



# MANUEL D'INSTALLATION ET DE MAINTENANCE

Projet n°2 : Création du site web de l'école maternelle Évariste Galois



COMMANDITAIRES : FLORENCE CORRIHONS – EMILIE LANNES

PROFESSEUR TUTEUR : PATRICK ETCHEVERRY

ETUDIANTS : LOÏC BARBE, MAËLYS BRAUD, SIMON FAVARD



# TABLE DES MATIÈRES :

Table des figures :	3
INSTALLATION	4
1. La base de données	4
2. Les fichiers du site	4
MAINTENANCE	5
1. Outils technologiques	5
1.1 Langages de programmation	5
1.2 Environnement de développement	6
2. Architecture du site	6
2.1 Architecture WordPress	6
2.2 Arborescence des fichiers	7
3. Modifications liées au code et à la base de données	8

## TABLE DES FIGURES :

FIGURE 1 : BASE DE DONNÉES	4
FIGURE 2 : INTERFACE CLOUD9	6
FIGURE 3 : DOSSIERS / FICHIERS THÈME ENFANT	7
FIGURE 4 : REDIRECTION UTILISATEUR	10
FIGURE 5 : CHANGEMENT URL	10
FIGURE 6 : PRIORITÉ THÈME ENFANT	11
FIGURE 7 : CRÉATION DE RÔLE	11
FIGURE 8 : PLUGIN ENFANT	12
FIGURE 9 : PLUGIN ENFANT	12
FIGURE 10 : FIL D'ARIANE	13
FIGURE 11 : APPEL DU FIL D'ARIANE	13
FIGURE 12 : SERVEUR D'HÉBERGEMENT	14
FIGURE 13 : REDIRECTION ADMIN	14
FIGURE 14 : SCROLL	14
FIGURE 15 : DÉCLARATION ADMIN.CSS	15
FIGURE 16 : PRIORITÉ THÈME ENFANT	15
FIGURE 17 : MODIFICATION TEXTE RESPOSITIVITÉ	15

# INSTALLATION

L'installation de notre site s'est déroulée en plusieurs étapes. Nous avons créé notre site WordPress sur "Cloud9", nous l'avons placé ensuite sur le serveur de production chez OVH.

À noter que les commanditaires ont souscrit, sous nos conseils et ceux de notre professeur tuteur M. ETCHEVERRY, un abonnement chez OVH.

## 1. LA BASE DE DONNÉES

Pour transférer la base de données, la première étape est d'exporter la base de données WordPress. Pour que la base de données soit fonctionnelle avec le nouveau site, il faut modifier quelques lignes : dans la table `egalois_options` "siteurl" et "home". Il faut remplacer le contenu de ces champs par la nouvelle URL du site.

De plus, il faut vérifier chaque page pour enlever l'ancienne URL, pour cela, nous avons utilisé pour la majorité des modifications la requête suivante :

```
UPDATE egalois_posts set post_content = REPLACE(post_content, 'https://projet-evariste-galois-lbarbe.c9users.io', 'http://adartza.iutbayonne.univ-pau.fr/~lbarbe/evariste-galois');
```

Ensuite, il faut intégrer cette base sur les serveurs d'OVH pour cela créer une base de données vierge sur le site d'OVH pour le nom de domaine choisi. Enfin, la dernière étape est d'importer la base donnée, précédemment exportée depuis le site en développement, sur cette base de données vierge.

## 2. LES FICHIERS DU SITE

Pour transférer les fichiers du site, il faut les prendre depuis notre serveur de développement et les modifier pour qu'ils soient fonctionnels sur le nom de domaine.

Dans le fichier `wp-config.php` situé à la racine du site, il faut changer les différents champs qui relient les fichiers du site à la base de données.

```
23  /** The name of the database for WordPress */
24  define('DB_NAME', 'maternellv');
25
26  /** MySQL database username */
27  define('DB_USER', 'maternellv');
28
29  /** MySQL database password */
30  define('DB_PASSWORD', '');
31
32  /** MySQL hostname */
33  define('DB_HOST', 'maternellv.mysql.db');
```

*Figure 1 : Base de données*

Après s'être connecté au serveur OVH grâce aux données fournies par les commanditaires, nous avons mis les fichiers de notre site via connexion FTP. De la même façon qu'avec le changement de la base de données, il faut ensuite vérifier le bon fonctionnement en testant chacune des pages sur le site.

