

Dossier de Projet

Mlle Flo

Michel Grillon 07-08 / 2024



Dossier de projet *M'Zlle Flo*

*Pas en ligne actuellement, en test local et en
test sur le serveur du CEFJJ :*

https://cefii-developpements.fr/michel1385/Projet_final/mzlleflo/accueil/

Juillet-août 2024

Stagiaire : Grillon Michel

Tuteur de stage : aucun

Formation : Développeur Web Web Mobile

Table des matières

I. Compétences couvertes par le projet.....	4
II. Résumé en français.....	5
III. Cahier des charges.....	7
A. Description de la demande.....	8
i. Contexte	8
ii. Objectifs.....	9
iii. Cas d'utilisation.....	9
iv. Scénario	11
B. Arborescence	11
C. Zoning	12
D. Spécifications fonctionnelles	14
IV. Spécifications techniques	17
A. Outils de développement.....	17
i. JDE	17
ii. CMS.....	17
B. Production.....	11
i. Logiciel.....	19
ii. Réseau.....	20
C. Point sécurité	20
D. Adaptation pour le web mobile.....	21
V. Réalisation	23
A. Conception	23
i. Structure de données	23
B. Développement	48
i. Architecture	48
ii. Exemple de code	58
VI. Test.....	58
i. Tests du front-end (l'interface graphique).....	58
ii. Tests du back-end (persistance des données)	58
B. Déploiement.....	59
VII. Veille technique	60
VIII. Recherche technique sur des sites anglophones	60
IX. Recherche technique : extrait du site anglophone.....	63
X. Conclusion.....	65

<i>A. Bilan technique</i>	65
<i>B. Bilan humain</i>	65
<i>XI. Annexes</i>	65

1. Compétences couvertes par le projet

Liste des compétences du référentiel qui sont 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 :

En amont de la réalisation à proprement parlé de l'application, nous avons travaillé avec la cliente, sur l'ergonomie, la navigation, les users-stories et les documents de conceptions (voir quelques exemples plus loin) et penser aux différents formats visés (mobile, tablette, desktop)

Réaliser une interface utilisateur web statique et adaptable :

Au vu de la clientèle visée, qui couvre plusieurs tranches d'âges, il était nécessaire d'avoir une interface simple et facile d'utilisation, peu importe le niveau de compétence en informatique des clients et par la même, l'endroit et donc le type d'appareils utilisés

Développer une interface utilisateur web dynamique :

Après le choix de WordPress qui participe au dynamisme du site, nous avons décidé la création originale du thème et de ses extensions et plugins afin d'apporter un plus pour les utilisateurs, ainsi que quelques goodies (recherche via une simple icône qui déplie le champ et bouton de recherche, flèche de retour en haut de page...).

Réaliser une interface utilisateur avec une solution de gestion de contenu e-commerce

Le choix ayant été fait d'utiliser WordPress, l'installation en plus de Woo Commerce et la personnalisation des deux nous permettent cette solution.

Le projet initial devait être full PHP (en MVC et donc POO) : le choix de la cliente et le timing serré me concernant pour rendre les dossiers en ont décidé autrement.

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 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

WordPress se chargeant de créer ses tables (articles, utilisateurs, produits, etc...) et donc n'étant pas à l'origine des tables de bases, je documenterais les tables créées par mes extensions et en particulier "Art-Gallery" car étant la plus complète à ce niveau, ainsi que certaines personnalisations de WordPress et Woo Commerce.

II. Résumé en français

Une artiste indépendante (création de bijoux, de peintures, son statut est en cours de création) me demande de lui faire un site vitrine avec une petite boutique, une galerie, plus des tutoriels et ses travaux en cours.

C'est son premier site, jusqu'à présent elle se servait de Facebook et Tumblr pour montrer ses œuvres et de sites marchands divers pour les vendre, ce qu'elle ne souhaite plus (passage d'amateur en professionnel, augmentations des commissions sur certains de ces sites) et elle souhaite ajouter la location de certaines de ses œuvres, au mois, en plus de la vente.

La cliente souhaite augmenter sa visibilité, se démarquer (par rapport aux vendeurs des sites marchands) et donc étendre sa clientèle à partir du site. L'ouverture du site (mise en ligne) est planifiée pour début novembre.

Ce site doit avoir :

- *gestions des commandes et clients*
- *gestions des stocks et des articles en vente*
- *gestion des articles/news*

La clientèle est principalement des particuliers, mais peut aussi intéresser les entreprises (via SCE par exemple, pour des salles de réunions, hall d'accueil, etc...).

Nous décidons, en particulier la cliente, de partir sur WordPress pour le site, ce qui m'arrange, ayant un délai plus court que prévu (désistement deux mois avant l'examen, du contact et projet initial), mais :

- *Le thème soit totalement créé (pas d'utilisation de thème déjà existant)*
- *La partie administration de WordPress (Back-office) et en particulier la personnalisation, soit amélioré*
- *Utilisation d'extensions supplémentaires le plus réduit possible ou alors gratuites (mis à part pour la sécurité et des cas de figures qui seront développés plus loin) exception faite d'extensions Woo Commerce, pour certains paiements en particulier.*

Je lui propose en outre, la création d'extensions et de plugins supplémentaires afin de lui faciliter certaines tâches et, pour moi, de me rapprocher des attentes de l'examen.

III. Cahier des charges

La demande de la cliente porte sur :

Design :

Le design, adapté aux mobiles, doit être simple, intuitif et refléter les activités de l'artiste (taches de peintures, esquisses, etc.).

Les polices Google choisies par la cliente sont Great Vibes pour les titres, Send Flowers pour les catégories/menu et Playwrite Norge pour les textes.

La cliente a déterminé les couleurs (du texte d'entête, d'arrière-plan, du texte) et fournis les différentes images (fond de pages, articles/créations, produits).

Je lui ai créé le logo.

Contenu du site :

Le site comprendra plusieurs pages :

Accueil, Galerie, Boutique, Contact, Actualités/News, Partenaires, Tutoriels, À propos, Conditions générales de vente et d'utilisation, Mentions légales, Confidentialité, Cookies, Plan du site, Mon compte et Mon panier.

Fonctionnalités :

Parmi les fonctionnalités, on retrouve un espace client sécurisé, une administration accessible pour l'artiste, une galerie marchande combinée avec une galerie classique, un thème original et personnalisé, la gestion des commandes, clients, stocks et articles via Woo commerce.

Une option de recherche sera disponible sur toutes les pages, sous le menu, ainsi qu'une flèche permettant le retour en haut de page.

Plugins :

La création d'extensions et widgets prévus :

Extensions :

- *Pour l'intégration des réseaux sociaux (Facebook, Tumblr),*
- *Une page de contact avancée avec formulaire personnalisable, intégration Google Maps et notifications par e-mail,*
- *Et une galerie d'art interactive permettant d'ajouter et gérer ses œuvres.*

Widgets :

- *Catégories d'art : affichage des créations, par catégories, sur une ou plusieurs pages*
- *Archives des Créations : affichages des dernières Créations, sur une ou plusieurs pages*

Les créations des extensions et widgets seront expliqués plus loin.

Manipulations de la base de données (extensions) :

Je ne développerais pas la manipulation de la BDD de l'ensemble du site (et donc de WordPress) mais celles d'une extension créée (voir plus loin)

La base de données WordPress sera manipulée pour l'intégration des réseaux sociaux, la page de contact avancée et la galerie d'art interactive (qui sera l'exemple des manipulations plus loin).

Sécurité :

La sécurité sera renforcée par la validation et sécurisation des données, l'utilisation des fonctions d'API WordPress, un mot de passe robuste, la modification du préfixe "wp" des tables, la suppression des plugins et thèmes inutilisés, la mise à jour de la version WP, du thème et des plugins, la sauvegarde de la base de données et des contenus, et la protection du fichier de configuration de WordPress et .htaccess, ainsi que le masquage des dossiers sensibles (.htaccess).

A. Description de la demande

i. Contexte : *Présentation de l'entreprise, du service, de l'équipe ... :*

Actuellement, une seule personne : la créatrice, basée en Loire Atlantique (Dréfféac), elle travaille dans son atelier.

En plus de la création, elle gèrera :

- *Les expéditions (et les retours pour la location)*
- *La mise à jour du site (ajout d'articles, photos, etc.)*
- *La gestion des utilisateurs*
- *etc.*

Aidé de sa famille et son comptable... et moi pour le SAV technique.

ii. Objectifs

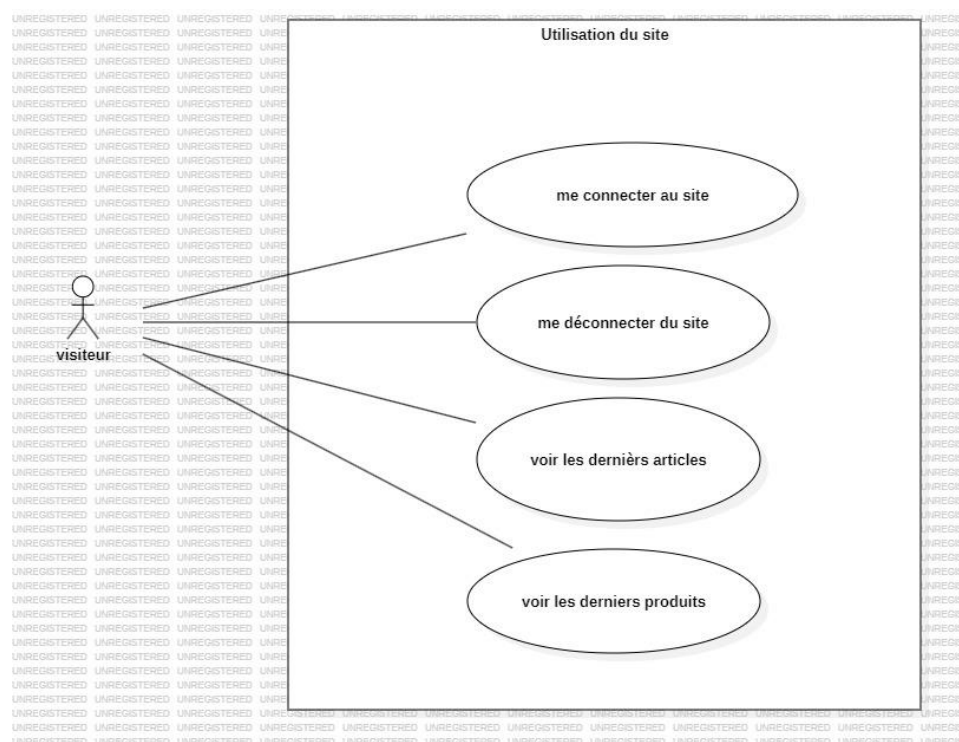
Comme indiqué précédemment (Résumé), la cliente souhaite augmenter sa visibilité, se démarquer (par rapport aux vendeurs des sites marchands) et donc étendre sa clientèle à partir du site. De plus, la création de son statut (en cours, finalisation prévue pour octobre-novembre 2024) et l'ajout d'une option de location au mois pour certains tableaux l'on amenée à la création d'un site vitrine. Son objectif de départ, la visibilité, le temps de se lancer, et ensuite les ventes, une fois le réseau mis en place.

