

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ Zinck  
*Nom d'usage* ▶  
*Prénom* ▶ Nicolas  
*Adresse* ▶ 2 Rue Louis Pasteur Mehun-Sur-Yèvre 18500

## Titre professionnel visé

Développeur(se) web et web mobile

### MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.  
**Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]*

### Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

*Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.*



<http://travail-emploi.gouv.fr/titres-professionnels>

## Sommaire

### Exemples de pratique professionnelle

#### Intitulé de l'activité-type n° 1 : Développer la partie front-end d'une application web ou web mobile

p.

5

► Intitulé de l'exemple n° 1 : Installer et configurer son environnement de travail en fonction du projet web ou web mobile .....p.

p.

► Intitulé de l'exemple n° 2 : Maquetter des interfaces utilisateur web ou web mobile .....p.

p.

► Intitulé de l'exemple n° 3 : Réaliser des interfaces utilisateur statiques web ou web mobile .....p

p.

► Intitulé de l'exemple n° 4 : Développer la partie dynamique des interfaces utilisateur web ou web mobile

p.

#### Intitulé de l'activité-type n° 2 Développer la partie back-end d'une application web ou web mobile sécurisée

p.

► Intitulé de l'exemple n° 1 : Mettre en place une base de données relationnelle.....p.

p.

► Intitulé de l'exemple n° 2 : Développer des composants d'accès aux données SQL et NoSQL .....p.

p.

► Intitulé de l'exemple n° 3 : Développer des composants métier coté serveur.....p

p.

► Intitulé de l'exemple n° 4 : Documenter le déploiement d'une application dynamique web ou web mobile .....p

p.

#### Titres, diplômes, CQP, attestations de formation (facultatif)

p.

#### Déclaration sur l'honneur

p.

#### Documents illustrant la pratique professionnelle (facultatif)

p.

#### Annexes (Si le RC le prévoit)

p.

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

**Exemple n°1** ► Installer et configurer son environnement de travail en fonction du projet web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de ma formation Développeur Web et Web Mobile à la Chambre des Commerce de Bourges, j'ai dû installer et configurer un environnement de travail complet afin de développer un site dynamique en PHP.

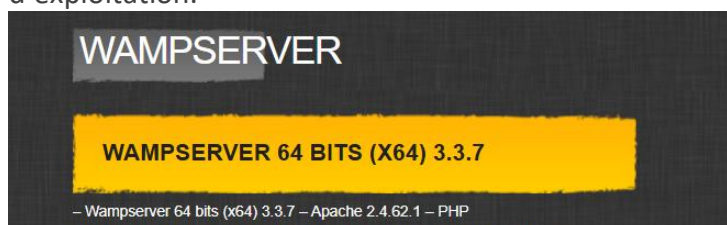
Pour cela, j'ai choisi d'utiliser WAMP pour simuler un serveur local, Visual Studio Code comme éditeur de code, et Git pour assurer le versionnement de mon projet. Ce choix d'outils m'a permis de travailler dans des conditions proches de celles rencontrées en entreprise et de mettre en pratique les compétences acquises tout au long de ma formation.

#### 1.1 Installation et configuration de WAMP

J'ai installé WAMP (Windows Apache MySQL PHP), un environnement de développement local, depuis le site officiel : <https://www.wampserver.com/>.



Dans la section téléchargement, j'ai sélectionné WampServer 64 BITS, correspondant à mon système d'exploitation.

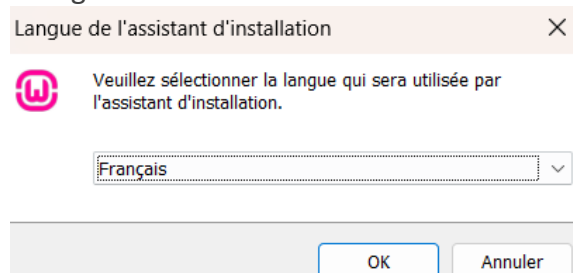


Une fenêtre POP-IN est apparue, me proposant de remplir un formulaire pour recevoir les actualités d'Alter Way. Comme cette étape est facultative, j'ai choisi de cliquer sur « Passer au téléchargement direct »

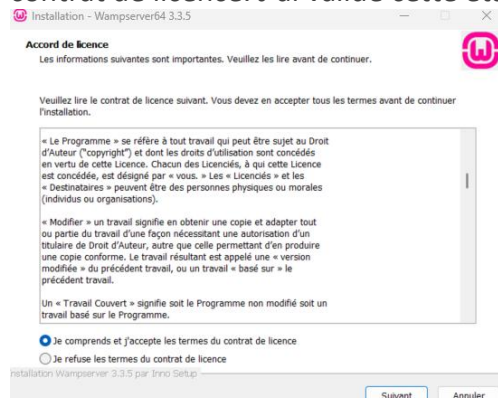
WampServer est disponible gratuitement (sous licence GPL). Vous pouvez recevoir les actualités formation d'Alter Way, société éditrice, ainsi que toutes les informations, vous pouvez [passer au téléchargement direct](#).

Prénom :  Nom :

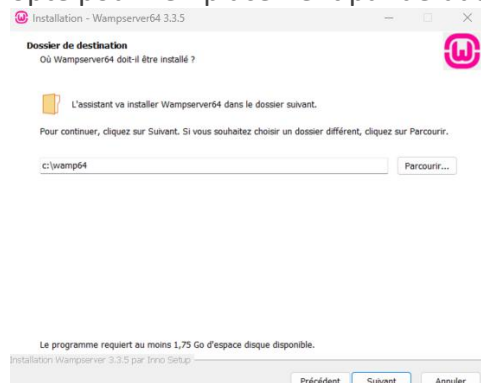
Une fois le fichier téléchargé (WampServer3.3.5\_64.exe), je l'ai ouvert et l'installation a débuté. Le logiciel m'a tout d'abord demandé de choisir la langue d'installation.



Après avoir sélectionné la langue d'installation, le logiciel m'a demandé d'accepter les conditions du contrat de licence. J'ai validé cette étape afin de poursuivre l'installation.



Ensuite, l'assistance d'installation m'a invité à choisir le répertoire d'installation de WampServer. J'ai opté pour l'emplacement par défaut, à la racine de mon disque dur, soit « c:/ », pour plus de simplicité



Une fois l'emplacement confirmé, l'installation s'est poursuivie automatiquement.

Durant le processus, le logiciel m'a également demandé de choisir mon navigateur web par défaut qui sera utiliser pour ouvrir les fichier Wamp. J'ai donc sélectionné GoogleChrome.

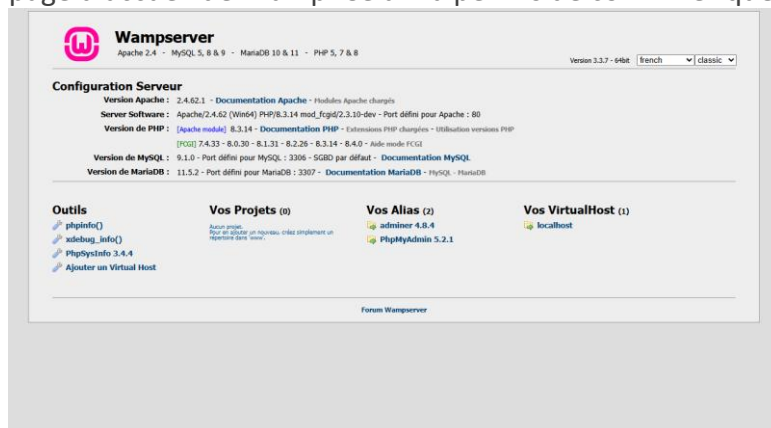
Le logiciel ma également demandé de choisir mon éditeur de textes j'ai donc choisis VisualCodeStudio, mon éditeur principal tout le long du projet.

Une fois toutes ces étapes terminées, l'installation s'est finalisée, et une icône Wamp est apparue dans ma barre des tâches.

Pour m'assurer du bon fonctionnement, j'ai lancé WAMP et vérifier que l'icône était bien verte, ce qui signifie que tous les services (Apache, MySQL) sont opérationnels



J'ai ensuite ouvert mon navigateur et accédé a l'adresse « http://localhost », qui ma redirigé vers la page d'accueil de Wamp. Cela ma permis de confirmer que le serveur local fonctionnait correctement.

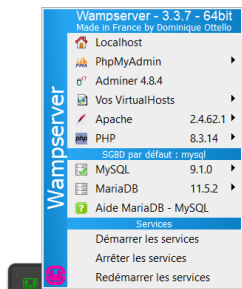


## 1.2 Création de virtualHost

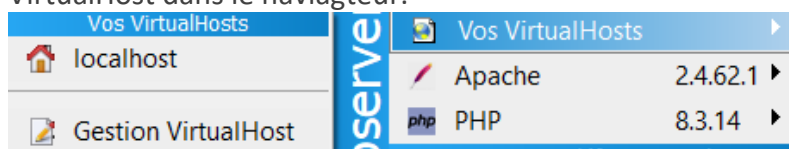
Pour faciliter l'accées à mon projet et simuler un environnement plus proche de celui d'un serveur en production, j'ai mis en place un VirtualHost. Cela m'a permis d'ouvrir mon site en local avec une URL personnalisée du type http://Elessard-Bijouterie.

Voici les étapes que j'ai suivies :

Tout d'abord il faut ouvrir le panneau de contrôle de WAMP, accessible depuis l'icône dans la barre des tâches, en bas à droite de l'écran.



Ensuite, dans le menu je sélectionne la rubrique « VirtualHost », ce qui m'ouvre le page de gestion des VirtualHost dans le navigateur.



Cette page me permet de :

- Consulter la liste des VirtualHosts déjà existants
- Accéder à un formulaire de création d'un nouveau VirtualHost

Pour créer un nouveau VirtualHost, je dois remplir ce formulaire.

Je commence par :

- Un nom de domaine local ici j'ai choisi comme nom « projet »
- Un chemin absolu du dossier contenant mon projet

Pour obtenir ce chemin, j'ouvre mon dossier de projet dans l'explorateur de fichiers, je copie le chemin d'accès dans la barre d'URL, puis je le colle dans le champ « Chemin complet absolu du dossier VirtualHost »

Attention : Ces deux champs sont obligatoires pour que la création du VirtualHost fonctionne.

Le reste du formulaire contient des champs optionnels.

Dans mon cas, j'ai également choisi la version de PHP 8.3.14, car c'est celle utilisée par défaut dans mon environnement WAMP et celles avec laquelle j'ai travaillé tout au long de ma formation.

