

**PhotoForYou**

**2022**

**Documentation technique**

Fait par Monchau Victoire Ioamra

## Sommaire

|  |    |
|--|----|
| 1. Présentation générale.....            | 3  |
| 1.1. Contexte.....                       | 3  |
| 1.2. Expression des besoins.....         | 3  |
| 1.2.a. Objectifs.....                    | 3  |
| 1.2.b. Cibles et caractéristiques.....   | 3  |
| 1.2.c. Description du contenu.....       | 4  |
| 1.3. Exigences fonctionnelles.....       | 4  |
| 1.3.a. Principales composantes.....      | 4  |
| 1.3.b. Fonctionnalités Front.....        | 4  |
| 1.3.c. Fonctionnalités Back.....         | 5  |
| 2. Analyse.....                          | 5  |
| 2.1. MCD / MLD.....                      | 5  |
| 2.2. Diagramme de cas d'utilisation..... | 6  |
| 3. Environnement de développement.....   | 6  |
| 4. Présentation du site.....             | 6  |
| 4.1. Page d'accueil.....                 | 6  |
| 4.2. Achat de crédits.....               | 9  |
| 4.3. Le panier.....                      | 10 |
| 4.4. Profil Utilisateur.....             | 12 |
| 4.5. Historique Achat / Vente.....       | 14 |
| 4.6. Déposer des images.....             | 15 |
| 4.7. Gestion utilisateur.....            | 15 |
| 4.8. gestion catégorie.....              | 17 |
| 4.9. Inscription.....                    | 18 |
| 4.10. Connexion.....                     | 20 |

# 1. Présentation générale

## 1.1. Contexte

La société **AtelierLumière** est un laboratoire photographique professionnel situé à Paris (14ème arrondissement) et créé en 2006. Composé d'une équipe de quatre personnes : *Flippo Raverdi* (dirigeant et photographe), *Sophie Pierre* (partie administrative), *Jean-Marc Tomé et Bernard Guillaume* (retouches et tirages). **AtelierLumière** est spécialisé dans le tirage photographique et la retouche pour les professionnels. (Tirages lors d'expositions, retouches pour des magazines...)

Ses principales activités sont les suivantes :

- numérisation de tout support
- développement de fichiers RAW
- la retouche créative
- Le tirage jet d'encre d'exposition et l'épreuve

La société AtelierLumière véhicule son image autour de l'artisanal. En effet elle s'est fait une solide réputation chez les photographes auteurs comme chez les professionnels de la mode.

Afin de diversifier son activité et faire profiter de son savoir-faire à un plus grand nombre: la société AtelierLumière voudrait proposer la vente en ligne de photographies d'auteur. La vente en ligne de photographies est déjà assurée par d'autres concurrents comme : [Fotolia] (<http://fr.fotolia.com>) , [DarqRoom] (<http://www.darqroom.com>), etc. Il sera donc important de se démarquer de ces concurrents en se basant sur la notoriété de la société et son expérience mais aussi en axant sur l'aspect photo d'art !

---

## 1.2. Expression des besoins

### 1.2.a. Objectifs

- Vente et achat de photographies.
- Attirer de nouveaux clients. Augmenter le CA et les bénéfices.

### 1.2.b. Cibles et caractéristiques

- Vendeurs : photographes d'auteur (travaillant avec la société ou pas).
- Acheteur : magazine, journaux, industriels, particuliers, ...

### **1.2.c. Description du contenu**

- Enregistrement sur le site en tant que photographe ou client.
- Les comptes photographe doivent attendre une validation de l'administrateur pour pouvoir se connecter.
- Les informations des utilisateurs doivent être modifiables à tous moments.
- L'administrateur doit pouvoir gérer l'état des utilisateurs (banni / valide / en attente de validation)
- Catalogue des photos à vendre classer par catégorie.
- Moteur de recherche rapide, avancé/multicritères
- Gestion du caddie
- Les photos achetées sont archivées chez le client et téléchargeables à tous moments
- L'historique d'achat et de vente doit être disponible à tous moments.
- Le photographe fixe le prix de vente en nombre de crédit. (minimum 250 crédits-maximum 25 000 crédits)
- Les photos sont en vente de manière exclusive (un seul achat après retrait du catalogue)
- L'achat se fait à partir de crédits que l'on doit acheter. (50 crédit = 1 euros)

