en vain) et créera une redirection. Je voulais vraiment absolument éviter la redirection car si l'humain est rediriger le bot aussi.

J'ai donc trouvé une autre solution, après de nombreuses recherches j'ai vu que c'était une mauvaise pratique de supprimer le fichier wp-login.php. J'ai du combiner plusieurs de mes recherches ainsi que les discussions avec mon maître de stage pour trouver la solution finale. J'ai décidé d'ajouter une fonction avec un paramètre GET pour ajouter un mot de passe par l'url si cette condition n'est pas remplis alors je détruis le login avant l'affichage du formulaire de connexion Combiner à ma nouvelle page de login pour les clients. Comme cela les bots seront face à un mur, vu qu'ils n'auront pas l'url correcte.

Pour se faire j'ai créé une fonction pour couper le formulaire de login wp-login que je cible grâce à la variable globale \$_SERVER. Le formulaire se coupera si l'une des deux conditions (if) que je définis est remplie.

Il faudra que passwordkey ainsi que ça valeur password soi présente dans l'url si nous ne retrouvons pas ses valeurs alors le formulaire sera coupé grâce au exit.

```
<?php
2 references
function killLoginForm( $errors, $redirect_to ){
    $passwordkey = 'security';
    $password = 'yata';
    if (strpos($_SERVER[REQUEST_URI],"wp-login.php")!==false) {
        if (!isset($_GET[$passwordkey])) {
            exit;
        }
        if ($_GET[$passwordkey]!=$password) {
            exit;
        }
    }
    return $errors;
}

add_filter( 'wp_login_errors', 'killLoginForm', 90, 2 );
```

mise en place de fonctionnalités WordPress

Une fois la nouvelle page login installer sur le serveur il m'a suffi d'utiliser les Hooks de WordPress pour insérer le contenu de mon développement.

J'ai utilisé un hook natif: le login-footer , il génère le pied de page de la page de connexion . Il a fallu que je passe mon contenu "footer" dans une fonction qui correspond dans mon cas à mon bloc de nouveau produit . Et ensuite l'appeler grâce au add_action (qui ajoute une fonction à notre hook).(voir la fonction entière en annexe)

```
add_action( 'login_footer', 'slider_produit');
```

Pour les autres blocs droits et gauches , il n'y avait pas de hook natif j'ai donc dû faire la même manipulation que plus haut en ajoutant do_action à l'endroit où je voulais que mon hook se lance.

```
do_action('bloc-gauche');
```

Cette fonction permet de dire où le add_action doit s'exécuter.

Adaptation a l'environnement de travail

Ici la partie la plus compliquée, à mes yeux , fut de s'organiser avec un univers de travail inconnue. Chaque agence a son fonctionnement et ça propre organisation qui n'est pas forcément la nôtre. Ici par exemple le fichier function.php ne fait que de l'inclusion de fichier pour ne pas surcharger ce fichier qui serait trop long et peu intuitif. J'ai donc dû morceler mon travail par bloc chaque bloc a son fichier, avec ça fonction attitrée, pour des raisons d'organisation de lisibilité et de compréhension. Je n'avais jamais travaillé avec autant de fichiers, j'ai dû être d'autant plus rigoureuse et organiser dans mon travail.

3. Intégration et développement de la page d'administration

Intégration

Dans cette deuxième partie du projet qui m'a été donné, j'ai dû faire l'intégration de la maquette réalisée dans les premières semaines de mon stage, pour la page d'administration de WordPress.

La grande difficulté que j'ai dû surpasser ici, fut de travailler avec une base de code déjà créée, on ne mis à pas obliger. Ici, mon maître de stage m'a laissé le choix de partir sur une page vierge ou de garder le code déjà créé. Pour une question d'efficacité j'ai décidé de garder une partie du code déjà présent. J'ai donc pu voir, que chaque intégrateur a sa façon de faire, sa façon de nommer les éléments, d'écrire, et globalement une organisation qui lui est propre. Même s'il est de rigueur d'adapter son code pour qu'il soit lu et compris par le plus grand nombre, chacun à tous de même son propre fonctionnement.

