

*Présentation du projet
« M'Zlle Flo »*



Michel Grillon 07-08/2024 - Formation "Développeur Web et Web Mobile" - CEFJJ Angers

Sommaire

1. *Présentation*
2. *Présentation du projet*
3. *Contexte*
4. *Résumé du projet*
5. *Compétences couvertes par le projet*
6. *Cahier des charges*
7. *Sécurité*
8. *Création de la maquette*
9. *Extensions et widgets*
10. *Structure de données , base de données : MCD, MLD et MPD*
11. *Développement*
12. *Déploiement*
13. *Veille technique*
14. *Conclusion*

Présentation

Michel Grillon, 55 ans,

Formations académiques :

CAP-BEP Commerce-vente / CAP monteur-raccordeur télécom / CAP métallier-serrurier

Expériences professionnelles :

Divers métiers puis 17 ans dans l'informatique (technicien de support puis superviseur), reconversion professionnelle en tant que soudeur puis retour vers l'informatique en raison de problèmes de santé.

Motivations pour cette formation :

j'ai toujours été attiré par l'informatique, l'infographie, la création de sites internet et la programmation, mais je l'ai fait en autodidacte. J'ai voulu aller plus loin dans l'apprentissage du code afin d'en devenir professionnel.

Présentation du projet

Une artiste (création de bijoux, de peintures) souhaite 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.

Elle souhaite ajouter la location de certaines de ses œuvres, au mois, en plus de la vente.

Ceci pour 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 (en même temps que son statut).

Contexte

J'ai réalisé ce projet seul, en télétravail, en parallèle de mon activité professionnelle, le soir, le week-end et pendant mes congés.

La cliente et moi communiquions par email, téléphone, visio-conférence et nous avons régulièrement des entretiens en présentiel dans son atelier.

Je présentais le travail du site sur un serveur local (créer pour l'occasion avec WampServer) via un portable lors des déplacements, et un pc de bureau pour le reste du temps.

Par la suite, après la mise en place sur le serveur du CEFJJ, je passais de l'un à l'autre (tests en local et si ok envois sur serveur distant).

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

L'administration du site :

https://cefii-developpements.fr/michel1385/Projet_final/mzlleflo/wp-admin/index.php

Contexte

Ce projet n'était pas le projet initial ...

Le projet initial consistait en la création d'un site vitrine pour ADPG49, un atelier spécialisé dans la transformation et la réparation de véhicules de loisir.

Le site devait inclure plusieurs fonctionnalités : présentation des transformations des véhicules, newsletter, formulaires de demande de rendez-vous et de devis, ainsi qu'un intranet pour la gestion interne (fournisseurs, outils, consommables, suivi des véhicules).

Le projet devait être développé en PHP orienté objet avec une architecture MVC, Bootstrap, et du JavaScript.

Cependant, le client cesse son activité fin 2024, ce qui a conduit à l'arrêt du projet.

Après deux semaines de recherches, un nouveau projet a été trouvé (celui présenté), mais avec un délai plus court pour la présentation à l'examen.

Compétences couvertes par le projet

Synthèse des points clés concernant le développement de la partie frontend d'une application web ou web mobile, en intégrant les recommandations de sécurité :

Maquettage de l'application :

- *Collaboration étroite avec la cliente*
- *Conception de l'ergonomie et de la navigation*
- *Élaboration des user-stories*
- *Création de documents de conception*

Réalisation d'une interface utilisateur web statique et adaptable :

- *Approche mobile-first (utilisation de HTML, CSS et templates PHP) = adaptation à différents formats*

Objectifs principaux :

- *Interface simple et facile d'utilisation*
- *Accessibilité pour une clientèle multigénérationnelle*
- *Adaptation à divers niveaux de compétence informatique*
- *Flexibilité d'utilisation sur différents appareils*

= créer une application web accessible et conviviale, répondant aux besoins d'une clientèle diversifiée tout en assurant une expérience utilisateur optimale sur l'ensemble des plateformes.

Compétences couvertes par le projet

Synthèse des points clés concernant le développement de la partie frontend d'une application web ou web mobile, en intégrant les recommandations de sécurité :

Développement d'une interface utilisateur web dynamique et la réalisation d'une interface avec une solution de gestion de contenu :

- *Choix de WordPress comme plateforme de base*
- *Création d'un thème personnalisé et dédié et développement d'extensions et plugins spécifiques*

Fonctionnalités spécifiques ajoutées :

- *Champ de recherche dynamique avec icône dépliant et Bouton de retour en haut de page (JavaScript)*
- *Formulaire de contact intégré avec Intégration de Google Maps*

Réalisation d'une interface avec solution de gestion de contenu et e-commerce :

Combinaison de WordPress et WooCommerce : personnalisation des 2 plateformes pour répondre aux besoins spécifiques

Objectifs de la personnalisation :

- *Se démarquer de la concurrence*
- *Éviter l'utilisation de thèmes préexistants*
- *Apporter une valeur ajoutée aux utilisateurs*

= créer une expérience utilisateur unique et fonctionnelle, en combinant les avantages d'un CMS populaire avec des développements sur mesure. Cela permet d'obtenir une solution flexible et personnalisée pour la gestion de contenu et le commerce électronique.

Compétences couvertes par le projet

Synthèse des points clés concernant le développement de la partie backend d'une application web ou web mobile, en intégrant les recommandations de sécurité :

Développement de la partie backend :

- *Intégration avec le système de WordPress*
- *Personnalisation de WP et WooCommerce pour répondre aux besoins spécifiques*

Création de la base de données :

- *Utilisation de la structure de base de WordPress, manipulations des tables et données*
- *WP gère automatiquement les tables principales (articles, utilisateurs, produits, etc.)*

Développement des composants d'accès aux données :

- *Focus sur les extensions personnalisées, notamment "Art-Gallery"*
- *Documentation des tables manipulées par les extensions*

Aspects de sécurité :

- *Utilisation des mécanismes de sécurité intégrés à WP et généraux*
- *Attention particulière lors du développement d'extensions personnalisées*

Compétences couvertes par le projet

Synthèse des points clés concernant le développement de la partie backend d'une application web ou web mobile, en intégrant les recommandations de sécurité :

Élaboration et mise en œuvre des composants dans une application de gestion de contenu ou e-commerce :

- *Utilisation combinée de WordPress (CMS) et WooCommerce (e-commerce)*
- *Développement d'extensions sur mesure pour étendre les fonctionnalités*

Documentation :

- *Accent mis sur la documentation des tables et fonctionnalités ajoutées ou modifiées*
- *Documentation détaillée de l'extension "Art-Gallery" en tant qu'exemple principal*
- *Permet de bénéficier de la robustesse et de la sécurité de WordPress tout en développant des fonctionnalités sur mesure.*
- *Met l'accent sur l'extension des capacités du CMS plutôt que sur la création d'une structure de base de données entièrement nouvelle,*
- *Offrant ainsi un bon équilibre entre personnalisation et utilisation des fonctionnalités existantes.*

Cahier des charges

Voici une synthèse des exigences et des contraintes pour le développement du site web :

Fonctionnalités principales :

- *Gestion des commandes et clients / des stocks et des produits en vente / des articles/news et d'une galerie*
- *Formulaire de contact*
- *Pages légales*

Cible clientèle :