Voici à quoi ressemble le formulaire une fois complété :



**VirtualHost déjà définis :**

ServerName : localhost - Directory : c:/wamp642/www

Windows hosts : C:/WINDOWS/system32/drivers/etc/hosts

Nom du Virtual Host : Pas d'espace - Pas de tiret bas (\_) **Requis**

projet

**Chemin** complet absolu du **dossier** VirtualHost - Exemples : C:/wamp/www/projet/ ou E:/www/site1/ **Requis**

C:\Users\znico\Desktop\Projet CCI

☐ **SI** vous voulez utiliser PHP en mode CGI **Versions acceptées** ci-dessous **Optionnel** Aide mode CGI

PHP : 8.3.14

Si vous voulez utiliser un "Listen port" autre que celui par défaut, vous devez ajouter un Listen Port à Apache par Clic-Droit Outils **Optionnel**

**SI** vous voulez utiliser les VirtualHost par IP : **IP locale** 127.x.y.z **Optionnel**

Démarrer la création ou la modification du VirtualHost (Peut prendre un certain temps)

## 2.Choix des technologies utilisées dans mon environnement de travail

Dans cette deuxième partie, je vais présenter les outils et technologies que j'ai choisis pour développer mon projet, en expliquant les raisons de ces choix.

### Visual Studio Code :

J'ai choisis Visual Studio Code comme environnement de développement car il est léger, gratuit et très complet.

- **Gratuit, léger et multiplateforme**

VSCode est un logiciel libre rapide à installer, et disponible sur tous les systèmes d'exploitation (Windows, MacOS, Linux) ce qui le rend accessible à tous les développeurs.

- **Interface claire et moderne**

Son interface est simple à prendre en main avec une bonne organisation des fichiers et des onglets, ce qui facilite la lecture et l'écriture du code.

- **Extensions très puissantes**

VSCode permet d'ajouter des extensions selon ses besoins. Par exemple, j'ai utilisé :

- **Live Server :**

Pour visualiser mes fichiers HTML dans le navigateur en temps réel

**PHP Intelephense :** Cette extension améliore considérablement le confort de développement en PHP. Elle offre une autocomplétion intelligente, une détection des erreurs en temps réel, ainsi qu'un accès rapide à la documentation des fonctions. Grâce à elle, j'ai pu coder plus rapidement et efficacement, tout en résolvant les erreurs de syntaxe ou d'appel de fonction.

- **Markdown Preview Enhanced :**

J'ai également cette extension, qui me permet de prévisualiser mes fichiers « .md » directement dans Visual Code Studio.

C'est particulièrement utile pour la rédaction de documentation technique ( comme le README d'un projet), car cela me permet de voir en temps réel le rendu final du texte formaté en Markdown.

- **Support de nombreux langages :**

Il prend en charge HTML, CSS, JavaScript, PHP (et bien d'autres) ce qui correspond parfaitement aux technologies que j'ai utilisé durant ma formation et dans mon projet.

- **Terminal intégré**

VSCode intègre un terminal, ce qui me permet d'exécuter des commandes Git, GitBash, ou d'utiliser des outils comme Composer pour installer PHPMailer sans quitter l'éditeur de code

- Très utilisé dans le milieu professionnel

C'est un IDE largement utilisé dans le monde du développement web. Le maîtriser représente donc un atout pour mon intégration dans le monde professionnel.

### Git :

Dans le cadre de mon projet et de ma formation j'ai utilisé Git, un système de gestion de versions très utilisé dans le développement. Cet outil m'a permis de garder une trace de l'évolution de mon code de travailler plus proprement et de corriger mes erreurs facilement

Voici pourquoi Git est indispensable dans un projet :

- **Suivis des modifications**

Git me permet d'enregistrer chaque changement apporté à mon code (ajout, suppression, modification de fichier). Cela crée un historique clair de mon projet, que je peux consulter ou revenir en arrière à tout moment.

- **Sauvegarde et sécurité**

Grâce à Git, je peux sauvegarder régulièrement mon travail avec des Commits. En cas de bug ou d'erreur, je peux revenir à une version précédente stable du projet, ce qui limite les risques de perte de données.

- **Organisation du code**

Git fonctionne avec des Branches, ce qui me permet de tester de nouvelles fonctionnalités sans toucher au code principal (Branch Main). Une fois la fonctionnalité terminée et testée, je peux la fusionner proprement dans le projet.

- **Travail en équipe**

Même si j'ai travaillé seul sur mon projet, Git est l'outil standard utilisé en entreprise pour le travail en équipe. Il facilite la collaboration, évite les conflits de version, et permet de centraliser le code dans un dépôt distant (comme GitHub)

- **Professionnalisation**

Utiliser Git m'a permis de me familiariser avec un outil utilisé par tous les développeurs. C'est une compétence essentielle pour travailler en entreprise, participer à des projets collaboratifs, ou contribuer à des projets open-source.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Wamp pour exécuter mon projet PHP en local, VSCode comme éditeur de code avec des extensions utiles (Live Server, Intelephense, Markdown Preview Enhanced ), et Git pour versionner et suivre l'évolution du projet.

## 3. Avec qui avez-vous travaillé ?

Pour cette partie j'ai travailler seul.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Chambre de Commerce de Bourges*

Chantier, atelier, service ► *Projet réaliser au cours de ma formation*

Période d'exercice ► Du : *18/09/2024* au : *04/07/2025*

## 5. Informations complémentaires (facultatif)

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

Exemple n°1 ► Maquetter des interfaces utilisateur web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

#### Maquetter une interface utilisateur.

Dans le cadre de la réalisation de mon site dynamique « Elessar Bijouterie », j'ai dû concevoir une maquette représentant l'interface utilisateur de mon projet. Cette étape de conception visuelle est essentielle pour structurer les différentes pages, anticiper l'expérience utilisateur (UX), et organiser les éléments graphiques de façon claire et cohérente.

Pour cela, j'ai utilisé Figma, un outil de design d'interface collaboratif en ligne. Il m'a permis de créer un prototype de mon site, consultable à tout moment et d'effectuer facilement des ajustements tout au long du processus de création.

Afin de respecter au mieux les besoins réels de l'utilisateur final, j'ai réalisé cette maquette en collaboration directe avec le bijoutier lui-même. En plus d'être un ami, il m'a apporté sa vision professionnelle du métier, ses attentes en matière de design, et ses exigences fonctionnelles. Cela m'a permis de produire une interface fidèle à son image de marque, moderne et élégante.

#### Maquette Réalisées

Au total, j'ai réalisé 12 maquettes de pages qui structurent le site complet.

Voici une présentation de quelques maquettes :

- Page d'Accueil : Cette page est la vitrine principale du site. Elle met en avant une sélection de produits phares via un carrousel, ainsi que les derniers articles ajoutés. Elle donne un aperçu rapide de l'univers du bijoutier et redirige vers les différentes sections du site grâce à un menu clair visible. (Voir annexe : page 1)
- Page Boutique :  
La page boutique affiche l'ensemble des créations disponibles ou non. L'utilisateur peut filtrer les produits par collection (crânes, Cristaux, Pierre volcanique) ou par type de bijou (Bracelet, Pendentif etc.). Il peut aussi trier les créations selon leur prix. Cette page facilite la recherche et la navigation dans les articles. (Voir annexe : page 2)

# DOSSIER PROFESSIONNEL (DP)

- Page Produit

La page produit affiche les détails d'un bijou sélectionné : nom, collection, prix, disponibilité en stock, description et options de quantité. Elle comprend également un bouton « Ajouter au panier » et des boutons pour changer la quantité. Son objectif est d'offrir un maximum d'informations de manière clair et attractive. (Voir annexe : page 3)

- Page Panier

La page panier permet à l'utilisateur de visualiser les articles sélectionnés avant de passer à l'achat. Elle affiche un récapitulatif des produits, leurs quantités (qui peut être changée), le prix total et propose de valider la commande. (Voir annexe : page 4)

- Page Connexion

Cette page permet aux clients de se connecter à leur espace personnel afin de consulter leurs commandes ou modifier leur profil. Elle comprend un formulaire de connexion classique avec email et mot de passe. (Voir annexe : page 5)

La réalisation de ces maquettes m'a permis d'anticiper l'architecture globale du site, de structurer efficacement les contenus et de garantir une expérience utilisateur fluide et intuitive. Grâce à l'utilisation de Figma et à la collaboration directe avec le client, j'ai pu concevoir une interface moderne, en adéquation avec l'univers du bijoutier et les besoins réels des utilisateurs.

## 2. Précisez les moyens utilisés :

Pour cette partie j'ai utilisé Figma.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé avec le bijoutier pour la réalisation de la maquette

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ellessard Bijouterie*

Chantier, atelier, service ► Fait dans le cadre d'une demande de site web

Période d'exercice ► Du : *12/04/2025* au : *13/04/2025*

# DOSSIER PROFESSIONNEL (DP)

## 5. Informations complémentaires *(facultatif)*

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

**Exemple n°1** ► Réaliser des interfaces utilisateur statiques web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

#### Réaliser des interfaces utilisateur statiques web

Dans le cadre du développement de mon projet « Elessar Bijouterie », j'ai réalisé plusieurs interfaces utilisateur statiques en m'appuyant sur les maquettes conçues préalablement avec Figma. Ces interfaces représentent les pages principales du site et permettent de structurer visuellement le contenu avant d'y intégrer la partie dynamique en PHP

- **Technologies utilisées**

Pour cela, j'ai utilisé principalement les langages suivants :

- HTML5 : pour structurer les pages et les différents blocs de contenu
- CSS3 : pour la mise en forme des éléments (couleurs, typographies, espacements, responsive...)
- JavaScript : ponctuellement pour certaines interactions comme le menu burger, la gestion du panier ou le carrousel
- PHP : dans le cadre de mon architecture MVC, les fichiers « .php » dans le dossier Views contiennent le HTML de mes vues

- **Méthode de travail**

j'ai travaillé avec une architecture MVC (Modèle-Vue-Contrôleur), ce qui m'a permis d'organiser mon code proprement.

Toutes les parties HTML ont été intégrées dans les fichiers de vues views/, tandis que les feuilles de style CSS sont centralisées dans le dossier assets/. Cette organisation respecte la séparation des responsabilités et permet une meilleure lisibilité du code

Je suis partie de mes maquettes et j'ai codé chaque page en reprenant leur structure :

- Header et Footer réutilisables sur toutes les pages
- Sections bien délimitées pour chaque partie (ex : Carrousel, mise en avant des derniers articles, formulaire...)
- Mise en page adaptée aux différents formats d'écran (responsive design avec des Media Queries en CSS)

- **Exemples de pages intégrées**

J'ai intégré plusieurs pages statiques à partir de mes maquettes, en respectant leur structure, leur style et leur cohérence graphique. Voici quelques représentatifs de mon travail d'intégration :

**Page d'accueil :**

Cette page met en avant l'univers de la bijouterie avec un carrousel en première section, suivi d'une deuxième section qui affiche les dernier article publier par l'administrateur (Voir annexe page 6)

**Page Boutique :**

Elle affiche l'ensemble des créations disponibles sous forme de grille. Des filtres permettent de trier les articles par collection ou par type de bijou, ainsi qu'un tri par prix. (voir annexe page 7)

- **Page Produit :**

Chaque produit est présenté sur une page dédiée ave son nom, sa description, son prix, sa disponibilité, une ou plusieurs images et un système de sélection de quantité. L'utilisateur peut facilement l'ajouter au panier. (Voir annexe page 8)

- **Page Panier :**

Cette page liste les articles ajoutés par l'utilsiateur avec la possibilité de modifier les quantités. Elle affiche le total du panier et propose un bouton pour passer à la validation de la commande. (Voir annexe page 9)

- **Page de connexion :**

Une page simple et épurée avec un formulaire de connexion comportant les champs email et mot de passe. Elle permet aux clients d'accéder à leur espace personnel. (Voir annexe page 10)

Chaque page a été testée dans le navigateur pour vérifier le rendu visuel, la compatibilité et l'accessibilité.