Une fois que j'ai eu compris le fonctionnement de l'intégration déjà mis en place, j'ai pu m'organiser, trier les éléments que je souhaitais garder et commencer à mettre en place ma propre structure.

Comme pour la page de login, cette page devait comprendre les contacts maintenance les nouveaux produits et les dernières réalisations. Mais elle devait comprendre aussi un bandeau d'accès rapide, ainsi qu'une explication des outils que YataPress propose à son utilisateur.

J'ai beaucoup réutilisé le code que j'avais créé pour le login, pour l'adapter à cette page. Il n'y avait pas de fonctionnalité différente. Cela permet à l'utilisateur de se retrouver dans un environnement qu'il connaît pour plus de clartés. Par contre en terme de visuel l'intégration était totalement différente de celle du login.

J'ai commencé par intégrer les éléments les plus simples et re-designer les éléments déjà présents dans la page, car j'ai jugé que ce serait le plus rapide.

J'ai travaillé par bloc pour bien fractionner chaque partie avec du display flex sur chaque bloc. Pour plus de faciliter et de rapidité, de plus cela rend plus simple l'intégration du responsive.

Le logo et la phrase d'accroche devaient être remplacés et centrer en haut. Ensuite les actions rapides étaient déjà présentes j'ai dû jouer avec le css présent pour les adapter à ma maquette.

Le bloc de support technique était quasiment pareil que pour le login en terme de contenu mise à part la forme de vague qui sert d'arrière-plans et pictogrammes dans les boutons.

Comme pour le login j'ai dû réaliser un slider pour les « produits yata! »

Le bloc photo et explication de yatapress n'avait comme le logo besoin que d'être remplacé pour coller à ma maquette.

Et enfin pour le slider des dernières réalisations, j'ai réutilisé mon code pour appeler les données de la rest API de WordPress dont j'avais besoin et j'ai changé la mise en place de mes éléments.

L'élément technique de l'intégration encore une fois fut l'intégration des sliders avec swiper.js

J'ai créé deux fichiers php et js séparer pour mes sliders pour ne pas surcharger mon code dans l'admin aussi pour être plus lisible et organiser dans mon travail.

Pour des raisons de clarté dans mon travail j'ai décidé de travailler en local pour le développement de ces éléments.

J'ai commencé par créer ma structure HTML qui contiendrait mon slider ainsi que le style avec des données factices. Après cela j'ai lié mes fichiers js à mon HTML pour être sûr que tous fonctionnaient avec mon intégration. Et pour finir j'ai rajouté les bonnes données qui venaient de l'API de WordPress et ma fonction de réduction de texte. Mon intégration a été quelque peu modifier par rapport a ma maquette de départ pour des raisons d'accessibilité.



Ici , mon slider comme je l'avais créé dans ma maquette.



Ici , mon slider après intégration, j'y ai rajouté un aplat noir en transparence, car sur certaine slides les textes n'étaient pas forcément lisibles complètement à cause de la photo d'arrière plan.

J'ai donc pris le temps de me renseigner sur l'accessibilité numérique, ce que cela impliquait et comment je pouvais intégrer cela dans mon projet.

L'accessibilité numérique

Pendant mes recherches, j'ai compris qu'il était vraiment extrêmement important de développer en ayant toujours en tête les norme WCAG (Web Content Accessibility Guidelines) qui sont : perception, utilisable, compréhensible, robuste. L'accessibilité numérique c'est le fait de rendre tous les contenue accessible à tous. Un code perceptible: Nous devons pouvoir faciliter la perception visuelle et auditive le code doit Utilisable: notre code doit pouvoir être lu par des technologies d'assistance,

Évidemment durant la formation on nous avait évoqué ces problématiques, mais je pense que grâce à la mise en situation professionnelle , j'ai pu me rendre compte de l'impact que pouvait avoir mon développement sur les consommateurs des sites.