Principalement des particuliers, potentiel pour les entreprises (ex : salles de réunions, halls d'accueil)

Contraintes de développement :

- *Création d'un thème entièrement original (pas de thème existant)*
- *Amélioration du back-office WordPress*

Cahier des charges

Voici une synthèse des exigences et des contraintes pour le développement du site web :

Limitation des extensions supplémentaires :

- *Privilégier les extensions gratuites*
- *Exceptions pour la sécurité et certains cas spécifiques*
- *Utilisation autorisée d'extensions WooCommerce pour les paiements*

Propositions supplémentaires :
Création d'extensions et plugins personnalisés pour :

- *Faciliter certaines tâches pour la cliente*
- *Répondre aux attentes de l'examen*
- *Viser à créer une solution e-commerce personnalisée et adaptée, en minimisant la dépendance aux extensions tierces.*
- *L'accent est mis sur le développement sur mesure pour répondre aux besoins spécifiques de la cliente et démontrer mes compétences.*

Cahier des charges

Synthèse des éléments de design pour le site web de l'artiste :

Concept général :

- *Design adapté aux mobiles (responsive)*
- *Style simple et intuitif*
- *Reflète des activités de l'artiste (taches de peintures, esquisses, etc.)*

Typographie (polices Google) :

- *Titres : Great Vibes*
- *Catégories/menu : Send Flowers*
- *Textes : Playwrite Norge*

Couleurs : Palette de couleurs déterminée par la cliente pour :

- *Texte d'en-tête*
- *Arrière-plan*
- *Texte principal*

Cahier des charges

Synthèse des éléments de design pour le site web de l'artiste :

Éléments visuels :

Logo créé spécifiquement pour la cliente

Images fournies par la cliente :

- *Fonds de pages*
- *Illustrations pour articles/créations*
- *Photos de produits*

Viser à créer une identité visuelle cohérente et personnalisée, reflétant l'univers artistique de la cliente et son image de marque.

Le mélange d'éléments fournis par la cliente (couleurs, images) et créés par le développeur (logo, mise en page) permet d'obtenir un design unique et parfaitement adapté aux besoins et à l'esthétique souhaités.

Cahier des charges

Contenu du site :

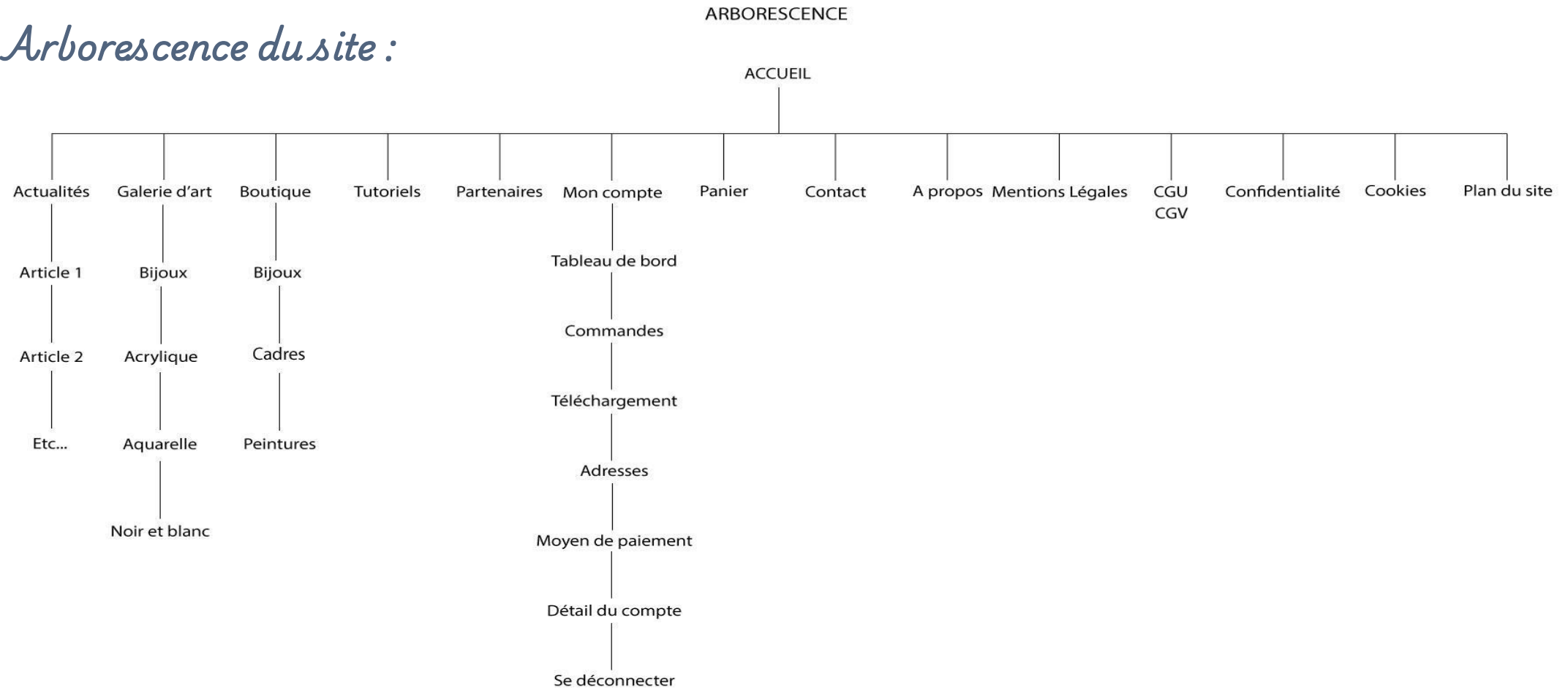
- *Accueil,*
- *Galerie d'art*
- *Boutique,*
- *Contact,*
- *Actualités/News,*
- *Partenaires,*
- *Tutoriels,*
- *À propos,*
- *Plan du site,*
- *Mon compte et Mon panier.*

Ainsi que les pages légales et obligatoires :

- *Conditions générales de vente et d'utilisation (CGV-CGU dans la même page),*
- *Mentions légales,*
- *Confidentialité,*
- *Cookies*