Ses clients (particuliers principalement mais aussi potentiellement les entreprises intéressées via leur CSE ou directement le site).

iii. Cas d'utilisation

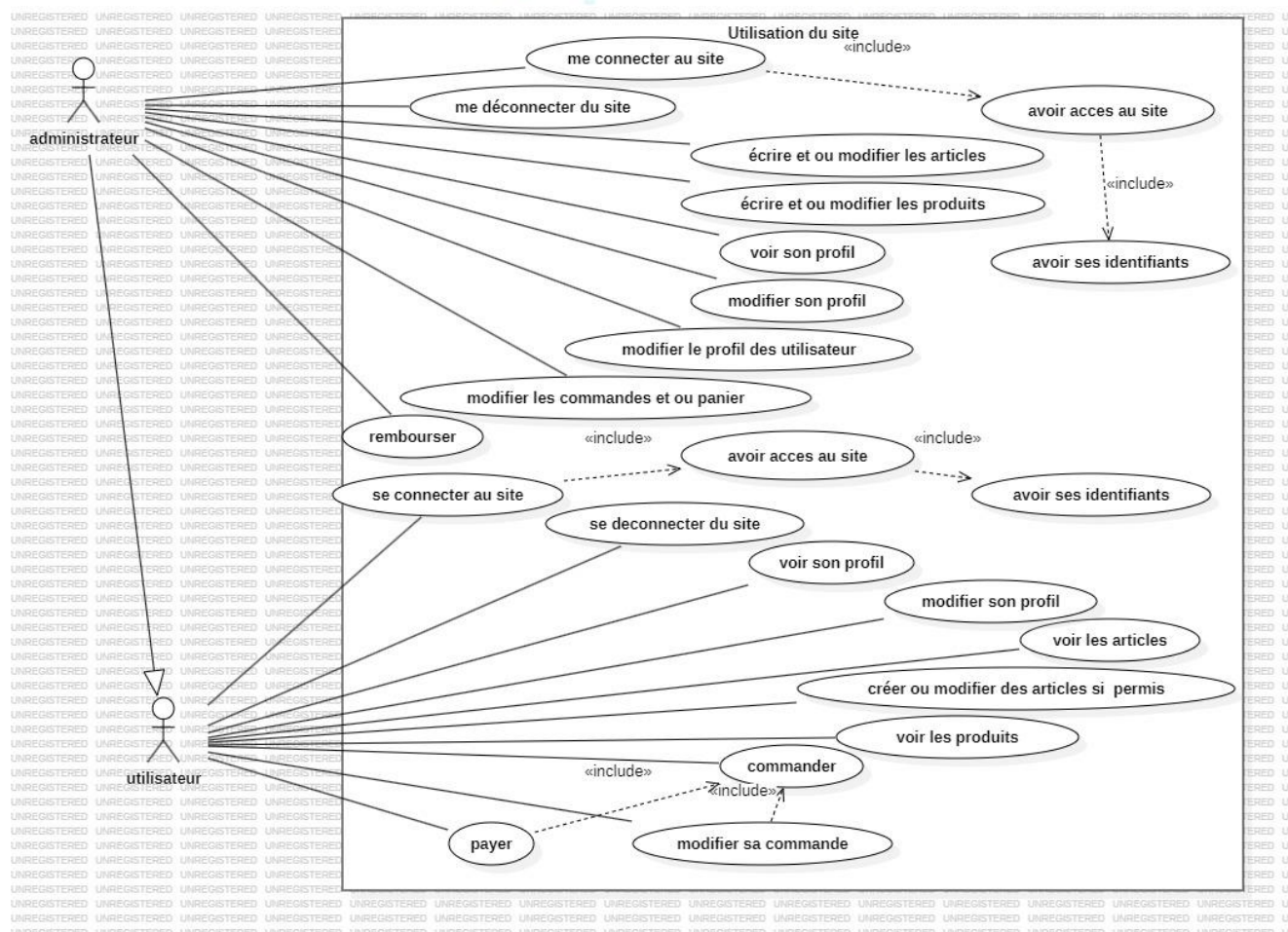
Diagramme UML accompagné d'un texte descriptif :

Pour un visiteur : celui-ci pourra se connecter ou se déconnecter s'il à créer son espace, regarder les dernières actualités ou les derniers produits ajoutés

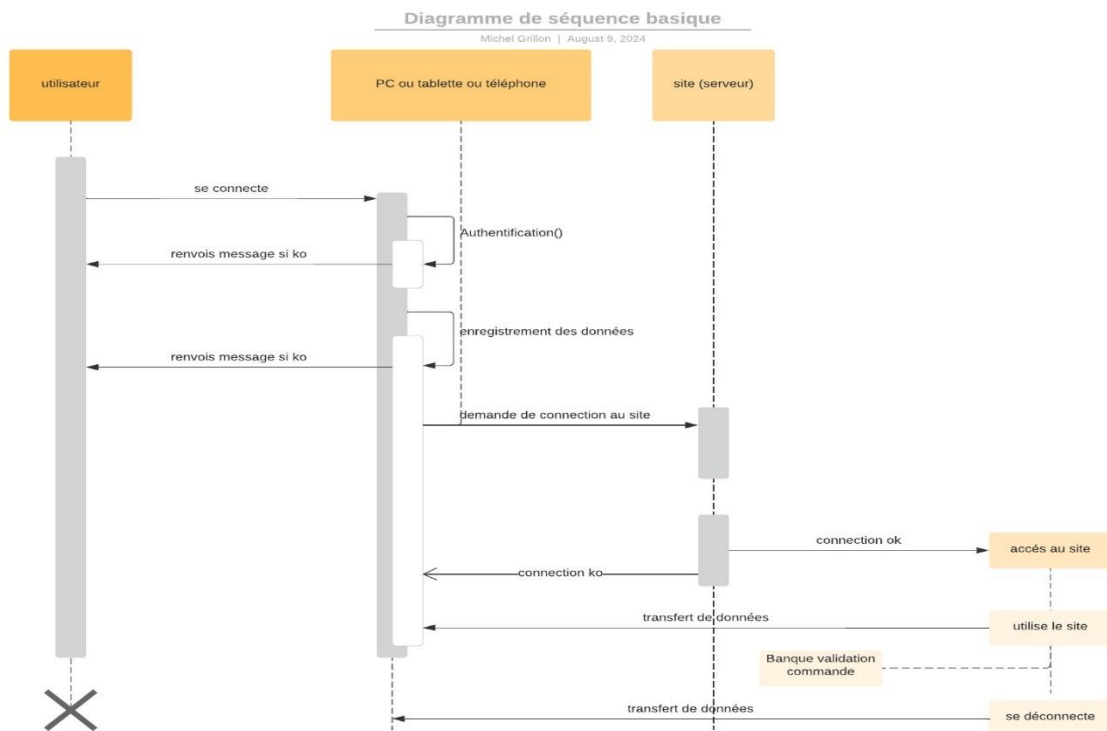


Coté utilisateur (connecté) : En plus des associations du visiteur, il pourra modifier toutes les informations de son profil (adresse, mdp, etc.) passer commandes ...

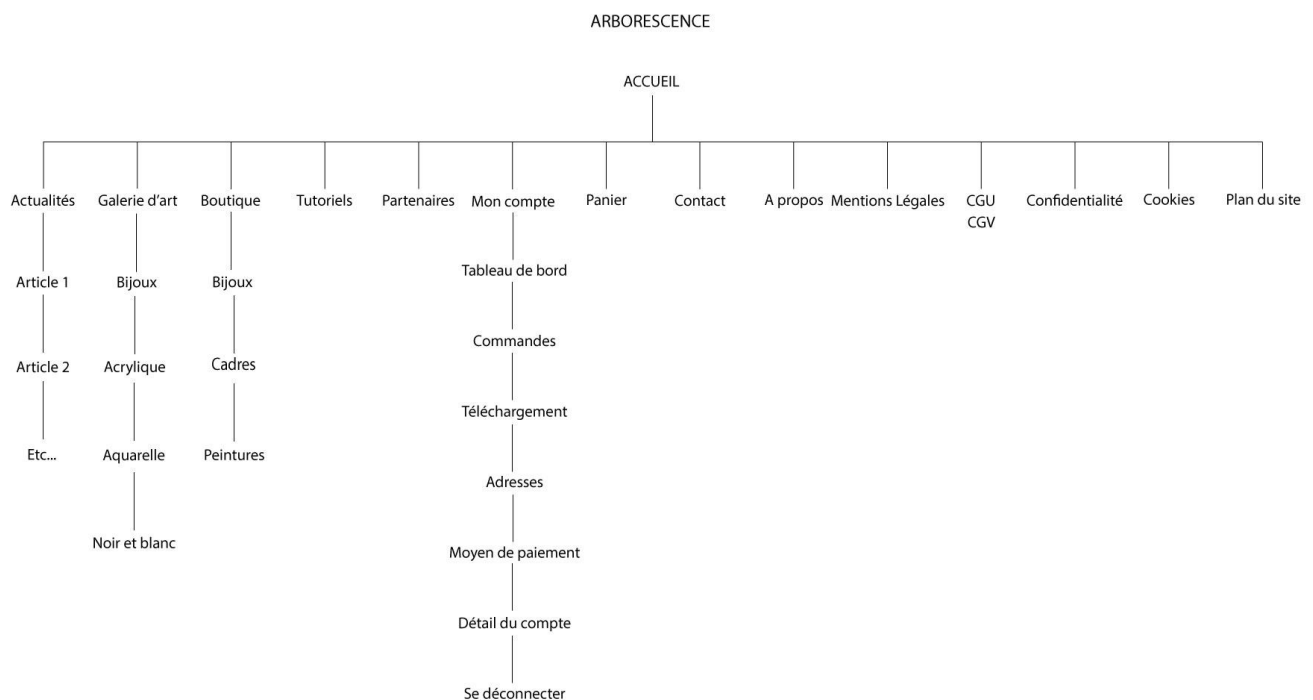
Et pour finir l'administrateur du site, qui pourra tout gérer, des utilisateurs en passant par les articles, les produits et donc la boutique et le contenu du site, aux utilisateurs (modifier des informations voir les supprimer à leurs demandes, etc.).



iv. Scénario : Diagramme UML de séquence basique :