Pour l'exemple en France sur 12 millions de personnes on compte 2 millions de personnes en déficiences visuelles (aveugles ou malvoyante). Ce qui fais 1 personnes sur 5 concernée par les problèmes d'accessibilité. Si cette partie de la population ne peut pas accéder au site internet on se retrouve en situation de fracture numérique donc d'exclusion. Il est d'ailleurs estimé qu'en France moins de 10% des sites sont accessibles. (information tirée de la vidéo « qu'est-ce que l'accessibilité numérique et pourquoi c'est important? » De l'association Valentin Hauy avec les aveugles et malvoyants)

J'ai donc décidé avec le temps qu'il me restait de mettre en place certainnes bonnes pratiques pour rendre mon projet plus accessible.

Tout d'abord, j'ai commencé par quelques choses de simple mais finalement fondamental, j'ai vérifié que les « alt » de mes images étaient tous correctement remplis avec un contenu assez descriptif et complet. Les alt ne sont pas obligatoires mais extrêmement utiles pour l'accessibilité. C'est ce qui sera utilisé par les lecteurs d'écrans pour expliquer ce que l'image représente pour les personnes aveugles ou malvoyante.

De plus j'ai faits en sorte que mes textes soient lisibles, et que les contrastes sois corrects même si je devais revenir sur ma maquette initiale.

Le sujet est très vaste et avec le temps impartie je n'ai pas pus faire de miracle. Mais c'est un sujet que j'aimerais vraiment approfondir dans le futur.

Responsive

Le responsive fut une des partie attendu de mon travail sur la page d'administration car les visuels crée pour écran d'ordinateur n'étais pas forcement adapté au mobile et à la tablette. Notamment la carrousel. Car le carrousel que j'avais penser pour l'administration desktop était à l'horizontal. J'ai donc du tous repasser a la vertical pour une question d'ergonomie. La forme de vague qui me servais de cadre n'étais pas adapter aussi pour les version mobile/tablette j'ai donc crée aplat jaune a la place et mis mon élément en display none pour que mon élément ne soit pas présent sur ses versions.



4- Changement de direction

Durant les dernières semaines de mon stage, mon projet a subi quelques modifications dans le contenu. Je tenais à vous présenter l'entièreté de mon travail et le cheminement pour arriver au résultat final.

Après rendez-vous avec la direction pour vérifier mon travail quasiment achevez, l'un des blocs à dû être modifié il ne convenait plus. Cet évènement m'a appris à m'adapter aux demandes client, à être efficace et réactive.

Le bloc des « produits Yata! » à dû être supprimé On m'a demandé de le remplacer par un bloc qui parlerait des services proposer par l'agence.

Il devrait comprendre : - audit d'éco-conception

- optimisation du référencement naturel
- évolution graphique
- création de brochures / flyers / cartes de visite
- hébergement vert

- gestion de noms de domaines

Je n'ai pas « réinventé la roue » j'ai gardé la structure déjà mise en place et je l'ai réadapté à la nouvelle demande. J'ai supprimé le slider mis en place car il n'avait plus d'intérêt, ainsi que les pictogrammes. J'ai entouré mes différents services dans des balises <a> pour rediriger le client vers des pages correspondant aux services ciblés.

En terme de design, j'ai beaucoup simplifié. Il m'a été demandé de créer des boutons sur les différents services pour amener le client à cliquer. J'ai supprimé l'aplat jaune que je trouvais incohérent avec le nouveau bloc. J'ai réutiliser le design utilisé dans la page admin avec les ombres portées pour amener au click