## **1.3. Exigences fonctionnelles**

### **1.3.a. Principales composantes**

- Gestion des utilisateurs
- Identification des utilisateurs
- Gestion de l'espace client/photographe
- Gestion des commandes
- Gestion de paiements
- Gestion du catalogue

### **1.3.b. Fonctionnalités Front**

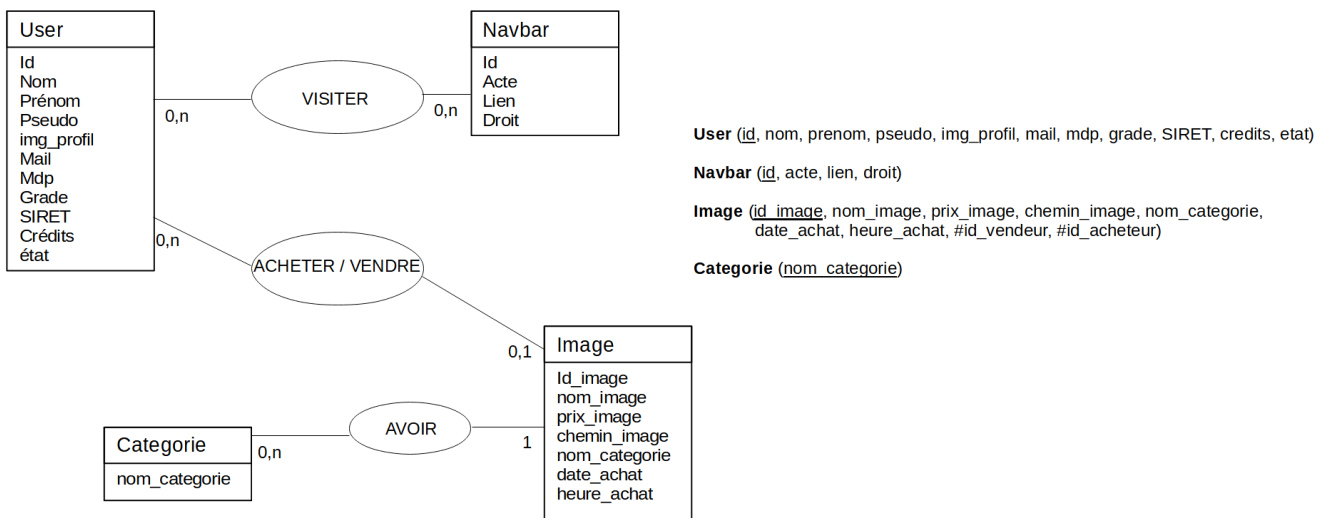
- Identification
- Catalogue des photos
- Gestion de l'espace photographe
- Gestion de l'espace client
- Gestion des commandes