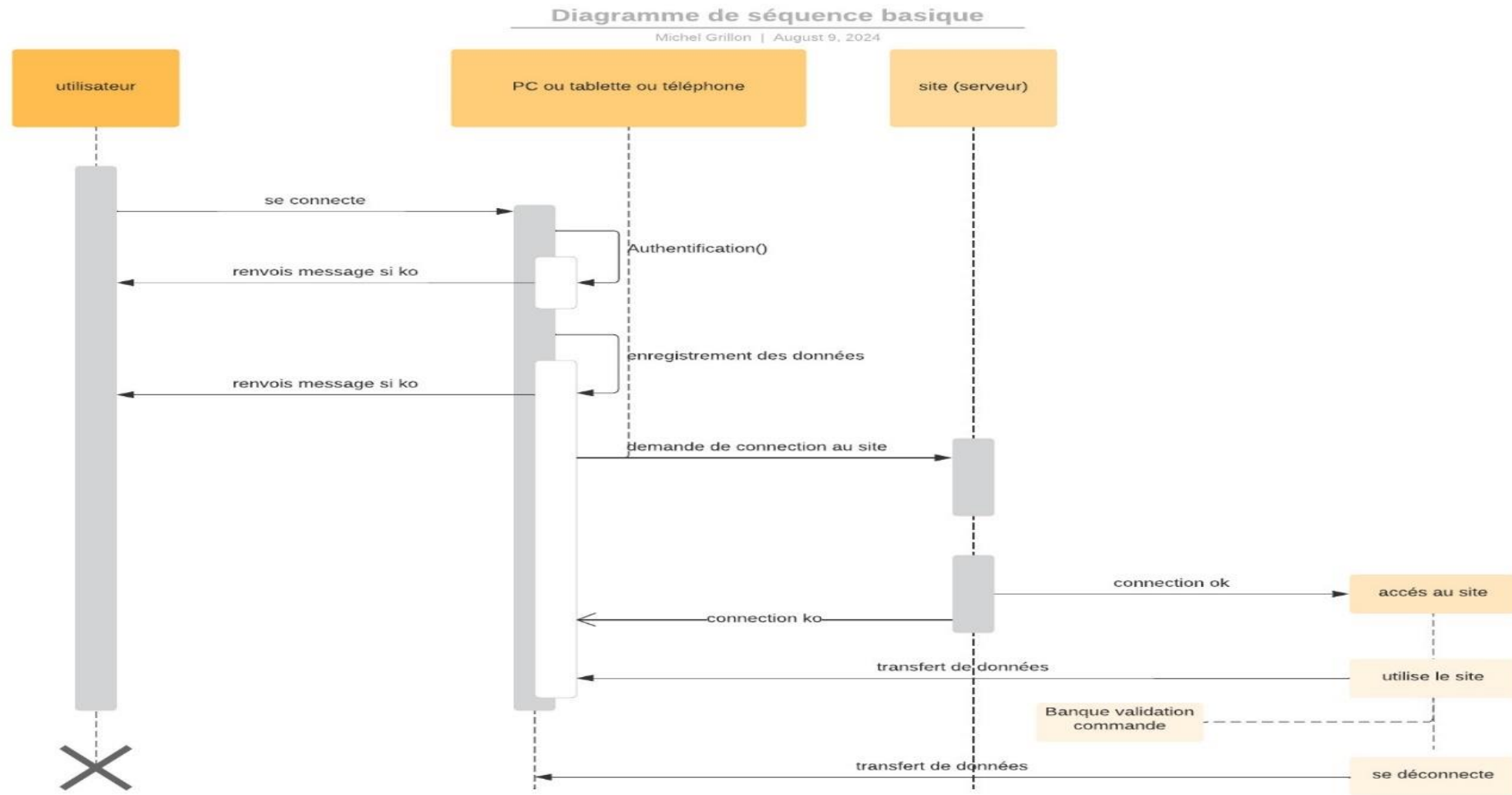
Cahier des charges

Arborescence du site :



Cahier des charges

UML



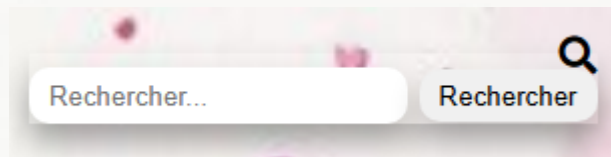
Cahier des charges

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/personnalisable,*
- ✓ *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. JavaScript a été utilisé à ces effets.



Cahier des charges

La création d'extensions et widgets prévus ...

Extensions :

- Pour l'intégration des réseaux sociaux (Facebook, Tumblr, actuellement en stand-by),*
- 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.

Sécurité

La sécurité sera renforcée par :

- *l'utilisation des fonctions d'API WordPress (comme `wp_insert_post`, `wp_update_post`, etc.), avec la validation et sécurisation des données (fonctions de validation et de nettoyage des données comme `sanitize_text_field`, `esc_html`, `esc_url`, etc., voir slide suivant),*
- *mot de passe robuste,*
- *modification du préfixe "wp" des tables (prévention des attaques SQL.) (via `wp-config.php`, avant installations de WP),*
- *suppression des plugins et thèmes inutilisés,*
- *mise à jours des versions WP et des extensions,*
- *sauvegarde de la base de données et des contenus,*
- *protection du fichier de configuration de WordPress et `.htaccess`, ainsi que le masquage des dossiers sensibles (`.htaccess`)(en cours avant mise en ligne définitive, voir note)*
- *limitation de tentatives de connexion (3 tentatives, via `functions.php`) pendant 15mn*
- *utilisation d'HTTPS (en attente de l'hébergeur)*

Sécurité

Sécurité bis : pas dans les dossiers remis... car fait après l'impression de ceux-ci.

- *Correction XSS (Cross-Site Scripting)*

*WordPress utilise des fonctions intégrées pour échapper les données et prévenir les attaques XSS.
(ex : functions.php ligne 83 ou taxonomy-art_category.php ligne 14)*

Objectif : Éviter l'injection de scripts malveillants dans les pages web.

Fonctions utilisées :

- *esc_html() : Échappe les caractères HTML spéciaux dans les données avant affichage.*
- *esc_attr() : Échappe les caractères spéciaux dans les attributs HTML.*
- *esc_url() : Échappe les URL pour éviter les injections.*
- *esc_js() : Échappe les données utilisées dans des contextes JavaScript.*

Exemple

```
$user_input = $_GET['name'];  
echo esc_html($user_input);
```

Protection contre les attaques CSRF (Cross-Site Request Forgery)

- *Objectif : Prévenir les actions non autorisées via des requêtes falsifiées.*
- *Fonctions utilisées :*
 - *wp_nonce_field() : Ajoute un champ caché avec un nonce dans les formulaires pour vérifier les requêtes.*
 - *wp_verify_nonce() : Vérifie le nonce pour authentifier les actions.*
 - *check_admin_referer() : Vérifie le nonce pour les actions administratives.*

- *Exemple :*

```
<?php $nonce = wp_create_nonce('my-nonce-action'); ?>
```

```
<form method="post" action="<?php echo esc_url(admin_url('admin-post.php')); ?>">
```

```
<?php wp_nonce_field('my-nonce-action'); ?>
```

```
<!-- Champs du formulaire -->
```

```
</form>
```

• *(ex. functions.php ligne 81 et Ajout d'une fonction pour vérifier le nonce lors de la sauvegarde des données du customizer. Ligne 20)*

Correction contre l'injection SQL

- *Objectif*: Prévenir l'exécution de requêtes SQL malveillantes.
- *Méthode utilisée*: `$wpdb->prepare()`: Prépare les requêtes SQL avec des paramètres sécurisés pour éviter les injections.
- *Exemple*:

```
global $wpdb;  
$user_id = $_GET['id'];  
$post = $wpdb->get_row($wpdb->prepare(  
    "SELECT * FROM $wpdb->posts WHERE ID = %d",  
    $user_id  
));
```

Sécurisation des uploads de fichiers

- *Objectif*: Vérifier et sécuriser les fichiers téléchargés pour éviter les risques.
- *Fonctions utilisées*: `wp_handle_upload()`: Gère le téléchargement et le stockage des fichiers.
 - `wp_check_filetype()`: Vérifie le type de fichier pour assurer sa sécurité.