# MAINTENANCE

Ce manuel permet de décrire toute l'architecture technique du produit final livré au commanditaire. Il devra alors être remis à ce dernier, qui sera responsable de l'évolution et de la maintenance du site.

Ce manuel sera sollicité dès que le site connaît un problème technique, et utilisé par un technicien extérieur qui n'a pas eu connaissance des moyens utilisés pour la fabrication du site.

Tout au long du projet, nous avons donc opéré à des modifications dans le code et la base de données afin que l'ensemble du site soit fonctionnel.

## 1. OUTILS TECHNOLOGIQUES

### 1.1 Langages de programmation

Pour la création technique du site, nous avons utilisé plusieurs langages de programmation :

#### HTML 5 :

Le langage HTML (HyperText Markup Language) est un langage permettant d'écrire des documents hypertexte pour le web. C'est un langage très simple, basé sur des balises (tag). Les balises servent à mettre en forme le texte, à réaliser des liens vers d'autres pages, à insérer des images, etc. . .

L'utilisation de la version HTML5 permet l'usage de nouvelles balises différentes des anciennes versions permettant d'intégrer des éléments par section (<section> ; <article> ; <nav> ; <aside> ; <header> ; <footer>). L'HTML5 dispose ainsi d'éléments bien pensés pour un meilleur balisage sémantique de l'information.

#### CSS 3 :

Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML. On le traduit en français par feuilles de style en cascade.

Le CSS3 introduit différents modules tels media queries, de nouveaux sélecteurs d'éléments, les animations (keyframes), l'ajout d'effets sur les textes. L'usage du CSS3 nous paraissait ainsi évident.

#### JAVASCRIPT :

Le Javascript est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Le standard pour JavaScript est ECMAScript. En 2012, tous les navigateurs modernes supportent complètement ECMAScript 5.1. Les anciens navigateurs supportent au minimum ECMAScript 3.

#### PHP 7 :

Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par tout le monde de façon totalement gratuite.

Sur un plan technique, le PHP s'utilise la plupart du temps côté serveur. Il génère du code HTML, CSS ou encore XHTML, des données (en PNG, JPG, etc.) ou encore des fichiers PDF. Il fait, depuis de nombreuses années, l'objet d'un développement spécifique et jouit aujourd'hui une bonne réputation en matière de fiabilité et de performances.

## 1.2 Environnement de développement

### AWS Cloud9

AWS Cloud9 vous permet d'exécuter votre environnement de développement sur une instance Amazon EC2 gérée ou un serveur UNIX existant prenant en charge SSH. Vous pouvez ainsi écrire, exécuter et déboguer des applications à l'aide d'un simple navigateur et sans avoir à installer ou à gérer un IDE local.

Cloud9 permet de collaborer facilement sur du code. il est possible de partager un environnement de développement avec une équipe en quelques clics et d'effectuer des programmations en groupe. Lorsqu'ils collaborent, les membres de l'équipe peuvent voir les entrées des autres participants du projet en temps réel et discuter instantanément les uns avec les autres depuis l'IDE.

