

# Titre développeur Web & Web mobile

---



---

**Dossier de Projet :**

**Réalisation d'une boutique en ligne.**

**Adrien COQUILLAT**

# Sommaire

- I. Le projet
  - 1 Compétences du référentiel couvertes par le projet
  - 2 Résumé
  - 3 Présentation personnelle
  - 4 Présentation formation
  - 5 Périmètre du projet
  - 6 Arborescence du projet
  
- II. Les spécifications fonctionnelles
  - 1.Espace utilisateur
    - 1.1 Inscription
    - 1.2 Authentification
    - 1.3 Fiche produit
    - 1.4 Panier client
    - 1.5 Fonctionnalité de recherche produit
    - 1.6 Historique des commandes
    - 1.7 Moyen de paiement
  - 2.Espace administrateur
    - 2.1 Back office
    - 2.2 Gestion des produits
    - 2.3 Gestion des utilisateurs
  
- III. Les spécifications techniques
  - 1.Réalisation : commencement projet
    - 1.1 Charte graphique
    - 1.2 Maquette
    - 1.3 Base de données
  - 2.Le Code
    - 2.1 Outil
    - 2.2 Architecture
    - 2.3 Rooter
    - 2.4 Technos utilisées Front-end
    - 2.5 Technos utilisées Back-end
    - 2.5 Sécurité

# I. LE PROJET

## 1. Compétences du référentiel couvertes par le projet

Pour l'activité 1, **"Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité."** :

- Maquetter une application
- Réaliser une interface utilisateur web ou mobile statique et adaptable
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Pour l'activité 2, **"Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité."** :

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

## 2. Résumé

Développez une boutique en ligne. Il faut au préalable concevoir la base de données et la maquette de l'application. Ce projet regroupe un nombre conséquent de compétences indispensables à l'obtention de votre titre. Ce projet a été réalisé avec Robin ARBONA . Le but était de réaliser une boutique en ligne, et de faire comme si nous avions un client. Nous avons dans un premier temps choisi le domaine dans lequel nous voulions faire notre boutique : la vente d'objets sexuels. Cela n'enlève pas le temps et le sérieux que nous avons mis sur ce projet qui a été conséquent pour mon apprentissage.

Nous avons réalisé une charte graphique, ce qui nous a permis de suivre une ligne de conduite et de ne pas en sortir.

Nous avons 1 mois pour travailler dessus. Nous avons utilisé principalement du PHP comme langage de programmation, ainsi que ses fonctionnalités diverses comme PDO. Ce projet a été très enrichissant pour notre apprentissage.

Nous nous sommes aussi intéressés à la sécurité ainsi qu'à inclure un système de paiement. Nous avons un délai d'un mois pour réaliser ce projet. Nous avons mis en place un système de routeur, avec un design pattern : le MVC.

Nous avons réalisé une boutique de e-commerce, permettant à un administrateur de pouvoir changer sa page d'accueil sans forcément avoir besoin de connaissance en code . Nous avons intégré une partie paiement avec Stripe. Pour l'organisation des produits, nous avons 3 catégories d'objets, ainsi que 2 sous catégories. Un système de recherche d'objets a été mis en place avec la méthode AJAX.

### **3. Présentation personnelle**

Je suis Adrien COQUILLAT , j'ai actuellement 30 ans et je suis en reconversion professionnelle dans le web développement et web mobile. J'ai toujours été passionné par la technologie et j'ai découvert les bases du code grâce à la Plateforme sur Marseille.

Le côté artistique et la liberté du front m'a plu et j'ai découvert ensuite la force des machines et toutes les possibilités qui s'offraient à nous en tant que développeur.

### **4. Présentation formation**

La Coding School est une formation web qui s'adresse à tous ceux qui souhaitent s'ouvrir les portes des métiers du numérique. Le modèle pédagogique unique de l'École la Plateforme, membre du label GEN, Grande École du Numérique, s'adapte aux besoins de chacun. Les évaluations se font par des contrôles continus sur des projets webs concrets réalisés seuls ou en groupes. L'intégration professionnelle est réalisée au travers de projets professionnels tutorés, portés par des entreprises. La coding school revendique une pédagogie active et inductive centrée sur l'apprenant et orientée projet. La formation ( 1200 h/an) s'effectue en présentiel, dans un lieu dédié spécifiquement pour catalyser l'apprentissage au 8 rue d'Hozier 13002 Marseille. Le programme vise à dispenser les connaissances et compétences nécessaires pour acquérir l'obtention du titre professionnel : technologies du web, maquettage d'applications, modélisation de bases de données, développement de sites web statiques, dynamiques et responsives,

déploiement de CMS, base d'algorithme, projet professionnel. Cette année de formation est divisée en 4 Units destinées à apprendre de nouvelles technologies :

- Unit 1 / HTML-CSS
- Unit 2 / PHP MySql
- Unit 3 / Javascript
- Unit 4 / Projet Pro.

J'ai ainsi appris à coder avec les technologies suivantes : HTML / CSS, PHP, MySql, Javascript, JQuery, Ajax, Codeigniter et utilisé des logiciels collaboratifs comme github ou figma.

## **5. Périmètre du projet**

Ce projet est fait pour un navigateur Web et est adaptable sur mobile. Le but était de rendre accessible et multiplateforme notre application. Cela s'adresse aux personnes qui ont envie de s'évader et de découvrir de nouvelles sensations, sans genre sexuel attribué.

## **6. Arborescence du projet**

Nous avons donc les pages suivantes pour notre site ;

- ❖ Une page d'accueil
- ❖ Une page d'inscription
- ❖ Une page avec tous les produits en ventes
- ❖ 3 pages pour les 3 catégories d'objets que nous vendons
- ❖ Une page descriptif pour chaque objet
- ❖ Une page profil qui donne accès (sous forme d'onglet) :
  - Panier
  - Modification du profil
  - Historique des commandes
- ❖ Un panel Admin
- ❖ Une page de paiement
- ❖ Une page de validation de paiement

## II. Les spécifications fonctionnelles

### 1 Espace utilisateur

#### 1.1 Inscription

Pour pouvoir s'inscrire, nous avons demandé de remplir un formulaire avec les informations suivantes :

- Login qui ne pourra pas être changé
- Mot de passe
- Adresse
- Mail

Nous avons stocké ces informations dans une base de données à l'aide des requêtes SQL.

#### 1.2 Authentification

Pour s'authentifier, l'utilisateur a besoin d'un identifiant ( le login ) et d'un mot de passe. Nous vérifions à l'aide d'une requête SQL, si l'utilisateur est présent dans la base de données. Nous utilisons les SESSIONS pour permettre la récupération des données, et permettre de savoir si l'utilisateur est bien connecté.

#### 1.3 Fiche produit

Nous avons organisé les produits avec 3 catégories principales. Pour chaque produit, nous avons une page spécifique qui permet d'avoir une description, un prix, la quantité que nous voulons mettre au panier si l'utilisateur souhaite l'acheter.

#### 1.4 Panier client

Un utilisateur non authentifié peut avoir un panier : celui va être créé avec un identifiant temporaire, mis en base de données. Lorsque l'utilisateur voudra passer au paiement, il devra alors s'inscrire ou se connecter. Le panier sera donc transféré au nouvel utilisateur.

#### 1.5 Fonctionnalité de recherche produit

Pour la recherche de produit, nous avons mis en place un système d'autocomplétion, qui permettra à l'utilisateur de voir si le produit qu'il tape existe, avant d'envoyer la recherche, sans rafraîchissement.