### 1.3.c. Fonctionnalités Back

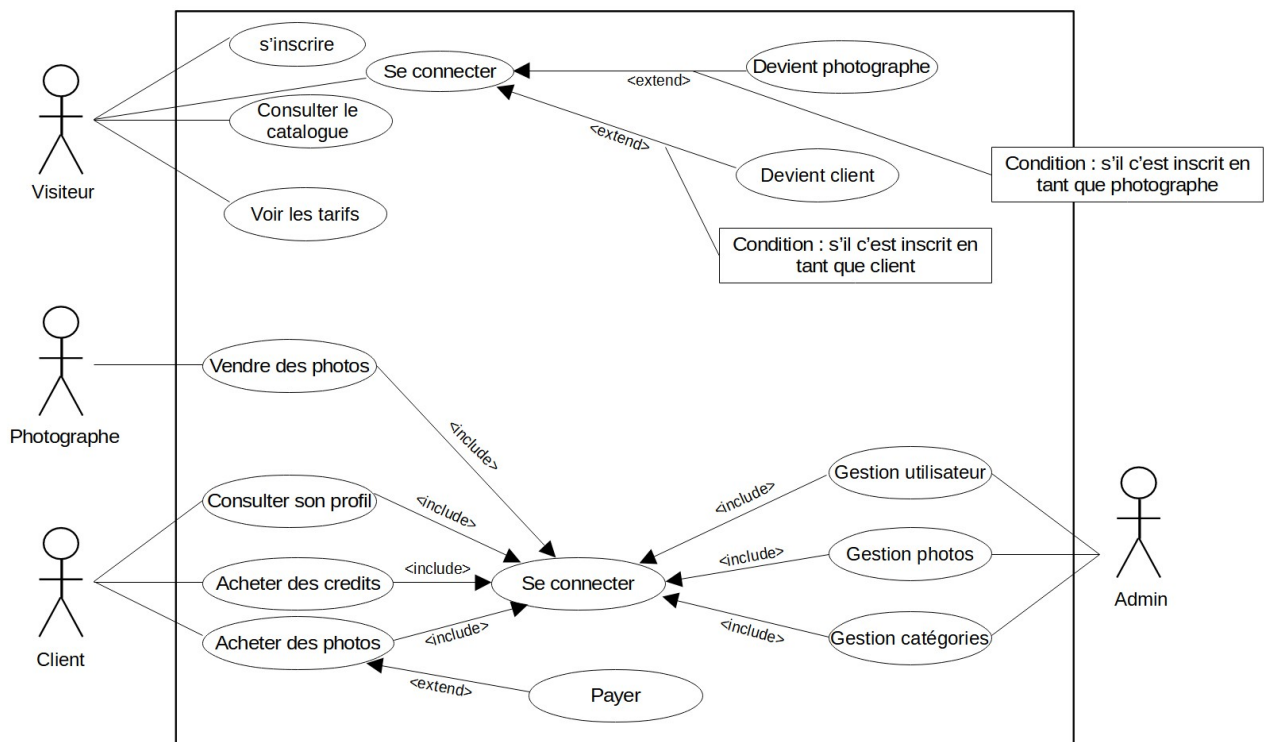
- Gestion des utilisateurs
- Gestion du contenu informationnel
- Gestion du catalogue

## 2. Analyse

### 2.1. MCD / MLD



## 2.2. Diagramme de cas d'utilisation



## 3. Environnement de développement

Langage utilisé : HTML, CSS, JavaScript, PHP, SQL

Framework/Librairie : JQuery, BootStrap 5

## 4. Présentation du site

### 4.1. Page d'accueil

Par défaut, affiche la liste des catégories avec l'image achetable la plus récente de chaque catégorie.

Chaque image est un lien vers la catégorie correspondante.

La gestion de l'affichage des catégories et de la liste des images de chaque catégorie est faites via les fonctions `fetchSearchImage()` et `fetchSearchCategorie()`

Dans fetch-image.js :

`fetchSearchImage(categorie, recherche) : void`

Cette fonction fait appel à l'api seach.php et appelle la fonction `getImages()`  
Les données récupérées sont alors mis en forme et affiché sur la page `index.php`

`fetchSearchCategorie(recherche): void`

Cette fonction fait appel à l'api seach.php et appelle la fonction `getCategorie()`  
Les données récupérées sont alors mis en forme et afficher sur la page `index.php`

Dans search.php :

`getImages(categorie, recherche)`

Affiche un objet JSON contenant toutes les informations des images non acheter qui corresponde à la catégorie et à la recherche.

`getCategorie(recherche)`


Affiche un objet JSON contenant, pour chaque catégorie les informations de l'image la plus récente et qui n'est pas encore acheté.

**Photo For You** Tarif


Connexion Inscription

### Catégorie


Rechercher une catégorie



**Animaux**



**Océan**



**Paysage**

Mail : [iom.monchau@gmail.com](mailto:iom.monchau@gmail.com)  
© Fait par Ioamra Monchau

[Revenir à la liste des catégories](#)

## Animaux

**Aigle**

1700 crédits

**Renard**

500 crédits

**Tigre**

400 crédits

Mail : [iom.monchau@gmail.com](mailto:iom.monchau@gmail.com)

© Fait par Ioamra Monchau

Chaque image de la catégorie choisie est un lien vers la page d'achat de cette image. Tous les utilisateurs peuvent visiter ces pages mais seuls les clients peuvent acheter les images.