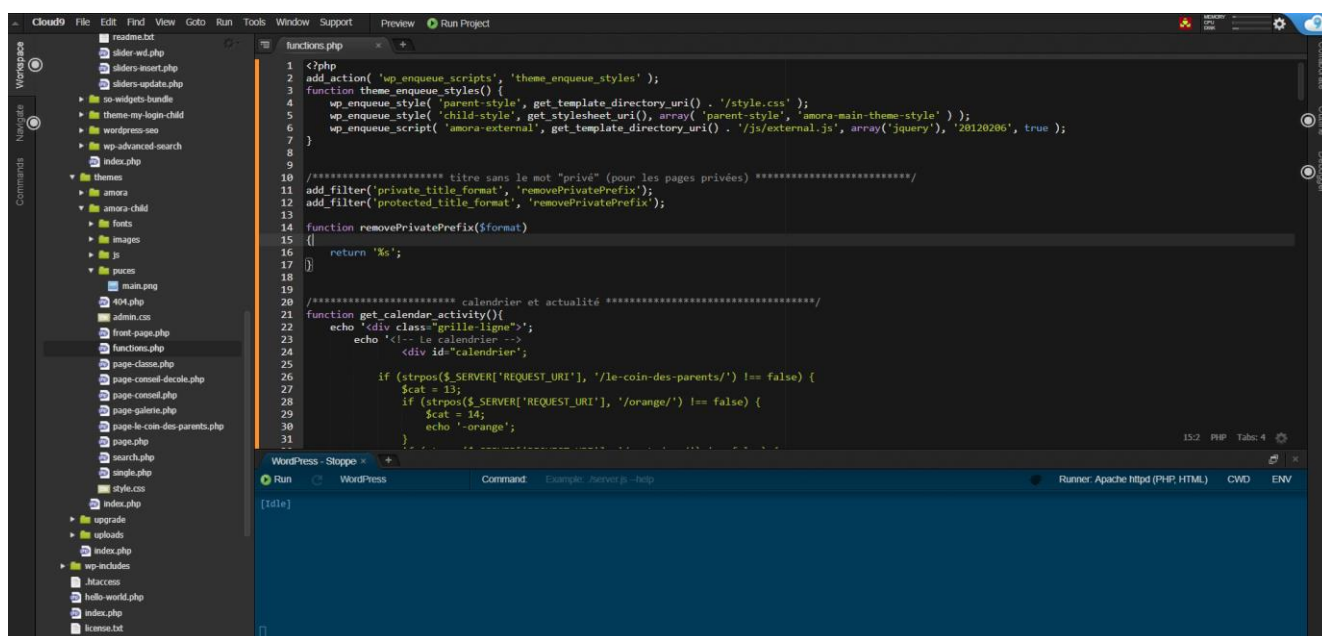


Figure 2 : Interface Cloud9

## 2. ARCHITECTURE DU SITE

### 2.1 Architecture WordPress

Afin de mieux comprendre WordPress, il est nécessaire de connaître au mieux l'architecture de celui-ci, pour cela il existe un site du nom de [wphierarchy.com](http://wphierarchy.com).

Ce site vous permettra de comprendre au mieux comment interagissent l'ensemble des fichiers fonctionnant sous WordPress, chaque clic vous amènera sur la page du codex WordPress correspondant afin de vous apporter des précisions supplémentaires.

## 2.2 Arborescence des fichiers

Afin de gérer les données et les fichiers du site, une interface de travail est proposée par Cloud9 comportant une arborescence de fichiers. L'ensemble des fichiers et des dossiers doivent être correctement rangés et nommés.