#### 1.6 Profil / Historique des commandes

Pour avoir accès à l'historique des commandes, l'utilisateur aura accès à une page profil :

Il pourra avoir accès à l'aide de 3 onglets à ses historiques de paiement, son panier et la page permettant de modifier ses identifications.

#### 1.7 Moyen de paiement

Pour le système de paiement, nous avons utilisé Stripe. Pour permettre d'utiliser la librairie stripe, nous avons utilisé Composer qui permet d'installer et mettre à jour des librairies.

## 2. Espace administrateur

### 2.1 Back office

Nous avons mis en place un système de back office. Cela permettra aux administrateurs du site, de pouvoir changer le contenu de leur site sans forcément avoir de connaissance en code, comme un CMS.

### 2.2 Dashboard

Depuis le panel administrateur, l'admin aura accès au dashboard. Le dashboard se compose de toutes les informations présentes dans notre base de données et celui-ci pourra donc modifier l'entièreté des tables, s'il le souhaite.

Il pourra donner à un utilisateur l'accès au dashboard s'il le souhaite, et le nouvel admin pourra lui aussi modifier le prix d'un produit ou encore en supprimer.

### 2.3 Image handler

Permet de stocker des images en base de données à l'aide d'un formulaire.

### 2.4 Page d'accueil

L'administrateur pourra modifier le contenu de sa page d'accueil comme par exemple la vitesse de son carrousel ou les produits phares, jusqu'au titre du site, juste à l'aide de formulaire qu'il doit remplir.

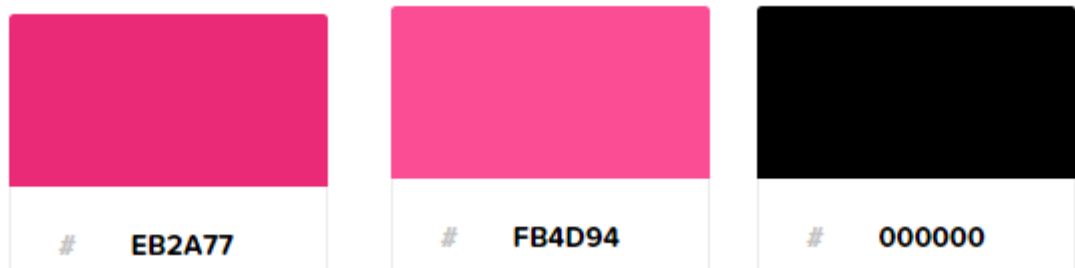


### III. Les spécifications techniques

#### 1. Réalisation : Commencement projet

##### 1.1 Charte graphique

Nous avons réalisé une charte graphique en nous inspirant des sites qui existaient déjà. Nous avons voulu partir sur 3 couleurs différentes :



Pour le logo, j'ai utilisé une tablette graphique ainsi qu'un logiciel de photo Affinity photo. J'ai travaillé sur une tablette graphique de la gamme gaomon. Pour les couleurs, j'ai respecté le design du site.



AFFINITY  
PHOTO



##### 1.2 Maquettage

Pour la mise en page, nous avons utilisé le logiciel web figma. Il nous a permis d'avoir un avant goût du site et d'être certains de nos choix. Nous pouvons voir que pour passer du wireframe au site réalisé, le résultat n'est pas si différent. L'avantage de faire ces étapes nous a permis de nous mettre sur une ligne à suivre et de gagner du temps par la suite pour l'organisation et la réalisation du site.



### 1.3 Base de données

Pour la réalisation de la base de données relationnelles, nous avons utilisé la source suivante : <https://ineumann.developpez.com/tutoriels/merise/initiation-merise/>

Nous avons utilisé la méthode merise. Dans un premier temps, nous avons établi **les règles de gestion métier** :

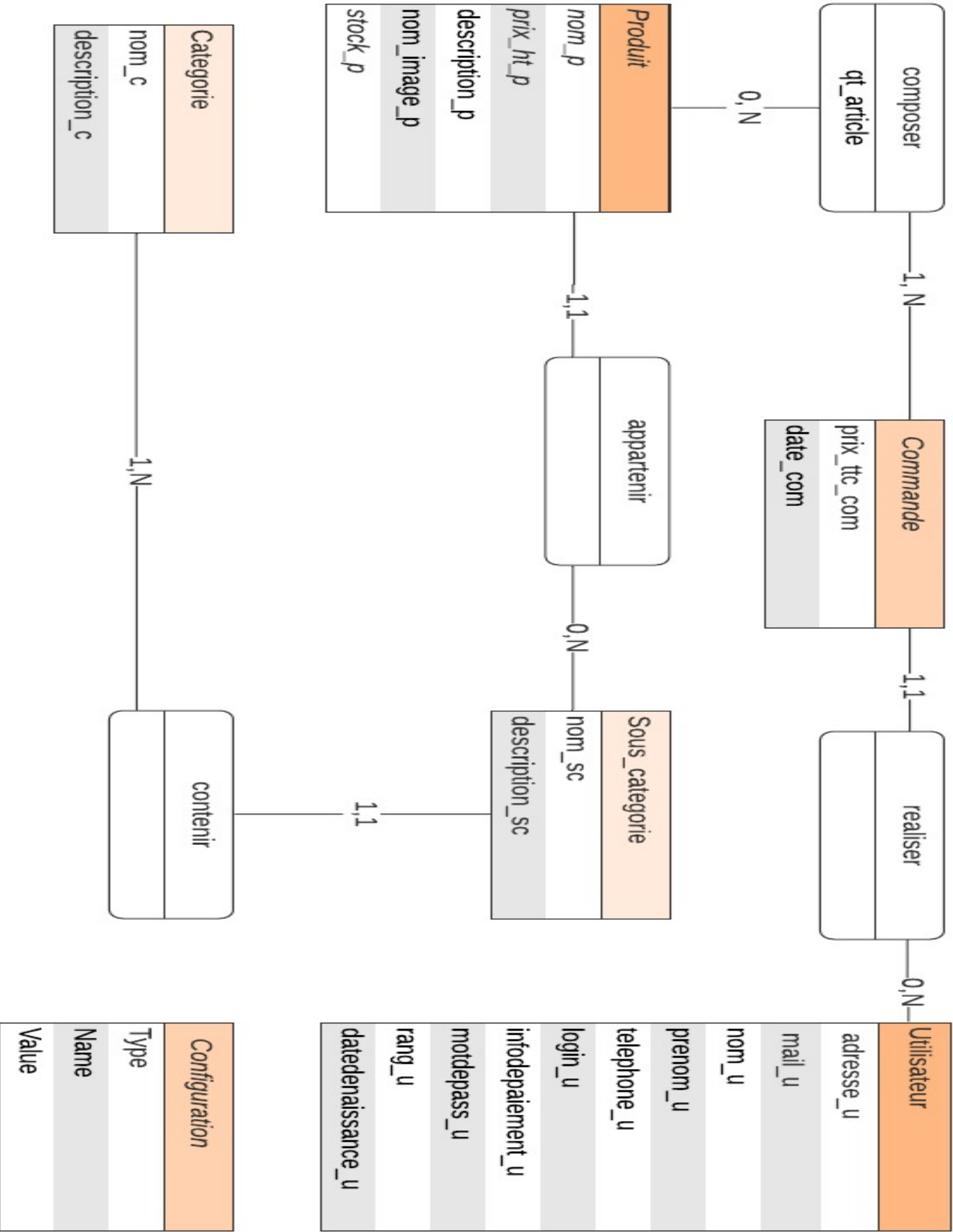
- un produit à un nom, une image, un prix, et une description
- un panier est propre à utilisateur
- les produits sont classés par catégories et sous catégories

Nous avons par la suite réalisé le dictionnaire des données. Cela consiste à donner le type de données, "N" lorsque la donnée est un nombre (entiers ou réels), "A" lorsqu'elle est composée de caractères alphabétiques, "AN" pour alphanumérique, elle est à la fois composée de caractères alphabétiques et numériques. Date quand c'est une date, et booléen pour true ou false. Cela nous a permis de gagner du temps lors de l'implémentation dans PHPmyAdmin, notre base de données. La taille correspond au nombre de caractères autorisés. Bien vérifier la date, car le format peut bloquer.