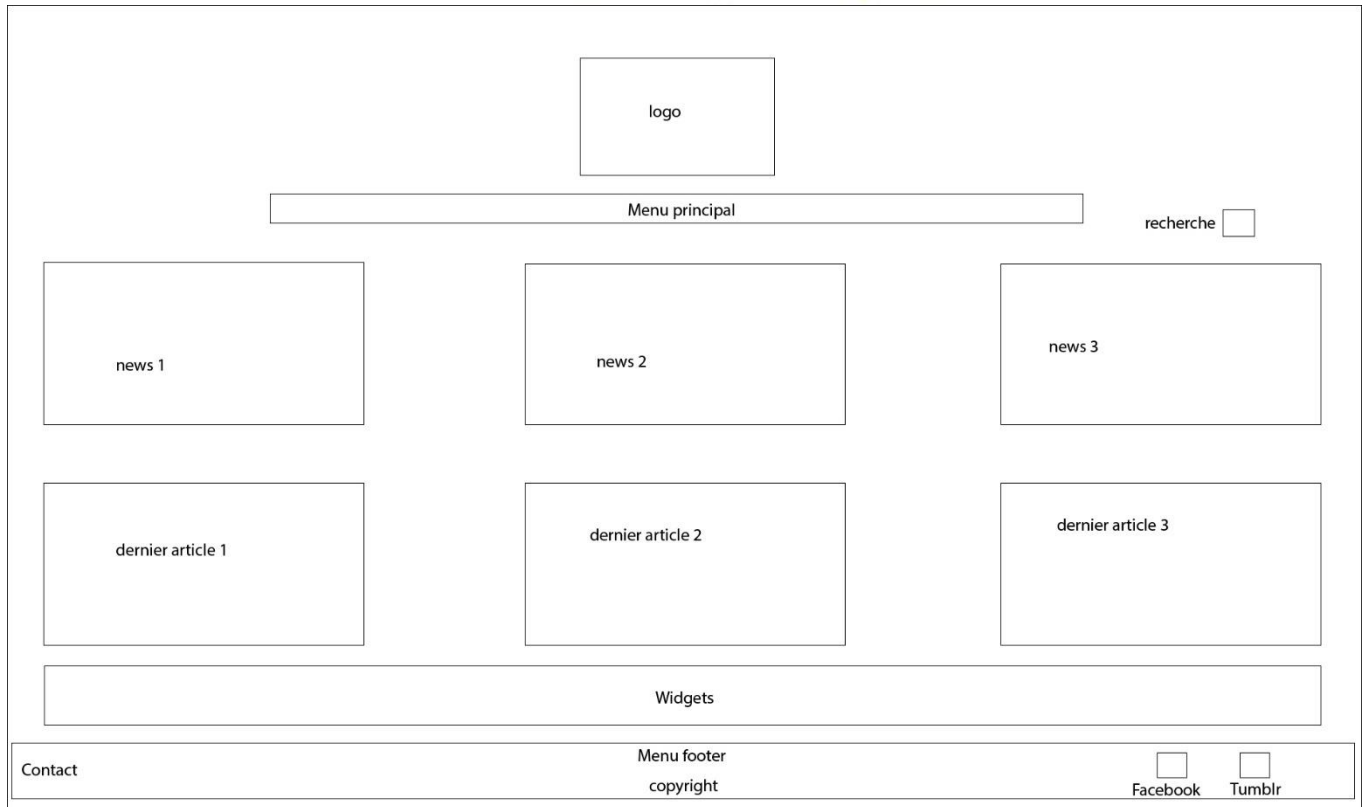


B. Arborescence

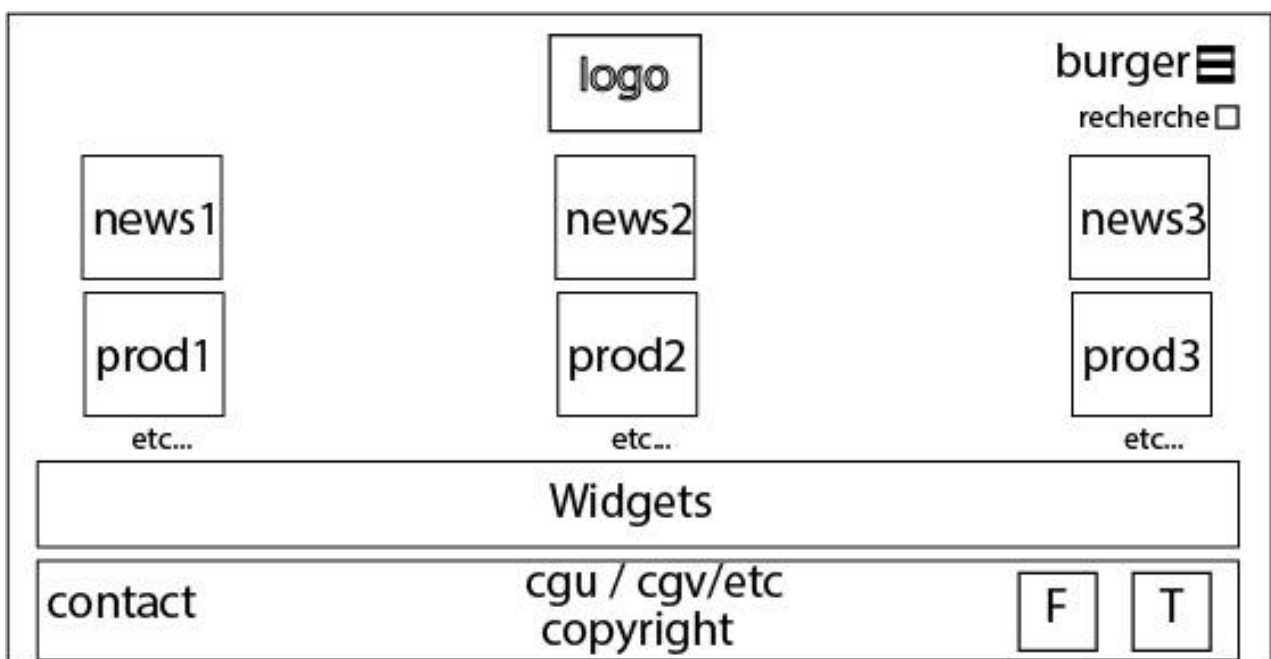


C. Zoning

Exemple page d'accueil desktop:



Autre exemple page d'accueil tablette :



Et la version mobile :

<div>F</div>	<div>T</div>	<div>logo</div>	<div><div></div><div>burger</div></div>
<div>news 1</div>			
<div>news 2</div>			
<div>news 3</div>			
<div>nouvel article 1</div>			
<div>nouvel article 2</div>			
<div>nouvel article 3</div>			
<div>contact</div>	<div>copyright</div>	<div>cgu / cgv/etc</div>	

A. Description détaillé des pages du site et spécifications fonctionnelles

Les articles, produits, et autres contenus seront mis à jour par la cliente. La fréquence de mise à jour dépendra du temps d'exécution de ses créations, des retours de location et donc des disponibilités des produits, de ses recherches de news et tutoriels...

Page 1 : Accueil

Description du contenu et des fonctionnalités sur cette page :

- *Derniers articles (5),*
- *Derniers produits (9),*
- *Dernières créations (5),*
- *Les types de créations (4),*
- *Widgets (produits en boutique, archives des créations, Catégories d'art, mots-clés)*

Mis à jour par la cliente dès que possible (pas de fréquence définie)

Page 2 : Actualités

Articles sur les créations en cours d'exécution

Page 3 : Galerie d'art

Affichage par sous-menus répartis en catégories, des différentes créations :

- *Bijoux*
- *Acrylique*
- *Aquarelle*
- *Noir et blanc*
- *Cadre*

Page 4 : Boutique

Comme son nom le laisse supposer, la porte d'entrée vers la vente et location de ses œuvres, réparties la aussi en catégories :

- *Bijoux,*
- *Cadres,*
- *Peintures*
 - o *Acrylique*
 - o *Aquarelle*
 - o *Noir et blanc*

Page 5 : Tutoriels

Des articles (et des liens) vers des tutoriels en relations avec la création, peintures etc.

Page 6 : Partenaires

Page regroupant des liens vers les fournisseurs de l'artiste

Page 7 : Mon compte

Pour toute la gestion du compte, la connexion ...

Dans le détail :

- *Se connecter, se déconnecter,*
- *Changer de mot de passe (volontairement ou suite à oubli)*
- *Modifier coordonnées*
- *Moyen de paiements*
- *Accéder à ses téléchargements*
- *Accéder à ses commandes*

Page 8 : Mon panier

Pour toute la gestion du panier, produits, etc.

Page 9 : Contact

Pour contacter la cliente.

> Accède à un formulaire de contact

> Oblige les internautes à renseigner au moins 4 champs : nom, prénom, email, ville

Insertion d'une Google maps

Page 10 : A propos

Pour en savoir plus sur la cliente

Page 11 : Plan du site

Comme son nom l'indique, le référencement de toutes les pages du site

Sans oublier les pages obligatoires (via menu footer) :

- *Mentions Légales*
- *CGU - CGV*
- *Confidentialité*
- *Cookies*

Et l'accès aux réseaux sociaux de la cliente via la aussi le menu footer .

IV. Spécifications techniques

Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

A. Outils de développement

i. IDE : Visual Studio Code

Extensions:

- *WordPress Snippets (wpprotocols.io) =>*
 - *Extraits pour chaque fonction, classe et constante WordPress.*
 - *Complétion automatique facile avec des indications de type et des tabulations aux bons endroits.*
 - *Abréviations de style Emmet pour toutes les fonctions. Par exemple, tapez gti pour la get_the_ID fonction. (2 lettres minimum)*
- *WordPress VS Code Extension Pack => Une collection d'extensions pour travailler avec des sites WordPress dans VS Code*

ii. CMS : WordPress

Extensions :

Pendant le développement (local et distant) :

- *Show Current Template :*

Extension permettant d'afficher le template utilisé pour la page en cours, au sein de la barre supérieure.

- *Theme Check :*

Extension permettant de tester un thème, si celui-ci est conforme.

- *Query Monitor :*

permet de visualiser les templates mais aussi les requêtes SQL, les fichiers CSS ou JS, les appels AJAX et bien plus.

*Attention cette extension masque par défaut les erreurs PHP ; pour les réactiver, il faut ajouter dans le fichier **wp-config.php** :*

```
define('QM_DISABLE_ERROR_HANDLER', true);
```

Hors développement (pour l'utilisation) :

WooCommerce

*Création de deux fonctions dans **functions.php** : 'custom_payment_gateways' et 'custom_deposit_calculation'*

- *custom_payment_gateways :*

- pour gérer les passerelles de paiement en fonction des variations paiement location et achat : paiement d'une commande par CB (pour la vente) ou par prélèvement automatique (pour les locations)

- *custom_deposit_calculation :*

- Pour gérer les dépôts ou les cautions : mandat de prélèvement de la caution (30% du prix de vente)

- modifications des fichiers :

wp-content/plugins/woocommerce/templates/single-product/add-to-cart/variable.php vers wp-content/themes/m3lleFlo/woocommerce/single-product/add-to-cart/variable.php : pour afficher les informations de paiement.

wp-content/plugins/woocommerce/templates/checkout/review-order.php vers wp-content/themes/m3lleFlo/woocommerce/checkout/review-order.php : pour afficher les

informations de paiement.

`wp-content/plugins/woocommerce/templates/cart/cart-totals.php` vers `wp-content/themes/m3lleFlo/woocommerce/cart/cart-totals.php` : pour afficher la caution.

=> ne fonctionne pas actuellement, en cours de résolution => utilisations d'extensions dédiées (pour WooCommerce) si toujours ko, lors de la mise en ligne du site.

Extensions pour WooCommerce :

- *WooPayments*

Si pas de résolutions des fichiers :

- *WooCommerce Conditional Payment Gateways* => pour gérer les passerelles de paiement conditionnelles basées sur les catégories de produits.

- "*WooCommerce Deposits*" ou "*YITH WooCommerce Deposits and Down Payments*" => permet de gérer les dépôts ou les cautions.

Production

i. Logiciels :

Windows 11

En plus de VSC, Notepad++ : en simultané, chacun sur un écran

Bloc-notes : comme pense bête, pour des copier-coller spécifiques, etc.

Photoshop et Illustrator (pour les images : création et retouches), Illustrator : wireframe et zonage

StarUML (évaluation) pour faire l'UML

Word : pour les dossiers et conversion en PDF

PowerPoint : pour le dossier oral

Adobe Acrobat : correction et finalisation des PDF

Wampserver 3.3.5 / Aache 2.4.56 / PHP 8.3.6 / MySql 8.3.0 et PhpMyAdmin pour le travail en local

JMerise (MCD, MLD ...)

ii. Réseau

FileZila pour les transferts FTP vers le serveur de test du CEFII et et PhpMyAdmin du serveur

B. Point sécurité

La sécurité sera renforcée par :

- la validation et sécurisation des données,
- l'utilisation des fonctions d'API WordPress,
- un mot de passe robuste (sauf actuellement en test, prévu avant la mise en ligne)
- la modification du préfixe "wp" des tables,
- la suppression des plugins et thèmes inutilisés,
- la mise à jour de la version WP, du thème et des plugins,
- la sauvegarde de la base de données et des contenus (actuellement sur un disque externe et sur le cloud, soit deux sauvegardes)
- et la protection du fichier de configuration de WordPress et .htaccess, ainsi que le masquage des dossiers sensibles (.htaccess) : pas encore fait, prévu en septembre, avant mise en ligne.