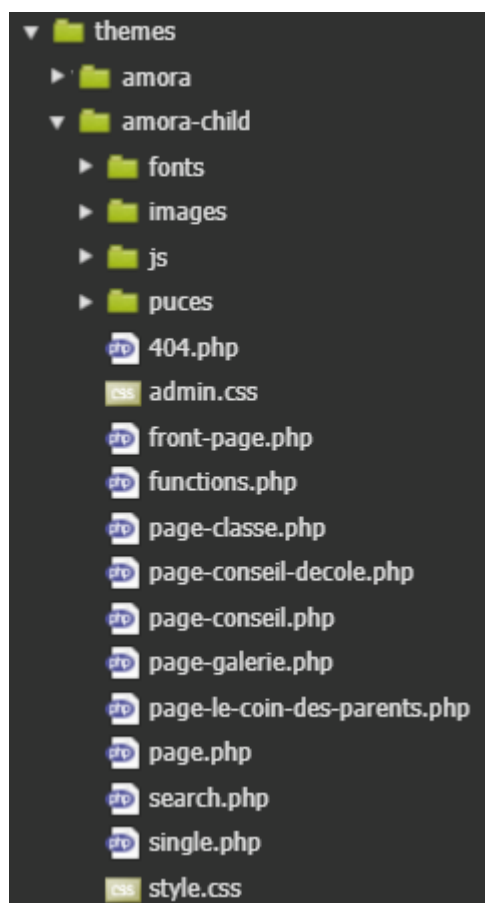


Figure 3 : Dossiers / fichiers thème enfant

### 3. MODIFICATIONS LIÉES AU CODE ET À LA BASE DE DONNÉES

#### MODIFICATION DU PRÉFIXE DES TABLES

*Un hacker peut accéder à la base de données en utilisant des failles connues de Wordpress ou de certains plugins (via les préfixes). Il pourrait alors ajouter du contenu sur le site, des commentaires, des spams ; détruire le site ou le faire planter ; extraire des informations.*

Avant de modifier les préfixes, il est recommandé d'effectuer une sauvegarde de toute la base de données. Le nouveau préfixe défini est : **egalois\_**

Tout d'abord, il faut remplacer le préfixe dans le fichier wp-config.php :

```
$table_prefix = 'wp_'; — $table_prefix = 'egalois_';
```

Ensuite, plusieurs modifications sont à suivre :

2. Changer le préfixe des tables grâce à une commande SQL pour le renommage de masse :  
Rename table wp\_commentmeta to egalois\_commentmeta;

3. Éditer la table wp\_options et la table wp\_usermeta :

```
UPDATE `egalois_options`  
SET `option_name` = REPLACE( option_name, 'wp_', 'egalois_' )  
WHERE `option_name` LIKE 'wp_%';
```

```
UPDATE `egalois_usermeta`  
SET `meta_key` = REPLACE( meta_key, 'wp_', 'egalois_' )  
WHERE `meta_key` LIKE 'wp_%';
```

**Attention** : Certains plugins peuvent avoir ajouté des tables possédant le préfixe wp\_. Il faut donc effectuer une recherche globale du term wp\_ afin de modifier les éléments restants wp\_.

Dès lors ces actions effectuées, il faut penser à créer un backup de la nouvelle base de données ; tester les différents types de pages du site ; tester l'interface administrateur ; tester les fonctionnalités et les plugins du site.

#### MODIFICATION DU LIEN HTTP EN HTTPS

Le HTTPS permet de sécuriser la connexion entre le client et le serveur en chiffrant les données qui transitent entre le navigateur web et le serveur web.

Le protocole HTTPS permet donc de vérifier l'identité du serveur web visité et permet donc à son visiteur de s'assurer d'être sur le bon site internet et de ne pas avoir été redirigé (par un malware par exemple) vers un faux serveur. De plus, les sites sécurisés via SSL bénéficient en outre, d'un avantage pour leur référencement dans les pages de résultats de recherche.

4. Mettre en place le HTTPS sur le site Wordpress



Il faut acquérir un certificat SSL qui peut être gratuit ou payant ; l'installer sur le serveur ; changer les permaliens HTTP en HTTPS. L'URL d'installation (images jointes, css, fichier js) doit également être modifiée.

À noter que le certificat SSL est inclus dans l'offre d'hébergement Web chez OVH.

Le plugin permettant de faciliter l'installation du certificat SSL sur le serveur WordPress est Really Simple SSL.

Après l'installation d'un SSL sur le site, il faut :

5. Tout rediriger vers le site en HTTPS
6. Remplacer les URL en HTTP
7. Vérifier les ressources chargées par le thème (contenus mixtes)
8. Forcer le HTTPS dans l'administration -> dans le fichier wp-config, il faut inscrire :  
`define('FORCE_SSL_ADMIN', true);`
9. Mettre à jour le fichier robots.txt : ajouter un "s" à l'adresse du sitemap
10. Mettre à jour le site dans Google Search Console : tutoriel intégré au site
11. Mettre à jour Google Analytics : Admin - Propriété - Paramètre de la propriété - Sélectionner https pour le champ URL par défaut.
12. Tester le certificat SSL : <https://www.ssllabs.com/ssltest/>

## CHANGEMENT DE L'URL /WP-ADMIN

La page de connexion WordPress est la porte d'entrée vers le back-office, elle vous permet d'accéder à la gestion de votre site internet mais surtout de votre page d'Accueil, d'ajouter des médias, de modifier votre design, de modifier vos Widgets, d'écrire des articles, des pages... Pour que les hackers n'y accèdent pas, nous l'avons modifié.