## Conclusion

Cette étape m'a permis de transformer mes maquettes en pages Web statiques fidèles au design initial. J'ai pu structurer les contenu de manière claire, avec un code propre, maintenable et prêt à accueillir la partie dynamique de mon site



# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

- Visual Studio Code : Comme éditeur de code.
- HTML5 CSS3 JavaScript et PHP : langages utilisées pour l'intégration les animation et la structure du projet .
- Figma : Pour consulter la maquette lors de l'intégration.
- Navigateur Chrome : Pour tester l'affichage des pages en version desktop et mobile
- Outils de développement navigateur : Pour simuler l'affichage mobile et corriger les erreur CSS et JavaScript

## 3. Avec qui avez-vous travaillé ?

Pour cela j'ai travaillé principalement seul

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Elessar Bijouterie*

Chantier, atelier, service ► Création de la partie statique d'un site web

Période d'exercice ► Du : *14/04/2025* au : *04/05/2025*

## 5. Informations complémentaires (facultatif)

## Activité-type 1 Développer la partie front-end d'une application web ou web mobile

**Exemple n°1** ► Développer la partie dynamique des interfaces utilisateur web ou web mobile

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

#### Développer la partie dynamique des interfaces utilisateur

Une fois l'interface graphique du site réalisée, j'ai intégré la partie dynamique afin de rendre les pages interactives et connectées à une base de données. Cette étape permet d'adapter le contenu de la boutique en fonction des disponibilités. L'utilisateur peut ensuite consulter la page du produit, ajouter au panier et ensuite faire diminuer les stocks après une commande.

Dans mon projet « Elessar Bijouterie », seul l'administrateur a la possibilité d'ajouter, modifier ou supprimer des créations via une interface de gestion. L'utilisateur, quant à eux, peuvent uniquement parcourir les produits, consulter les détails, ajouter des articles à leur panier et finaliser une commande. Lors de cette validation, le stock du bijou est automatiquement mis à jour.

J'ai développé cette logique à l'aide de PHP en suivant une architecture MVC (Modèle-Vue-Contrôleur), qui me permet de bien organiser mon code, séparer les responsabilités du projet et faciliter la maintenance du projet.

#### Explication de l'architecture MVC :

##### Le Modèle :

Contient la logique de gestion des données (requêtes SQL, connexions à la base de données etc..)

**Le Contrôleur :** Reçoit les actions de l'utilisateur (comme cliquer sur un bouton ou soumettre un formulaire) peut aussi Controller si le formulaire est correctement rempli par exemple et fait le lien entre la vue et le modèle.

**La Vue :** La Vue est chargée d'afficher dynamiquement les informations à l'utilisateur

#### Exemple 1 : Affichage des créations dans la Boutique

J'ai créé une table création dans ma base de données, contenant les informations suivantes : Nom, Description, Prix, Stock et une relation vers la table Image\_Creation pour stocker le chemin vers les images enregistrées (car une création peut avoir plusieurs images).

Grâce au modèle associé, je récupère en PHP toutes les créations à afficher sur la page Boutique, et je boucle sur ces données pour générer dynamiquement les fiches produit (voir annexe page 11).

## Exemple 2 : Détail d'un produit

Quand un utilisateur clique sur un bijou, son ID est passé en GET dans l'URL. Ce paramètre permet de récupérer en base les informations précises du produit sélectionné, qui seront ensuite affichées dans un page dédiée (voir annexe page 12)

## Exemple 3 : ajout au panier et gestion du stock

L'utilisateur peut ajouter un article à son panier. Une fois la commande validée, le stock du produit est automatiquement décrémenté dans la base de données. Ce processus est géré dans un contrôleur qui traite les données du panier et met à jour le stock via une requête SQL(voir annexe page 13 )

## Gestion de l'administration

L'administrateur a accès à une interface dédiée qui lui permet de :

- Ajouter une Création / Article (via un formulaire)
- Modifier ou supprimer des Créations ou Articles
- Gérer les commandes passées

Ces actions sont sécurisées et réservées uniquement aux comptes administrateurs. Chaque action agit sur la base de données via les modèles PHP.(voir annexe page 14)

## Conclusion

Grâce à l'intégration dynamique des données, mon site offre une expérience fluide et personnalisée à l'utilisateur, tout en permettant un gestion complète côté administrateur. L'utilisation du MVC m'a permis de structurer mon code de façon clair, modulable et évolutive.

## 2. Précisez les moyens utilisés :

PHP pour les méthode  
HTML pour la structure de l'affichage  
MySQL pour la base de donnée  
Visual Studio Code comme Editeur de code  
Wamp comme serveur local

## 3. Avec qui avez-vous travaillé ?

Pour effectuer cette partie j'ai travailler seul

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Elessar Bijouterie*

Chantier, atelier, service ► Création de la partie dynamique

Période d'exercice ► Du : 04/05/2025 au : 20/05/2025

## 5. Informations complémentaires (facultatif)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 1 ► Mettre en place une base de données relationnelle

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

#### Mettre en place une base de données relationnelle

L'objectif est de concevoir et mettre en place une base de données relationnelle permettant de stocker, structurer et interroger efficacement les informations nécessaires au bon fonctionnement de l'application web.

#### Compétence mobilisée

- **Identification des entités principale du projet**  
J'ai commencé par identifier les entités essentielles à la structure de mon application, telles que les utilisateurs, les créations, les catégories et les commandes. Cela m'a permis de bien comprendre les besoins en données du projet
- **Définition des relations entre ces entités**  
Une fois les entités définies, j'ai établi les liens logiques entre elles. Par exemple, une Création comporte au minimum une image et au maximum N image.
- **Choix adaptés des types de données**  
J'ai sélectionné les types de données les plus pertinents selon les champs (ex : VARCHAR pour les noms, INT pour les quantités, DECIMAL pour les prix). Cela garantit une base de données optimisée et cohérente
- **Mise en place des clés primaires et étrangères**  
Pour assurer l'unicité et la cohérence des données, chaque table possède une clé primaire. J'ai également mis en place des clés étrangères pour relier les entités entre elles.
- **Application des règles de normalisation**  
J'ai appliqué les principes de normalisation afin d'éviter les redondances inutiles. Chaque information est stockée à un seul endroit, ce qui facilite la maintenance et l'évolution de la base
- **Rédaction de requêtes SQL pour créer et interroger la base**  
J'ai rédigé des requêtes SQL pour créer les tables, insérer des données et faire des sélections. Cela m'a permis de manipuler efficacement les informations stockées dans la base

## Étapes de réalisation

- **Analyse fonctionnelle :**

J'ai identifié les différentes informations à gérer dans l'application, telles que les produits, les catégories, les utilisateurs, ou encore les commandes

- **Modélisation de données :**

J'ai réalisé un MCD (Modèle Conceptuel de Données) pour représenter les entités et leurs relations, puis je l'ai transformé en MLD (Modèle Logique de Données) en vue de la création des tables (voir annexe page15)

- **Création des tables en SQL :**

Dans le cadre de la modélisation de la base de données, j'ai procédé à la création d'une table permettant de stocker les informations relatives aux différentes créations proposées sur le site. Cette table est liée à d'autres tables grâce à des clés étrangères, ce qui permet de structurer les données de manière cohérente.

```
1 CREATE TABLE creations (  
2     id_Creation INT AUTO_INCREMENT PRIMARY KEY,  
3     Nom_Creation VARCHAR(100) NOT NULL,  
4     Description_Creation TEXT,  
5     Stock_Creation INT NOT NULL,  
6     Prix_Creation DECIMAL(10,2) NOT NULL,  
7     ID_Categorie INT,  
8     ID_Matiere INT,  
9     FOREIGN KEY (ID_Categorie) REFERENCES categorie_creation(id_Categorie),  
10    FOREIGN KEY (ID_Matiere) REFERENCES matiere(id_Matiere)  
11 );|
```

- **Insertion des données de test :**

Pour valider la structure de la base, j'ai inséré des données d'exemple :

```
1 INSERT INTO creations (Nom_Creation, Description_Creation, Stock_Creation, Prix_Creation,  
2 ID_Categorie, ID_Matiere)  
3 VALUES (  
4     'Sautoir pierre blanche et crâne doré',  
5     'Sautoir crâne doré et perles blanches. Ce sautoir au style audacieux mêle élégance et  
6 caractère. Composé de perles blanches mates, il est orné de plusieurs têtes de mort dorées, dont un  
7 crâne central imposant, accompagné d'un pompon noir pour une touche originale et raffinée. Un bijou  
8 au design fort, idéal pour affirmer sa personnalité avec style.',  
9     3,  
10    180,  
11    1,  
12    2  
13 );
```

- **Tests de cohérence et exploitation des données :**

J'ai effectué des requêtes de sélection avec des jointures pour m'assurer du bon fonctionnement des relation :

```
1 SELECT
2     creation.ID_Creation,
3     creation.Nom_Creation,
4     creation.Description_Creation,
5     creation.Stock_Creation,
6     creation.Prix_Creation,
7     categorie_creation.Nom_Categorie,
8     matiere.Nom_Matiere
9 FROM creation
10 INNER JOIN categorie_creation ON creation.ID_Categorie = categorie_creation.ID_Categorie
11 INNER JOIN matiere ON creation.ID_Matiere = matiere.ID_Matiere;
```

## Conclusion

Cette base de données relationnelle me permet de structurer mes informations de manière fiable, d'assurer l'intégrité des données, et d'exploiter facilement les données dans mon application web via des requêtes SQL. Elle est conçue de manière à être maintenable et évolutif en fonction des futurs besoins du projet

## 2. Précisez les moyens utilisés :

### Moyen et outils utilisés :

- **SGBD** : MySQL
- **Interface d'administration** : PhpMyAdmin
- **Langage** : SQL
- **Outils de modélisation** : Looping pour le MCD et MLD

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour cette partie

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Elessard Bijouterie*

Chantier, atelier, service ► Création de la base de donnée

Période d'exercice ► Du : *01/05/2025* au : *20/05/2025*

## 5. Informations complémentaires (facultatif)



## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

**Exemple n° 1** ► Développer des composants d'accès aux données SQL et NoSQL

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

#### Développer des composants d'accès aux données SQL

Dans le cadre de ce projet, j'ai mis en place des composants PHP permettant d'interagir efficacement avec la base de données. L'objectif était de structurer et sécuriser les échanges de données en utilisant des requêtes préparées, tout en centralisant la gestion de la connexion. Pour cela, j'ai utilisé une classe « Database » basée sur le design pattern Singleton, afin de garantir une connexion unique à la base tout au long de l'application. Ce choix permet de rendre le code plus maintenable ? clair et facilement réutilisable