Ici voilà le slider du début du projet



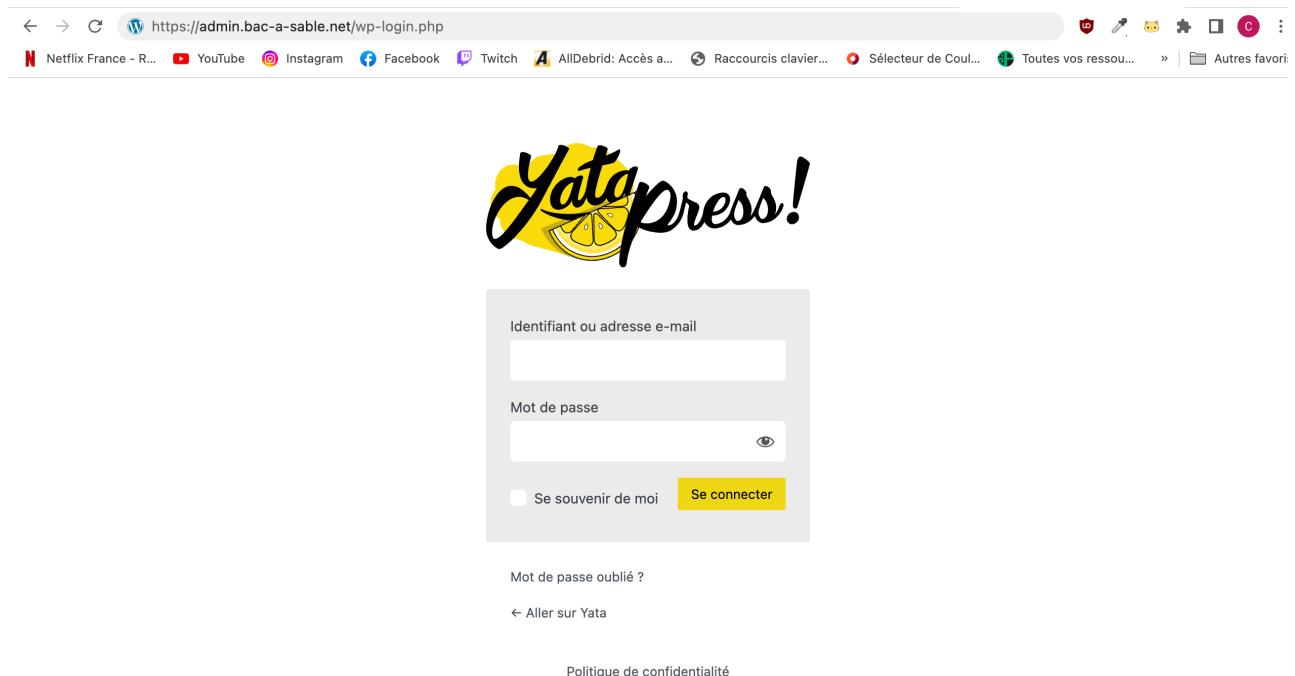
Et dans la deuxième photo voilà le résultat après les changements qui m'ont été demander.

Pour l'administration j'ai cassé et gardé le même visuel que ce que j'avais créé initialement mais je n'ai finalement pas fait de carrousel !

Jeu d'essai de la fonctionnalité la plus représentative

La fonctionnalité la plus représentative de mon projet fut la sécurisation de la page wp-login contre les attaques brut force.

Mon but était de couper l'utilisation du formulaire de connexion wp-login.php et de créer mon propre fichier de login pour qu'il ne soit pas impacté par les mises à jour WordPress. Donc après beaucoup de recherches j'ai décidé de sécuriser l'URL, sans la clef de mot de passe et le mot de passe dans l'url, le formulaire se coupera.



Ici nous avons l'état initial du wp-login.php avec l'url propre, sans sécurité.

