

2022

PhotoForYou

DOCUMENTATION TECHNIQUE
CLEMENT MOYNARD

BTS SIO 2 – LYCEE PAUL SABATIER - 2022

Sommaire

| | |
|---|----|
| Description du site..... | 1 |
| Contexte | 1 |
| Expression des besoins..... | 2 |
| Exigences fonctionnelles | 3 |
| Partie analyse – Diagrammes | 5 |
| MLD | 5 |
| SCD | 5 |
| Cas d'utilisations..... | 6 |
| Description de l'environnement | 7 |
| Environnement utilisé | 7 |
| Installation de l'environnement..... | 7 |
| Installation du site et de la BDD : | 8 |
| Présentation du site | 10 |
| Exemples de pages | 10 |
| Exemple de code | 14 |
| Difficultés rencontrées | 17 |
| Changements à prévoir | 17 |
| Annexe..... | 18 |
| Github du projet : | 18 |

Description du site

Contexte

La société *AtelierLumière* est un laboratoire photographique situé à Paris, créé en 2006. Composé de 4 personnes, *AtelierLumière* est spécialisé dans le tirage photographique et la retouche pour les professionnels.

Ses secteurs d'activités sont les suivant :

- Numérisation de tout support
- Retouche créative
- Tirages multiples lors d'expositions

La société *AtelierLumière* véhicule son image autour de l'artisanal, avec une solide réputation chez les photographes auteurs comme chez les professionnels de la mode.

Afin de diversifier son activité et faire profiter son savoir faire à un plus grand nombre : *AtelierLumière* voudrait proposer la vente en ligne de photographies d'auteurs.

La vente en ligne de photographies est déjà assurée par d'autres concurrents, il sera donc important de se démarquer en se basant sur la notoriété de la société mais aussi sur l'aspect photographie d'art.

Expression des besoins

1. **Objectifs :**

Vente et achat de photographies.

Attirer de nouveaux clients, augmenter les bénéfices et le CA.

2. **Cibles et caractéristiques :**

Vendeurs : photographes d'auteurs (travaillant ou non avec la société)

Acheteurs : professionnels (magazines, industriels), particuliers...

3. **Description du contenu**

- Enregistrement sur le site en tant que photographe ou client.
- La désinscription (avec la perte des crédits et des photos de catalogue si photographe)
- Catalogue de photos à vendre
- Les photos achetées sont archivées sur le profil client, et retiré du catalogue après achat.
- Le photographe fixe le prix de vente d'une photo et touche 70% des crédits à l'achat.
- L'achat se fait avec des crédits, ceux-ci peuvent être achetés (1 crédit = 1 euro)

Exigences fonctionnelles

1. **Principales composantes :**

- Gestion des utilisateurs (Bannissement, suppression définitif)
- Identification des utilisateurs
- Gestion de l'espace client
- Gestion de l'espace photographe
- Gestion du catalogue

2. **Fonctionnalités en Front :**

- Gérer les identifications des utilisateurs, utiliser des mots de passe cryptés.
- Affichage des photos et de leurs caractéristiques dans une bonne organisation, avec :
Nom de la photo, Description, Prix, Nom de l'auteur.
- Gestion de l'espace photographe avec affichage des photos déposés et vendus, et ajout des photos pour le catalogue.
- Gestion de l'espace client avec achat de photos avec assez de crédits, affichage des photos achetés, rajout de crédits sur le compte, suppression du compte.
- Gestion des commandes, avec le débit du compte client des crédits lors de l'achat, et crédit sur le compte du photographe de 70% du prix.
- Gestion du catalogue avec des vignettes montrant les photos avec leurs informations et leur prix en crédit et l'achat d'une photo d'un simple clic, suppression des photos déjà achetés de la galerie.