Protection contre les failles d'inclusion (include())

- *Objectif: Éviter l'inclusion de fichiers non sécurisés ou malveillants.*
- *Pratiques recommandées :*
 - *Validation et échappement des chemins de fichiers.*
 - *Utilisation de `sanitize_file_name()` pour nettoyer les noms de fichiers avant de les inclure*

Résumé

- *XSS: Utilisation des fonctions d'échappement de WordPress (`esc_html()`, `esc_attr()`, etc.).*
- *CSRF: Utilisation des jetons CSRF (`wp_nonce_field()`, `wp_verify_nonce()`, etc.).*
- *Injection SQL: Utilisation de `$wpdb->prepare()` pour les requêtes SQL.*
- *Upload: Utilisation de `wp_handle_upload()` et `wp_check_filetype()` pour gérer les téléchargements.*
- *Inclusion: Validation et échappement des chemins de fichiers avant de les inclure..*

Création de la maquette

J'ai réalisé la maquette à l'aide d'Adobe Illustrator.

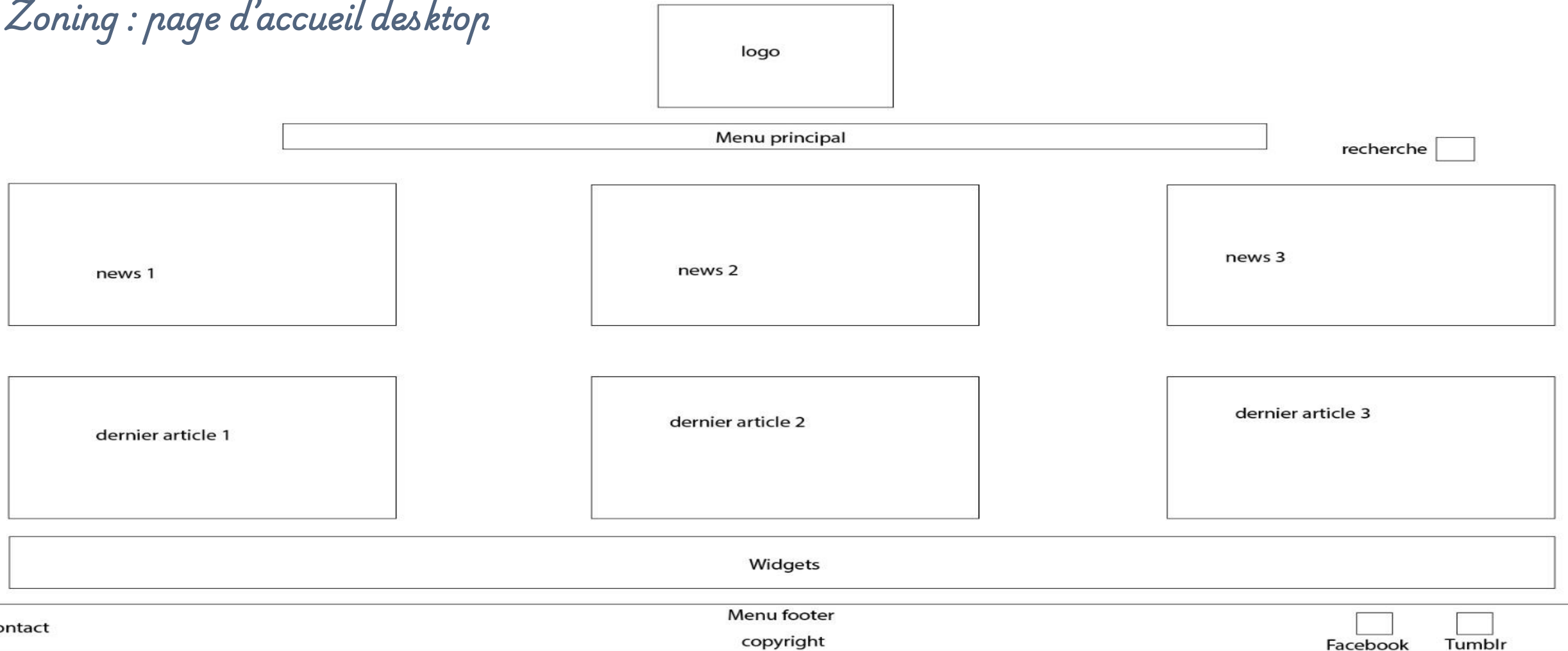
L'idée était d'avoir un menu de navigation en haut de page, un footer en bas de page et le contenu du site entre ces deux éléments.

La bannière titre comporte une image de fond, non représentée sur la maquette, car modifiable depuis l'espace administration.

Sur mobile, le menu de navigation se retrouve dans un menu "hamburger" créé en Javascript et une icône « Font Awesome, tandis que le menu prend toute la hauteur de l'écran.