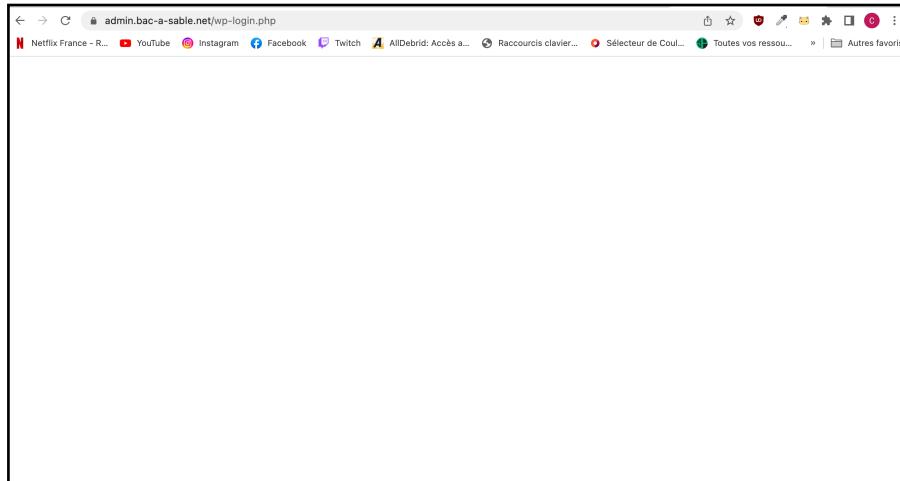
Grâce à cette fonction comme je l'ai déjà expliqué plus haut je réussit à conditionner l'action de fermeture du formulaire en y ajoutant des conditions et en le restreignant au fichier wp-login.php et pas à mon autre login.

```

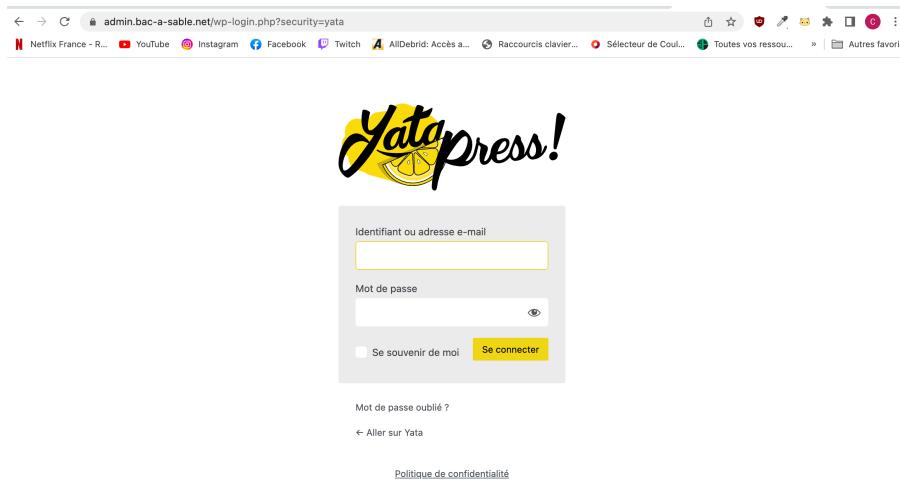
<?php
2 references
function killLoginForm( $errors, $redirect_to ){
    $passwordkey = 'security';
    $password = 'yata';
    if (strpos($_SERVER[REQUEST_URI],"wp-login.php")!==false) {
        if (!isset($_GET[$passwordkey])) {
            exit;
        }
        if ($_GET[$passwordkey]!=$password) {
            exit;
        }
    }
    return $errors;
}

add_filter('wp_login_errors', 'killLoginForm', 90, 2 );

```



Ici on peut voir que l'url ne contient pas la clef et le mot de passe donc la page est blanche à l'affichage



Alors qu'ici la clef et le mot de passe sont présents donc nous avons l'affichage du login.

Pour finir on peut voir ici la page de formulaire Yata_login.php qui n'est pas impacté par ma fonction de sécurité.

The screenshot shows a web browser window with the URL `admin.bac-a-sable.net/yata_login.php` in the address bar. The page has a yellow and white color scheme. On the left, there's a sidebar with a section titled "Nos dernières réalisations" featuring a graphic of a hand holding a glass. Below it, a section titled "J'aime mes bouteilles" describes a project involving motion design videos. In the center, the "Yata press!" logo is displayed above the login form. The login form includes fields for "Identifiant ou adresse e-mail" and "Mot de passe", with a "Se souvenir de moi" checkbox and a "Se connecter" button. To the right, a yellow box titled "Nous contacter" provides contact information: "Notre support est disponible via:" followed by "Mail: support@yata.fr", "Téléphone: 03-84-24-42-35", and "Une demande? Contacter nous! hello@yata.fr". At the bottom, links for "Politique de confidentialité" and "Aller sur Yata" are visible.