Code mnémonique	Désignation	Type *	Taille	Remarque
id_p	Identification numérique du produit	N		
nom_p	Nom du produit	A	255	
prix_ht_p	Prix unitaire hors taxe du produit	N	10	
stock_p	Nombre de produit en stock	N	10	
description_p	Description du produit	AN	1000	
nom_image_p	Nom du fichier image du produit	AN	255	
id_com	Identification de la commande	N		
prix_ttc_com	Prix ttc de la commande	N	10	
date_com	Date de la commande	Date		AAAA-MM-JJ

id_u	Identification numérique de l'utilisateur	N		
adresse_u	Adresse physique de l'utilisateur	AN	255	
mail_u	Adresse email de l'utilisateur	AN	255	
login_u	Login utilisateur	AN	50	
nom_u	Nom utilisateur	A	50	
prenom_u	Prénom de l'utilisateur	A	50	
telephone_u	Téléphone de l'utilisateur	N	10	
infodepaiement_u	Information de paiement de l'utilisateur	AN	50	
motdepasse_u	Mot de passe de l'utilisateur	AN	100	
rang_u	Rang de l'utilisateur	AN	50	
datedenaissance_u	Date de naissance de l'utilisateur	Date		AAAA-MM-JJ
id_c	Id de la catégorie	N		
nom_c	Nom de la catégorie	A	50	
description_c	Description de la catégorie	AN	1000	
id_sc	Id de la sous-catégorie	N		
nom_sc	Nom de la sous-catégorie	A	50	
description_sc	Description de la sous-catégorie	AN	1000	

Nous avons ensuite élaboré un MCD. (Modèle Conceptuel des Données). Le MCD est une représentation graphique de haut niveau qui permet facilement et simplement de comprendre comment les différents éléments sont liés entre eux à l'aide de diagrammes codifiés. Il permet de voir nos relations entre nos tables, et si elles peuvent avoir plusieurs données.



commande	produit	utilisateur	composer	sous_categorie	categorie	configuration
id com	id p	id u	id comp	id sc	id c	id
prix ttc com	nom p	login u	id com	nom sc	nom c	type
date com	prix ht p	adresse u	id p	description sc	description c	name
id u	description p	mail u	qt_article	id c		value
	nom image p	nom u				
	stock p	pre nom u				
	id sc	telephone u				
		infodepalement u				
		motdepasse u				
		rang u				
		datedenaissance u				

Les tables que nous avons créé pour la réalisation de notre site :

- Commande -> avoir les informations d'une commande d'un utilisateur
- Produit -> informations d'un produit, image, description, prix ...
- Utilisateur -> informations de l'utilisateur
- Composer -> avoir la quantité de l'id de la commande et du produit
- Sous-catégorie -> trier les produits par sous catégories, qui appartiennent à une catégorie
- Catégorie -> ranger les produits dans des catégories
- Configuration -> permettre de stocker des informations par rapport aux types de données, et lui attribuer une valeur. Permet de transférer le panier à un nouvel inscrit

## 2. Le Code

### 2.1 Outil

J'ai travaillé sur visual code studio sur un ordinateur avec windows 10 comme système d'exploitation et on a pu travailler en équipe avec Github.

Nous avons choisi d'utiliser le langage back PHP et la P.O.O dans du MVC.

### 2.2 Architecture

Pour l'architecture de notre application, nous avons décidé d'utiliser un pattern design : le MVC. Il est très répandu et a permis de répondre à certaines problématiques d'organisation. Il fait partie des conceptions les plus répandues et les plus utilisées. Cela m'a permis de bien différencier les parties de mes codes, qui pouvaient faire de très longues pages. Nous l'avons un peu remanié pour éviter d'avoir trop de lignes. Nous avons rajouté un dossier library et Entity.

Un modèle est une donnée utilisée par un programme. Cela peut être une base de données, un fichier, ou un objet simple, tel qu'un icône ou un personnage dans un jeu vidéo.

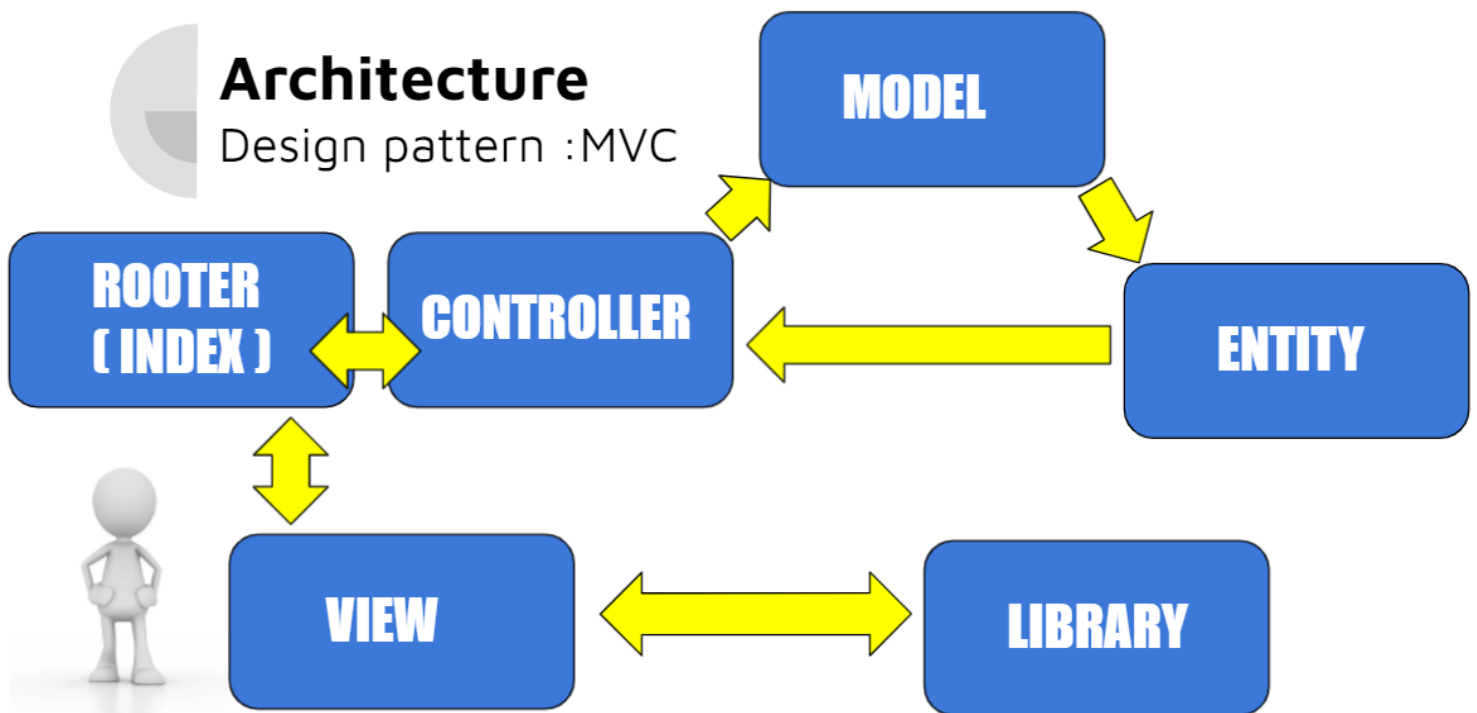
Une vue est le moyen d'afficher des objets dans une application. Les exemples incluent l'affichage d'une fenêtre ou des boutons ou du texte dans une fenêtre. Cela inclut tout ce que l'utilisateur peut voir.