#### Connexion à la base de données avec le pattern Singleton

La classe « Database » permet d'assurer une connexion unique à la base de données grâce au Singleton. Ce modèle empêche la création de multiples connexions inutiles et centralise les paramètres de connexion (hôte, nom de base, identifiants). En cas d'erreur, un message clair est retourné pour faciliter le débogage. Voici un aperçu de cette classe :

```
class Database {
    // Stocke une seule instance de la connexion PDO (Singleton)
    private static ?PDO $instance = null;

    // Constructeur privé pour empêcher l'instanciation directe de la classe
    private function __construct() {
        // Configuration des informations de connexion à la base de données
        $host = "localhost";
        $dbname = "elessardbijouterie";
        $username = "root";
        $password = "";

        try {
            // Création de l'instance PDO pour la connexion à la base de données
            self::$instance = new PDO("mysql:host=$host;dbname=$dbname;port=3306", $username, $password);
            // Configuration de PDO pour afficher les erreurs sous forme d'exceptions
            self::$instance->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        } catch (PDOException $e) {
            // En cas d'erreur de connexion, afficher un message et arrêter l'exécution du script
            die("Erreur de connexion : " . $e->getMessage());
        }
    }

    // Méthode statique pour récupérer l'instance unique de la connexion PDO
    public static function getInstance(): PDO {
        if (self::$instance === null) {
            new self(); // Appelle le constructeur privé pour initialiser l'instance PDO
        }
        return self::$instance;
    }
}
```

Cette classe est ensuite utilisée dans les autres fichier PHP grâce à l'appel :

```
$dbh = Database::getInstance();
```

## Requêtes SQL préparées

Pour éviter les injections SQL et renforcer la sécurité de l'application, j'ai utilisé des requêtes préparées avec la méthode « prepare() » de PDO. Cela permet d'envoyer des requêtes tout en liant les valeurs de manière sécurisée.

Exemple d'insertion d'un produit dans la table création :

```
public function insererCreation(): bool {
    $db = Database::getInstance();
    try {
        // Insertion dans la table creation (sans ID_Matiere)
        $stmt = $db->prepare("INSERT INTO creation (Nom_Creation, Description_Creation, Stock_Creation, Prix_Creation, ID_Categorie) VALUES (?, ?, ?, ?, ?)");
        $stmt->execute([
            $this->nom,
            $this->description,
            $this->stock,
            $this->prix,
            $this->id_categorie
        ]);
        $this->id = $db->lastInsertId();
        // Insertion des images dans image_creation
        foreach ($this->images as $chemin) {
            $stmtImg = $db->prepare("INSERT INTO image_creation (URL_Image, ID_Creation) VALUES (?, ?)");
            $stmtImg->execute([$chemin, $this->id]);
        }
        // Insertion dans la table de liaison matiere/creation
        if ($this->id_matiere) {
            $stmtMat = $db->prepare("INSERT INTO renfermer_creationmatiere (ID_Creation, ID_Matiere) VALUES (?, ?)");
            $stmtMat->execute([$this->id, $this->id_matiere]);
        }
    }
}
```

## Requêtes SQL avec jointures

Pour récupérer l'ensemble des créations avec leurs images associées, j'ai développé une méthode « getAllCreations() » qui exploite une jointure entre les tables « creation » et « image\_creation ». Grâce à l'instruction « GROUP\_CONCAT », il est possible de regrouper toutes les URL des images liées à une même création dans un seul champ. Ensuite, ces images sont transformées en tableau via « explode() » pour être intégrées dans un objet Création. Cela permet d'afficher plusieurs images par création tout en optimisant la requête SQL avec un seul appel à la base de données.