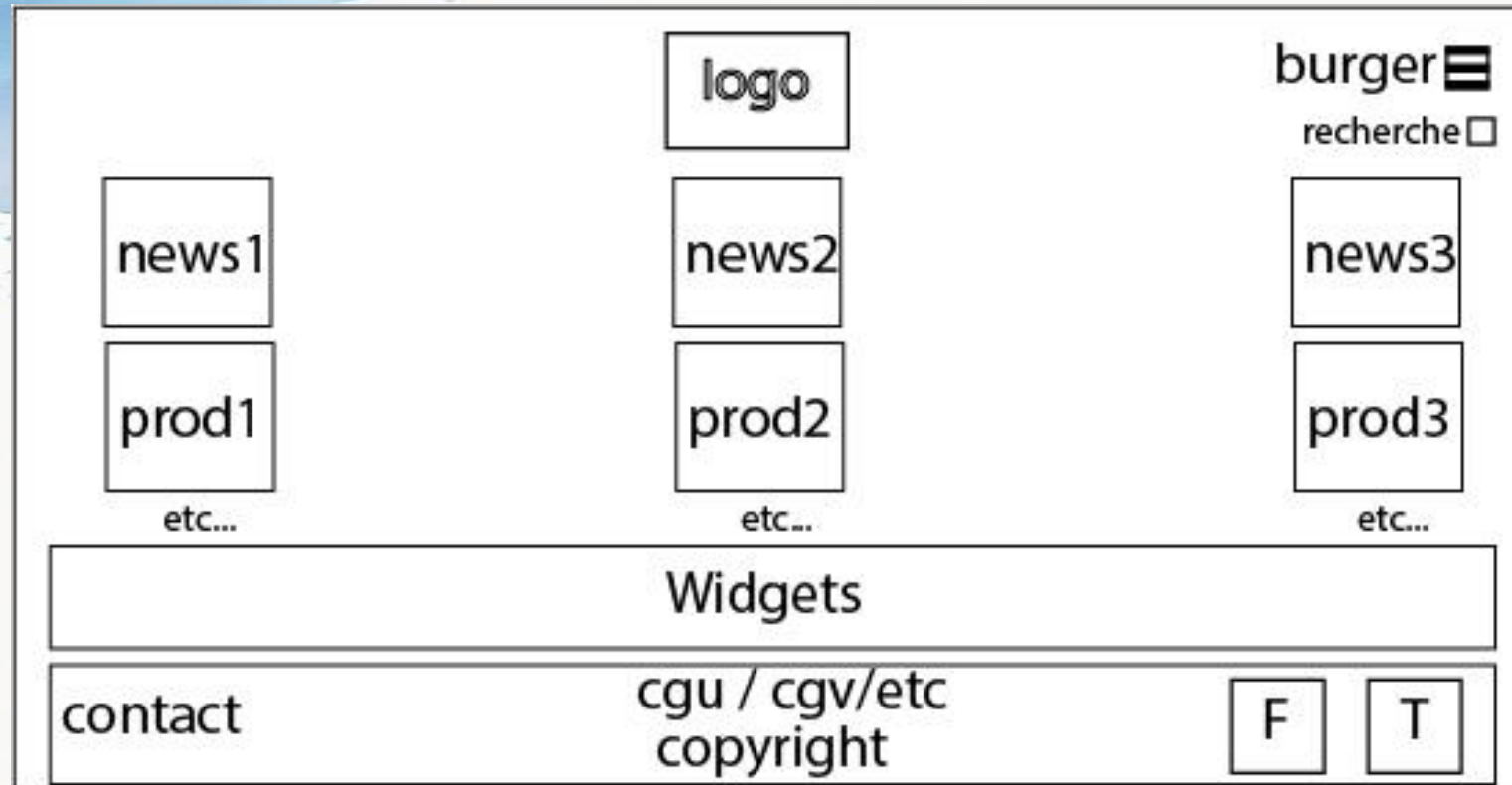
Création de la maquette

Zoning : page d'accueil desktop




Création de la maquette

Zoning : page d'accueil tablette



Création de la maquette

Zoning : page d'accueil mobile

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

Extensions et widgets

Plugin-art-gallery :

Voici une synthèse des fonctionnalités :

- *Gestion des œuvres d'art : Interface pour ajouter et gérer les œuvres d'art.*
- *Affichage : Présentation en grille avec un léger effet de zoom sur les images.*
- *Intégration Woo Commerce : Synchronisation avec Woo Commerce pour récupérer ou utiliser les informations des œuvres d'art comme produits de vente, permettant un gain de temps.*
- *Création automatique : Lorsqu'une œuvre d'art est créée dans l'administration WordPress (via un type de publication personnalisé "artwork"), un produit Woo Commerce correspondant est automatiquement généré s'il n'existe pas déjà.*
- *Mise à jour synchronisée : Toute mise à jour d'une œuvre d'art entraîne également la mise à jour des informations du produit Woo Commerce associé.*

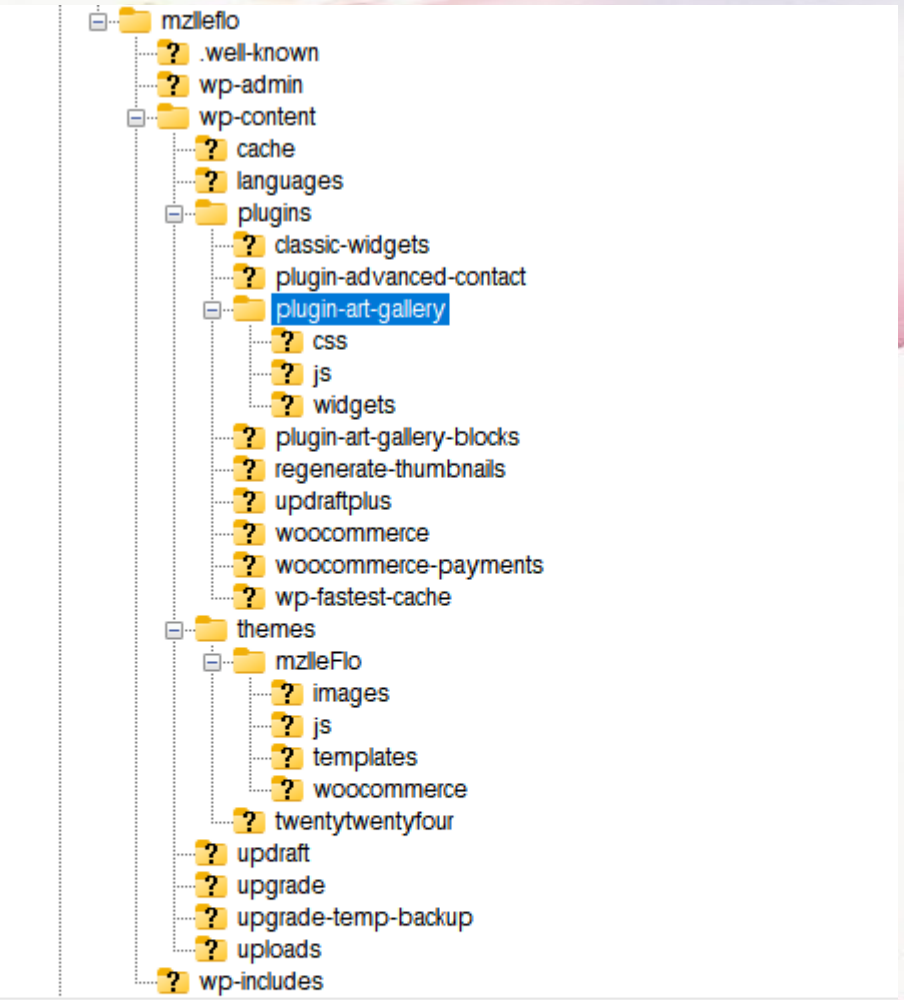
Extensions et widgets

Plugin-art-gallery :

Création du fichier *plugin-art-gallery.php* dans le dossier du même nom, lui-même dans le dossier *plugins* de *wp-content* de WordPress.

Modification du fichier *archive-product.php* de Woo commerce 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 Woo Commerce : Inclure la logique pour afficher les œuvres d'art personnalisées.



Nom de fichier	Taille de fi...	Type de fic...
..		Dossier de...
css		Dossier de...
js		Dossier de...
widgets		Dossier de...
plugin-art-gallery.php	13 806	Fichier sour...
render-functions.php	2 413	Fichier sour...

Extensions et widgets

Plugin-art-gallery :

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.

Extensions et widgets

Plugin-art-gallery :

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

- *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 Woo Commerce :*
 - *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`.*

On pourrait dire que ces opérations sont CRUD :

Création / Read (lecture) / Update (mettre à jour) / Delete (supprimer)

Extensions et widgets

Plugin-art-gallery :

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 *Woo Commerce*.

Les fonctions vues précédemment sont écrites dans le fichier *plugin-art-gallery.php*.

Si *WordPress* était structuré de manière stricte selon le modèle *MVC*, les *Custom Post Types* (CPT) seraient considérés comme des *modèles*.

Dans une architecture *MVC* appliquée à *WordPress* :

- **Modèle (Model)** : Représente les données et la logique métier. Un *Custom Post Type* agit comme un modèle, définissant la structure et les interactions avec la base de données.
- **Vue (View)** : Gère la présentation des données à l'utilisateur. Les fichiers de thème (*Templates*) (*single-{post_type}.php*, *archive-{post_type}.php*) jouent ce rôle en définissant l'affichage des *Custom Post Types*.
- **Contrôleur (Controller)** : Gère les interactions utilisateur et les commandes au modèle. Dans *WordPress*, il est partiellement implémenté via les *Hooks*, actions, filtres, et fonctions personnalisées.