### 13. Rediriger les utilisateurs après une connexion réussie

```

/***** Redirection *****/
//Rediriger les utilisateurs après une connexion réussie.
function my_login_redirect( $redirect_to, $request, $user ) {
    //is there a user to check?

    if ( isset( $user->roles ) && is_array( $user->roles ) ) {
        //check for admins

        if ( in_array( 'administrator', $user->roles ) || in_array( 'editor', $user->roles ) ) {
            return $redirect_to;
        } else {
            $lien =(isset($_SERVER['HTTPS']) ? "https" : "http") . "://" . $_SERVER["HTTP_HOST"] . "/le-coin-des-parents/";
            return $lien;
        }
    } else {
        //si il y a une erreur de connexion
        if (array_key_exists('incorrect_password', $user->errors)){
            $user->errors['incorrect_password'][0] = "<strong>ERREUR</strong> : Ce mot de passe ne correspond pas à l'identifiant de connexion";
        }
        if (array_key_exists('invalid_username', $user->errors)){
            $user->errors['invalid_username'][0] = "<strong>ERREUR</strong> : Nom d'utilisateur non valide.";
        }

        if(strpos($_SERVER['HTTP_REFERER'], '/le-coin-des-enseignants') !== false){
            $lien = home_url() . "/le-coin-des-enseignants/";
            return $lien;
        } else {
            return $redirect_to;
        }
    }
}
add_filter( 'login_redirect', 'my_login_redirect', 10, 3 );

```

Figure 4 : Redirection utilisateur

Dans le fichier functions.php :

### 14. Changement de l'URL de connexion

Dans le fichier ".htaccess" :

```
1 RewriteRule ^admingalois$ https://maternellegalois.fr/wp-login.php [NC,L]
```

Figure 5 : Changement URL

## DÉVELOPPER UN THÈME ENFANT WORDPRESS

Un thème “enfant” WordPress, c’est un thème qui va reprendre exactement les fonctionnalités, le design et la mise en page d’un thème “parent” et permettre d’apporter des modifications à celui-ci sans s’attaquer à son intégrité.

Un thème enfant permet :

15. D’empêcher la perte des modifications d’un thème si le thème utilisé est mis à jour
16. D’accélérer le temps de développement

Le thème actuel (Amora) possède un fichier .css par défaut, qui est “amora-main-theme-style”.

Dans le fichier functions.php du thème enfant :

```
1 <?php
2 add_action( 'wp_enqueue_scripts', 'theme_enqueue_styles' );
3 function theme_enqueue_styles() {
4     wp_enqueue_style( 'parent-style', get_template_directory_uri() . '/style.css' );
5     wp_enqueue_style( 'child-style', get_stylesheet_uri(), array( 'parent-style', 'amora-main-theme-style' ) );
6     wp_enqueue_script( 'amora-external', get_template_directory_uri() . '/js/external.js', array('jquery'), '20120206', true );
7 }
```

Figure 6 : Priorité thème enfant

## CRÉATION DE RÔLE AVEC CERTAINES PERMISSIONS

Sachant que les parents doivent se connecter pour accéder à du contenu privé, nous avons créé un rôle sur mesure.

Nous avons donc ajouté un rôle qui ne peut que lire les pages, et lire les pages privées.