En prévision avant la mise en ligne définitive donc :

- Adresse de connexion changée (Plugin Hide Login)
- Version de WordPress masquée (via functions.php)
- Certificat SSL/TLS / WooPayments (carte bleu/mastercard/etc) dès que l'hébergeur aura été choisi
- mot de passe à modifier (quand fin de phase de test finie et avant mise en ligne)
- la protection du fichier de configuration de WordPress et .htaccess, ainsi que le masquage des dossiers sensibles (.htaccess)

c. Adaptation pour le web mobile

Elle se fait principalement par le fichier `style.css`, découpé en 4 "parties" :

- Partie globale, pour les styles qui ne dépendront pas des résolutions et seront communs à tous les types
- Partie Desktop qui ne gèrera que les grandes résolutions
- Partie tablette
- Partie mobile

Exemple :

Desktop :

```
/* Styles pour les écrans de bureau (largeur minimale de 769px) */
@media screen and (min-width: 769px) {
  /* Style pour les éléments de liste dans le menu principal */
  nav#menuPrincipal ul li {
    display: inline-block;
    /* Affiché en ligne */
  }
}
```

Tablette :

```
/* Styles pour les écrans de largeur maximale de 768px */
@media screen and (max-width: 768px) {
  /* Style pour les éléments de liste dans le menu */
  nav#menuPrincipal ul li {
    display: block;
    text-align: center;
    padding: 10px 0;
    margin-bottom: .5em;
    position: relative;
  }
}
```


Mobile :

```
/* Styles pour les écrans de largeur maximale de 768px */
@media screen and (max-width: 768px) {

/* Style pour les éléments de liste dans le menu */
nav#menuPrincipal ul li {
    display: block;
    text-align: center;
    padding: 10px 0;
    margin-bottom: .5em;
    position: relative;
}
```

Global :

```
/* Style pour les éléments de liste dans le menu */
nav#menuPrincipal ul li {
    display: block;
    margin-bottom: .5em;
    position: relative;
    font-family: "Playwrite NO", cursive;
    font-weight: bold;
}
```

Cela concerne bien évidemment d'autres éléments de la page, comme l'apparition du menu burger et de son icône, lorsque l'on passe de desktop vers les résolutions inférieures, alors que ceux-ci sont cachés en mode desktop.

v. Réalisation

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

A. Conception

i. Structure de données

Schémas conceptuels Merise : MCD, MLD, MPD

Comme indiqué plus tôt, je m'attache, concernant la base de données, aux tables relatives à l'extension "plugin-art-gallery" car c'est la plus complète des extensions faites. Avant d'expliquer sa structure, explication de l'extension :

C'est une extension pour créer une galerie d'art interactive où les visiteurs peuvent visualiser les œuvres de l'artiste avec des descriptions détaillées.

Fonctionnalités :

- Interface pour ajouter et gérer les œuvres d'art.*
- Affichage en grille avec zoom sur les images (léger effet).*
- Relation avec WooCommerce afin de récupérer les informations de la création pour en faire un produit de vente avec les informations déjà rentrées (gain de temps) ou inversement : utiliser le produit pour en faire une création.*
- Création d'une œuvre d'art : Lorsqu'une nouvelle œuvre d'art est créée dans l'admin de WordPress (utilisant le type de publication personnalisé artwork), un produit WooCommerce correspondant est automatiquement généré si aucun produit n'est encore associé à cette œuvre d'art.*
- Mise à jour de l'œuvre d'art : Lorsque vous mettez à jour une œuvre d'art, les informations du produit WooCommerce associés sont également mises à jour pour refléter les changements.*

Pour cela :

- Création du fichier **plugin-art-gallery.php** dans le dossier du même nom, lui-même dans le dossier **plugins** de **wp-content** de **WordPress**
- Affichage de la galerie d'art : Un **shortcode** `[art_gallery]` permet d'afficher la galerie d'art sur une page. Cette galerie devait inclure des fonctionnalités de **lightbox** pour afficher les images en grand format, ainsi que le **chargement infini** pour charger plus d'œuvres au fur et à mesure que l'utilisateur fait défiler la page. (la cliente à fini annuler ces effets).

1. Création d'une œuvre d'art et du produit **WooCommerce** associé :

```
2. // Créer un type de publication personnalisé pour les œuvres d'art
3. add_action('init', 'create_art_post_type');
4. function create_art_post_type()
5. {
6.     // Enregistrer le type de publication personnalisé 'artwork'
7.     register_post_type('artwork', array(
8.         'labels' => array(
9.             'name' => __('Galerie d\'art', 'textdomain'), // Nom au pluriel
10.            'singular_name' => __('Galerie d\'art', 'textdomain') // Nom au singulier
11.        ),
12.        'public' => true, // Accessible publiquement
13.        'has_archive' => true, // A une archive
14.        'supports' => array('title', 'editor', 'thumbnail'), // Supporte le titre, l'éditeur et l'image
        à la une
15.        'show_in_rest' => true, // Disponible dans l'API REST
16.        'taxonomies' => array('art_category'), // Supporte les catégories personnalisées
17.    ));
18.}
```

Tout comme les articles classiques de **WordPress** ou les produits de **Woo Commerce**, ont créés des catégories pour les œuvres d'arts :

```
// Enregistrer les catégories personnalisées pour les œuvres d'art
add_action('init', 'register_art_creation_taxonomy');
function register_art_creation_taxonomy()
{
```

```
$labels = array(
    'name' => __x('Art Categories', 'taxonomy general name'),
    'singular_name' => __x('Art Category', 'taxonomy singular name'),
    'search_items' => __('Search Art Categories'),
    'all_items' => __('All Art Categories'),
    'parent_item' => __('Parent Art Category'),
    'parent_item_colon' => __('Parent Art Category:'),
    'edit_item' => __('Edit Art Category'),
    'update_item' => __('Update Art Category'),
    'add_new_item' => __('Add New Art Category'),
    'new_item_name' => __('New Art Category Name'),
    'menu_name' => __('Art Categories'),
);

$args = array(
    'hierarchical' => true,
    'labels' => $labels,
    'show_ui' => true,
    'show_admin_column' => true,
    'query_var' => true,
    'rewrite' => array('slug' => 'art-category'),
);

register_taxonomy('art_category', array('artwork'), $args);
}

// Ajouter les catégories personnalisées à la section "Ajouter des éléments de menu"
add_filter('wp_nav_menu_item_post_type_meta_boxes',
    'add_art_categories_to_menu_meta_boxes');
function add_art_categories_to_menu_meta_boxes($post_types)
{
    $post_types[] = 'art_category'; // Utiliser les catégories personnalisées
    return $post_types;
}

// Ajouter une meta box pour lier les œuvres d'art aux produits WooCommerce
add_action('add_meta_boxes', 'add_artwork_product_meta_box');
function add_artwork_product_meta_box()
{
    add_meta_box(
        'artwork_product_meta_box', // ID de la meta box
```

```
__('Produit WooCommerce associé', 'textdomain'), // Titre de la meta box
'display_artwork_product_meta_box', // Fonction de rappel pour afficher la meta box
'artwork', // Type de publication auquel la meta box est ajoutée
'side', // Position de la meta box (côté)
'default' // Priorité de la meta box
);
}
```

Lorsqu'une nouvelle œuvre d'art est publiée, la fonction ***create_woocommerce_product_from_artwork*** est appelée. Cette fonction crée un produit WooCommerce associé à l'œuvre d'art, en utilisant les informations de l'œuvre (titre, description, image).

```
// Créer un produit WooCommerce lorsque l'œuvre d'art est publiée
add_action('publish_artwork', 'create_woocommerce_product_from_artwork');
function create_woocommerce_product_from_artwork($post_id)
{
    // Vérifier si le type de publication est 'artwork'
    if (get_post_type($post_id) != 'artwork') {
        return;
    }

    // Récupérer l'ID du produit WooCommerce associé à l'œuvre d'art
    $product_id = get_post_meta($post_id, '_artwork_product_id', true);

    // Si aucun produit WooCommerce n'est associé, créer un nouveau produit
    if (empty($product_id)) {
        $post = get_post($post_id);
        $title = $post->post_title;
        $content = $post->post_content;
        $image_id = get_post_thumbnail_id($post_id);

        // Créer un nouveau produit WooCommerce
        $product = new WC_Product_Simple();
        $product->set_name($title);
        $product->set_description($content);
        $product->set_status('publish');
        $product->set_catalog_visibility('visible');

        // Si une image à la une est définie, l'associer au produit
```

```

if ($image_id) {
    $product->set_image_id($image_id);
}

// Sauvegarder le produit et mettre à jour la meta donnée avec l'ID du nouveau produit
$new_product_id = $product->save();
update_post_meta($post_id, '_artwork_product_id', $new_product_id);
}

```

Il faut ensuite pouvoir afficher les œuvres :

```

function display_artwork_product_meta_box($post)
{
    // Récupérer l'ID du produit WooCommerce associé à l'œuvre d'art
    $product_id = get_post_meta($post->ID, '_artwork_product_id', true);

    // Afficher le champ de sélection pour lier l'œuvre d'art à un produit WooCommerce
    echo '<label for="artwork_product_id">' . __('Sélectionner le produit WooCommerce', 'textdomain') . '</label>';
    echo '<select name="artwork_product_id" id="artwork_product_id">';
    echo '<option value="">' . __('Aucun', 'textdomain') . '</option>';

    // Récupérer tous les produits WooCommerce
    $products = wc_get_products(array('limit' => -1));
    foreach ($products as $product) {
        // Afficher chaque produit dans le champ de sélection
        echo '<option value="" . $product->get_id() . "" . selected($product_id, $product->get_id(), false) . ">' . $product->get_name() . '</option>';
    }
    echo '</select>';
}

```

Et une sauvegarde est effectuée :

```

// Sauvegarder la meta box lors de la sauvegarde de la publication
add_action('save_post', 'save_artwork_product_meta_box');
function save_artwork_product_meta_box($post_id)
{
    // Vérifier si l'ID du produit WooCommerce est défini

```

```
if (isset($_POST['artwork_product_id'])) {
    // Mettre à jour la meta donnée avec l'ID du produit WooCommerce
    update_post_meta($post_id, '_artwork_product_id',
    sanitize_text_field($_POST['artwork_product_id']));
}
}
```

2. Mise à jour de l'œuvre d'art :

Lorsque l'œuvre d'art est mise à jour, la fonction `update_woocommerce_product_from_artwork` met à jour les informations du produit WooCommerce associé pour correspondre aux nouvelles informations de l'œuvre d'art.

```
// Mettre à jour le produit WooCommerce lorsque l'œuvre d'art est mise à jour
add_action('save_post', 'update_woocommerce_product_from_artwork', 20, 1);
function update_woocommerce_product_from_artwork($post_id)
{
    // Vérifier si le type de publication est 'artwork'
    if (get_post_type($post_id) != 'artwork') {
        return;
    }

    // Récupérer l'ID du produit WooCommerce associé à l'œuvre d'art
    $product_id = get_post_meta($post_id, '_artwork_product_id', true);

    // Si un produit WooCommerce est associé, mettre à jour ses informations
    if (!empty($product_id)) {
        $post = get_post($post_id);
        $title = $post->post_title;
        $content = $post->post_content;
        $image_id = get_post_thumbnail_id($post_id);

        // Récupérer le produit WooCommerce
        $product = wc_get_product($product_id);
        if ($product) {
            // Mettre à jour les informations du produit
            $product->set_name($title);
            $product->set_description($content);

            // Si une image à la une est définie, l'associer au produit
```



```

        if ($image_id) {
            $product->set_image_id($image_id);
        }

        // Sauvegarder le produit
        $product->save();
    }
}
}

// Charger les œuvres d'art supplémentaires pour le chargement infini via Ajax
add_action('wp_ajax_load_more_artworks', 'load_more_artworks');
add_action('wp_ajax_nopriv_load_more_artworks', 'load_more_artworks');

```

3. Affichage de la galerie d'art

Le shortcode [art_gallery] génère la galerie d'art, affichant les œuvres sous forme de vignettes. Annulé => Chaque vignette est liée à une lightbox pour afficher l'image en grand format. Le bouton "Charger plus" utilise AJAX pour charger plus d'œuvres sans recharger la page.

=> création du fichier **art-gallery.js** (voir annexe) et de son dossier dans wp-content/plugins (et des sous-dossier "js" et "css" dans le dossier de l'extension, et sous-dossier widgets)

=> mise à jour de scripts.js pour ajout lightbox (actuellement commenté, à la demande de la cliente)

=> import de <https://cdnjs.cloudflare.com/ajax/libs/lightbox2/2.11.4/js/lightbox.min.js> et de <https://cdnjs.cloudflare.com/ajax/libs/lightbox2/2.11.4/css/lightbox.css> => création de lightbox.js (dossier js) et de lightbox.css (dossier css du plugin). => annulé par la cliente, plus d'utilisation de Lightbox.

=> créations des widgets associés pour utiliser les catégories et archives créés par l'extension, comme des archives et catégories classiques : dans la partie Apparence/Widget et mise à jour des widgets pour fonctionner aussi comme des blocs afin d'être placés n'importe où...

Reste à faire : code pour vérifier les mises à jour, créer l'API de mise à jour (JSON), ajout d'un fichier de version (format zip téléchargeable) ...