Extensions et widgets

Plugin-art-gallery :

Custom Post Type (CPT ou type de contenu personnalisé) dans WP :

- *Types de contenu personnalisés en plus des types de contenu par défaut (articles, pages, etc.), utilisé pour structurer et organiser des contenus spécifiques qui ne rentrent pas dans les catégories standards.*

Dans le contexte de WP, un CPT est considéré comme un modèle, mais : on pourrait aussi argumenter qu'ils ont certaines caractéristiques d'entités :

- *Identité unique : Chaque instance d'un CPT a un identifiant unique, comme une entité.*
- *Cycle de vie : Les CPT ont un cycle de vie (CRUD) géré par WordPress, similaire à des entités dans un système de persistance.*

Pour l'extension : Elle crée un CPT appelé "artwork" pour gérer les œuvres d'art, afin de séparer les œuvres d'art des autres types de contenu dans WordPress.

Métadonnées :

- *Sont des informations supplémentaires associées aux types de contenu de WP (articles, pages, CPT, etc.).*
- *Permettent de stocker des données spécifiques pour chaque élément de contenu (pour chaque œuvre d'art, l'ID du produit Woo Commerce associé par exemple).*

Les métadonnées ne sont ni des entités ni des objets en elles-mêmes, mais plutôt des attributs ou des propriétés qui décrivent ou enrichissent des entités (comme les CPT).

Extensions et widgets

Plugin-art-gallery : Hooks

Les Hooks dans WordPress permettent de personnaliser et d'étendre ses fonctionnalités sans modifier le code source principal. Ils se divisent en deux types principaux :

- ***Actions** : Exécutées à des moments spécifiques du cycle de vie de WordPress (comme le chargement d'un thème ou la publication d'un article). Elles permettent d'ajouter des fonctionnalités, comme envoyer un email après la publication d'un article.*
- ***Filtres** : Utilisés pour modifier des données avant qu'elles ne soient affichées ou enregistrées. Ils permettent d'altérer des éléments comme le contenu d'un post ou les titres.*

Ces Hooks sont généralement ajoutés dans le fichier `functions.php` du thème.

Extensions et widgets

Plugin-art-gallery : Hooks, exemple ...

Modifier la longueur des extraits :

Fonctionnement : Cette fonction définit la longueur des extraits d'articles (*excerpts*) à 25 mots.

Utilisation du Hook : `add_filter('excerpt_length', 'nbMotsExtraits');`
Le Hook `excerpt_length` permet de personnaliser le nombre de mots dans l'extrait des articles. La fonction `nbMotsExtraits` fixe ce nombre à 25.

```
function nbMotsExtraits($length)
{
    return 25;
}
add_filter('excerpt_length', 'nbMotsExtraits');
```

Modifier le texte "Lire la suite" des extraits :

Fonctionnement : Cette fonction modifie le texte "Lire la suite" qui apparaît à la fin d'un extrait d'article pour le remplacer par ` [→]`.

Utilisation du Hook : `add_filter('excerpt_more', 'finExtrait');`
Le Hook `excerpt_more` contrôle le texte ou l'élément HTML qui suit un extrait. La fonction `finExtrait` le remplace par une flèche (→).

```
function finExtrait($more)
{
    return '&nbsp;[&rarr;]';
}
add_filter('excerpt_more', 'finExtrait');
```

Extensions et widgets

Les autres extensions : 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, Tumblr).

Publication automatique des nouveaux produits avec images et descriptions.

Interface pour programmer des publications.

En stand-by : 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, donc 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 dans l'attente de la décision de la cliente.

De mon côté, je cherche une alternative (JavaScript ? ou autre).

Extensions et widgets

Les autres extensions : 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`, et ajout de short code :

- *`[advanced_contact_form]` pour le formulaire*
- *`[google_maps address=« adresse » zoom="15" width="100%" height="400px"]` pour la carte.*

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

Extensions et widgets

Widgets : Après les extensions, les widgets ...

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

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](#)

Structure de données, Base de données :

Comme indiqué plus tôt, je m'attache, concernant la bdd, aux tables relatives à l'extension "plugin-art-gallery" car c'est la plus complète des extensions faites.

Elle effectue plusieurs manipulations de la base de données WordPress pour gérer les œuvres d'art et les produits Woo Commerce associés (comme vu slide 35)

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 Woo Commerce).*
- *Création et mise à jour des produits Woo Commerce :*
 - *Création de nouveaux produits Woo Commerce lorsque de nouvelles œuvres d'art sont publiées.*
 - *Mise à jour des produits Woo Commerce existants lorsque les œuvres d'art sont mises à jour.*
- *Sauvegarde des métadonnées :*
 - *Enregistrement de l'ID du produit Woo Commerce associé en tant que métadonnée de l'œuvre d'art.*

Structure de données, Base de données :

Exemple : création et enregistrement d'un type de publication personnalisé

//Ajoute l'action create_art_post_type à l'initialisation de WordPress pour créer un CPT « artwork » pour les œuvres d'art
add_action('init', 'create_art_post_type');

function create_art_post_type()

{

// Enregistre un nouveau CPT nommé artwork avec les paramètres spécifiés.

register_post_type('artwork', array(

// Définit les labels pour le CPT

'labels' => array(

// Nom au pluriel

'name' => __('Galerie d\'art', 'textdomain'),

// Nom au singulier

'singular_name' => __('Galerie d\'art', 'textdomain')

),

// Accessible publiquement

'public' => true,

// Active les archives pour ce type de contenu

'has_archive' => true,

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

'supports' => array('title', 'editor', 'thumbnail'),

// Rend le CPT disponible dans l'API REST de WordPress

'show_in_rest' => true,

// Supporte les catégories personnalisées

'taxonomies' => array('art_category'),

));

}

Structure de données, Base de données :

Nous avons vu précédemment les CPT, c'est maintenant qu'ils vont servir, en détails :

Lorsqu'un CPT est créé et que des contenus sont ajoutés, WordPress stocke ces contenus dans la table `wp_posts` de la bdd. (voir page 39 du dossier de projet pour plus de détail sur cette table).

La table `wp_posts` contient une colonne `post_type` qui indique le type de contenu.

Pour les CPT, cette colonne contiendra la valeur spécifiée lors de l'enregistrement du CPT (par exemple, `artwork`).

Exemple de structure de la table `wp_posts` pour un CPT :

ID	post_title	post_content	post_type	...
101	Mon œuvre d'art	Contenu de l'œuvre	artwork	...
102	Un autre contenu	Contenu standard	post	...

Structure de données, Base de données :

Au tour des métadonnées :

Les métadonnées associées à un contenu sont stockées dans la table `wp_postmeta`.

Cette table contient des informations supplémentaires sous forme de paires clé-valeur, où chaque enregistrement est lié à un élément de contenu spécifique via l'ID du post.

Exemple de structure de la table `wp_postmeta` :

meta_id	post_id	meta_key	meta_value
1	101	_artwork_product_id	303
2	102	_other_meta_key	Some value

Explications :

`post_id` : Identifie l'ID du post auquel les métadonnées sont associées (par exemple, 101 pour une œuvre d'art).

`meta_key` : Indique la clé des métadonnées (par exemple, `_artwork_product_id`).

`meta_value` : Contient la valeur des métadonnées (par exemple, 303 pour l'ID du produit WooCommerce associé).

Structure de données, Base de données :

Exemples de requêtes pour manipuler et interagir avec les CPT et les métadonnées

Explications :

- *Requêtes simples* : Utilisent des clauses de base comme **SELECT**, **WHERE**, et **INSERT**.
- *Requêtes multiples* : Utilisent plusieurs conditions dans la clause **WHERE**.
- *Requêtes croisées* : Utilisent des jointures (**JOIN**) pour combiner des données de plusieurs tables (par exemple, `wp_posts` et `wp_postmeta`).
- *Mise à jour et suppression* : Utilisent des clauses **UPDATE** et **DELETE** pour modifier ou supprimer des enregistrements existants.