Un contrôleur met à jour les modèles et les vues. Il accepte la contribution et effectue la mise à jour correspondante. Par exemple, un contrôleur peut mettre à jour un modèle en modifiant les attributs d'un utilisateur de notre site. Il peut modifier la vue en affichant les nouvelles informations d'utilisateur sur notre page d'accueil.

La librairie va nous servir à avoir du code html, et être appelé par notre view. Elle nous permet de réduire nos lignes de codes.

L'entity va nous permettre de vérifier nos entrées de formulaires rentrées par l'utilisateur. Par exemple, le nombre de caractères minimum du mot de passe .

Les trois principales parties de MVC sont interconnectées. La vue affiche le modèle pour l'utilisateur. Le contrôleur accepte les entrées utilisateur et met à jour le modèle et affiche les informations en conséquence. Bien que MVC ne soit pas requis dans la conception d'applications, de nombreux langages de programmation et IDE soutiennent l'architecture MVC, ce qui en fait un choix commun pour les développeurs.





## 2.3 Router

Nous avons mis en place un routeur. Celui-ci va nous permettre d'éviter la répétition de code et de faire tous les appels directement grâce à notre index.php.

```
<?php

session_start();

use controller\BackController;
use controller\FrontController;
use library\Display;
use model\Model;
```

Plus besoin de répéter sur nos pages le start des session, étant donné que tout passe par la page index, qui est notre "routeur".

```
if (method_exists($controller, $method)) { //if method exist, use controller where method name like p
    try {
        $data = $controller->$method($_POST); //controller use method where $_POST is use for registe
    } catch (Exception $e) {
        $msg = "<div class='alert alert-danger' role='alert'>{$e->getMessage()}</div>";
        $data = $controller->categorie();
    }
    if (is_array($data)) {
        extract($data);
    }
}
```

On appelle notre controller grâce à la fonction method\_exists, si elle existe, notre méthode ( fonction de la classe appelée, grâce à l'url ) sera alors chargée.

On a utilisé la méthode GET pour permettre la navigation des pages. Nous resterons du coup sur la page index.php, et nous naviguerons de page en page en fonction de la valeur GET.

```

if (empty($_GET)) { //if GET is empty, $page became home -> index
    $page = 'home';
} else {
    if (isset($_GET['page'])) { //if GET contain something, $page redirection to the $page (index.php?page=panier)
        $page = $_GET['page'];
        if (!file_exists("view/$page.php")) { //if the $page doesn't exist in the folder view, $page redirection to 404
            $page = '404';
        }
    }
}
}

```

En fonction de la valeur de la variable page dans notre GET, celui va appeler dans notre view, la page demandé. Si elle n'existe pas, l'utilisateur sera alors redirigé vers une page erreur 404.

Pour charger la vue, nous avons utilisé la fonction ob\_start. On "require" la vue qui sera stockée dans une mémoire tampon et on appel notre contenu avec ob\_get\_clean.

```

ob_start(); //we stock on buffer(tampon) the element of variable
require("view/$page.php");
$content = ob_get_clean(); //we post the variable
require("view/template/template.php"); //require the template (header footer + html/css/bootstrap)

```

Pour le chargement des classes, on a utilisé deux autoloader. Le premier permet de charger les librairies tierce ( stripe ) . Le second utilise la fonction myautoload écrite . Cette fonction permet de charger automatiquement les classes et d'appeler notre fichier php dont on a besoin.

```

require_once('vendor/autoload.php');
function myautoload($className)
{
    require(str_replace('\\', '/', $className) . '.php');
}
spl_autoload_register('myautoload');

```

## 2.4 Technos utilisées Front-end

### a. Bootstrap

Pour réaliser notre front, on a utilisé la technologie bootstrap. Cela nous a permis de gagner du temps malgré les désavantages que peut avoir ce framework. En effet, ce framework est plutôt simple d'utilisation. Il nous a permis de réaliser notre HTML sans forcément avoir besoin d'énormément de lignes de code en CSS. Nous l'avons utilisé principalement pour le positionnement de nos composants, et nos boutons. Cela nous a clairement simplifié la vie pour les formulaires et le plus gros avantage reste le responsive que je vais détailler plus bas.

L'inconvénient de ce framework est qu'il a des contraintes au niveau de la personnalisation. En effet, pour le changement des couleurs, nous sommes obligés d'utiliser la fonction !important dans notre CSS pour prioriser celle-ci. Au niveau des marges, nous avons seulement 5 effets de marge et souvent, la plus longue marge ne convient pas.

```
<div class="form-floating col-5 ms-5">
  <input type="password" class="form-control" id="
  <label for="floatingPassword">Mot de passe</labe
</div>
<div class="form-floating col-5 ms-5">
  <input type="password" class="form-control" id="
  <label for="floatingCpassword">Confirmation du m
</div>
```

Sur l'exemple ci-dessus, nous avons utilisé la méthode grid. Nous avons juste à rajouter les classes déjà pré faites de bootstrap pour pouvoir rajouter du margin.