[Revenir à la liste des Animaux](#)

## Aigle

1440 px X 960 px

1700 Credits

de Lareyku

Mail : [iom.monchau@gmail.com](mailto:iom.monchau@gmail.com)

© Fait par Ioamra Monchau



## 4.2. Achat de crédits

La conversion EUR Crédits se fait automatiquement.

```
/* Conversion du prix
$('#input-switch-money-1').on('keyup', function() {
  if ($('#span-switch-money-1').html() == 'EUR') {
    let val = $('#input-switch-money-1').val() * 50;
    $('#input-switch-money-2').val(val);
  } else {
    let val = $('#input-switch-money-1').val() / 50;
    $('#input-switch-money-2').val(val);
  }
});
```

### Acheter des crédits

|   |                                  |                                  |
|---|----------------------------------|----------------------------------|
| <input type="text" value="10"/>               | <input type="button" value="↔"/> | <input type="text" value="500"/> |
| EUR   |                                  | Crédits                          |
| Nom du détenteur de la carte                  |                                  |                                  |
| <input type="text"/>                          |                                  |                                  |
| Numéros de la carte                           |                                  |                                  |
| <input type="text"/>                          |                                  |                                  |
| Date d'expiration                             |                                  |                                  |
| <input type="text"/>                          |                                  |                                  |
| Code de sécurité                              |                                  |                                  |
| <input type="text"/>                          |                                  |                                  |
| <input type="button" value="Validé l'achat"/> |                                  |                                  |