Description de la veille concernant la sécurité

Wordpress étant le CMS (Content Management System, c'est-à-dire système de gestion de contenu) quasiment le plus utiliser au monde avec 43% de tous les sites web, en d'autres termes, WordPress propulse plus d'un tiers du web! Il est donc la cible potentielle de nombreuses attaques de sécurité. Mais aussi c'est une cible de choix car le nombre de tentatives de connexion est illimité et qu'il est très simple d'accéder à la page de connexion toujours accessible avec le slash wp_admin dans url. Tout d'abord je vais expliquer brièvement ce que sont les attaques brut force et qu'elles sont les problèmes qu'il en retourne. Car je ne connaissais pas réellement de quoi retournait et comment fonctionnait ce type d'attaque avant mes recherches.

Les brute force attack (ou attaque de force brut en français) sont des attaques faite part des bots (ou robots en français) leurs buts est de se rendre sur nos logins et de découvrir nos identifiants et mots de passe. Une fois trouver les hackers ont pleins pouvoir sur nos administrations, nos données , les informations personnelles, les données clients. Ils peuvent récupérer, supprimer des données ou ajouter des logiciels malveillants à leurs guises. De plus, même si le pirate ne parvient pas à trouver nos mots de passe grâce aux bots, la multiplication des raquettes HTTP peut créer de problème de performance voire même rendre le site indisponible. Ces attaques peuvent toucher n'importe quels sites mais comme dit avant wordpress est une cible de choix.

J'ai donc voulu chercher une solution pour créer une nouvelle page login avec une URL propre, sans utiliser de redirection ni ajouter un nouveau plugin (ce qui m'était interdis) et en cohabitant avec les plugins déjà mis en place par l'agence. De plus pour mon intégration il m'était nécessaire d'avoir un nouvelle url pour utiliser mes hooks customs ainsi avoir un travail qui pourrait perdurer au cours des mises à jour WordPress.

Le début de mes recherches a commencé sur :

-<https://wpmarmite.com/brute-force-wordpress/>



Ensuite:

-<https://www.seedprod.com/change-wordpress-admin-login-url-without-plugin/>

-https://developer.wordpress.org/reference/hooks/login_url/

Puis:

-<https://www.inmotionhosting.com/support/edu/wordpress/lock-down-wordpress-admin-login-with-htaccess/>

-<https://wpcerber.com/how-to-hide-wp-admin-and-wp-login-php-from-possible-attacks/>

Et enfin:

-<https://www.daggerhartlab.com/how-to-disable-the-wordpress-login-form/>

Les solutions que j'ai pu trouver dans mes différentes recherches, sont toutes des solutions viables. La vraie difficulté ici est de faire marcher les solutions avec les plugins mis en place. J'ai donc aussi fais beaucoup de recherches sur les plugins de sécurité utilisées dans l'agence Yata! notamment Cerber.

Cerber , est un plugin de sécurité et d'anti-spam pour wordpress. Il protège les sites contre les pirates et contre les attaques brut force. Il effectue des analyses automatiques de logiciels malveillants et des virus. Suit l'activité des utilisateurs et des intrus avec des notifications par courrier électronique et mobile.

Malheureusement les recherche faite sur Cerber on été veine et les solutions pas assez satisfaisante à mes yeux.

De plus , la sécurité WordPress est un sujet très vaste et assez complexe. Je n'ai , ici traité d'une infime partie du problème ainsi que ces solutions.

Situation de travail ayant nécessité une recherche en anglais et extrait

Durant mes recherches et ma veille concernant la sécurité WordPress et les attaques brut forces, je n'ai quasiment eux de réponses que sur des sites anglophones. Comme je l'ai expliquer plus haut, je voulais sécuriser la page wp-login des attaques brut force.