Autres opérations liées à cette extension :

*Modification du fichier **archive-product.php** de woocommerce en template dans le dossier mon dossier thème :*

- Header des Produits : Gardez le header original pour afficher le titre de la page et la description*
- Filtres de la Galerie d'Art : Ajoutez des filtres pour les catégories de produits.*
- Boucle de Produits WooCommerce : Inclure la logique pour afficher les œuvres d'art personnalisées.*

*Créations de deux widgets : "**class-art-categories-widget.php**" et "**class-artwork-archives-widget**" dans un sous dossier "widgets" : **wp-content/plugins/plugin-art-gallery/widgets/***

*Création de l'extension "**plugin-art-gallery-block**" : permet de créer un block pour les archives et un bloc pour les catégories, créer par l'extension "**plugin-art-gallery**" et à partir de ses deux widgets.*

Ces blocs sont donc plaçable n'importe où, contrairement aux widgets qui ne sont plaçables que dans les sidebars et footer.

On a donc :

Pour plugin-art-gallery :

wp-content/plugins/plugin-art-gallery/plugin-art-gallery.php,

puis :

wp-content/plugins/plugin-art-gallery/render-functions.php

puis :

wp-content/plugins/plugin-art-gallery/css/lightbox.css

puis :

wp-content/plugins/plugin-art-gallery/js/art-gallery.js

puis :

wp-content/plugins/plugin-art-gallery/js/lightbox.js

puis :

wp-content/plugins/plugin-art-gallery/widgets/class-art-categories-widget.php

puis :

wp-content/plugins/plugin-art-gallery/widgets/class-artwork-archives-widget.php

Pour plugin-art-gallery-blocks :

wp-content/plugins/plugin-art-gallery-blocks/plugin-art-gallery-blocks.php

puis :

wp-content/plugins/plugin-art-gallery-blocks/css/art-gallery-blocks.css

puis :

wp-content/plugins/plugin-art-gallery-blocks/js/art-gallery-blocks.js

Manipulation de la base de données pour cette extension :

L'extension effectue plusieurs manipulations de la base de données WordPress pour gérer les œuvres d'art et les produits WooCommerce associés. Voici un résumé des manipulations principales :

Création de publications personnalisées :

Enregistrement d'un type de publication personnalisé (artwork).

Enregistrement des métadonnées associées aux œuvres d'art (comme l'ID du produit WooCommerce).

Création et mise à jour des produits WooCommerce :

Création de nouveaux produits WooCommerce lorsque de nouvelles œuvres d'art sont publiées.

Mise à jour des produits WooCommerce existants lorsque les œuvres d'art sont mises à jour.

Sauvegarde des métadonnées :

Enregistrement de l'ID du produit WooCommerce associé en tant que métadonnée de l'œuvre d'art.

Annulé => Requêtes AJAX pour le chargement infini des œuvres d'art.

Détail des manipulations spécifiques :

Création et enregistrement d'un type de publication personnalisé :

Utilisation de la fonction `register_post_type` pour créer un type de publication artwork.

Sauvegarde des métadonnées :

Utilisation de la fonction **`add_post_meta`** pour ajouter des métadonnées.

Utilisation de **`update_post_meta`** pour mettre à jour les métadonnées existantes.

Création et mise à jour des produits WooCommerce :

Création de nouveaux produits avec **`WC_Product_Simple`** et enregistrement avec **`save`**.

Mise à jour des produits existants en récupérant le produit avec **`wc_get_product`** et en mettant à jour les propriétés avant d'appeler **`save`**.

Codes concernés

Dans le fichier **`plugin-art-gallery.php`**, les lignes ayant une interaction avec la base de données (création, modification ou suppression de tables ou données) sont les suivantes :

Création d'un type de publication personnalisé pour les œuvres d'art :

```
add_action('init', 'create_art_post_type');
function create_art_post_type()
{
    // Enregistrer le type de publication personnalisé 'artwork'
    register_post_type('artwork', array(
        'labels' => array(
            'name' => __('Galerie d\'art', 'textdomain'), // Nom au pluriel
            'singular_name' => __('Galerie d\'art', 'textdomain') // Nom au singulier
        ),
        'public' => true, // Accessible publiquement
        'has_archive' => true, // A une archive
        'supports' => array('title', 'editor', 'thumbnail'), // Supporte le titre, l'éditeur et l'image à la une
        'show_in_rest' => true, // Disponible dans l'API REST
        'taxonomies' => array('art_category'), // Supporte les catégories personnalisées
    ));
}
```

Enregistrement des catégories personnalisées pour les œuvres d'art :

```
add_action('init', 'register_art_creation_taxonomy');
function register_art_creation_taxonomy()
{
    $labels = array(
        'name' => _x('Art Categories', 'taxonomy general name'),
        'singular_name' => _x('Art Category', 'taxonomy singular name'),
        'search_items' => __('Search Art Categories'),
    );
```

```
'all_items' => __('All Art Categories'),
'parent_item' => __('Parent Art Category'),
'parent_item_colon' => __('Parent Art Category:'),
'edit_item' => __('Edit Art Category'),
'update_item' => __('Update Art Category'),
'add_new_item' => __('Add New Art Category'),
'new_item_name' => __('New Art Category Name'),
'menu_name' => __('Art Categories'),
);

$args = array(
    'hierarchical' => true,
    'labels' => $labels,
    'show_ui' => true,
    'show_admin_column' => true,
    'query_var' => true,
    'rewrite' => array('slug' => 'art-category'),
);

register_taxonomy('art_category', array('artwork'), $args);
}
```

Ajout d'une meta box pour lier les œuvres d'art aux produits WooCommerce :

```
add_action('add_meta_boxes', 'add_artwork_product_meta_box');
function add_artwork_product_meta_box()
{
    add_meta_box(
        'artwork_product_meta_box', // ID de la meta box
        __('Produit WooCommerce associé', 'textdomain'), // Titre de la meta box
        'display_artwork_product_meta_box', // Fonction de rappel pour afficher la meta box
        'artwork', // Type de publication auquel la meta box est ajoutée
        'side', // Position de la meta box (côté)
        'default' // Priorité de la meta box
    );
}
```

Sauvegarde de la meta box lors de la sauvegarde de la publication :

```
add_action('save_post', 'save_artwork_product_meta_box');
function save_artwork_product_meta_box($post_id)
{
    // Vérifier si l'ID du produit WooCommerce est défini
    if (isset($_POST['artwork_product_id'])) {
```

```
// Mettre à jour la meta donnée avec l'ID du produit WooCommerce
update_post_meta($post_id, '_artwork_product_id',
sanitize_text_field($_POST['artwork_product_id']));
}
}
```

Création d'un produit WooCommerce lorsque l'œuvre d'art est publiée :

```
add_action('publish_artwork', 'create_woocommerce_product_from_artwork');
function create_woocommerce_product_from_artwork($post_id)
{
    // Vérifier si le type de publication est 'artwork'
    if (get_post_type($post_id) != 'artwork') {
        return;
    }

    // Récupérer l'ID du produit WooCommerce associé à l'œuvre d'art
    $product_id = get_post_meta($post_id, '_artwork_product_id', true);

    // Si aucun produit WooCommerce n'est associé, créer un nouveau produit
    if (empty($product_id)) {
        $post = get_post($post_id);
        $title = $post->post_title;
        $content = $post->post_content;
        $image_id = get_post_thumbnail_id($post_id);

        // Créer un nouveau produit WooCommerce
        $product = new WC_Product_Simple();
        $product->set_name($title);
        $product->set_description($content);
        $product->set_status('publish');
        $product->set_catalog_visibility('visible');

        // Si une image à la une est définie, l'associer au produit
        if ($image_id) {
            $product->set_image_id($image_id);
        }

        // Sauvegarder le produit et mettre à jour la meta donnée avec l'ID du nouveau produit
        $new_product_id = $product->save();
        update_post_meta($post_id, '_artwork_product_id', $new_product_id);
    }
}
```

Mise à jour du produit WooCommerce lorsque l'œuvre d'art est mise à jour :

```
add_action('save_post', 'update_woocommerce_product_from_artwork', 20, 1);
function update_woocommerce_product_from_artwork($post_id)
{
    // Vérifier si le type de publication est 'artwork'
    if (get_post_type($post_id) != 'artwork') {
        return;
    }

    // Récupérer l'ID du produit WooCommerce associé à l'œuvre d'art
    $product_id = get_post_meta($post_id, '_artwork_product_id', true);

    // Si un produit WooCommerce est associé, mettre à jour ses informations
    if (!empty($product_id)) {
        $post = get_post($post_id);
        $title = $post->post_title;
        $content = $post->post_content;
        $image_id = get_post_thumbnail_id($post_id);

        // Récupérer le produit WooCommerce
        $product = wc_get_product($product_id);
        if ($product) {
            // Mettre à jour les informations du produit
            $product->set_name($title);
            $product->set_description($content);

            // Si une image à la une est définie, l'associer au produit
            if ($image_id) {
                $product->set_image_id($image_id);
            }

            // Sauvegarder le produit
            $product->save();
        }
    }
}
```

Shortcode pour afficher la galerie d'art :

```
add_shortcode('art_gallery', 'display_art_gallery');
function display_art_gallery()
{
    ob_start();
    ?>
    <div class="art-gallery">
    <?php
```



```
$args = array(
    'post_type' => 'artwork',
    'posts_per_page' => get_option('posts_per_page'),
    'paged' => 1,
);
$query = new WP_Query($args);

if ($query->have_posts()) {
    while ($query->have_posts()) {
        $query->the_post();
        $product_id = get_post_meta(get_the_ID(), '_artwork_product_id', true);
        ?>
        <div class="art-gallery-item">
            <a href="<?php echo get_the_post_thumbnail_url(get_the_ID(), 'large'); ?>"
class="lightbox">
                <?php the_post_thumbnail('large'); ?>
                <h2><?php the_title(); ?></h2>
                <p><?php echo get_post_meta($product_id, '_price', true); ?></p>
            </a>
        </div>
        <?php
        }
        wp_reset_postdata();
    }
    ?>
</div>
<button id="load-more" data-next-page="2"><?php _e('Charger plus', 'textdomain'); ?></button>
<?php
return ob_get_clean();
}
```

Pour ces fonctionnalités, nous utilisons des *Custom Post Types* et des *métadonnées* pour stocker et récupérer les informations sur les œuvres d'art et leurs associations avec des produits WooCommerce.

Qu'est-ce qu'un Custom Post Type ?

Un Custom Post Type (CPT) est une fonctionnalité de WordPress qui permet de créer des types de contenu personnalisés en plus des types de contenu par défaut (articles, pages, etc.). Les CPT sont utilisés pour structurer et organiser des contenus spécifiques qui ne rentrent pas dans les catégories standards.

Custom Post Type (CPT) pour les œuvres d'art :

L'extension crée un type de publication personnalisé appelé "artwork" pour gérer les œuvres d'art. Cela permet de séparer les œuvres d'art des autres types de contenu dans WordPress.

Création du CPT :

```
add_action('init', 'create_art_post_type');
function create_art_post_type()
{
    register_post_type('artwork', array(
        'labels' => array(
            'name' => __('Œuvres d\'art'),
            'singular_name' => __('Œuvre d\'art')
        ),
        'public' => true,
        'has_archive' => true,
        'supports' => array('title', 'editor', 'thumbnail'),
        'show_in_rest' => true,
    ));
}
```

Explications :

`add_action('init', 'create_art_post_type')` : Ajoute l'action `create_art_post_type` à l'initialisation de WordPress.

`register_post_type('artwork', ...)` : Enregistre un nouveau Custom Post Type nommé `artwork` avec les paramètres spécifiés.

`labels` : Définit les labels pour le CPT.

`public` : Rend le CPT public.

`has_archive` : Active les archives pour ce type de contenu.

`supports` : Indique les fonctionnalités supportées par le CPT (titre, éditeur, image à la une).

`show_in_rest` : Rend le CPT disponible via l'API REST de WordPress.

Métadonnées associées :

Qu'est-ce que les métadonnées ?

Les métadonnées sont des informations supplémentaires associées aux types de contenu de WordPress (articles, pages, CPT, etc.).

Elles permettent de stocker des données spécifiques pour chaque élément de contenu.