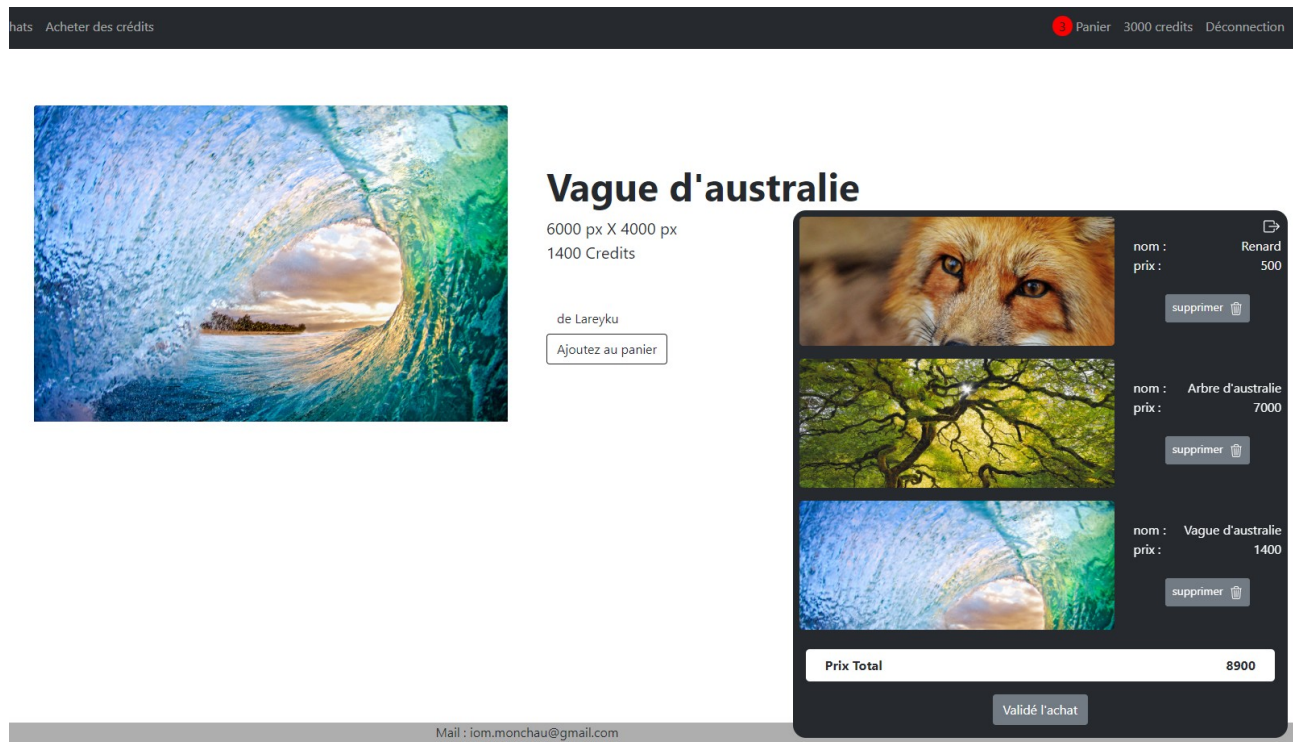
### Acheter des crédits

|   |                                  |                                 |
|---|----------------------------------|---------------------------------|
| <input type="text" value="3000"/>             | <input type="button" value="↔"/> | <input type="text" value="60"/> |
| Crédits                                       |                                  | EUR                             |
| RIB   |                                  |                                 |
| <input type="text"/>                          |                                  |                                 |
| <input type="button" value="Validé l'achat"/> |                                  |                                 |



### 4.3. Le panier

Les clients peuvent ajouter des images au panier, dès qu'un élément est ajouté au panier, un lien dans la barre de navigation apparaît avec le nombre d'éléments dans le panier sur un fond rouge.



Dans panier.js

`togglePanier() : void`

Si le panier est affiché, le panier disparaît avec une animation vers la droite.

Si le panier n'est pas affiché, le panier apparaît depuis la droite et prend un % de l'écran en fonction de la taille de l'écran.

`ajouterAuPanier() : void`

Cette fonction récupère les données de l'image que l'on veut ajouter au panier et lance la fonction `addPanier()` avec celle-ci (sous forme de JSON), puis lance la fonction `actuPanier()`.

`actuPanier() : void`

Change le contenu HTML du panier, permet d'actualiser le panier après chaque modification. Cette fonction est appelée au chargement de chaque page.

`savePanier(panier) : void`

Sauvegarde la variable panier dans localStorage.

getPanier() : Array

Retourne localStorage['panier'], un tableau d'objet, chaque objet correspondant à une image dans le panier. Si le panier est vide, renvoi un tableau vide.

addPanier(produit) : void

exemple de produit : {id:1, nom:'aaa', prix:400, url:'aaa', categorie:'aaa'}

Si le produit existe déjà dans localStorage, une alerte apparaît avec un message d'erreur, sinon le produit est ajouter au localStorage

removePanier(produit):void

exemple de produit : {id:1}

Retire le produit du panier dans localStorage.

getPrixTotal():int

Retourne le prix total des produits dans le panier.

getNombreProduit():int


Retourne le nombre de produits dans le panier.

#### 4.4. Profil Utilisateur

Si un client consulte son profil, il peut voir les images qu'il a précédemment achetées.

Si le compte consulté est un compte photographe, les photos actuellement en vente par celui-ci sont afficher.

De plus si le compte consulté correspond à celui connecter, l'utilisateur peut modifier les informations de son compte.



Belorik


Editer le profil

0  
Photos

Mes information

|           |                  |
|-----------|------------------|
| Nom :     | Duval            |
| Prenom :  | Jordan           |
| Pseudo :  | Belorik          |
| Mail :    | client@gmail.com |
| Credits : | 3000             |

Photos achetée



Lion

## Mes information

|           |                       |
|-----------|-----------------------|
| Nom :     | Maroud                |
| Prenom :  | Kevin                 |
| Pseudo :  | Lareyku               |
| Mail :    | photographe@gmail.com |
| Credits : | 300                   |

## Photos en ventes



### Corail

300 crédits

Océan



### Corail étrange

500 crédits

Océan



### Fond marin

250 crédits

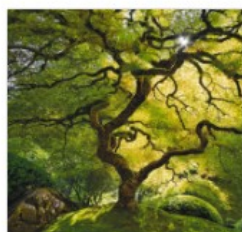
Océan



### Paysage Canadien

1100 crédits

Paysage



### Arbre d'australie

7000 crédits

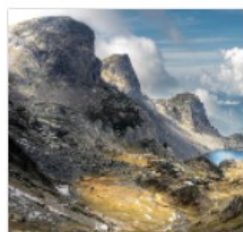
Paysage



### Lac d'hiver

600 crédits

Paysage



### Massif de Belledonne

2200 crédits

Paysage



### Aigle

1700 crédits


Animaux

## 4.5. Historique Achat / Vente

Selon la personne qui est connectée l'historique disponible est différent.