Dans le fichier functions.php :

```
91 // Ajout d'un role Parent
92 function egalois_add_role() {
93     add_role( 'parent', 'Parent', // son identifiant et son nom visible
94         array(
95             'read',
96             'read_private_pages' // On liste les droits à donner
97         )
98     );
99 }
100 }
101
102 add_action( 'init', 'egalois_add_role' ); // On lance la création de notre fonction
103
```

Figure 7 : Création de rôle

## CRÉATION D'UN PLUGIN ENFANT : "THEME MY LOGIN CHILD"

Il faut copier les fichiers du plugin de base et modifier le nom du plugin dans le fichier theme-my-login.php afin que WordPress ne mette pas à jour le plugin et que

Pour gérer le fait qu'il y ait deux formulaires de connexion sur le site (pour l'accès au coin des parents et depuis le coin des enseignants)

Modification de templates/login-form.php :

```
10 <form name="loginform" id="loginform"><?php $template->the_instance(); ?>"
11 action="<?php $template->the_action_url( 'login', 'login_post' ); ?>" method="post">
12 <?php if(strpos($ _SERVER['REQUEST_URI'], '/le-coin-des-enseignants') !== false):?>
13 <p class="tml-user-login-wrap">
14 <label for="user_login"><?php $template->the_instance(); ?><?php
15 if ( 'username' == $theme_my_login->get_option( 'login_type' ) ) {
16 _e( 'Identifiant', 'theme-my-login' );
17 } elseif ( 'email' == $theme_my_login->get_option( 'login_type' ) ) {
18 _e( 'E-mail', 'theme-my-login' );
19 } else {
20 _e( 'Identifiant ou E-mail', 'theme-my-login' );
21 }
22 ?></label>
23 <input type="text" name="log" id="user_login"><?php $template->the_instance(); ?>"
24 class="input" value="" size="20" />
25 </p>
26 <?php else: ?>
27 <p class="tml-user-login-wrap" hidden="hidden">
28 <label for="user_login"><?php $template->the_instance(); ?><?php
29 if ( 'username' == $theme_my_login->get_option( 'login_type' ) ) {
30 _e( 'Username', 'theme-my-login' );
31 } elseif ( 'email' == $theme_my_login->get_option( 'login_type' ) ) {
32 _e( 'E-mail', 'theme-my-login' );
33 } else {
34 _e( 'Username or E-mail', 'theme-my-login' );
35 }
36 ?></label>
37 <input type="text" name="log" id="user_login"><?php $template->the_instance(); ?>"
38 class="input" value="Parent" hidden="hidden" readonly="readonly" size="20" />
39 </p>
40 <?php endif; ?>
```

Figure 8 : Plugin enfant

## CRÉATION D'UN PLUGIN ENFANT : "CALENDAR CATEGORY CHILD"

Il faut copier des fichiers et modifier le nom du plugin dans le fichier calendar-category.php afin que WordPress ne mette pas à jour le plugin créé.

Dans calendar-category.php :

```
116 // Récupération du mois et de l'année pour mettre à jour le calendrier lors d'un changement de mois
117 if(isset($_GET['M'])){
118     $monthnum = $_GET['M'];
119     $year = $_GET['Y'];
120 }
```

Figure 9 : Plugin enfant

## INTÉGRATION D'UN FIL D'ARIANE VIA LE CODE

Le fil d'ariane est une aide à la navigation pour les visiteurs de votre site internet. Il sert de repère et permet de se situer sur le site. C'est un des éléments qui améliorent l'accessibilité d'un site web.

Afin de l'intégrer, il a d'abord fallu créer une fonction `page_breadcrumb()`.

Dans le fichier `functions.php` :

```
150 /***** fil d'ariane *****/
151 function page_breadcrumb(){
152     global $post;
153
154     echo "<div class='fil_d_ariane'>";
155
156     // Navigation
157     if (is_page() && !is_front_page() || is_single() || is_category()) {
158
159         echo ' <a title="Accueil" rel="nofollow" href="'.get_option('siteurl').'">Accueil</a> - ';
160
161         if (is_page()) {
162             $ancestors = get_post_ancestors($post);
163
164             if ($ancestors) {
165                 $ancestors = array_reverse($ancestors);
166                 foreach ($ancestors as $crumb) {
167                     echo ' <a href="'.get_permalink($crumb).'">'.get_the_title($crumb). '</a> - ';
168                 }
169             }
170         }
171
172         // Page actuelle
173         if (is_page() || is_single()) {
174             echo get_the_title();
175         }
176
177     } elseif (is_front_page()) {
178         // Page d'accueil
179
180         echo ' <a title="Accueil" rel="nofollow" href="'.get_option('siteurl').'">Accueil</a>';
181     }
182     echo "</div>";
183 }
184 }
```