3. **Fonctionnalités en Back :**

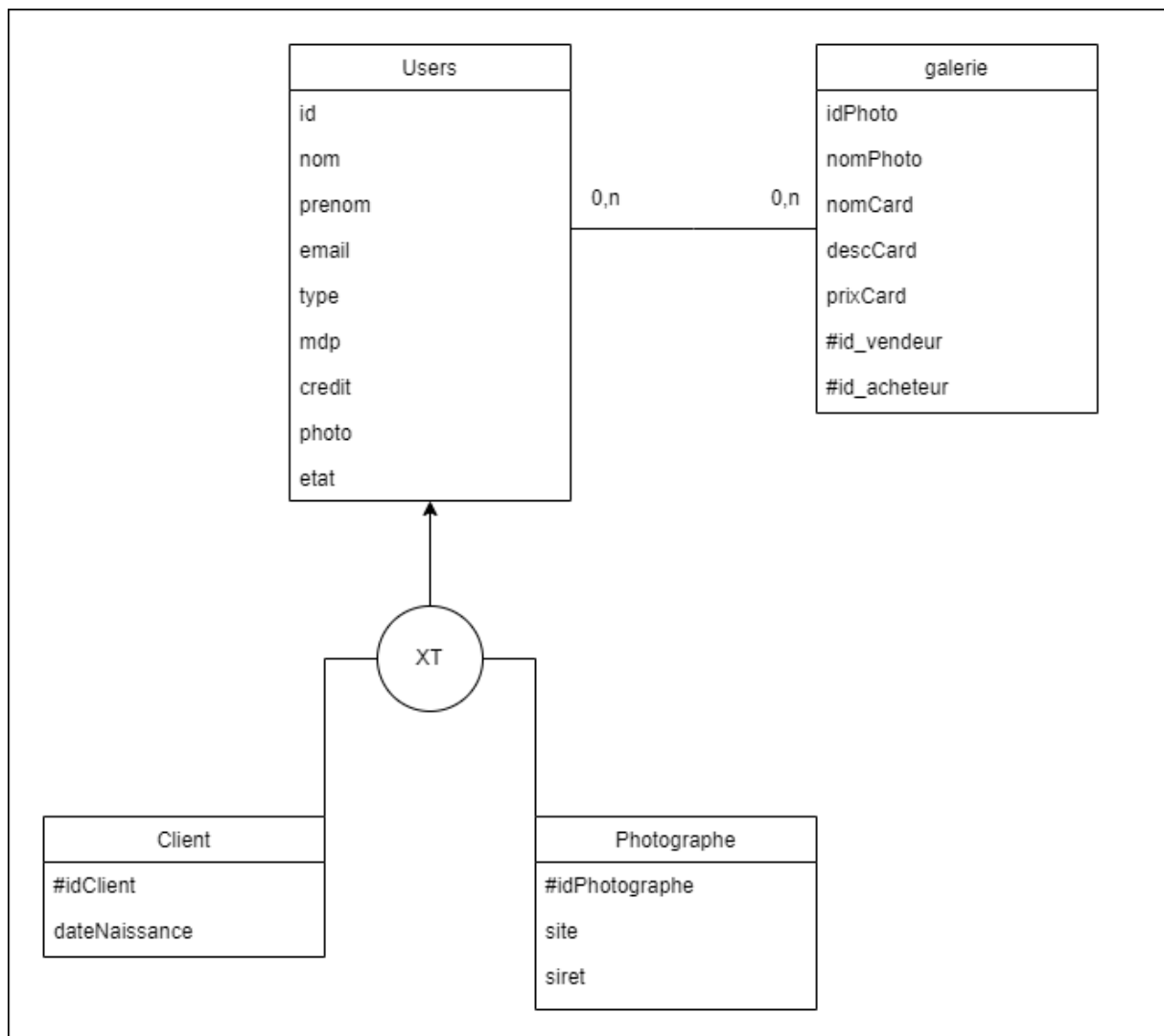
- Gestion des utilisateurs avec système de bannissement et de suppression d'un compte, ce qui entraîne la suppression de toutes les photos postées si le compte supprimé est un photographe.
- Gestion du catalogue avec Ajout ou Suppression de photos.