Voici la méthode utilisée :

```
public static function getAllCreations(): array {
    $db = Database::getInstance();
    $creations = [];

    try {
        $query = "SELECT creation.*, GROUP_CONCAT(image_creation.URL_Image) as images
        FROM creation
        LEFT JOIN image_creation ON creation.ID_Creation = image_creation.ID_Creation
        GROUP BY creation.ID_Creation";

        $stmt = $db->query($query);
        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

        foreach ($results as $row) {
            $images = $row['images'] ? explode(',', $row['images']) : [];
            $creations[] = new Creation([
                'id' => $row['ID_Creation'],
                'nom' => $row['Nom_Creation'],
                'description' => $row['Description_Creation'],
                'stock' => $row['Stock_Creation'],
                'prix' => $row['Prix_Creation'],
                'id_categorie' => $row['ID_Categorie'],
                'images' => $images
            ]);
        }

        return $creations;
    } catch (PDOException $e) {
        error_log("Erreur lors de la récupération des créations : " . $e->getMessage());
        return [];
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

## Conclusion

Grâce à la mise en place de composants d'accès aux données bien structurés et sécurisés, il est possible de gérer efficacement les interactions entre l'application et la base de données. L'utilisation de requêtes préparées, de jointures et de regroupements permet de récupérer des informations complètes tout en garantissant la performance, la lisibilité du code et évolutivité du projet. Ce travail constitue une base solide pour le bon fonctionnement de l'application et son enrichissement futur.

## 2. Précisez les moyens utilisés :

Pour cette partie j'ai utilisé le langage PHP et MySQL

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour cette partie

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Ellessar Bijouterie*

Chantier, atelier, service ► Développement de la partie back

Période d'exercice ► Du : *20/05/2025* au : *25/05/2025*

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

**Exemple n° 1** ► Développer des composants métier coté serveur

**1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :**

**2. Précisez les moyens utilisés :**

**3. Avec qui avez-vous travaillé ?**

**4. Contexte**

Nom de l'entreprise, organisme ou association ► Cliquez ici pour taper du texte.

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : Cliquez ici au : Cliquez ici

**5. Informations complémentaires (facultatif)**

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 1 ▶