Les métadonnées permettent de stocker des informations supplémentaires pour chaque œuvre d'art, comme l'ID du produit WooCommerce associé.

Sauvegarde des métadonnées :

```
add_action('save_post', 'save_artwork_product_meta_box');
function save_artwork_product_meta_box($post_id)
{
    if (isset($_POST['artwork_product_id'])) {
        update_post_meta($post_id, '_artwork_product_id',
            sanitize_text_field($_POST['artwork_product_id']));
    }
}
```

La création des tables de l'extension

Voici les scripts SQL pour créer les tables nécessaires :

Table wp_posts :

Cette table existe déjà dans WordPress, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_posts (
    ID bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    post_author bigint(20) UNSIGNED NOT NULL DEFAULT 0,
    post_date datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
    post_date_gmt datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
    post_content longtext NOT NULL,
    post_title text NOT NULL,
    post_excerpt text NOT NULL,
    post_status varchar(20) NOT NULL DEFAULT 'publish',
    comment_status varchar(20) NOT NULL DEFAULT 'open',
    ping_status varchar(20) NOT NULL DEFAULT 'open',
    post_password varchar(255) NOT NULL DEFAULT '',
    post_name varchar(200) NOT NULL DEFAULT '',
    to_ping text NOT NULL,
    pinged text NOT NULL,
    post_modified datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
    post_modified_gmt datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
    post_content_filtered longtext NOT NULL,
    post_parent bigint(20) UNSIGNED NOT NULL DEFAULT 0,
    guid varchar(255) NOT NULL DEFAULT '',
    menu_order int(11) NOT NULL DEFAULT 0,
    post_type varchar(20) NOT NULL DEFAULT 'post',
    post_mime_type varchar(100) NOT NULL DEFAULT '',
    comment_count bigint(20) NOT NULL DEFAULT 0,
    PRIMARY KEY (ID),
    KEY post_name (post_name(191)),
    KEY type_status_date (post_type,post_status,post_date,ID),
    KEY post_parent (post_parent),
    KEY post_author (post_author)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_postmeta :

Cette table existe déjà dans WordPress, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_postmeta (
  meta_id bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  post_id bigint(20) UNSIGNED NOT NULL DEFAULT 0,
  meta_key varchar(255) DEFAULT NULL,
  meta_value longtext,
  PRIMARY KEY (meta_id),
  KEY post_id (post_id),
  KEY meta_key (meta_key(191))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_terms :

Cette table existe déjà dans WordPress, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_terms (
  term_id bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  name varchar(200) NOT NULL DEFAULT "",
  slug varchar(200) NOT NULL DEFAULT "",
  term_group bigint(10) NOT NULL DEFAULT 0,
  PRIMARY KEY (term_id),
  KEY slug (slug(191)),
  KEY name (name(191))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_term_taxonomy :

Cette table existe déjà dans WordPress, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_term_taxonomy (
  term_taxonomy_id bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  term_id bigint(20) UNSIGNED NOT NULL DEFAULT 0,
  taxonomy varchar(32) NOT NULL DEFAULT "",
  description longtext NOT NULL,
  parent bigint(20) UNSIGNED NOT NULL DEFAULT 0,
  count bigint(20) NOT NULL DEFAULT 0,
  PRIMARY KEY (term_taxonomy_id),
  KEY term_id_taxonomy (term_id,taxonomy),
  KEY taxonomy (taxonomy)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_term_relationships :

Cette table existe déjà dans WordPress, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_term_relationships (  
  object_id bigint(20) UNSIGNED NOT NULL DEFAULT 0,  
  term_taxonomy_id bigint(20) UNSIGNED NOT NULL DEFAULT 0,  
  term_order int(11) NOT NULL DEFAULT 0,  
  PRIMARY KEY (object_id,term_taxonomy_id),  
  KEY term_taxonomy_id (term_taxonomy_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_woocommerce_order_items :

Cette table existe déjà dans WooCommerce, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_woocommerce_order_items (  
  order_item_id bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  order_id bigint(20) UNSIGNED NOT NULL,  
  order_item_name text NOT NULL,  
  order_item_type varchar(200) NOT NULL,  
  PRIMARY KEY (order_item_id),  
  KEY order_id (order_id),  
  KEY order_item_name (order_item_name(191))  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table wp_woocommerce_order_itemmeta :

Cette table existe déjà dans WooCommerce, mais voici une version simplifiée pour référence :

```
CREATE TABLE wp_woocommerce_order_itemmeta (  
  meta_id bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  order_item_id bigint(20) UNSIGNED NOT NULL,  
  meta_key varchar(255) DEFAULT NULL,  
  meta_value longtext,  
  PRIMARY KEY (meta_id),  
  KEY order_item_id (order_item_id),  
  KEY meta_key (meta_key(191))  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Modèle Conceptuel de Données (MCD)

Le MCD représente les entités et les relations entre elles. Voici les entités et les relations que nous pouvons identifier à partir du code :

Entités :

1. *Artwork* (Œuvre d'art)

- Attributs : ID, Titre, Contenu, Image à la une, Catégorie, Produit WooCommerce associé

2. *ArtCategory* (Catégorie d'art)

- Attributs : ID, Nom, Parent (si hiérarchique)

3. *WooCommerceProduct* (Produit WooCommerce)

- Attributs : ID, Nom, Description, Image, Prix

Relations :

1. *Artwork* appartient à une *ArtCategory* (relation de type "appartient à")
2. *Artwork* est associé à un *WooCommerceProduct* (relation de type "est associé à")

Description des Relations

1. *Artwork* appartient à *ArtCategory*

- Une œuvre d'art peut appartenir à une ou plusieurs catégories d'art.
- Une catégorie d'art peut contenir plusieurs œuvres d'art.

2. *Artwork* est associé à *WooCommerceProduct*

- Une œuvre d'art peut être associée à un produit WooCommerce.
- Un produit WooCommerce peut être associé à une œuvre d'art.

Modèle Logique de Données (MLD)

Le MLD représente les tables et les colonnes dans une base de données relationnelle. Voici les tables et les colonnes que nous pouvons identifier à partir du code :

Tables :

1. *wp_posts* (Table des publications WordPress)

- ID (Primary Key)
- post_title (Titre)
- post_content (Contenu)
- post_type (Type de publication, par exemple 'artwork')
- post_status (Statut de la publication)
- post_date (Date de la publication)
- post_author (Auteur de la publication)
- post_parent (Parent de la publication)

2. *wp_postmeta* (Table des métadonnées des publications)

- meta_id (Primary Key)
- post_id (Foreign Key vers wp_posts)
- meta_key (Clé de la métadonnée)
- meta_value (Valeur de la métadonnée)

3. *wp_terms* (Table des termes de taxonomie)

- term_id (Primary Key)
- name (Nom du terme)
- slug (Slug du terme)
- term_group (Groupe du terme)

4. *wp_term_taxonomy* (Table des taxonomies)

- term_taxonomy_id (Primary Key)
- term_id (Foreign Key vers wp_terms)
- taxonomy (Nom de la taxonomie, par exemple 'art_category')
- description (Description de la taxonomie)
- parent (Parent de la taxonomie)
- count (Nombre d'éléments dans la taxonomie)

5. *wp_term_relationships* (Table des relations entre les publications et les termes)

- object_id (Foreign Key vers wp_posts)
- term_taxonomy_id (Foreign Key vers wp_term_taxonomy)

6. *wp_woocommerce_order_items* (Table des éléments de commande WooCommerce)

- order_item_id (Primary Key)
- order_id (Foreign Key vers wp_posts)
- order_item_name (Nom de l'élément de commande)
- order_item_type (Type de l'élément de commande)

7. *wp_woocommerce_order_itemmeta* (Table des métadonnées des éléments de commande WooCommerce)

- meta_id (Primary Key)
- order_item_id (Foreign Key vers wp_woocommerce_order_items)
- meta_key (Clé de la métadonnée)
- meta_value (Valeur de la métadonnée)

Relations entre les Tables :

wp_posts et wp_postmeta

- *Relation : Une publication peut avoir plusieurs métadonnées.*
- *Clé étrangère : wp_postmeta.post_id -> wp_posts.ID*

wp_posts et wp_term_relationships

- *Relation : Une publication peut être associée à plusieurs termes de taxonomie.*
- *Clé étrangère : wp_term_relationships.object_id -> wp_posts.ID*

wp_term_relationships et wp_term_taxonomy

- *Relation : Un terme de taxonomie peut être associé à plusieurs publications.*
- *Clé étrangère : wp_term_relationships.term_taxonomy_id -> wp_term_taxonomy.term_taxonomy_id*

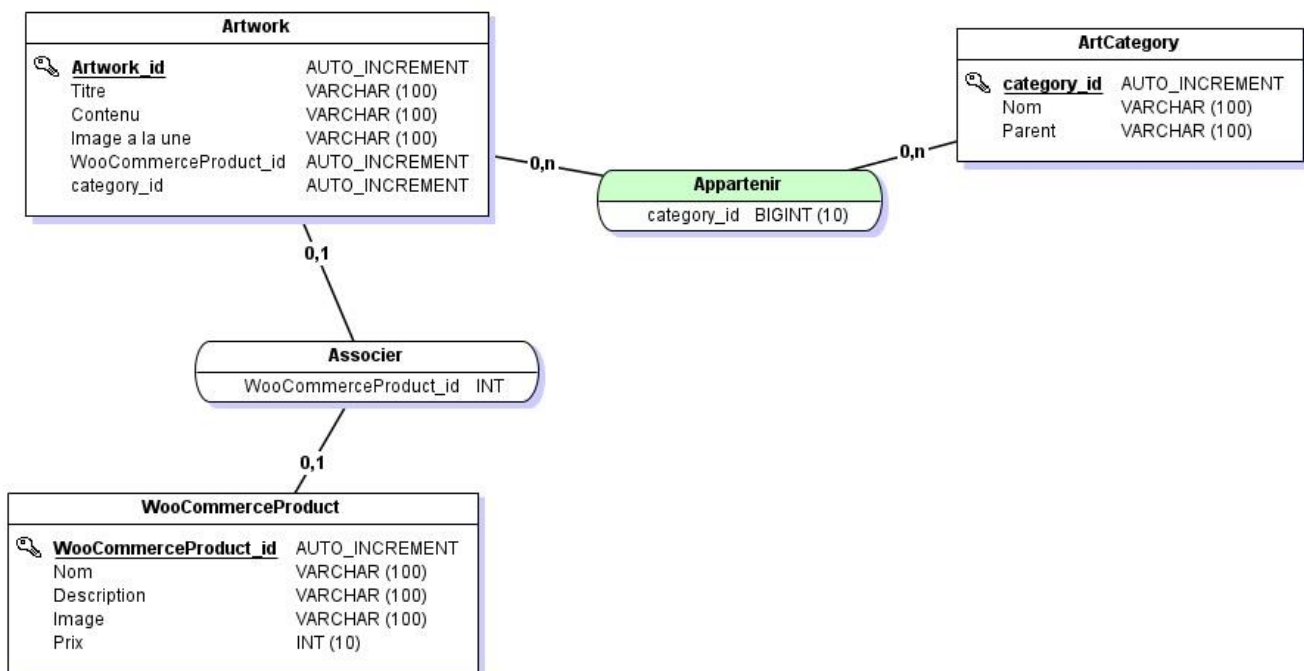
wp_term_taxonomy et wp_terms

- *Relation : Une taxonomie peut avoir plusieurs termes.*
- *Clé étrangère : wp_term_taxonomy.term_id -> wp_terms.term_id*

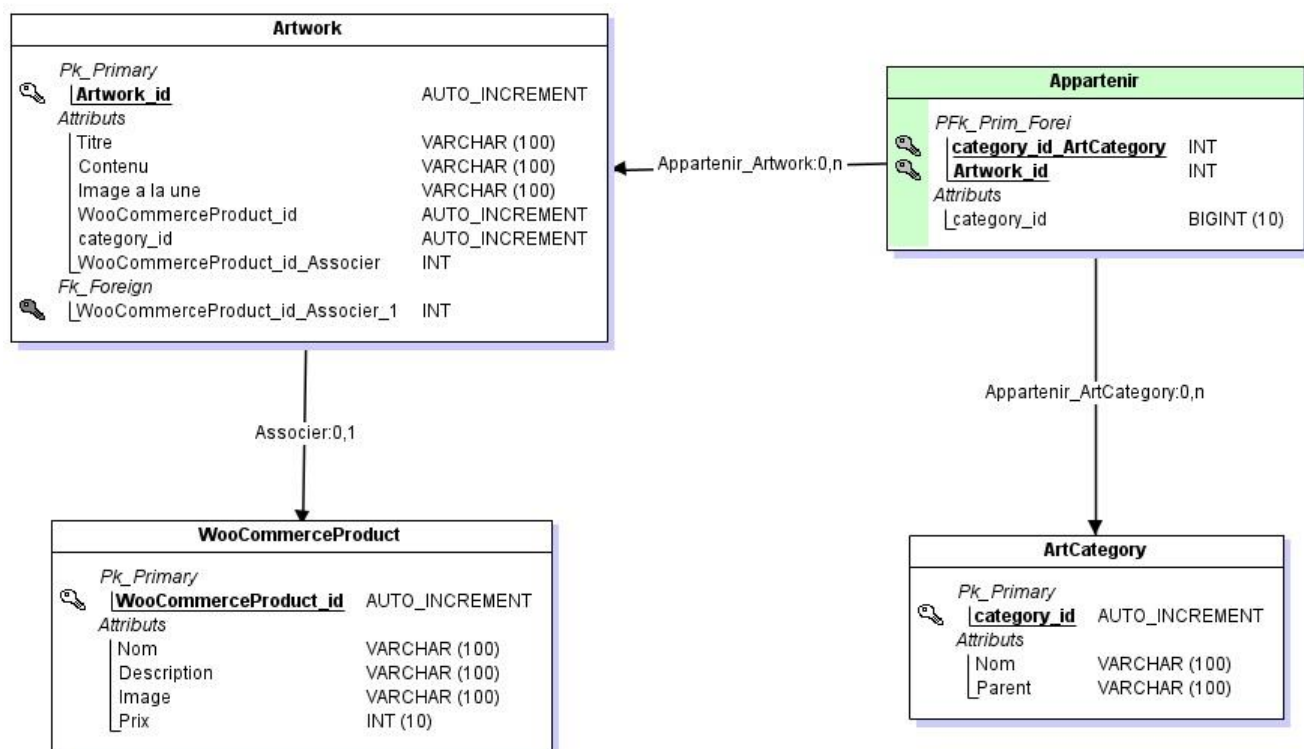
wp_woocommerce_order_items et wp_woocommerce_order_itemmeta

- *Relation : Un élément de commande peut avoir plusieurs métadonnées.*
- *Clé étrangère : wp_woocommerce_order_itemmeta.order_item_id -> wp_woocommerce_order_items.order_item_id*

Diagramme MCD



MCD:



MPD

Voici une représentation du MPD (via JMerise) :