Figure 10 : Fil d'Ariane

Appel dans `page.php` :

```
18 <!-- Fil d'ariane -->
19 <?php page_breadcrumb(); ?>
```

Figure 11 : Appel du fil d'Ariane

## CHANGEMENT DE SERVEUR D'HÉBERGEMENT

Dans le fichier `wp-config.php` :

Il faut modifier le nom, identifiant, mot de passe et hostname de la Base de données MySQL :

```
23  /** The name of the database for WordPress */
24  define('DB_NAME', 'c9');
25
26  /** MySQL database username */
27  define('DB_USER', substr(getenv('C9_USER'), 0, 16));
28
29  /** MySQL database password */
30  define('DB_PASSWORD', '');
31
32  /** MySQL hostname */
33  define('DB_HOST', getenv('IP'));
34
35  /** Database Charset to use in creating database tables. */
36  define('DB_CHARSET', 'utf8');
37
38  /** The Database Collate type. Don't change this if in doubt. */
39  define('DB_COLLATE', '');
```

Figure 12 : Serveur d'hébergement

Dans le fichier `.htaccess` :

Afin d'avoir un accès différent à la partie Administration du site :

```
1  RewriteRule ^admingalois$ https://maternellegalois.fr/wp-login.php [NC,L]
2
3  # BEGIN WordPress
4  <IfModule mod_rewrite.c>
5  RewriteEngine On
6  RewriteBase /
7  RewriteRule ^index\.php$ - [L]
8  RewriteCond %{REQUEST_FILENAME} !-f
9  RewriteCond %{REQUEST_FILENAME} !-d
10 RewriteRule . /index.php [L]
11 </IfModule>
12
13 # END WordPress
```

Figure 13 : Redirection admin

Dans `function.php` :

Pour gérer le positionnement du scroll lorsque l'on navigue dans le calendrier :

```
305 function my_custom_js() {
306     echo '<script type="text/javascript" src="https://maternellegalois.fr/wp-includes/js/jquery/jquery.js"></script>';
307     echo '<script type="text/javascript" src="https://maternellegalois.fr/wp-includes/js/jquery/jquery.cookie.js"></script>';
308     echo '<script type="text/javascript" src="https://maternellegalois.fr/wp-includes/js/jquery/jquery.rememberscroll.js"></script>';
309 }
310 // Add hook for front-end <head></head>
311 add_action('wp_head', 'my_custom_js');
```

Figure 14 : Scroll

Nous avons également modifié différentes adresses et liens manuellement dans différents éléments :

17. L'adresse d'envoi du formulaire de contact
18. Les liens du footer
19. Le plan du site (tous les liens)
20. L'emplacement de l'en-tête du site
21. L'emplacement des images des pages "Règlement intérieur" / "Organisation de l'école"

Pour remplacer la majeure partie des liens de la Base de Donnée, on peut insérer ce code SQL :

```
UPDATE egalais_posts set post_content = REPLACE(post_content, 'http://adartza.iutbayonne.univ-pau.fr/~lbarbe/evariste-galois', 'https://maternellegalois.fr');
```

## CRÉATION FICHIER CSS POUR LA PARTIE BACK OFFICE

Le fichier CSS permet des modifications visuelles des éléments d'une page. En effet, il permet de mettre en forme des fichier HTML. Nous avons donc créé en plus d'un fichier CSS correspondant à la partie FRONT OFFICE, un CSS permettant de modifier uniquement la partie BACK OFFICE.

Nous avons créé un fichier CSS nommé admin.css dans le dossier du thème enfant.