Documenter le déploiement d'une application dynamique web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

2. Précisez les moyens utilisés :

3. Avec qui avez-vous travaillé ?

### 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service ▶ Cliquez ici pour taper du texte.

Période d'exercice ▶ Du : Cliquez ici au : Cliquez ici

5. Informations complémentaires (facultatif)

## DOSSIER PROFESSIONNEL (DP)

### Titres, diplômes, CQP, attestations de formation

*(facultatif)*

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

### Déclaration sur l'honneur

---

Je soussigné(e) [prénom et nom] [Cliquez ici pour taper du texte.](#) ,  
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis  
l'auteur(e) des réalisations jointes.

Fait à [Cliquez ici pour taper du texte.](#) le [Cliquez ici pour choisir une date](#)  
pour faire valoir ce que de droit.

Signature :

### Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.

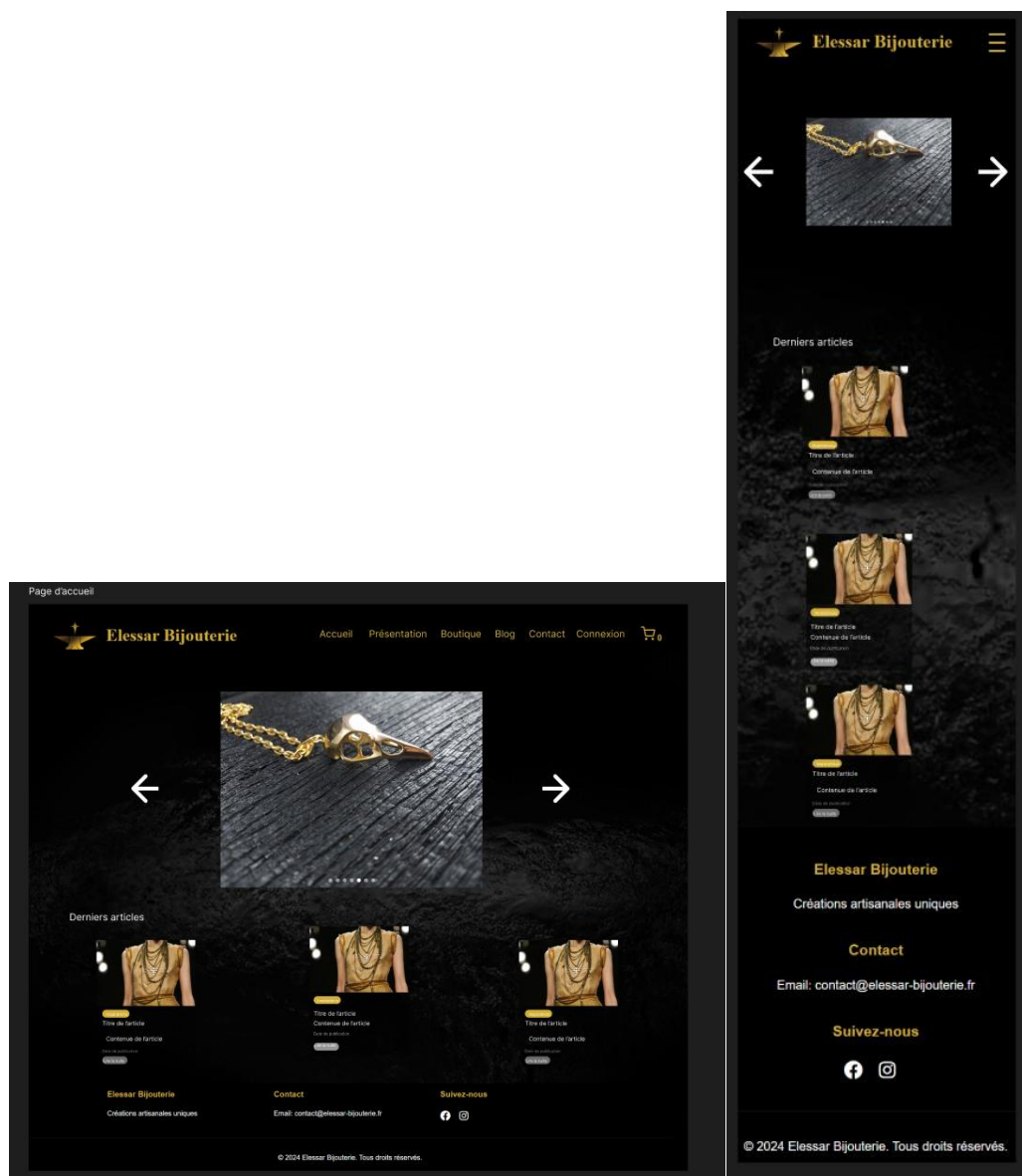


# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

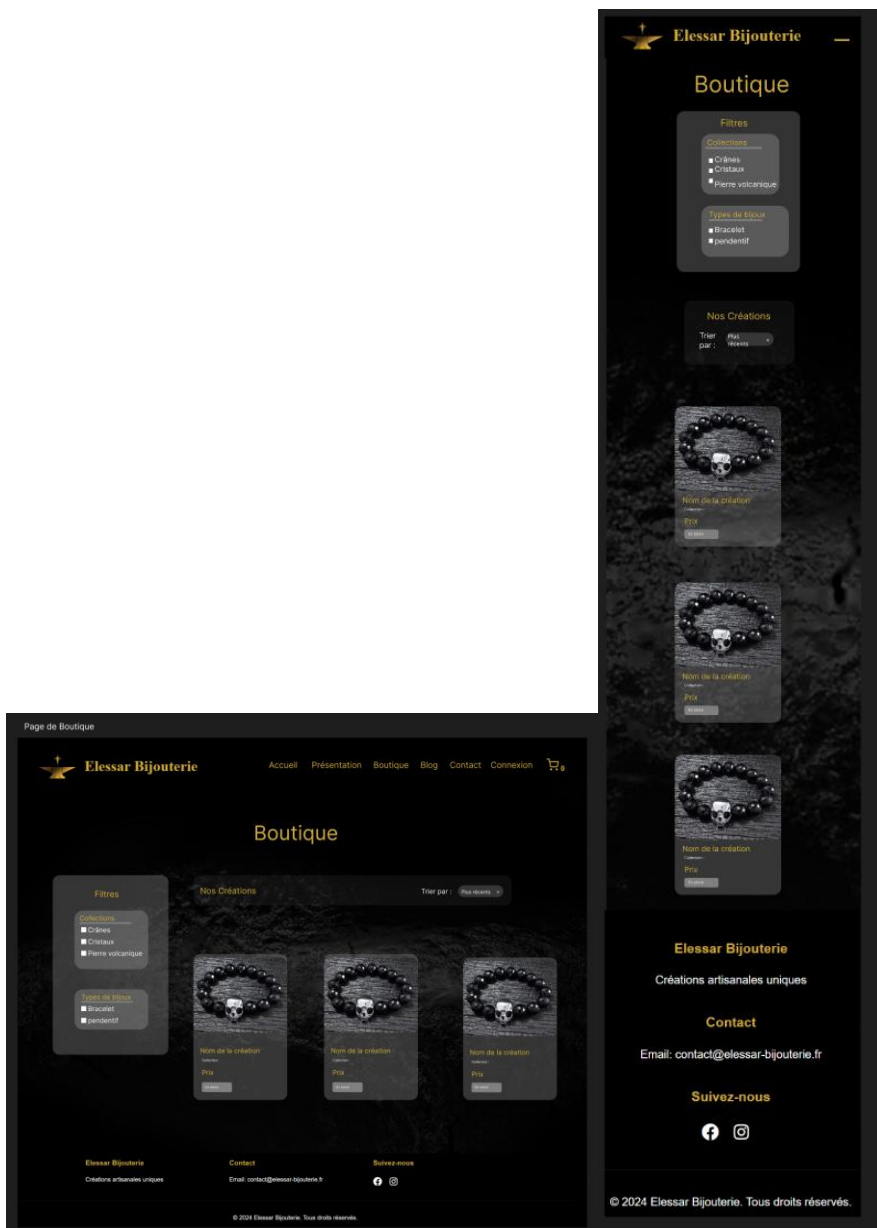
(Si le RC le prévoit)

Voici la maquette de la page d'accueil : (Version desktop et mobile)



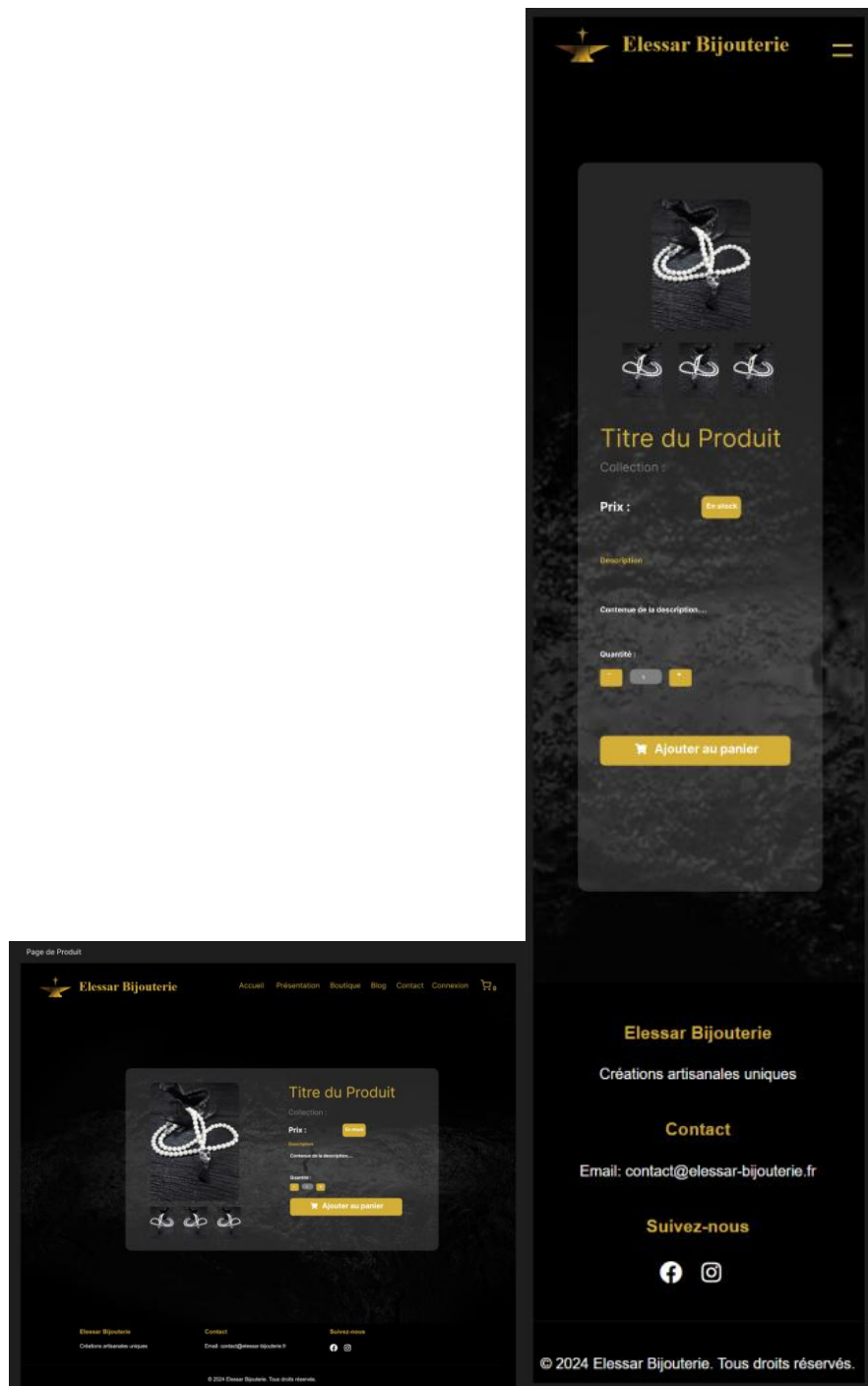
# DOSSIER PROFESSIONNEL (DP)

Voici la maquette de la page Boutique : (Version desktop et mobile)



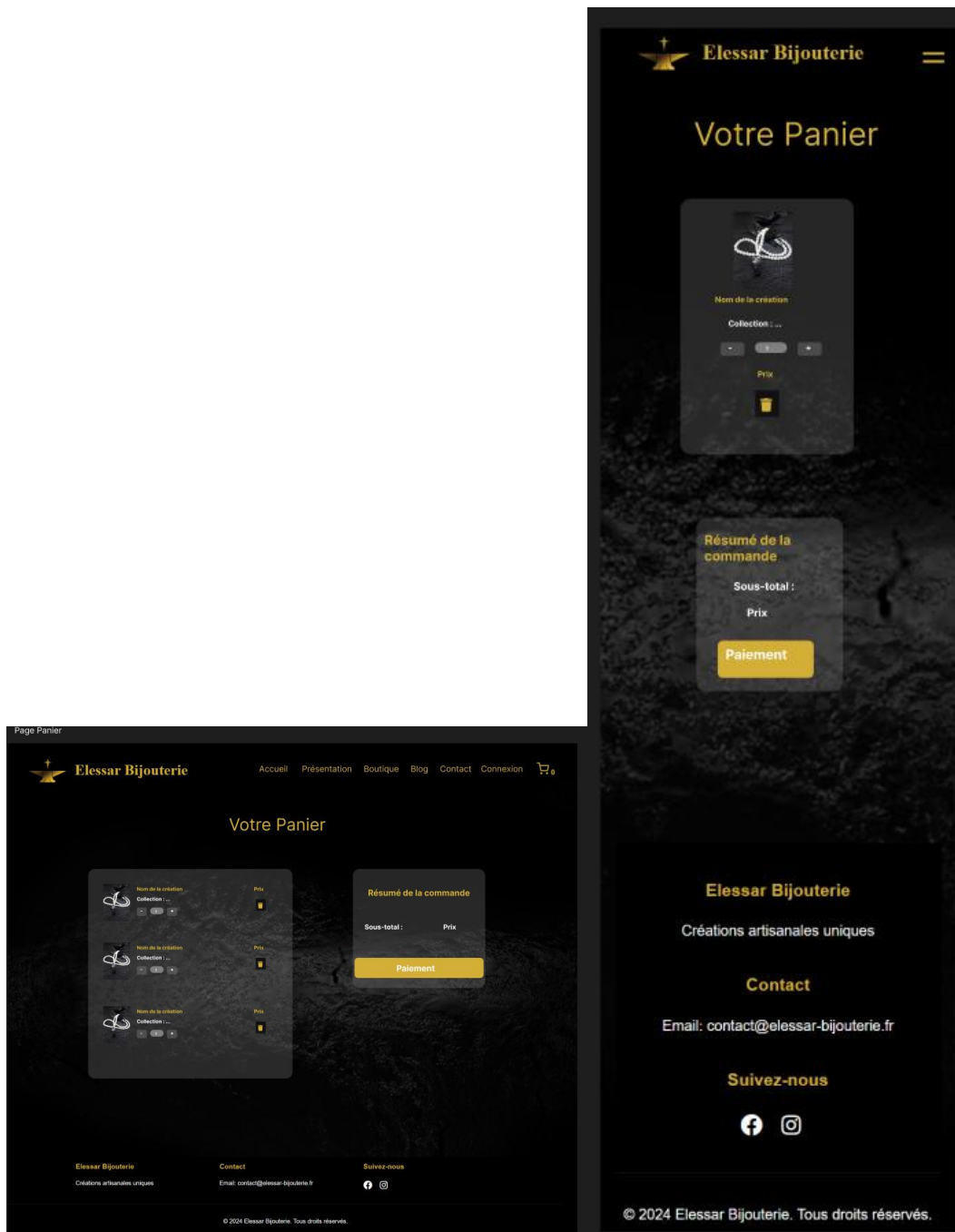
# DOSSIER PROFESSIONNEL (DP)

Voici la maquette de la page Produit : (Version desktop et mobile)



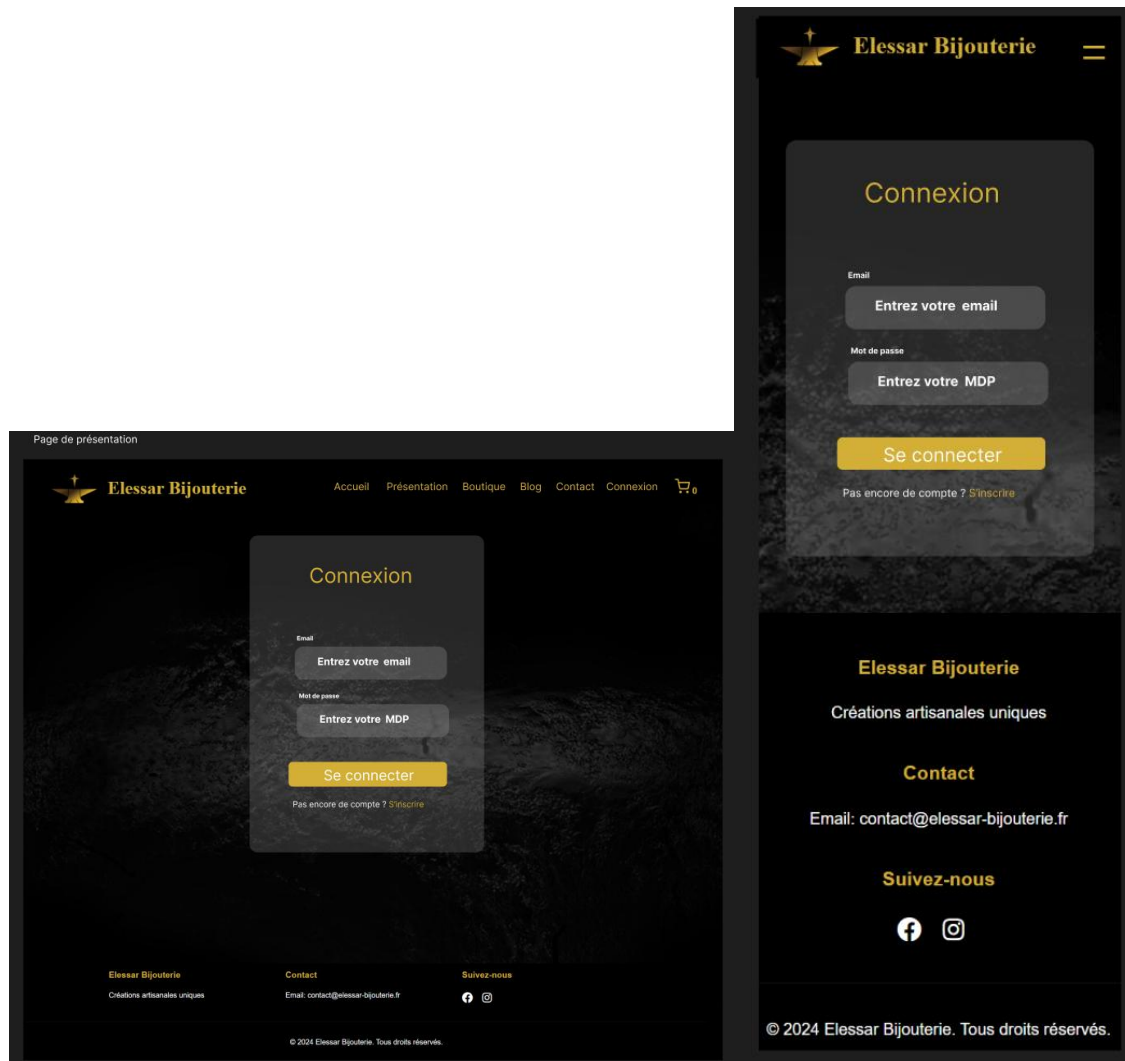
# DOSSIER PROFESSIONNEL (DP)

Voici la maquette de la page Panier : (Version desktop et mobile)



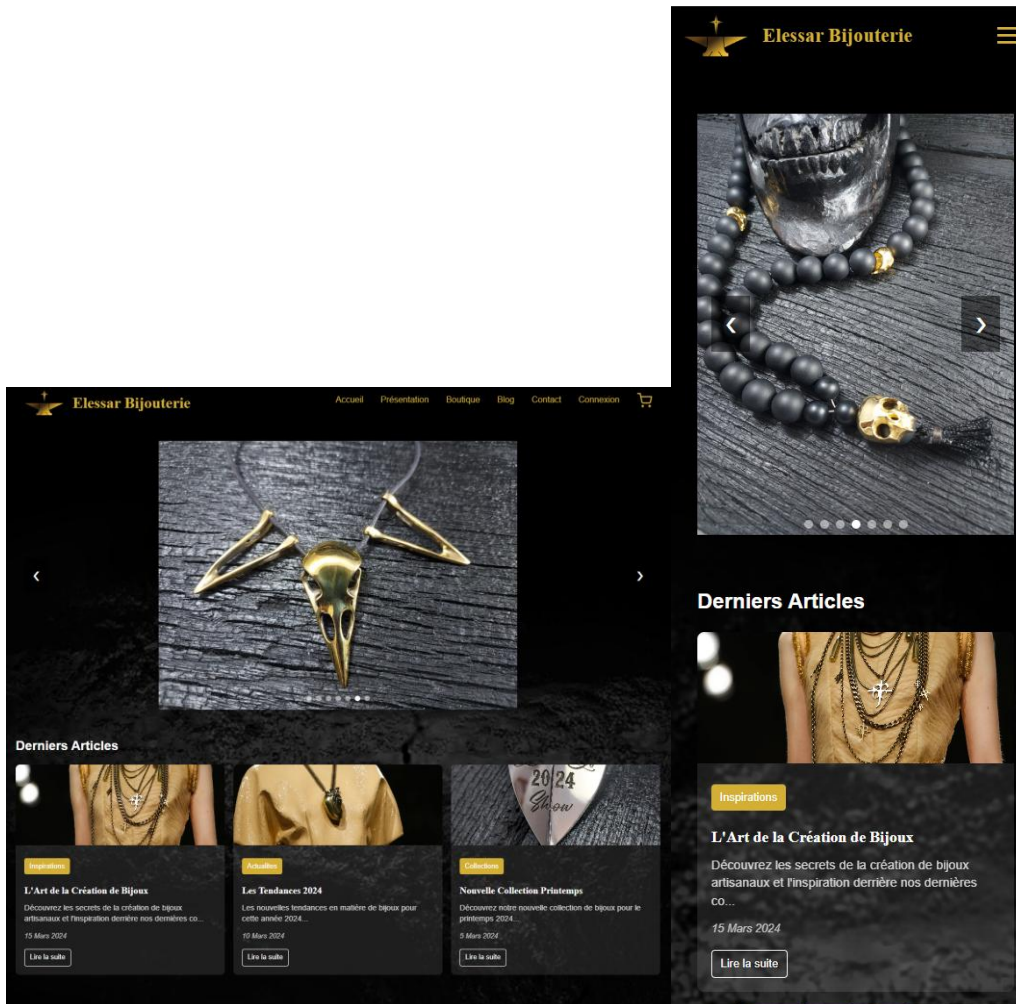
# DOSSIER PROFESSIONNEL (DP)

Voici la maquette de la page Connexion : (Version desktop et mobile)



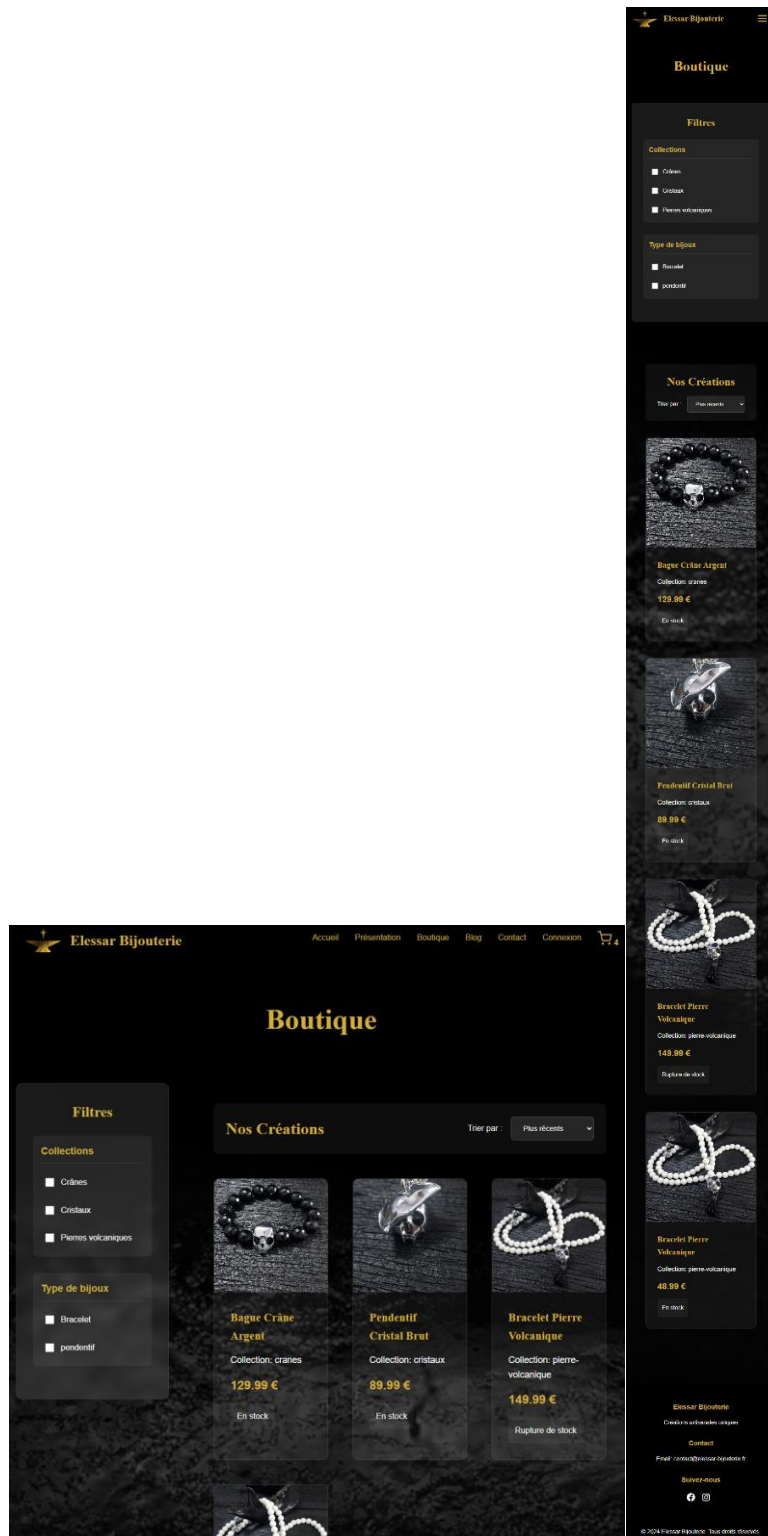
# DOSSIER PROFESSIONNEL (DP)

Voici la page d'accueil en statique : (version Desktop et mobile)



# DOSSIER PROFESSIONNEL (DP)

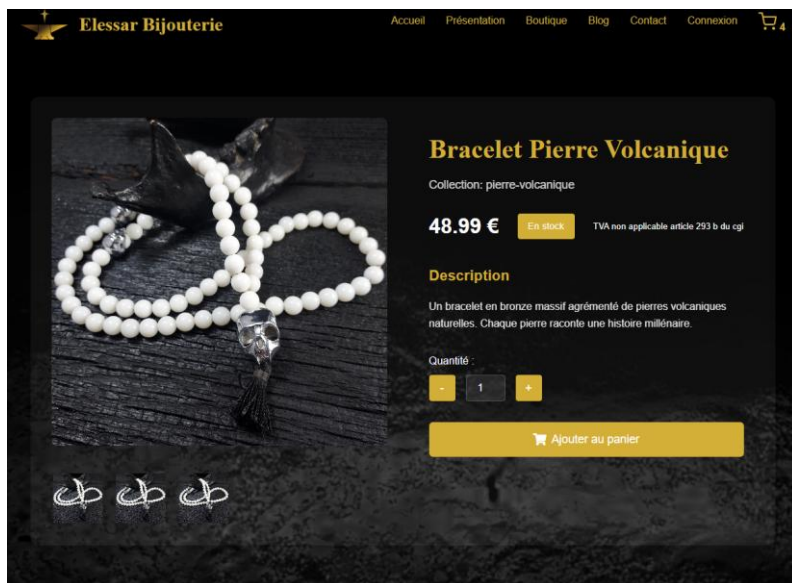
Voici la page boutique en statique : (version Desktop et mobile)





# DOSSIER PROFESSIONNEL (DP)

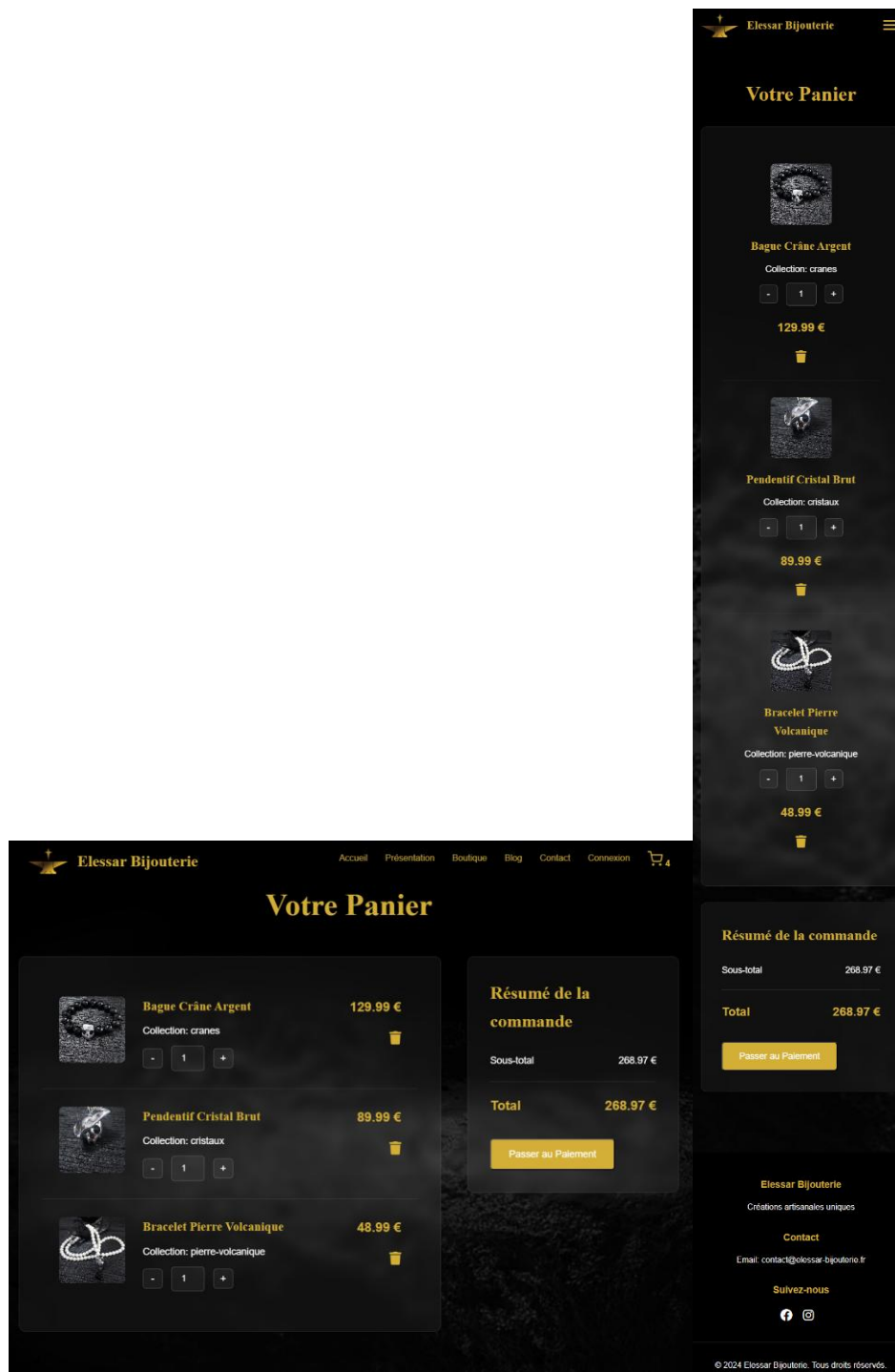
Voici la page produit en statique : (version Desktop et mobile)





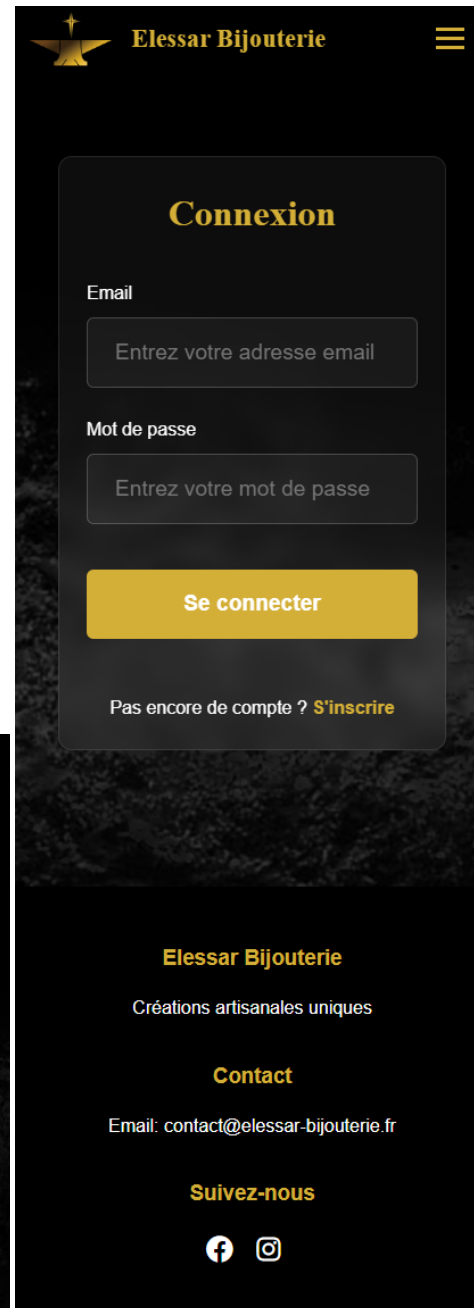
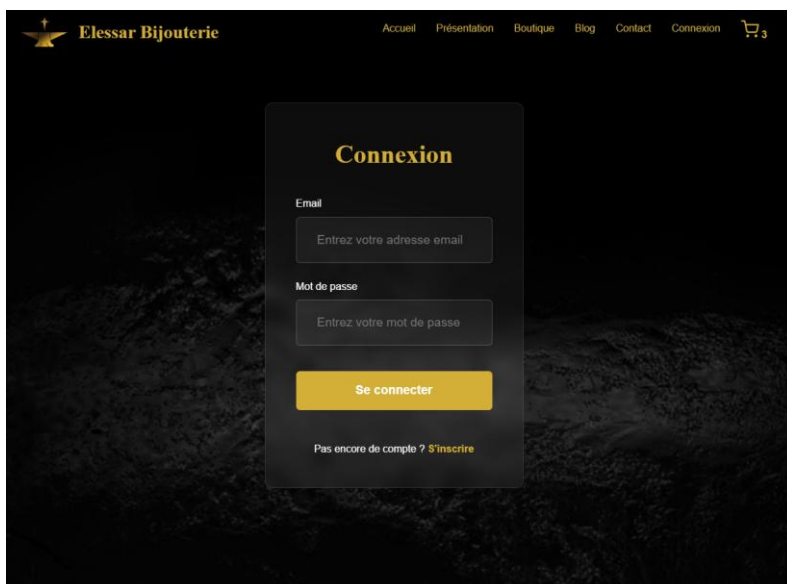
# DOSSIER PROFESSIONNEL (DP)

Voici la page panier en statique : (version Desktop et mobile)

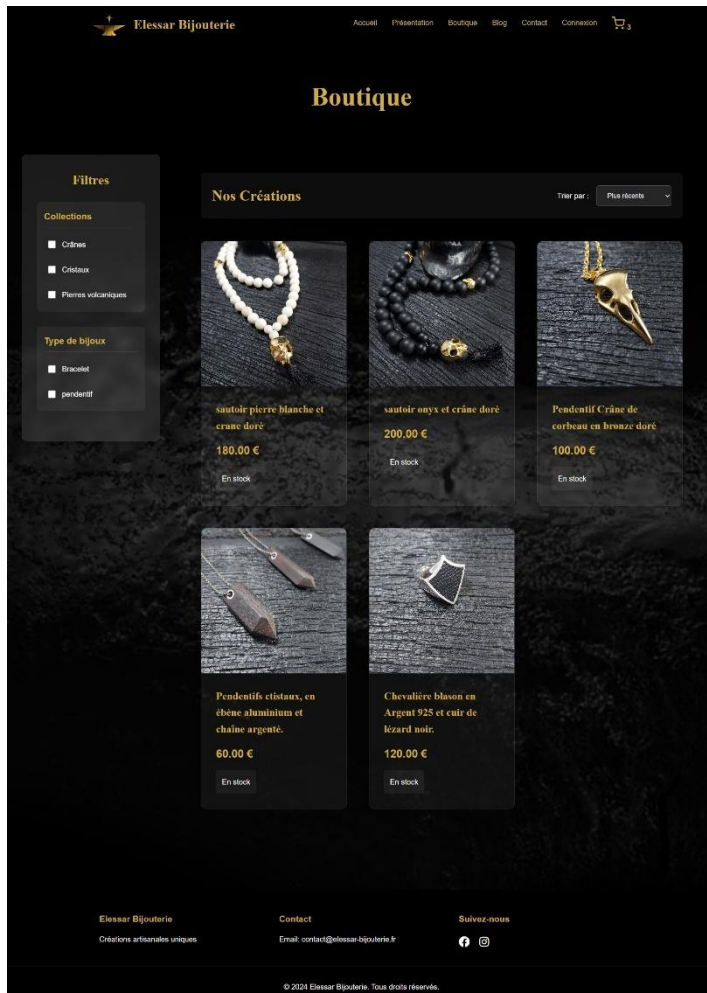


# DOSSIER PROFESSIONNEL (DP)

Voici la page connexion en statique : (version Desktop et mobile)



Voici la capture d'écran de la page Boutique avec les créations affichées dynamiquement :



Ensuite voici ma boucle ForEach qui affiche les produits dans la page Boutique :