## b. Composants réutilisables

Pour faire réaliser notre front, nous avons créé le dossier library qui contient le fichier display.php . On a créé des objets comme par exemple une carte produit qu'on va appeler lors de la recherche de l'utilisateur ou dans son panier par exemple. Il contient du code HTML avec du bootstrap et du CSS.

```
public function productCard(object $produit, $width = 21, $height = 55, $textlength = 200)
{
    >>

    <div class="col-sm d-flex justify-content-center mb-4 ">
        <div class="card text-decoration-none text-body" style="width: <?=$width ?>rem;" id="shadowproduct">
            <img style="height: <?=$height ?>vh;" class="img-card-custom border-img-top" src="public/img/<?=$produit->nom_image_p ?>" alt="...">
            <div class="card-body d-flex flex-column justify-content-between " method="POST" action="index.php?page=produit&id_p=<?=$produit->id_p ?>">
                <div class="d-flex col-12">
                    <div class="col-9">
                        <h5 class="card-title"><?=$produit->nom_p ?></h5>
                    </div>
                    <div class="col-3 fs-6 fw-bold" style="font-family:Montserrat, sans-serif;"><?=$produit->prix_ht_p ?>,00€</div>
                </div>
                <p class=" card-text description-produit"><?=$produit->troncateText($produit->description_p, $textlength) ?></p>

                <form class="d-flex justify-content-between mt-4" method="POST" action="index.php?page=produit&id_p=<?=$produit->id_p ?>">
                    <input type="hidden" name="fromPage" value="<?=$_SERVER["QUERY_STRING"] ?>">

                    <button class="btn btn-primary custom2 fw-bold fs-5" style="font-family: 'Ubuntu', sans-serif;" type="submit" name="add">
                        <svg width=" 44" height="40" viewBox="0 0 44 40" fill="none" xmlns="http://www.w3.org/2000/svg">
                            <path d="M16.5457 37.65C17.5499 37.65 18.3639 36.8862 18.3639 35.944C18.3639 35.0019 17.5499 34.2381 16.5457 34.2381C15.5415 34.2381 14.7275 35.0019 14.7275 35.944C14.7275 36.8862 15.5415 37.65 16.5457 37.65" style="fill:#000000;"/>
                            <path d="M36.5457 37.65C37.65 37.65 38.3639 36.8862 38.3639 35.944C38.3639 35.0019 37.5499 34.2381 36.5457 34.2381C35.5415 34.2381 34.7275 35.0019 34.7275 35.944C34.7275 36.8862 35.5415 37.65 36.5457 37.65" style="fill:#000000;"/>
                            <path d="M2 29.2727C14.1455 24.3161C14.3117 25.0834 14.7671 25.7727 15.4319 26.2632C16.0967 26.7538 16.9285 27.0144 17.5457 27.0144C18.1629 27.0144 18.7701 26.7538 19.3773 26.2632C20.0000 25.7727 20.6072 25.0834 21.1136 24.3161C21.6200 23.5488 21.1136 22.7815 20.6072 22.0142C20.0000 21.2469 19.3773 20.7531 18.7701 20.5000C18.1629 20.2469 17.5457 20.5000 16.9285 20.7531C16.3213 21.0000 15.6938 21.2469 15.0469 21.5000C14.4000 21.7531 13.7727 22.0142 13.1655 22.2727C12.5583 22.5311 11.9511 22.7815 11.3438 23.0311C10.7366 23.2815 10.1294 23.5311 9.5222 23.7815C8.9150 24.0311 8.3078 24.2815 7.6905 24.5311C7.0833 24.7815 6.4761 25.0311 5.8689 25.2815C5.2617 25.5311 4.6545 25.7815 4.0473 26.0311C3.4401 26.2815 2.8329 26.5311 2.2257 26.7815C1.6185 27.0311 0.9113 27.2815 0.2041 27.5311C-0.5031 27.7815 -1.2103 28.0311 -1.9175 28.2815C-2.6247 28.5311 -3.3319 28.7815 -4.0391 29.0311C-4.7463 29.2815 -5.4535 29.5311 -6.1607 29.7815C-6.8679 30.0311 -7.5751 30.2815 -8.2823 30.5311C-8.9895 30.7815 -9.6967 31.0311 -10.4039 31.2815C-11.1111 31.5311 -11.8183 31.7815 -12.5255 32.0311C-13.2327 32.2815 -13.9399 32.5311 -14.6471 32.7815C-15.3543 33.0311 -16.0615 33.2815 -16.7687 33.5311C-17.4759 33.7815 -18.1831 34.0311 -18.8903 34.2815C-19.5975 34.5311 -20.3047 34.7815 -21.0119 35.0311C-21.7191 35.2815 -22.4263 35.5311 -23.1335 35.7815C-23.8407 36.0311 -24.5479 36.2815 -25.2551 36.5311C-25.9623 36.7815 -26.6695 37.0311 -27.3767 37.2815C-28.0839 37.5311 -28.7911 37.7815 -29.4983 38.0311C-30.2055 38.2815 -30.9127 38.5311 -31.6199 38.7815C-32.3271 39.0311 -33.0343 39.2815 -33.7415 39.5311C-34.4487 39.7815 -35.1559 39.5311 -35.8631 39.2815C-36.5703 39.0311 -37.2775 38.7815 -37.9847 38.5311C-38.6919 38.2815 -39.3991 38.0311 -40.1063 37.7815C-40.8135 37.5311 -41.5207 37.2815 -42.2279 37.0311C-42.9351 36.7815 -43.6423 36.5311 -44.3495 36.2815C-45.0567 36.0311 -45.7639 35.7815 -46.4711 35.5311C-47.1783 35.2815 -47.8855 35.0311 -48.5927 34.7815C-49.3000 34.5311 -49.9999 34.2815 -50.7000 34.0311C-51.4000 33.7815 -52.1000 33.5311 -52.8000 33.2815C-53.5000 33.0311 -54.2000 32.7815 -54.9000 32.5311C-55.6000 32.2815 -56.3000 32.0311 -57.0000 31.7815C-57.7000 31.5311 -58.4000 31.2815 -59.1000 31.0311C-59.8000 30.7815 -60.5000 30.5311 -61.2000 30.2815C-61.9000 30.0311 -62.6000 29.7815 -63.3000 29.5311C-64.0000 29.2815 -64.7000 29.0311 -65.4000 28.7815C-66.1000 28.5311 -66.8000 28.2815 -67.5000 28.0311C-68.2000 27.7815 -68.9000 27.5311 -69.6000 27.2815C-70.3000 27.0311 -71.0000 26.7815 -71.7000 26.5311C-72.4000 26.2815 -73.1000 26.0311 -73.8000 25.7815C-74.5000 25.5311 -75.2000 25.2815 -75.9000 25.0311C-76.6000 24.7815 -77.3000 24.5311 -78.0000 24.2815C-78.7000 24.0311 -79.4000 23.7815 -80.1000 23.5311C-80.8000 23.2815 -81.5000 23.0311 -82.2000 22.7815C-82.9000 22.5311 -83.6000 22.2815 -84.3000 22.0311C-85.0000 21.7815 -85.7000 21.5311 -86.4000 21.2815C-87.1000 21.0311 -87.8000 20.7815 -88.5000 20.5311C-89.2000 20.2815 -89.9000 20.0311 -90.6000 19.7815C-91.3000 19.5311 -92.0000 19.2815 -92.7000 19.0311C-93.4000 18.7815 -94.1000 18.5311 -94.8000 18.2815C-95.5000 18.0311 -96.2000 17.7815 -96.9000 17.5311C-97.6000 17.2815 -98.3000 17.0311 -99.0000 16.7815C-99.7000 16.5311 -100.4000 16.2815 -101.1000 16.0311C-101.8000 15.7815 -102.5000 15.5311 -103.2000 15.2815C-103.9000 15.0311 -104.6000 14.7815 -105.3000 14.5311C-106.0000 14.2815 -106.7000 14.0311 -107.4000 13.7815C-108.1000 13.5311 -108.8000 13.2815 -109.5000 13.0311C-110.2000 12.7815 -110.9000 12.5311 -111.6000 12.2815C-112.3000 12.0311 -113.0000 11.7815 -113.7000 11.5311C-114.4000 11.2815 -115.1000 11.0311 -115.8000 10.7815C-116.5000 10.5311 -117.2000 10.2815 -117.9000 10.0311C-118.6000 9.7815 -119.3000 9.5311 -120.0000 9.2815C-120.7000 9.0311 -121.4000 8.7815 -122.1000 8.5311C-122.8000 8.2815 -123.5000 8.0311 -124.2000 7.7815C-124.9000 7.5311 -125.6000 7.2815 -126.3000 7.0311C-127.0000 6.7815 -127.7000 6.5311 -128.4000 6.2815C-129.1000 6.0311 -129.8000 5.7815 -130.5000 5.5311C-131.2000 5.2815 -131.9000 5.0311 -132.6000 4.7815C-133.3000 4.5311 -134.0000 4.2815 -134.7000 4.0311C-135.4000 3.7815 -136.1000 3.5311 -136.8000 3.2815C-137.5000 3.0311 -138.2000 2.7815 -138.9000 2.5311C-139.6000 2.2815 -140.3000 2.0311 -141.0000 1.7815C-141.7000 1.5311 -142.4000 1.2815 -143.1000 1.0311C-143.8000 0.7815 -144.5000 0.5311 -145.2000 0.2815C-145.9000 0.0311 -146.6000 -0.2187 -147.3000 -0.4688C-148.0000 -0.7188 -148.7000 -0.9688 -149.4000 -1.2188C-150.1000 -1.4688 -150.8000 -1.7188 -151.5000 -1.9688C-152.2000 -2.2188 -152.9000 -2.4688 -153.6000 -2.7188C-154.3000 -2.9688 -155.0000 -3.2188 -155.7000 -3.4688C-156.4000 -3.7188 -157.1000 -3.9688 -157.8000 -4.2188C-158.5000 -4.4688 -159.2000 -4.7188 -159.9000 -4.9688C-160.6000 -5.2188 -161.3000 -5.4688 -162.0000 -5.7188C-162.7000 -5.9688 -163.4000 -6.2188 -164.1000 -6.4688C-164.8000 -6.7188 -165.5000 -6.9688 -166.2000 -7.2188C-166.9000 -7.4688 -167.6000 -7.7188 -168.3000 -7.9688C-169.0000 -8.2188 -169.7000 -8.4688 -170.4000 -8.7188C-171.1000 -8.9688 -171.8000 -9.2188 -172.5000 -9.4688C-173.2000 -9.7188 -173.9000 -9.9688 -174.6000 -10.2188C-175.3000 -10.4688 -176.0000 -10.7188 -176.7000 -10.9688C-177.4000 -11.2188 -178.1000 -11.4688 -178.8000 -11.7188C-179.5000 -11.9688 -180.2000 -12.2188 -180.9000 -12.4688C-181.6000 -12.7188 -182.3000 -12.9688 -183.0000 -13.2188C-183.7000 -13.4688 -184.4000 -13.7188 -185.1000 -13.9688C-185.8000 -14.2188 -186.5000 -14.4688 -187.2000 -14.7188C-187.9000 -14.9688 -188.6000 -15.2188 -189.3000 -15.4688C-190.0000 -15.7188 -190.7000 -15.9688 -191.4000 -16.2188C-192.1000 -16.4688 -192.8000 -16.7188 -193.5000 -16.9688C-194.2000 -17.2188 -194.9000 -17.4688 -195.6000 -17.7188C-196.3000 -17.9688 -197.0000 -18.2188 -197.7000 -18.4688C-198.4000 -18.7188 -199.1000 -18.9688 -199.8000 -19.2188C-200.5000 -19.4688 -201.2000 -19.7188 -201.9000 -19.9688C-202.6000 -20.2188 -203.3000 -20.4688 -204.0000 -20.7188C-204.7000 -20.9688 -205.4000 -21.2188 -206.1000 -21.4688C-206.8000 -21.7188 -207.5000 -21.9688 -208.2000 -22.2188C-208.9000 -22.4688 -209.6000 -22.7188 -210.3000 -22.9688C-211.0000 -23.2188 -211.7000 -23.4688 -212.4000 -23.7188C-213.1000 -23.9688 -213.8000 -24.2188 -214.5000 -24.4688C-215.2000 -24.7188 -215.9000 -24.9688 -216.6000 -25.2188C-217.3000 -25.4688 -218.0000 -25.7188 -218.7000 -25.9688C-219.4000 -26.2188 -220.1000 -26.4688 -220.8000 -26.7188C-221.5000 -26.9688 -222.2000 -27.2188 -222.9000 -27.4688C-223.6000 -27.7188 -224.3000 -27.9688 -225.0000 -28.2188C-225.7000 -28.4688 -226.4000 -28.7188 -227.1000 -28.9688C-227.8000 -29.2188 -228.5000 -29.4688 -229.2000 -29.7188C-229.9000 -29.9688 -230.6000 -30.2188 -231.3000 -30.4688C-232.0000 -30.7188 -232.7000 -30.9688 -233.4000 -31.2188C-234.1000 -31.4688 -234.8000 -31.7188 -235.5000 -31.9688C-236.2000 -32.2188 -236.9000 -32.4688 -237.6000 -32.7188C-238.3000 -32.9688 -239.0000 -33.2188 -239.7000 -33.4688C-240.4000 -33.7188 -241.1000 -33.9688 -241.8000 -34.2188C-242.5000 -34.4688 -243.2000 -34.7188 -243.9000 -34.9688C-244.6000 -35.2188 -245.3000 -35.4688 -246.0000 -35.7188C-246.7000 -35.9688 -247.4000 -36.2188 -248.1000 -36.4688C-248.8000 -36.7188 -249.5000 -36.9688 -250.2000 -37.2188C-250.9000 -37.4688 -251.6000 -37.7188 -252.3000 -37.9688C-253.0000 -38.2188 -253.7000 -38.4688 -254.4000 -38.7188C-255.1000 -38.9688 -255.8000 -39.2188 -256.5000 -39.4688C-257.2000 -39.7188 -257.9000 -39.9688 -258.6000 -40.2188C-259.3000 -40.4688 -260.0000 -40.7188 -260.7000 -40.9688C-261.4000 -41.2188 -262.1000 -41.4688 -262.8000 -41.7188C-263.5000 -41.9688 -264.2000 -42.2188 -264.9000 -42.4688C-265.6000 -42.7188 -266.3000 -42.9688 -267.0000 -43.2188C-267.7000 -43.4688 -268.4000 -43.7188 -269.1000 -43.9688C-269.8000 -44.2188 -270.5000 -44.4688 -271.2000 -44.7188C-271.9000 -44.9688 -272.6000 -45.2188 -273.3000 -45.4688C-274.0000 -45.7188 -274.7000 -45.9688 -275.4000 -46.2188C-276.1000 -46.4688 -276.8000 -46.7188 -277.5000 -46.9688C-278.2000 -47.2188 -278.9000 -47.4688 -279.6000 -47.7188C-280.3000 -47.9688 -281.0000 -48.2188 -281.7000 -48.4688C-282.4000 -48.7188 -283.1000 -48.9688 -283.8000 -49.2188C-284.5000 -49.4688 -285.2000 -49.7188 -285.9000 -49.9688C-286.6000 -50.2188 -287.3000 -50.4688 -288.0000 -50.7188C-288.7000 -50.9688 -289.4000 -51.2188 -290.1000 -51.4688C-290.8000 -51.7188 -291.5000 -51.9688 -292.2000 -52.2188C-292.9000 -52.4688 -293.6000 -52.7188 -294.3000 -52.9688C-295.0000 -53.2188 -295.7000 -53.4688 -296.4000 -53.7188C-297.1000 -53.9688 -297.8000 -54.2188 -298.5000 -54.4688C-299.2000 -54.7188 -299.9000 -54.9688 -300.6000 -55.2188C-301.3000 -55.4688 -302.0000 -55.7188 -302.7000 -55.9688C-303.4000 -56.2188 -304.1000 -56.4688 -304.8000 -56.7188C-305.5000 -56.9688 -306.2000 -57.2188 -306.9000 -57.4688C-307.6000 -57.7188 -308.3000 -57.9688 -309.0000 -58.2188C-309.7000 -58.4688 -310.4000 -58.7188 -311.1000 -58.9688C-311.8000 -59.2188 -312.5000 -59.4688 -313.2000 -59.7188C-313.9000 -59.9688 -314.6000 -60.2188 -315.3000 -60.4688C-316.0000 -60.7188 -316.7000 -60.9688 -317.4000 -61.2188C-318.1000 -61.4688 -318.8000 -61.7188 -319.5000 -61.9688C-320.2000 -62.2188 -320.9000 -62.4688 -321.6000 -62.7188C-322.3000 -62.9688 -323.0000 -63.2188 -323.7000 -63.4688C-324.4000 -63.7188 -325.1000 -63.9688 -325.8000 -64.2188C-326.5000 -64.4688 -327.2000 -64.7188 -327.9000 -64.9688C-328.6000 -65.2188 -329.3000 -65.4688 -330.0000 -65.7188C-330.7000 -65.9688 -331.4000 -66.2188 -332.1000 -66.4688C-332.8000 -66.7188 -333.5000 -66.9688 -334.2000 -67.2188C-334.9000 -67.4688 -335.6000 -67.7188 -336.3000 -67.9688C-337.0000 -68.2188 -337.7000 -68.4688 -338.4000 -68.7188C-339.1000 -68.9688 -339.8000 -69.2188 -340.5000 -69.4688C-341.2000 -69.7188 -341.9000 -69.9688 -342.6000 -70.2188C-343.3000 -70.4688 -344.0000 -70.7188 -344.7000 -70.9688C-345.4000 -71.2188 -346.1000 -71.4688 -346.8000 -71.7188C-347.5000 -71.9688 -348.2000 -72.2188 -348.9000 -72.4688C-349.6000 -72.7188 -350.3000 -72.9688 -351.0000 -73.2188C-351.7000 -73.4688 -352.4000 -73.7188 -353.1000 -73.9688C-353.8000 -74.2188 -354.5000 -74.4688 -355.2000 -74.7188C-355.9000 -74.9688 -356.6000 -75.2188 -357.3000 -75.4688C-358.0000 -75.7188 -358.7000 -75.9688 -359.4000 -76.2188C-360.1000 -76.4688 -360.8000 -76.7188 -361.5000 -76.9688C-362.2000 -77.2188 -362.9000 -77.4688 -363.6000 -77.7188C-364.3000 -77.9688 -365.0000 -78.2188 -365.7000 -78.4688C-366.4000 -78.7188 -367.1000 -78.9688 -367.8000 -79.2188C-368.5000 -79.4688 -369.2000 -79.7188 -369.9000 -79.9688C-370.6000 -80.2188 -371.3000 -80.4688 -372.0000 -80.7188C-372.7000 -80.9688 -373.4000 -81.2188 -374.1000 -81.4688C-374.8000 -81.7188 -375.5000 -81.9688 -376.2000 -82.2188C-376.9000 -82.4688 -377.6000 -82.7188 -378.3000 -82.9688C-379.0000 -83.2188 -379.7000 -83.4688 -380.4000 -83.7188C-381.1000 -83.9688 -381.8000 -84.2188 -382.5000 -84.4688C-383.2000 -84.7188 -383.9000 -84.9688 -384.6000 -85.2188C-385.3000 -85.4688 -386.0000 -85.7188 -386.7000 -85.9688C-387.4000 -86.2188 -388.1000 -86.4688 -388.8000 -86.7188C-389.5000 -86.9688 -390.2000 -87.2188 -390.9000 -87.4688C-391.6000 -87.7188 -392.3000 -87.9688 -393.0000 -88.2188C-393.7000 -88.4688 -394.4000 -88.7188 -395.1000 -88.9688C-395.8000 -89.2188 -396.5000 -89.4688 -397.2000 -89.7188C-397.9000 -89.9688 -398.6000 -90.2188 -399.3000 -90.4688C-400.0000 -90.7188 -400.7000 -90.9688 -401.4000 -91.2188C-402.1000 -91.4688 -402.8000 -91.7188 -403.5000 -91.9688C-404.2000 -92.2188 -404.9000 -92.4688 -405.6000 -92.7188C-406.3000 -92.9688 -407.0000 -93.2188 -407.7000 -93.4688C-408.4000 -93.7188 -409.1000 -93.9688 -409.8000 -94.2188C-410.5000 -94.4688 -411.2000 -94.7188 -411.9000 -94.9688C-412.6000 -95.2188 -413.3000 -95.4688 -414.0000 -95.7188C-414.7000 -95.9688 -415.4000 -96.2188 -416.1000 -96.4688C-416.8000 -96.7188 -417.5000 -96.9688 -418.2000 -97.2188C-418.9000 -97.4688 -419.6000 -97.7188 -420.3000 -97.9688C-421.0000 -98.2188 -421.7000 -98.4688 -422.4000 -98.7188C-423.1000 -98.9688 -423.8000 -99.2188 -424.5000 -99.4688C-425.2000 -99.7188 -425.9000 -99.9688 -426.6000 -100.2188C-427.3000 -100.4688 -428.0000 -100.7188 -428.7000 -100.9688C-429.4000 -101.2188 -430.1000 -101.4688 -430.8
```