Ces requêtes couvrent des opérations courantes que l'on peut effectuer pour manipuler des Custom Post Types et des métadonnées dans une base de données WordPress.

Structure de données, Base de données :

Exemples de requêtes pour manipuler et interagir avec les CPT et les métadonnées

Requête simple pour récupérer des Custom Post Types / Récupérer tous les artworks (œuvres d'art) :

```
SELECT * FROM wp_posts WHERE post_type = 'artwork';
```

Requête multiple pour récupérer des Custom Post Types avec des conditions / Récupérer tous les artworks publiés (status publish) :

```
SELECT * FROM wp_posts WHERE post_type = 'artwork' AND post_status = 'publish';
```

Requête croisée pour récupérer des CPT avec leurs métadonnées / Récupérer toutes les œuvres d'art et leur ID de produit WooCommerce associé :

```
SELECT p.ID, p.post_title, pm.meta_value AS product_id  
FROM wp_posts p  
JOIN wp_postmeta pm ON p.ID = pm.post_id  
WHERE p.post_type = 'artwork' AND pm.meta_key = '_artwork_product_id';
```

Requête pour insérer un nouveau CPT / Insérer une nouvelle œuvre d'art :

```
INSERT INTO wp_posts (post_author, post_date, post_content, post_title, post_status, post_type)  
VALUES (1, NOW(), 'Description de l\'œuvre d\'art', 'Titre de l\'œuvre', 'publish', 'artwork');
```

Requête pour insérer des métadonnées / Insérer une métadonnée associée à une œuvre d'art :

```
INSERT INTO wp_postmeta (post_id, meta_key, meta_value)  
VALUES (101, '_artwork_product_id', '303');
```

Structure de données, Base de données :

Exemples de requêtes pour manipuler et interagir avec les CPT et les métadonnées

Requête pour mettre à jour un CPT / Mettre à jour le titre d'une œuvre d'art :

```
UPDATE wp_posts  
SET post_title = 'Nouveau titre de l'œuvre'  
WHERE ID = 101;
```

Requête pour mettre à jour des métadonnées / Mettre à jour l'ID du produit associé à une œuvre d'art :

```
UPDATE wp_postmeta  
SET meta_value = '404'  
WHERE post_id = 101 AND meta_key = '_artwork_product_id';
```

Requête pour supprimer un CPT / Supprimer une œuvre d'art :

```
DELETE FROM wp_posts  
WHERE ID = 101;
```

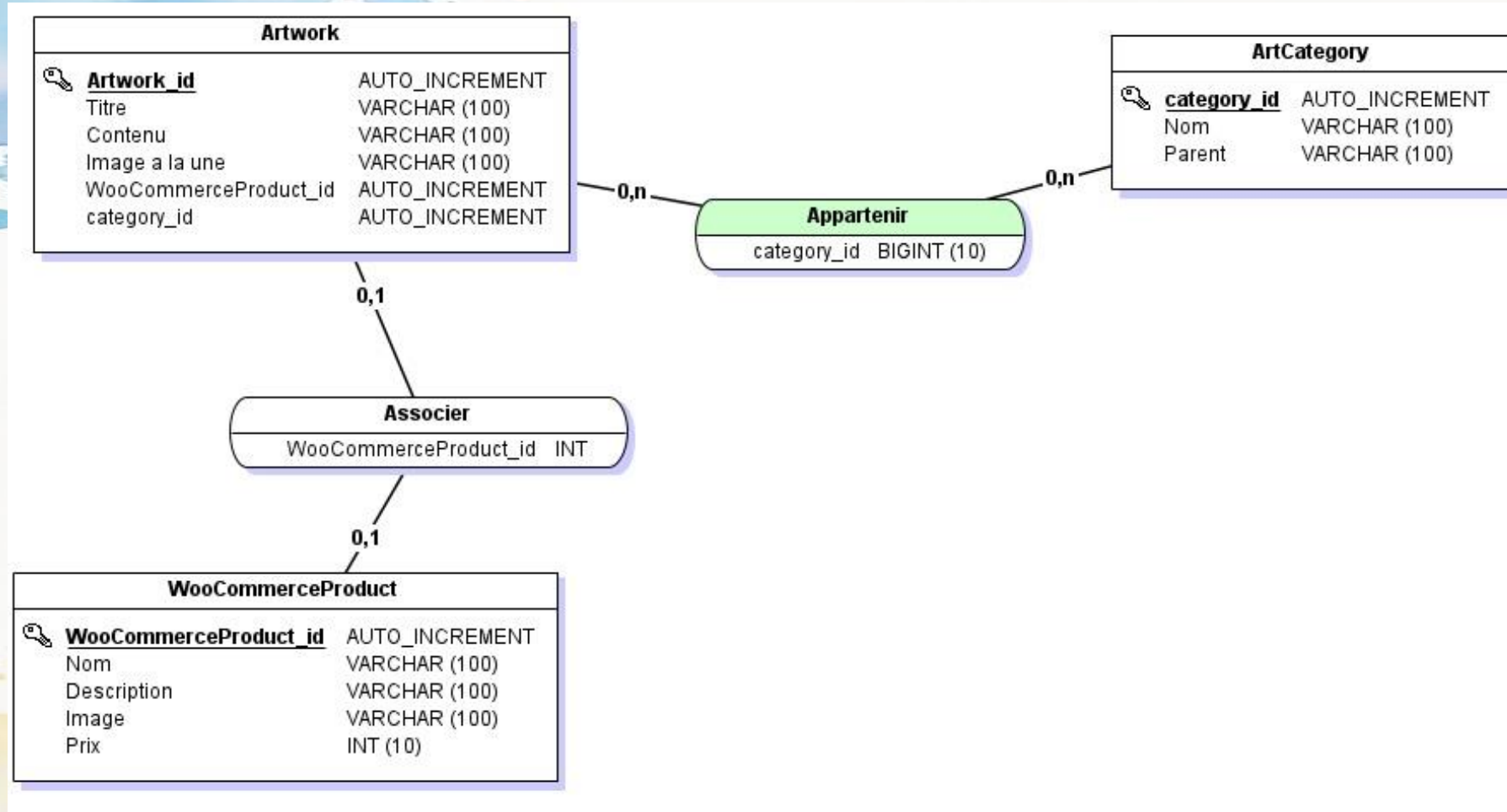
Requête pour supprimer des métadonnées / Supprimer des métadonnées associées à une œuvre d'art :

```
DELETE FROM wp_postmeta  
WHERE post_id = 101 AND meta_key = '_artwork_product_id';
```

Ces exemples ne montrent que 2 des 7 tables utilisées par l'extension (voir page 40 du dossier de projet pour toutes les tables concernées).