```
#-----
#   Script MySQL.
#-----

#-----
# Table: ArtCategory
#-----

CREATE TABLE ArtCategory(
  category_id Int Auto_increment NOT NULL ,
  Nom      Varchar (100) NOT NULL ,
  Parent   Varchar (100) NOT NULL
  ,CONSTRAINT ArtCategory_PK PRIMARY KEY (category_id)
)ENGINE=InnoDB;

#-----
# Table: WooCommerceProduct
#-----

CREATE TABLE WooCommerceProduct(
  WooCommerceProduct_id Int Auto_increment NOT NULL ,
  Nom      Varchar (100) NOT NULL ,
  Description Varchar (100) NOT NULL ,
  Image     Varchar (100) NOT NULL ,
  Prix      Int NOT NULL
  ,CONSTRAINT WooCommerceProduct_PK PRIMARY KEY (WooCommerceProduct_id)
)ENGINE=InnoDB;

#-----
# Table: Artwork
#-----

CREATE TABLE Artwork(
  Artwork_id      Int Auto_increment NOT NULL ,
  Titre           Varchar (100) NOT NULL ,
  Contenu         Varchar (100) NOT NULL ,
  Image_a_la_une  Varchar (100) NOT NULL ,
  WooCommerceProduct_id      Int Auto_increment NOT NULL ,
  category_id      Int Auto_increment NOT NULL ,
  WooCommerceProduct_id_Associer Int NOT NULL ,
  WooCommerceProduct_id_Associer_1 Int NOT NULL
  ,CONSTRAINT Artwork_PK PRIMARY KEY (Artwork_id)
```

```
,CONSTRAINT Artwork_WooCommerceProduct_FK FOREIGN KEY  
(WooCommerceProduct_id_Associer_1) REFERENCES WooCommerceProduct(WooCommerceProduct_id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Appartenir  
#-----
```

```
CREATE TABLE Appartenir(  
    category_id_ArtCategory Int NOT NULL ,  
    Artwork_id Int NOT NULL ,  
    category_id BigInt NOT NULL  
    ,CONSTRAINT Appartenir_PK PRIMARY KEY (category_id_ArtCategory,Artwork_id)  
    ,CONSTRAINT Appartenir_ArtCategory_FK FOREIGN KEY (category_id_ArtCategory) REFERENCES  
    ArtCategory(category_id)  
    ,CONSTRAINT Appartenir_Artwork0_FK FOREIGN KEY (Artwork_id) REFERENCES Artwork(Artwork_id)  
)ENGINE=InnoDB;
```

B. Développement

i. Architecture

Pour le thème :

Création de modèles de pages :

- *404.php* (fameux erreur 404...)
- *contact.php* : pour le formulaire de contact
- *header.php* et *footer.php* : header et footer personnalisés
- *page-full.php* (sans sidebar)
- *sidebar.php* : affichage des sidebars
- *archive.php* et *category.php* (page d'archives des articles classés par catégories)
- *single.php* (affichage d'un seul post)
- *taxonomy-art-category.php* : afficher le type de contenu personnalisé (archives, catégories) relatif à l'extension plugin-art-gallery (article "artwork")

Options personnalisées développées du thème :

- *Couleurs* (du texte d'entête, d'arrière-plan, du texte)
- *Typographie* : choix des fontes, tailles, Gras, Italique pour : menu, entête, texte global, catégories, footer
- *Art-gallery* (cf extension) : nombre d'image par page, taille des images, effet de transition, couleurs fond et bordure d'image, durée d'animation, affichage flèche navigation, bouton de fermeture, légende et numéros d'images
- *Page contact - Google map* : clé API, latitude, longitude, Zoom
- *Image d'entête*
- *Layout*: taille de la sidebar
- *Footer* : montrer le copyright ou pas, Téléphone, Adresse, URL facebook et Tumblr

Voici des aperçus :

×Publié

Vous personnalisez ?

M 'Z Ile Flo

Thème actif
MZLLeFlo Thème Changer

Identité du site >

Typographie >

Art Gallery >

Couleurs >

Page contact - Google map >

Image d'entête >

Layout >

Image d'arrière-plan >

Menus >

Widgets >

Réglages de la page d'accueil >

Footer >

WooCommerce >

CSS additionnel >

⏮ Masquer les controles 🖥 📱 📱

Typographie :

Personnalisation

Typographie

Police pour le menu

Playwrite NO

Taille de la police pour le menu

14

☐ Gras

☐ Italique

Police pour l'entête

Great Vibes

Taille de la police pour l'entête

50

☐ Gras

☐ Italique

Police pour le texte global

Playwrite NO

Taille de la police pour le texte global

19


☐ Gras

☐ Italique

Police pour les catégories


Masquer les controles

Extension Art Gallery :


 **Personnalisation**
Art Gallery

Nombre d'images par page


Taille des images

Moyenne 


Effet de transition

Fade 

Couleur de fond de la lightbox

 Sélectionner une couleur

Couleur de la bordure des images

 Sélectionner une couleur

☐ Afficher les légendes

☐ Afficher les numéros d'image

Durée de l'animation (ms)

☐ Afficher les flèches de navigation

☐ Afficher le bouton de fermeture

Page contact :

<

Personnalisation

Page contact - Google map

☐ Afficher la Google map ?
(page contact - juste avant le footer)

votre clé API
<https://developers.google.com/maps/documentation/embed/map>

Latitude

Longitude

Zoom

Sidebar :

<

Personnalisation

Layout

Largeur de la sidebar
20% à 40%

Footer :

[←](#) [Personnalisation](#)
[Footer](#)

Souhaitez-vous montrer le copyright ?
☒ Oui
☐ Non

Téléphone

Adresse

URL page Facebook

URL page Tumblr

Contact (codes court):

Contact

[/] **Code court**