Grâce à notre routeur et au namespace, pas besoin d'appeler notre fichier, étant donné que nous sommes toujours sur l'index et qu'il contient

```
: use library\Display; $display = new Display();
```

On utilise le display et on instancie l'objet.

### c. Responsive

Pour le responsive, nous avons utilisé principalement le framework bootstrap. Pour la police, j'ai utilisé Media Queries. Media queries fait partie d'un module CSS3, qui permet de rendre adaptable une div, un paragraphe en fonction de la taille de l'écran. Cela m'a permis de rendre la police adaptable sur grand écran et sur téléphone portable.

```
@media screen and (min-width: 684px) {
    .titre {
        font-size: 30px;
    }

    .titreblanc {
        font-size: 25px;
    }

    .textblanc {
        font-size: 20px;
    }

    .card {
        width: 10rem;
    }

    .imageprojet {
        max-width: 45vw;
    }

    .logo {
        width: 11vw;
        max-height: 90%;
    }
}

@media screen and (max-width: 683px) {
    .titre {
        font-size: 5vw;
    }

    .titreblanc {
        font-size: 4vw;
    }

    .textblanc {
        font-size: 3vw;
    }

    .card {
        width: 10rem;
    }

    .imageprojet {
        width: 85vw;
    }

    .logo {
        width: 18vw;
        max-height: 90%;
    }
}
```



## 2.5 Technos utilisées Back-end

### a. Création d'utilisateur

Pour la création d'utilisateur, nous avons utilisé le langage PHP avec PDO. PHP Data Objects est une extension définissant l'interface pour accéder à une base de données avec PHP. Elle est propre à PHP. On va pouvoir aller vérifier des informations avec une simple requête SQL.

Pour la création d'utilisateur, nous avons décidé que l'utilisateur doit s'enregistrer avec login unique. Pour vérifier si le login est disponible et non utilisé nous avons utilisé la fonction suivante :

```
public function isInDb($user)
{
    $sth = $this->db->prepare("SELECT * FROM utilisateur WHERE login_u = :login_u");
    $sth->bindParam(":login_u", $user['login_u']);
    $sth->execute();
    $result = $sth->fetch();
    return $result;
}
```

Cette fonction va nous donner un résultat true or false. Elle se situe dans notre modèle utilisateur.