Client :

### Mes achats

| Aperçu  | Nom de l'image | Date               | Pseudo du vendeur | Prix (Crédits) |                                     |
|---|----------------|--------------------|-------------------|----------------|-------------------------------------|
|  | Lion           | 18:07 - 02-05-2022 | Lareyku           | 300            | <a href="#">telecharger l'image</a> |

1

Photographe :


### Mes ventes

| Aperçu  | Nom de l'image | Date               | Pseudo de l'acheteur | Prix (Crédits) |
|---|----------------|--------------------|----------------------|----------------|
|  | Lion           | 18:07 - 02-05-2022 | Belorik              | 300            |

1

Administrateur :

### Historique Achat/Vente

| Aperçu  | Nom de l'image | Date               | Pseudo de l'acheteur | Pseudo du vendeur | Prix (Crédits) |
|---|----------------|--------------------|----------------------|-------------------|----------------|
|  | Lion           | 18:07 - 02-05-2022 | Belorik              | Lareyku           | 300            |

1

## 4.6. Déposer des images

Quand on choisit une image via le formulaire, l'image est prévisualisée sur la droite.

L'extension de l'image doit être jpeg, jpg ou png, et le prix entre 250 et 25 000 crédits.

### Déposer des images

Nom

Catégorie

Veillez choisir une catégorie ▾

Image

Choisir un fichier corail habité.jpg

Prix (entre 250 et 25 000 crédits)

Publier



## 4.7. Gestion utilisateur

Cette page n'est accessible que pour ceux qui ont le grade admin.

Elle permet de modifier l'état des utilisateurs depuis les boutons à droite qui pointe vers `modif-etat-user.php` avec des paramètres GET en fonction des modifications à faire.

### Gestion Utilisateur

| Id | Nom    | Prenom | Mail                  | Grade       | Credits | Etat     |
|----|--------|--------|-----------------------|-------------|---------|----------|
| 2  | Maroud | Kevin  | photographe@gmail.com | photographe | 300     | valid    |
| 3  | Duval  | Jordan | client@gmail.com      | client      | 3000    | valid    |
| 17 | Dupres | Simon  | zekles@gmail.com      | photographe | 0       | attValid |
| 16 | Filon  | Jean   | jean@gmail.com        | client      | 0       | banni    |

modif-etat-user.php

```
1  <?php
2      session_start();
3      if(empty($_SESSION['grade'])){
4          header('location:index.php');
5          exit();
6      }
7      if($_SESSION['grade'] != 'admin'){
8          header('location:index.php');
9          exit();
10     }else{
11
12         include "includes/bdd.php";
13
14         if (isset($_GET["id"])) {
15             if (isset($_GET["etat"])) {
16
17                 $id = $_GET["id"];
18                 $etat = $_GET["etat"];
19                 if($etat == 'valid'){
20                     // Valid devient banni
21                     $req = $bdd->prepare("UPDATE user SET etat='banni' WHERE id = '$id'");
22                     $req->execute();
23                 }
24                 if($etat == 'attValid'){
25                     // Attente de validation devient valid
26                     $req = $bdd->prepare("UPDATE user SET etat='valid' WHERE id = '$id'");
27                     $req->execute();
28                 }
29                 if($etat == 'banni'){
30                     // Banni devient valid
31                     $req = $bdd->prepare("UPDATE user SET etat='valid' WHERE id = '$id'");
32                     $req->execute();
33                 }
34                 header("location:gestion-utilisateur.php");
35             }
36         }
37     }
```



## 4.8. gestion catégorie

Cette page n'est accessible que pour ceux qui ont le grade admin.  
Elle permet de créer des catégories et d'en supprimer.  
Seules les catégories qui n'ont pas d'image peuvent être supprimé.

### Gestion Catégorie

Nom de la catégorie

Ajoutez

| Nom des catégories | Supprimer                 |
|--------------------|---------------------------|
| Animaux            | <a href="#">supprimer</a> |
| Océan              | <a href="#">supprimer</a> |
| Paysage            | <a href="#">supprimer</a> |

127.0.0.1 indique

Vous ne pouvez pas supprimer cette catégorie, il y a encore des images qui en font partie.