Le site ou j'ai trouver le plus de réponses à mes questions fut cette pages :

-<https://www.seedprod.com/change-wordpress-admin-login-url-without-plugin/>
Cette page nous explique comment (avec ou sans plugin) et pourquoi changer l'url du login de WordPress.

À la façon d'un tutoriel, étape par étape , il nous explique la marche à suivre pour crée une nouvelle url pour le login et diriger les bots vers une page d'erreurs 404. La page étant vaste, j'ai choisi de présenter l'en-tête qui explique pourquoi il faudrait changer l'url du login WordPress.

Why Change Your WordPress Login URL?

WordPress is one of the most popular content management systems. And because of that, it's also popular among hackers.

Moreover, anyone can **find your default login page** by adding "wp-admin" or "wp-login.php" to your site's URL. So it makes your login URL the primary way for hackers to carry out their attacks.

One of the easiest ways to **secure your WordPress site** from hackers is to create a custom login URL. That way, you can hide your login page from malicious activity and only grant access to people you trust.

Today we'll share 2 ways to change your WordPress admin login URL.

The first method should only be used by advanced WordPress users because it involves changing your core WordPress files. For that reason, we recommend using method 2 which doesn't involve changing any code if you don't know what you're doing.

Voilà la traduction que j'ai pu faire de cette extrait:

« Pourquoi changer l'url de votre login WordPress?

WordPress est l'un des CMS les plus populaires. Et c'est donc pour cela qu'il est aussi populaire auprès des pirates informatiques.

En outre, n'importe qui peut trouver votre page login par défaut en ajoutant « wp-admin » ou « wp-login.php » à votre url de site internet. Ce qui fait de votre page de login le premier chemin pour les pirates où réaliser leurs attaques.

L'une des voies les plus simples pour sécuriser votre site WordPress des pirates sont de créer une url de login personnalisé. De cette façon, vous pouvez cacher votre page de login des activités malveillantes et n'accorder l'accès qu'aux personnes de confiance.

Aujourd'hui nous allons partager deux façons de changer votre url admin login de WordPress.

La première méthode ne peut être utilisée que par les utilisateurs avancés de WordPress car il implique de changer le core des fichiers WordPress. Pour cette raison, nous recommandons d'utiliser la deuxième méthode qui n'implique de changer aucun code, si vous ne savez pas ce que vous faites. »

En Conclusion

Le projet qui m'a été donné fut extrêmement formateur et enrichissant pour moi et a couvert beaucoup d'aspects du développement web. Ce qui m'a permis de progresser dans le domaine et surtout m'aider à améliorer ma confiance en moi vis-à-vis de mon travail. J'ai pu consolider mes acquis back-ends avec Wordpress , Php et front-end avec Javascript.

J'ai pu me rendre compte que j'étais capable de beaucoup de choses par mes propres moyens, dont je ne me sentais pas capable avant.

J'ai aussi beaucoup appris à titre professionnel sur le fonctionnement d'une agence web, et la cohabitation entre les différents cores de métier qui la compose. Je pense qu'avec cette vision globale de tous les métiers d'une agence web je pourrais avoir un travail plus efficace et une capacité de conseil.

Remerciements

J'aimerais remercier toutes les personnes qui ont contribué au bon déroulement de ces six mois de formation et de ces deux mois de stage.

Pour commencer , un grand merci à mon maître de stage, Tom Tassez, pour m'avoir fait confiance et m'avoir accueillie chez Yata , mais aussi pour sa disponibilité, ses conseils, son adaptabilité et son soutien tout au long du stage, pour m'avoir poussé à toujours mieux. Merci de m'avoir donné toutes ses petites victoires!

Merci aussi à David Blanchard, pour m'avoir donné l'opportunité de venir me former dans son agence et pour m'avoir donné de son temps.

Merci à mes 3 collègues de bureau Daphnée, Laura et Charly.