Nous allons ensuite dans notre controller faire appel à cette méthode comme ce qui suit :

```
public function inscription($donnee_u)
{
    // if form is empty, stop fonction execution
    if (empty($donnee_u)) {
        return;
    }
    $user = new UtilisateurEntity($donnee_u);

    if (($msg = $user->checkData()) === TRUE) {

        $utilisateurModel = new UtilisateurModel();
        if ($utilisateurModel->isInDb($donnee_u)) {
            throw new Exception('Le login n\'est pas disponible');
        } else {
            $utilisateurModel->add($user);
            header('Location: index.php?page=home&error=successinscription');
        }
    } else {
        throw new Exception(implode('<br />', $msg));
    }
}
```

Avec le système des exception, en appelant notre fonction, si la fonction `isInDb` retourne `true`, alors le login ne sera pas disponible et nous faisons apparaître le message au dessus de notre formulaire d'inscription, sinon, l'utilisateur est rentré en base de donnée avec la fonction `add` que nous avons au préalable codé.

```
public function add($user, $table = NULL)
{
    foreach ($user as $key => &$value) {
        $value = trim(htmlspecialchars($value));
    }
    $SQL = "INSERT INTO utilisateur (login_u, nom_u, prenom_u, datedenaissance_u, adresse_u, mail_u, telephone_u, motdepass_u )
VALUES (:login_u, :nom_u, :prenom_u, :datedenaissance_u, :adresse_u, :mail_u, :telephone_u, :motdepass_u)";
    $sth = $this->db->prepare($SQL);
    $sth->bindParam(":login_u", $user->login_u, PDO::PARAM_STR, 255);
    $sth->bindParam(":nom_u", $user->nom_u, PDO::PARAM_STR, 255);
    $sth->bindParam(":prenom_u", $user->prenom_u, PDO::PARAM_STR, 255);
    $sth->bindParam(":adresse_u", $user->adresse_u);
    $sth->bindParam(":mail_u", $user->mail_u);
    $sth->bindParam(":telephone_u", $user->telephone_u);
    $sth->bindParam(":motdepass_u", $user->motdepass_u, PDO::PARAM_STR, 255);
    $sth->bindParam(":datedenaissance_u", $user->datedenaissance_u, PDO::PARAM_STR, 255);
    $sth->execute();
}
```

## b. Requête générique

Nous avons créé une requête générique pour nous permettre de récupérer toutes les informations de nos tables. Nous avons appelé cette fonction `getAll`. En mettant `$table = NULL`, cela nous permet d'avoir une erreur PHP.

```
public function getAll($table = NULL)
{
    $table = $table != NULL ? $table : $this->table;
    $SQL = "SELECT * FROM $table";
    return $this->fetchAll($SQL);
}
```

Une fois la fonction faite, nous pouvons récupérer toutes les informations des tables en mettant son nom dans l'argument de notre méthode.

```
$model = new Model();
$utilisateur = $model->getAll('Utilisateur');
```



## 2.6 Sécurité

Lorsque vous surfez sur Internet, votre navigateur utilise le protocole HTTP (Hypertext Transfer Protocol) pour afficher les pages web, et le protocole Transmission Control Protocol/Internet Protocol (TCP/IP) pour les transmettre.

Si le serveur web établit la connexion TCP avec le navigateur, une réponse avec le code statut et le fichier demandé (généralement le fichier index.html pour la page web) sera transmise. Mais dans notre cas, les données transitent en HTTP et pas en HTTPS...

Les données transitant en HTTP peuvent être interceptées, car elles transitent en clair. Comment éviter d'exposer les données sensibles en transit ?

- Utilisez le HTTPS pour l'ensemble de votre site, même s'il ne contient pas de données sensibles.
- Utilisez les requêtes GET pour récupérer les informations et POST pour modifier les informations.
- Sécurisez vos cookies pour qu'ils soient transmis par l'en-tête et via HTTPS.
- Sécurisez vos sessions en ajoutant une date d'expiration, en sécurisant l'ID et en ne mettant pas cet ID dans l'URL.

**Injection SQL :** La faille SQLi, abréviation de SQL Injection, soit injection SQL en français, est un groupe de méthodes d'exploitation de failles de sécurité d'une application interagissant avec une base de données. Elle permet d'injecter dans la requête SQL en cours un morceau de requête non prévu par le système et pouvant compromettre la sécurité.

Comment s'en prémunir ?

La sécurité du serveur sur lequel a cours votre système de gestion de base de données joue naturellement un rôle primordial dans la prévention contre les injections SQL. La première étape est donc de renforcer votre système d'exploitation selon le schéma établi :

- Installer ou activer seulement les applications et services qui sont pertinents pour faire fonctionner la base de données.
- Supprimer tous les comptes utilisateurs qui ne sont pas nécessaires.
- S'assurer que toutes les mises à jour des systèmes et programmes pertinents sont installés

Pour les Prepared Statements, il est largement recommandé de ne pas utiliser le module PHP mysql mais de choisir à la place mysqli ou PDO. De cette manière, vous pourrez par la même occasion vous protéger avec des codes sûrs. La fonction `mysqli_real_escape_string()` du script PHP permet par exemple d'éviter d'émettre des caractères spéciaux aux bases de données SQL sous leur forme originale.

Qu'est-ce que la faille XSS ?

Il faut absolument utiliser les fonctions php `htmlspecialchars()` qui filtre les '<' et '>' ou `htmlentities()` qui filtre toutes les entités html. Ces fonctions doivent être utilisées sur des entrées utilisateurs qui s'afficheront plus tard sur votre site. Si elles ne sont pas filtrées, les scripts injectés pourront s'exécuter.

<https://www.aurone.com/5-mesures-de-securite-php-implementer/>

<https://www.ionos.fr/digitalguide/serveur/securite/injection-sql-bases-et-mesures-pour-se-proteger/>

## 2.7 Anglais

Exemple de type de recherche :

<https://stackoverflow.com/questions/1179874/what-is-the-difference-between-bindparam-and-bindvalue>

### What is the difference between bindParam and bindValue?

What is the difference between [PDOStatement::bindParam\(\)](#) and [PDOStatement::bindValue\(\)](#)?

The answer is in the documentation for [bindParam](#):

Unlike `PDOStatement::bindValue()`, the variable is bound as a reference and will only be evaluated at the time that `PDOStatement::execute()` is called.

And [execute](#)

call `PDOStatement::bindParam()` to bind PHP variables to the parameter markers: bound variables pass their value as input and receive the output value, if any, of their associated parameter markers

Example:

```
$value = 'foo'; $s = $dbh->prepare('SELECT name FROM bar WHERE baz = :baz');  
$s->bindParam(':baz', $value); // use bindParam to bind the variable  
$value = 'foobarbaz'; $s->execute(); // executed with WHERE baz = 'foobarbaz'
```

or

```
$value = 'foo'; $s = $dbh->prepare('SELECT name FROM bar WHERE baz = :baz');  
$s->bindValue(':baz', $value); // use bindValue to bind the variable's value  
$value = 'foobarbaz'; $s->execute(); // executed with WHERE baz = 'foo'
```

Quelle est la différence entre bindParam et bindValue ?

La réponse est dans la documentation,

pour bindParam:

Contrairement à PDOStatement::bindValue(), la variable est liée à une référence et sera évaluée au moment où PDOStatement::execute() est appelé.

Et exécute

appelé PDOStatement::bindParam() pour lier des variables PHP aux marqueurs: les variables liées transmettent leur valeur en entrée et reçoivent la valeur de sortie, si disponible, de leurs marqueurs de paramètres associés

exemple:

```
$value = 'foo'; $s = $dbh->prepare('SELECT name FROM bar WHERE baz = :baz'); $s->bindParam(':baz', $value); // utilise bindParam pour lier la variable $value = 'foobarbaz'; $s->execute(); // exécuté avec WHERE baz = 'foobarbaz'
```

ou

```
$value = 'foo'; $s = $dbh->prepare('SELECT name FROM bar WHERE baz = :baz'); $s->bindValue(':baz', $value); // utilise bindValue pour lier la valeur de la variable $value = 'foobarbaz'; $s->execute(); // exécuté avec WHERE baz = 'foo'
```