```
<div class="products-list" id="products-list">
  <?php foreach (
    $creations as $creation): ?>
    <article class="product-card">
      <div class="product-image">
        getNom()); ?>" />
        <div class="product-overlay">
          <button class="btn-view" onclick="window.location.href='index.php?page=Produits&id=<?php echo $creation->getId(); ?>'>
            Voir détails
          </button>
          <button class="btn-add-cart" data-id="<?php echo $creation->getId(); ?>">
            Ajouter au panier
          </button>
        </div>
      </div>
      <div class="product-info">
        <h3 class="product-title"><?php echo htmlspecialchars($creation->getNom()); ?></h3>
        <p class="product-price"><?php echo number_format($creation->getPrix(), 2); ?> €</p>
        <p class="product-status">
          <?php echo $creation->getStock() > 0 ? 'En stock' : 'Rupture de stock'; ?>
        </p>
      </div>
    </article>
  <?php endforeach; ?>
</div>
```

Voici la capture d'écran de la page Produit qui affiche les informations du produit stocker en base de données :



Ensuite voici un extrait du code qui récupère les données via l'ID :

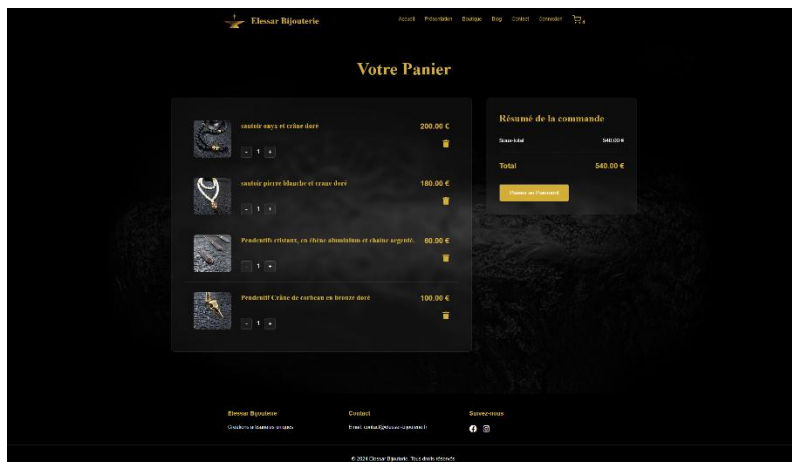
Ceci est mon contrôleur il vérifie qu'il y a bien un ID dans l'URL si c'est le cas il appelle la méthode de l'objet création getCreationById