OK

Comme pour la gestion des utilisateurs, la suppression des utilisateurs se fait sur la page `suppr-categorie.php`

```
if (isset($_GET["nom_categorie"])) {  
    $nom_categorie = $_GET["nom_categorie"];  
  
    $req = $bdd->query("SELECT * FROM image WHERE nom_categorie = '$nom_categorie'");  
    $nbImg = $req->rowCount();  
    if ($nbImg == 0) {  
        /* Supprime dans la bdd la categorie  
        $req = $bdd->prepare("DELETE FROM categorie WHERE nom_categorie = :nom_categorie");  
        $req->bindValue(':nom_categorie', $nom_categorie);  
        $req->execute();  
        /* supprime le dossier de la categorie  
        rmdir('upload/'.str_replace(" ", "-", $nom_categorie));  
        header("location:gestion-categorie.php");  
    } else {  
        echo "<script>alert('Vous ne pouvez pas supprimer cette catégorie, il y a encore des images q
```

4.9. Inscription

# Inscription

Nom

Prenom

Pseudo

Adresse Email

Image (pas obligatoire)

Choisir un fichier

Aucun fichier choisi

Mot de passe

Mot de passe


☒photographe☐client

Num SIRET

Image (pas obligatoire)

Choisir un fichier

cat-55824...340.webp



Mot de passe

Mot de passe

☐photographe☒client

S'inscrire

```

$nom = htmlspecialchars($_POST['nom']);
$prenom = htmlspecialchars($_POST['prenom']);
$mail = htmlspecialchars($_POST['mail']);
$grade = htmlspecialchars($_POST['grade']);
$pseudo = htmlspecialchars($_POST['pseudo']);

if(strlen($nom) < 3 && strlen($nom) > 15){ $validation = False; }
if(strlen($prenom) < 3 && strlen($prenom) > 15){ $validation = False; }
if(strlen($pseudo) < 3 && strlen($pseudo) > 15){ $validation = False; }
if(filter_var($mail, FILTER_VALIDATE_EMAIL) == False) { $validation = False; }

/* Vérification des doublons d'adresse mail
$req = $bdd->prepare("SELECT * FROM user WHERE mail = '$mail'");
$req->execute();
$result = $req->rowCount();
if($result > 0){ $validation = False; $mes_error = '<br/>Cette adresse mail est déjà utilisé.'; }

/* Vérification des doublons de pseudo
$req = $bdd->prepare("SELECT * FROM user WHERE pseudo = '$pseudo'");
$req->execute();
$result = $req->rowCount();
if($result > 0){ $validation = False; $mes_error = '<br/>Ce pseudo est déjà utilisé.'; }

/* Vérification des doublons de num SIRET
if(!empty($_POST['siret'])){
    $siret = $_POST['siret'];

    $req = $bdd->prepare("SELECT * FROM user WHERE SIRET = '$siret'");
    $req->execute();
    $result = $req->rowCount();
    if($result > 0){ $validation = False; $mes_error = '<br/>Ce numéro de SIRET est déjà utilisé.'; }
}

if($_POST['password1'] != $_POST['password2']){ $validation = False;

```

## 4.10. Connexion

# Connexion

Adresse Email

Mot de passe

Se connecter

[Vous n'avez pas de compte, inscrivez-vous !](#)

```
$mail = $_POST['mail'];
$mdp = sha1($_POST['password']);

$co = $bdd->prepare("SELECT mail, mdp FROM user WHERE mail = :mail AND mdp = :mdp");
$co->bindParam(':mail', $mail, PDO::PARAM_STR);
$co->bindParam(':mdp', $mdp, PDO::PARAM_STR);
$co->execute();
$user_exist = $co->rowCount();

if($user_exist == 1){
    /* Verifiacion de l'etat du comte : attValid / valid / banni
    $req = $bdd->prepare("SELECT etat FROM user WHERE mail = '$mail'");
    $req->execute();
    $result = $req->fetch(PDO::FETCH_OBJ);
    $etat = $result->etat;

    if($etat == 'attValid'){
        $mess = 'Votre comte est en cour de validation.';
    }
    if($etat == 'banni'){
        $mess = 'Votre comte à été banni.';
    }
}

/* Si le compte est valid
if($etat == 'valid'){
    session_start();
    $_SESSION['id'] = $id;
    $_SESSION['grade'] = $grade;
    $_SESSION['mail'] = $mail;
    $_SESSION['pseudo'] = $pseudo;
    header("location:index.php");
```