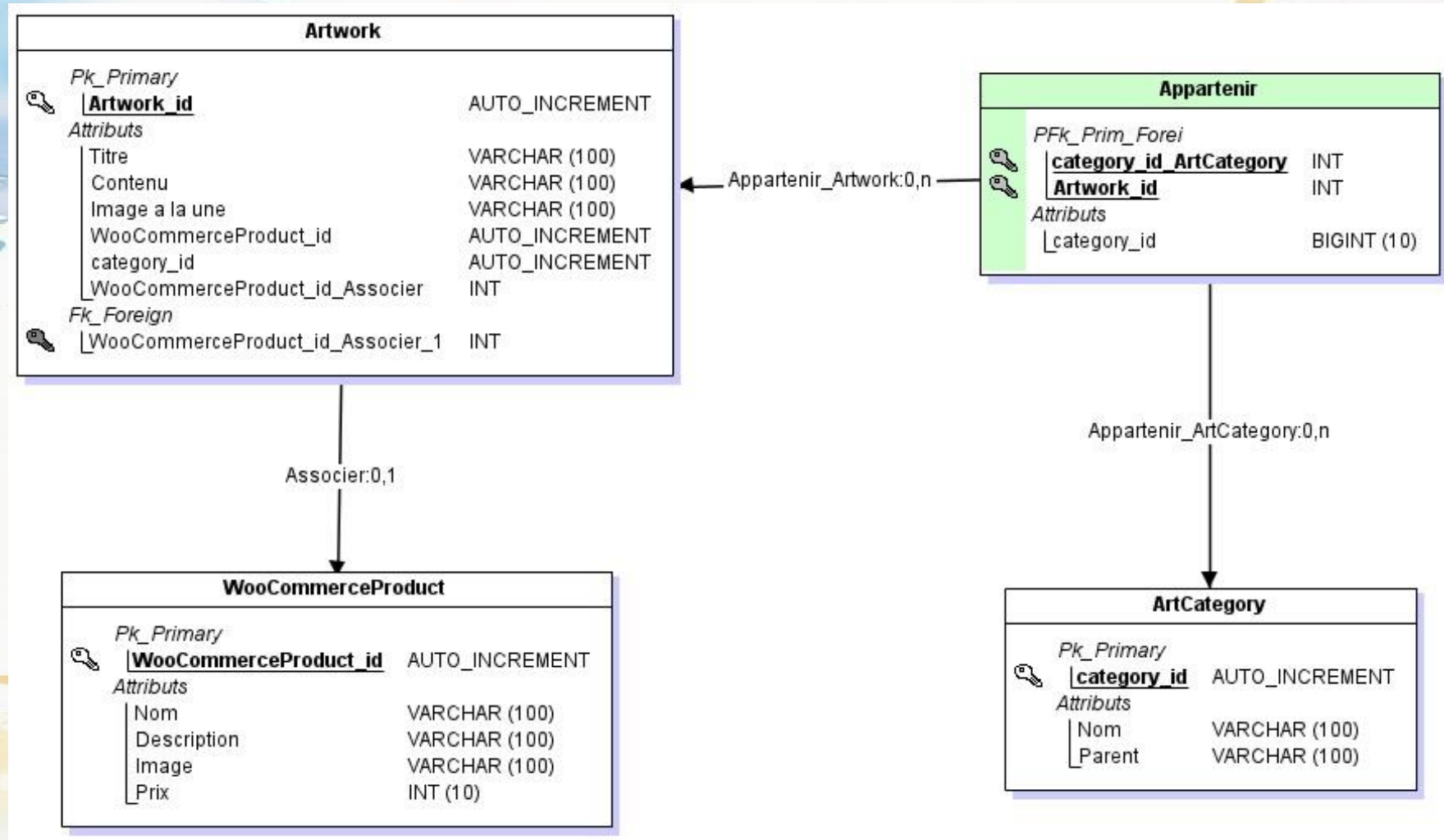
Base de données : MCD, MLD et MPD

Diagramme Modèle Conceptuel de Données (MCD)



Base de données : MCD, MLD et MPD

Modèle Logique de Données (MLD) Schématisation (voir page 43 du dossier de projet pour plus de détails)



Base de données : MCD, MLD et MPD

MPD : représentation 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;
```

Développement

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 personnaliser (archives, categories) relatif à l'extension plugin-art-gallery (article "artwork")

Développement

Architecture - pour le thème : 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*

Développement

Architecture - pour les extensions ,

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"]

Tests

Tests du frontend (l'interface graphique) : =>

Les tests du frontend 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 (ajout 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

Tests du backend (persistance des données) : =>

N°	Test	Technique de test
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

Plus :
Utilisation du fichier `debug.log` (`wp-content`) suite à ajout des lignes suivantes dans `wp-config.php` :

```
define('WP_DEBUG', true);  
define('WP_DEBUG_LOG', true);  
define('WP_DEBUG_DISPLAY', false);  
@ini_set('display_errors', 0);
```


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*

Prévu (fin septembre, avec la cliente) :

- *Choix de l'hébergeur (en cours)*
- *Configuration du domaine (enregistrement DNS et SSL, en cours)*

Déploiement

Plan de déploiement (acteur, planning, méthodologie, ...), Reste à faire :

- *Corriger deux bugs :*
 - *widget art-gallery (une erreur apparaît lors de l'initialisation du widget mais celui ci s'affiche quand même)...*
 - *et choix du type de paiement suivant location ou achat) voir page suivante...*
- *Modifier un fichier template de Wordpress (mauvaise disposition de vignette)*
- *En attente des informations supplémentaires des produits (dimensions, poids, etc.) par la cliente*
- *En attente des prix définitifs*
- *En attente de mise à jour suivant la cliente (changement d'avis ou autres impératifs...)*
- *En attente décision API Facebook*

Déploiement

Plan de déploiement (acteur, planning, méthodologie, ...), Reste à faire :

Bug choix du type de paiement dans Woo Commerce

Création de deux fonction 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 dans le dossier woocommerce : pour afficher les informations de paiement et 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.*

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 :

J'ai régulièrement consulté plusieurs sources telles que :

- *la CVE (Common Vulnerabilities and Exposures) : <https://www.cve.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.*

Veille technique

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.

Recherche effectuée sur des sites

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.*

Veille technique

D'autres erreurs ou bugs rencontrés ont nécessité soit des recherches sur Google, des manuels en ligne comme php.net/manual etc., Discord (canal DecClub), Reddit, etc. etc., soit (et combiné avec Google) l'utilisation de prompts avec IA...

A cet effet, j'ai utilisé :

- *ChatGPT (version free 3.5)*
- *Le chat Mistral*
- *Claude*
- *Et plus rarement Gemini et Copilot...*

L'utilisation des « chats » était faite en simultanée sur plusieurs onglets, et les réponses croisées et vérifiées (via mes cours, google, des forums officiels etc.) et à surtout servis à la vérification du code et en particulier les erreurs de syntaxes ou du nettoyage du code.

Conclusion

Bilan technique et humain :

Wordpress :

Vision initiale : Outil trop simple et peu flexible

Découverte : Similitudes avec MVC et CRUD, malgré une approche non stricte

Constat final : Plateforme puissante et personnalisable

Compétences démontrées :

- *Personnalisation de thème*
- *Développement de plugins*
- *Optimisation des performances*
- *Renforcement de la sécurité*

Conclusion :

WordPress s'est révélé être une solution adaptable pour des projets complexes.

Cependant, il est dommage que les circonstances ne m'aient pas permis de travailler sur du "full" PHP, car j'ai une préférence personnelle pour le MVC et la POO en PHP.

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*
- *Et vous, le jury, de m'avoir suivi pendant ces minutes...*