Daphnée Mendez Alija, la directrice artistique , pour sa patience, son calme, sa pédagogie et ces moments de re-motivation et de câlin pendant mes pertes de confiance !

Laura Dariel pour m'avoir permis de rester confiante pendant toute la période de stage!

Charly Dumont pour toutes ses blagues et sa bonne humeur

Merci à Fanny Berthaud pour m'avoir éclairée dans mes obscures problèmes de carrousel.

Mais aussi tout le reste de l'agence, pour leurs conseils, leurs techniques et leurs expertises dans leur domaine respectif.

Je tiens aussi à remercier OnlineFormaPro pour la chance que vous m'avez donné de pouvoir changer de vie dans un domaine que j'aime.

Pour ce qui est de la formation, un énorme merci à Alain Merucci , pour son apprentissage, sa pédagogie, sa bienveillance, pour toute la confiance qu'il met en nous, et surtout pour avoir cru en moi bien plus que moi-même. Tu fais partie de ma réussite ! Merci !

Et enfin, merci à tous les copains de la formation à votre soutien, votre aide, et toutes vos blagues ! Merci d'être les belles personnes que vous êtes !!

Annexe

Ici la fonction slider_produit() utiliser dans le hook login_footer de la page yata_login.php . Cette fonction comprend un tableau d'objet que j'ai exploité dans le slider des nouveaux produits sur la page de login ainsi que sur la page d'administration.

```
function slider_produit(){
    wp_enqueue_style('swiper-css', 'https://cdn.jsdelivr.net/npm/swiper@8/swiper-bundle.min.css', array(), YATAv, 'all');
    wp_enqueue_script('swiper-js', 'https://cdn.jsdelivr.net/npm/swiper@8/swiper-bundle.min.js', array(), YATAv, true);
    wp_enqueue_script('swiper-js-min', 'https://cdn.jsdelivr.net/npm/party-js@latest/bundle/party.min.js', array(), YATAv, true);
    $tableau = array(
        [
            'name' => 'Vos données SIT',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/data.svg',
        ],
        [
            'name' => 'Carte interactive',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/map.svg',
        ],
        [
            'name' => 'E-Commerce',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/grocery.svg',
        ],
        [
            'name' => 'Liste de vos communes',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/pins.svg',
        ],
        [
            'name' => 'Flux domaines skiables',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/Mountain.svg',
        ],
        [
            'name' => 'Espace Pro',
            'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore',
            'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/handShake.svg',
        ],
    )
}
```

```

[  

    'name' => 'Votre météo',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/sun.svg',  

],  

[  

    'name' => 'Vos horaires d\'ouverture',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/clock.svg',  

],  

[  

    'name' => 'Vos avis Google',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/star.svg',  

],  

[  

    'name' => 'Vos circuits de randonnées',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/Mountainman.svg',  

],  

]

```

```

[  

    'name' => 'Carnet de voyage',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/heartBook.svg',  

],  

[  

    'name' => 'Programme des animations',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/calendrier.svg',  

],  

[  

    'name' => 'Centrale de réservation',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/bell.svg',  

],  

[  

    'name' => 'Vos brochures',  

    'info' => 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

incididunt ut labore',  

    'picto' => get_template_directory_uri().'/config/admin/assets/img/picto/flyer.svg',  

],  

]

```

```
    );
?>
<section class="blocBas">
    <div class="produitYata swiper2">
        <div>
            <h2 class="produit">Les Produits Yata!</h2>
        </div>
        <div class="swiper-wrapper wrapper">
<?php
foreach ($tableau as $key => $value) {
    $name = $value['name'];
    $info = $value['info'];
    $picto = $value['picto'];
?>
        <div class="swiper-slide groupeProduit" >
            <div class="boitePicto">
                <img class="picto" src=<?= $picto ?>" alt="">
            </div>
            <div class="ProduitY">
                <p class="titreproduit"><?= $name ?></p>
                <p class="descproduit"><?= $info ?></p>
            </div>
        </div>
    </div>
</section>
<?php
}
add_action ('login_footer', 'slider_produit');
```