```
[advanced_contact_form]
```

[/] **Code court**

```
[google_maps address="25 Grande Rue 44530 Drefféac FR" zoom="15" width="100%" height="400px"]
```

Gallerie d'art :

M'Z'le Flo

Créer Voir les articles Vider le cache UpdraftPlus

Tableau de bord

Articles

Médias

Pages

Commentaires

Contact Submissions

Galerie d'art

Galerie d'art

Ajouter un article

Art Categories

WooCommerce

Produits

Paielements

Statistiques

Marketing

Apparence

Extensions

Comptes

Outils

Réglaes

WP Fastest Cache

Réduire le menu

Galerie d'art

[Ajouter un article](#)

Tous (41) | Publiés (41) | Corbeille (6)

Actions groupées Appliquer Toutes les dates Filtrer

<input type="checkbox"/> Titre	Art Categories
<input type="checkbox"/> Plastron polymère bleu-blanc bronze	Bijoux, Collier
<input type="checkbox"/> Pendentif polymère encre noir et rose	Bijoux, Collier
<input type="checkbox"/> Pendentif encre noir et rose	Bijoux, Collier
<input type="checkbox"/> Horloge sur toile	Bijoux, Collier
<input type="checkbox"/> Collier rock hidden magic bleu noir	Bijoux, Collier
<input type="checkbox"/> Collier polymère relief chocolat feuille dorée	Bijoux, Collier
<input type="checkbox"/> Collier polymere hidden magic bleu vert blanc argente	Bijoux, Collier
<input type="checkbox"/> Collier polymere hidden magic bleu vert blanc argente hematite	Bijoux, Collier
<input type="checkbox"/> Collier polymere hidden magic bleu blanc noir argente	Bijoux, Collier
<input type="checkbox"/> Collier polymère encre rose et noir toupie argente	Bijoux, Collier
<input type="checkbox"/> Collier polymère encre rose et noir argente	Bijoux, Collier

Et ses widgets :

Widgets

Gérer avec la prévisualisation en direct

Widgets disponibles

Pour activer un widget, glissez-le dans la colonne latérale ou cliquez dessus. Pour désactiver un widget et supprimer ses réglages, enlevez-le de la colonne latérale.

Archives

Une archive mensuelle des articles de votre site.

Archives des Créations

Affiche les archives des créations

☒ Barre latérale principale

Zone de dépôt en pied de page

Barre latérale Contact

Annuler

Ajouter le widget

Widgets

Gérer avec la prévisualisation en direct

Widgets disponibles

Pour activer un widget, glissez-le dans la colonne latérale ou cliquez dessus. Pour désactiver un widget et supprimer ses réglages, enlevez-le de la colonne latérale.

Archives

Une archive mensuelle des articles de votre site.

Archives des Créations

Affiche les archives des créations

Articles récents

Les articles les plus récents de votre site.

Avis récents sur les produits

Affiche une liste des avis récents publiés sur votre boutique.

Bloc

Un widget contenant un bloc.

Calendrier

Un calendrier des articles de votre site.

Catégories

Une liste ou un menu déroulant des catégories.

Catégories d'Art

Affiche les catégories d'art

☒ Barre latérale principale

Zone de dépôt en pied de page

Barre latérale Contact

Annuler

Ajouter le widget

Les autres extensions : Détails

Plugin-social-media :

- création d'une application Facebook (via Facebook for Developers : Créer une application)

Description : Un plugin pour automatiser la publication des nouveaux produits sur les réseaux sociaux.

Fonctionnalités :

Connexion aux comptes de réseaux sociaux (Facebook, Instagram, Twitter).

Publication automatique des nouveaux produits avec images et descriptions.

Interface pour programmer des publications.

Annulé : Facebook a restreint l'accès à certaines permissions, notamment celles permettant de publier directement sur le mur personnel d'un utilisateur. Sauf à faire une application commerciale ou de jeux, dont d'utiliser la version payante de l'API (for business...)...

L'application est créée mais n'a pas les droits nécessaires... désactivation de l'extension.

Page de Contact Avancée

Description : Un plugin pour créer une page de contact avec des fonctionnalités avancées.

Fonctionnalités :

Formulaire de contact avec options personnalisables (champs, validations, etc.).

Intégration avec **Google Maps** pour afficher l'atelier ou la galerie.

Notifications par e-mail pour chaque soumission de formulaire.

Création du modèle de page "contact.php", ajout de short code : `[advanced_contact_form]` pour le formulaire et `[google_maps address="123 Rue de l'Exemple, Ville, Pays" zoom="15" width="100%" height="400px"]` pour la carte.

Création d'une page de remerciement qui apparaît après la soumission du formulaire.

Widgets : Après les extensions, les widgets ...

- *Categories d'art : affichage des créations, par catégories*
- *Archives des Créations : affichages des dernières Créations*

Pour les deux widgets, voir la partie extension "Plugin-art-gallery" pour plus d'informations.

ii. Exemples de code

Voir annexe séparée.

VI. Test

i. Tests du front-end (l'interface graphique)

Les tests du front-end concernent la vérification de l'interface utilisateur. Il faut s'assurer que l'utilisateur peut interagir correctement avec l'application et que tout fonctionne comme prévu. Les tests peuvent être manuels ou automatisés.

N°	Test	Technique de test
1	Vérification navigation mobile	Test manuel
2	Vérification navigation tablette	Test manuel
3	Vérification navigation desktop	Test manuel
4	Vérifications des extensions (jout artwork)	Test manuel
5	Vérification formulaire de contact	Test manuel
6	Vérification achat (commande, panier, etc)	Test manuel
7	Vérifications widgets (insertion, utilisation)	Test manuel

ii. Tests du back-end (persistance des données)

1	Ajout d'un nouvel utilisateur	Test d'intégration
2	Modification d'un profil utilisateur	Test de validation de données
3	Modifications commandes	Test de validation de données

B. Déploiement

Plan de déploiement (acteur, planning, méthodologie, ...)

Fait :

Vérification du site local :

On s'assure que le site fonctionne correctement en local. Cela inclut la vérification de la compatibilité des plugins, des thèmes, et de la configuration générale du site.

Sauvegarde du site :

Le site est actuellement sauvegardé :

- *En local (disque externe)*
- *Sur le cloud*
- *sur le serveur du CEFJJ*

Utilisation d'un plugin à cet effet (dans WordPress : UdraftPlus)

Prévu (fin septembre, avec la cliente) :

Choix de l'hébergeur :

En cours par la cliente.

Pour WordPress, il existe des hébergeurs spécialisés comme SiteGround, Bluehost, ou OVH! faut s'assurer que l'hébergeur supporte PHP et MySQL, qui sont nécessaires pour WordPress. Idem pour le HTTPS...

Transfert de fichier :

Utilisation de FileZilla à cet effet

Modification du fichier wp-config.php pour l'adapter aux informations de connexion à la nouvelle base de données de l'hébergeur.

Importation de la base de données :

Via PhpMyAdmin

Configuration du domaine :

Enregistrement DNS et SSL

VII. Veille technique

Dans le cadre de ce projet, j'ai mené une veille technique active sur les vulnérabilités de sécurité liées à l'utilisation de WordPress et au développement d'interfaces utilisateurs web dynamiques. Cette veille a été essentielle pour assurer la sécurité et la robustesse de l'application développée..

Méthodologie de la veille

Pour mener cette veille, j'ai régulièrement consulté plusieurs sources fiables, telles que la [CVE (Common Vulnerabilities and Exposures)](<https://cve.mitre.org/>), le [NVD (National Vulnerability Database)](<https://nvd.nist.gov/>), ainsi que des blogs spécialisés comme le [WordPress Security Blog](<https://wordpress.org/news/category/security/>). J'ai également utilisé des outils de veille comme *Google Alerts* avec des mots-clés tels que « *WordPress security vulnerabilities* », « *XSS WordPress* », et « *SQL injection WordPress* » pour être informé des dernières menaces.

VIII. Recherche technique sur des sites anglophones

Recherche technique sur des sites anglophones :

Dans le cadre de mon projet, j'ai été confronté à plusieurs défis techniques nécessitant des recherches approfondies sur des sites anglophones. L'un des problèmes majeurs concernait une erreur de template et de mise en page dans un fichier `template-canvas.php`. Cette situation a directement impacté la présentation visuelle du site, ce qui est crucial pour une interface utilisateur.

Description de la situation

Lors de la personnalisation d'un thème WordPress, j'ai rencontré une erreur de mise en page

liée au fichier `template-canvas.php`. Cette erreur perturbait l'affichage du contenu, rendant le site difficilement utilisable. Étant donné que le fichier jouait un rôle central dans la structure de certaines pages, il était impératif de corriger ce problème rapidement.

Recherche effectuée sur des sites anglophones

Pour résoudre ce problème, j'ai effectué une recherche en anglais en utilisant des mots-clés spécifiques tels que "WordPress template-canvas.php layout issue". Les principaux sites que j'ai consultés incluent :

- ***[Stack Overflow](https://stackoverflow.com/)*** : où j'ai trouvé plusieurs discussions sur des problèmes similaires.
- ***[WordPress Codex](https://codex.wordpress.org/)*** : qui m'a fourni des informations sur la structure des fichiers de template.
- ***[WordPress Stack Exchange](https://wordpress.stackexchange.com/)*** : où j'ai découvert des solutions proposées par d'autres développeurs ayant rencontré des problèmes analogues.

Résolution du problème

Grâce aux informations recueillies, j'ai pu identifier que l'erreur provenait d'une incompatibilité entre le fichier `template-canvas.php` et certains éléments du thème. J'ai modifié le fichier selon les suggestions trouvées, notamment en ajustant les appels de fonctions et en réorganisant les sections de code pour mieux correspondre aux standards du thème utilisé.

Impact sur le projet

La résolution de ces problèmes a permis de restaurer l'intégrité visuelle du site, assurant une meilleure expérience utilisateur. De plus, en corrigeant les erreurs liées aux scripts et aux styles, j'ai amélioré les performances du site et évité des conflits futurs lors des mises à jour de WordPress ou des plugins.

Sans parler du temps de recherche, de test des solutions...

Conclusion et lien avec la compétence

Cette recherche technique a été essentielle pour garantir la qualité de l'interface utilisateur développée. Elle m'a permis d'améliorer ma compétence en « Réalisation d'une interface utilisateur avec une solution de gestion de contenu », en me familiarisant davantage avec les meilleures pratiques de WordPress pour gérer les templates et les scripts. La capacité à rechercher et appliquer des solutions techniques en anglais est cruciale dans le développement web, où une grande partie des ressources et de la documentation est disponible en langue anglaise.

De plus, j'ai rencontré une autre erreur concernant l'utilisation incorrecte de la fonction `wp_enqueue_script()`, ce qui entraînait l'affichage de messages d'alerte. En recherchant "Notice: la fonction `wp_enqueue_script()` a été appelée de façon incorrecte" sur Google, j'ai trouvé une explication détaillée sur ***[WPBeginner](https://www.wpbeginner.com/)*** qui m'a aidé à corriger l'appel de fonction.

D'autres recherches ont été effectuées, par exemple :

- "notice : la fonction `wp_enqueue_script()` a été appelée de façon incorrecte.
- le script « wp-editor » ne d' pour une erreur sur un plugin,
- "woocommerce conditional shipping and payments" pour savoir comment effectuer des types de paiements suivants des conditions,
- ou encore l'erreur "`wp_enqueue_style`"...

IX. Recherche technique : extrait du site anglophone

Recherche technique : extrait du site anglophone

Contexte et besoin d'information

Lors de la personnalisation du thème WordPress utilisé pour mon projet, j'ai rencontré une erreur de mise en page liée au fichier `template-canvas.php`. Pour corriger ce problème, j'avais besoin d'une information précise sur la manière dont ce fichier interagissait avec le reste du thème et les autres éléments du site.

Processus de recherche

Pour résoudre ce problème, j'ai utilisé les mots-clés suivants : "WordPress template-canvas.php layout issue". Après avoir consulté plusieurs sites, j'ai sélectionné un article sur **Stack Overflow** qui offrait une explication détaillée des erreurs courantes liées à ce fichier. J'ai choisi cet article en raison de la pertinence des réponses et de l'expérience des contributeurs.

Extrait du site anglophone

"The issue with `template-canvas.php` usually arises when the theme's structure doesn't match the WordPress core functions it relies on. To fix this, check if the template is properly enqueueing the necessary scripts and styles, and ensure that the canvas div has the appropriate classes that match the theme's CSS. If the issue persists, consider overriding the template in a child theme to avoid conflicts with future theme updates."

(Source : Stack Overflow)

Traduction en français

"Le problème avec `template-canvas.php` survient généralement lorsque la structure du thème ne correspond pas aux fonctions principales de WordPress sur lesquelles il repose. Pour résoudre ce problème, vérifiez si le template appelle correctement les scripts et les styles nécessaires, et assurez-vous que la div `canvas` possède les classes appropriées correspondant au CSS du thème."

Si le problème persiste, envisagez de remplacer le template dans un thème enfant pour éviter les conflits lors des futures mises à jour du thème."

Application de la solution

Grâce à cet extrait, j'ai pu identifier que l'erreur provenait de l'appel incorrect de certains scripts et styles. J'ai modifié le fichier `template-canvas.php` en suivant les conseils donnés, ce qui a résolu le problème de mise en page. J'ai également créé un thème enfant pour appliquer ces modifications, garantissant ainsi que les futures mises à jour du thème principal n'affecteraient pas ma correction.

Pour schématiser :

Pour effectuer mes recherches, j'adopte en grande partie le processus de recherche suivant :

- 1. Requête en anglais, plutôt qu'en français, sur google avec des termes simples*
- 2. Sélection des sources par rapport au classement personnel, ainsi que la date de création / mise à jour de la source*

3. Analyse critique des informations contenues dans les pages sélectionnées

En règle général je sélectionne les ressources dans l'ordre suivant:

- 1. La documentation de la technologie*
- 2. Un acteur ou un groupe d'acteurs connus et identifiés dans le domaine de la recherche, par exemple la fondation OWASP dans le cadre d'une recherche sur la sécurité*
- 3. Les sites ayant attiré à la programmation, exemple: medium, blog LogRocket, etc.*
- 4. Éventuellement des questions sur Stackoverflow,*

x. Conclusion

Bilan technique et humain

En conclusion, j'aimerais remercier :

- *La cliente, pour ses idées, sa patience et son écoute*
- *Mes formateurs, Olivier Pesce, Nolwenn Gouzien pour la aussi la patience, la pédagogie et les conseils et le professionnalisme*
- *L'équipe dirigeante du CEFJJ, Charly Patrault et Amandine Leroux, pour avoir cru en moi et la aussi la patience et l'écoute*

Bien que ce projet me posait des défis techniques, j'ai réussi à cumuler les recherches sur le fonctionnement de certaines technologies, avec le développement du projet.

J'appréhendais cette période de mise en place d'un projet (hors stage malheureusement) mais des entretiens réguliers avec la cliente (mails, visios, déplacements) quasiment chaque semaine (voir moins pour les mails) ont permis de structurer celui-ci, de tenir des délais (longs, profitons-en...) et donc d'avoir un cap.

Il reste quelques bugs (mineurs, pour la grande majorité d'entre eux) et éventuellement quelques ajustements esthétiques concernant la mise en page, suivant les demandes à venir de la cliente mais le délai restant permet d'y faire face et de trouver les solutions adéquates.

La cliente est satisfaite du résultat et c'est ce qui importe le plus.

xi. Annexes

Voir dossiers séparé "Annexe".