Dans le fichier functions.php :

```
299 function admin_style() {
300 wp_enqueue_style('admin-styles', get_stylesheet_directory_uri().'/admin.css');
301 }
302 add_action('admin_enqueue_scripts', 'admin_style');
```

Figure 15 : Déclaration admin.css

## MODIFIER LE TITRE "CHOISIR UNE PAGE" DANS LE MENU RESPONSIVE

Lors de l'affichage sur un écran plus petit qu'un écran d'ordinateur (style smartphone), l'élément permettant la navigation sur les pages affichait "Select a page". Il fallait donc modifier cet élément afin de l'afficher en français ("Choisir une page").

Nous avons copié le fichier external.js du thème parent amora dans le thème enfant amora-child.

Pour que le fichier soit pris en compte, nous avons utilisé cette fonction :

```
2 add_action( 'wp_enqueue_scripts', 'theme_enqueue_styles' );
3 function theme_enqueue_styles() {
4     wp_enqueue_style( 'parent-style', get_template_directory_uri() . '/style.css' );
5     wp_enqueue_style( 'child-style', get_stylesheet_uri(), array( 'parent-style', 'amora-main-theme-style' ) );
6     wp_enqueue_script( 'amora-external', get_template_directory_uri() . '/js/external.js', array('jquery'), '20120206', true );
7 }
```

Figure 16 : Priorité thème enfant

Nous avons ainsi modifié la ligne correspondante (21) dans external.js.

```
c+.'<option value="">Choisir une page</option>';
```

Figure 17 : Modification texte responsivité

## DÉ-RÉFÉRENCEMENT DES PAGES DE LA PARTIE PRIVÉE :

Pour que les informations et images de la partie du site ne soient pas visibles sur les moteurs de recherche, nous avons ajouté un fichier robots.txt qui empêche que les informations renseignées soient le résultat d'une recherche.

La syntaxe de ce fichier est la suivante :

User-agent : \*

Disallow:

Il faut ensuite ajouter après "disallow:" les chemins qui permettent d'accéder au page que l'on souhaite dé-référencer.

Nous avons également indiqué au plugin Yoast SEO, de ne pas les référencer (voir ci-dessous figure 18)

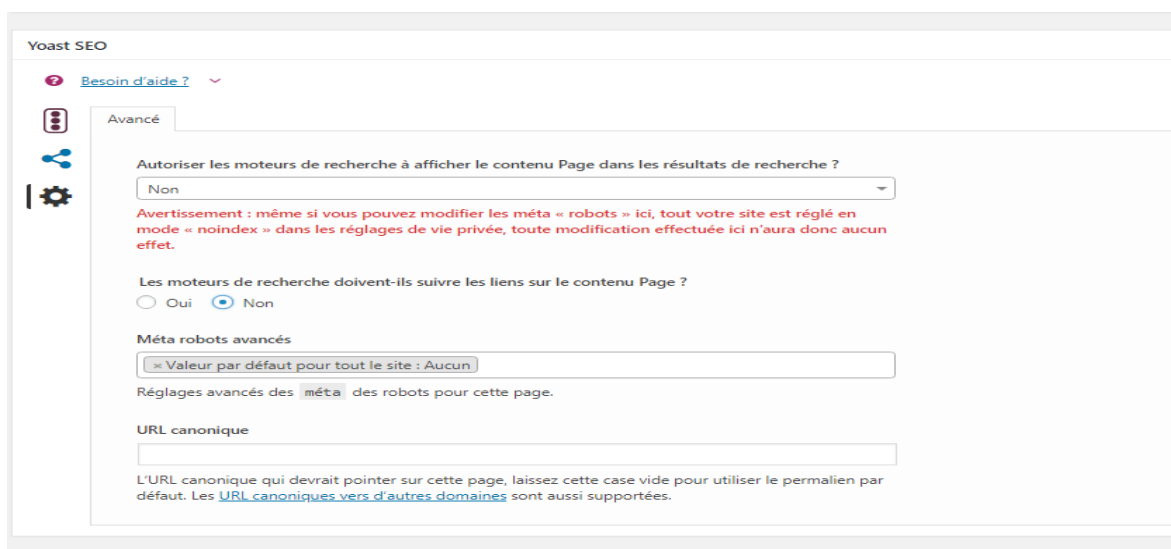


Figure 18 : Yoast SEO