```
if (isset($_GET['id'])) {  
    require_once __DIR__ . '/../models/Creation.php';  
    $creation = Creation::getCreationById($_GET['id']);  
}
```

Voici la méthode getCreationById

```
public static function getCreationById($id): ?self {  
    $db = Database::getInstance();  
    try {  
        $query = "SELECT creation.*, GROUP_CONCAT(image_creation.URL_Image) as images  
        FROM creation  
        LEFT JOIN image_creation ON creation.ID_Creation = image_creation.ID_Creation  
        WHERE creation.ID_Creation = ? GROUP BY creation.ID_Creation";  
        $stmt = $db->prepare($query);  
        $stmt->execute([$id]);  
        $row = $stmt->fetch(PDO::FETCH_ASSOC);  
        if ($row) {  
            $images = $row['images'] ? explode(',', $row['images']) : [];  
            return new Creation([  
                'id' => $row['ID_Creation'],  
                'nom' => $row['Nom_creation'],  
                'description' => $row['Description_creation'],  
                'stock' => $row['Stock_creation'],  
                'prix' => $row['Prix_creation'],  
                'id_categorie' => $row['ID_Categorie'],  
                'images' => $images  
            ]);  
        }  
        return null;  
    } catch (PDOException $e) {  
        error_log("Erreur lors de la récupération de la création : " . $e->getMessage());  
        return null;  
    }  
}
```

Voici la page Panier rempli dynamiquement :



Voici le contrôleur qui permet à l'utilisateur de modifier la quantité, de supprimer la quantité de l'article

```
// Gestion des actions du panier
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['add_to_cart'])) {
        $panier::addToCart(
            $_POST['id'],
            $_POST['nom'],
            $_POST['prix'],
            $_POST['image'],
            $_POST['quantite'] ?? 1
        );
    } elseif (isset($_POST['remove_from_cart'])) {
        $panier::removeFromCart($_POST['id']);
    } elseif (isset($_POST['update_quantity'])) {
        $panier::updateQuantity($_POST['id'], $_POST['quantite'] ?? 1);
    }
}
```

On peut retrouver ici les 2 méthodes qui permettent de modifier la quantité et supprimer l'article du panier

```
public static function removeFromCart($id) {
    if (session_status() === PHP_SESSION_NONE) session_start();
    if (!isset($_SESSION['panier'])) return;
    $_SESSION['panier'] = array_filter($_SESSION['panier'], function($item) use ($id) {
        return $item['id'] !== $id;
    });
}

public static function updateQuantity($id, $quantite) {
    if (session_status() === PHP_SESSION_NONE) session_start();
    if (!isset($_SESSION['panier'])) return;
    foreach ($_SESSION['panier'] as $item) {
        if ($item['id'] === $id) {
            $item['quantite'] = $quantite;
            break;
        }
    }
}
```

Voici le formulaire d'ajout de création uniquement accessible par l'administrateur :

The screenshot shows a web application interface for 'Elessar Bijouterie'. On the left is a sidebar menu with options like 'Tableau de bord', 'Formulaire de Création', 'Formulaire d'Article', 'Catégories', 'Commandes', 'Utilisateurs', and 'Paramètres'. The main area is titled 'Création d'un nouveau produit'. It contains a form with the following sections:

- Informations générales**:
  - 'Nom du produit': A text input field.
  - 'Description': A large text area.
- Détails du produit**:
  - 'Prix': A text input field with a Euro symbol (€).
  - 'Stock': A text input field with the value '0'.
  - 'Catégorie': A dropdown menu with the text 'Sélectionnez une catégorie'.
  - 'Matériau': A dropdown menu with 'Or' selected.
- Image du produit**:
  - Text: 'Images (JPG, PNG, GIF, max 5Mo chacune):'.
  - Buttons: 'Sélectionner fichiers' and 'Aucun fichier choisi'.
  - Text: 'Formats acceptés : JPG, PNG, GIF. Taille maximale : 5MB'.

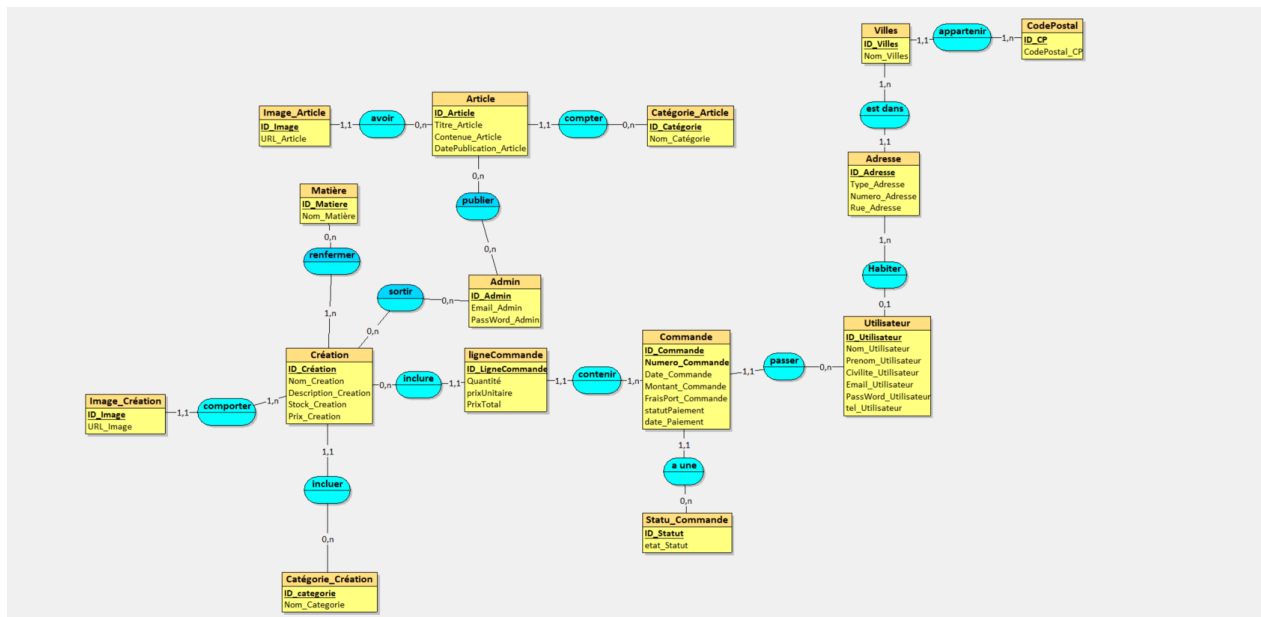
At the bottom of the form are two buttons: 'Créer le produit' and 'Annuler'.

Voici la methode pour ajouter une Création

```
public function insererCreation(): bool {
    $db = Database::getInstance();
    try {
        // Insertion dans la table creation (sans ID_Matiere)
        $stmt = $db->prepare("INSERT INTO creation (Nom_Creation, Description_Creation, Stock_Creation, Prix_Creation, ID_Categorie) VALUES (?, ?, ?, ?, ?)");
        $stmt->execute([
            $this->nom,
            $this->description,
            $this->stock,
            $this->prix,
            $this->id_categorie
        ]);
        $this->id = $db->lastInsertId();
        // Insertion des images dans image_creation
        foreach ($this->images as $chemin) {
            $stmtImg = $db->prepare("INSERT INTO image_creation (URL_Image, ID_Creation) VALUES (?, ?)");
            $stmtImg->execute([$chemin, $this->id]);
        }
        // Insertion dans la table de liaison matiere/creation
        if ($this->id_matiere) {
            $stmtMat = $db->prepare("INSERT INTO renfermer_creationmatiere (ID_Creation, ID_Matiere) VALUES (?, ?)");
            $stmtMat->execute([$this->id, $this->id_matiere]);
        }
        return true;
    } catch (PDOException $e) {
        // Pour debug : echo $e->getMessage();
        return false;
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

Voici mon MCD :



Voici mon MLD :