Partie analyse – Diagrammes

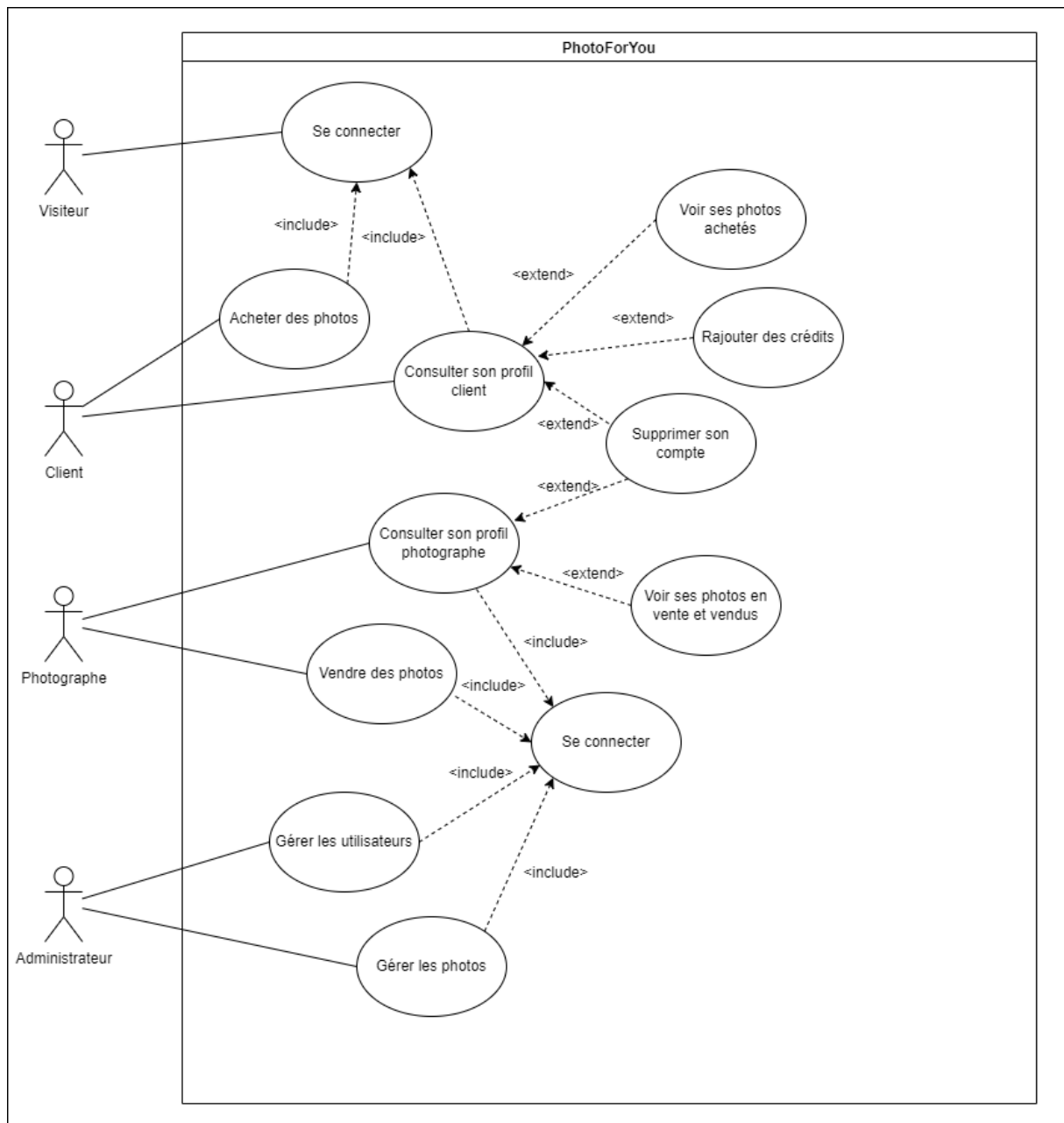
MLD

- Users (id, nom, prenom, email, type, mdp, credit, photo, etat)
- Galerie (idPhoto, nomPhoto, nomCard, descCard, prixCard, #id_vendeur, #id_acheteur)

SCD



Cas d'utilisations



Description de l'environnement

Environnement utilisé

Site web effectué seul en procédural, avec essentiellement du :

- **FRONT** : HTML, CSS, BOOTSTRAP
- **BACK** : PHP, JS, JQuery

Sur Windows en local, à l'aide de WampServer et de PhpMyAdmin pour la partie base de donnée.

Installation de l'environnement

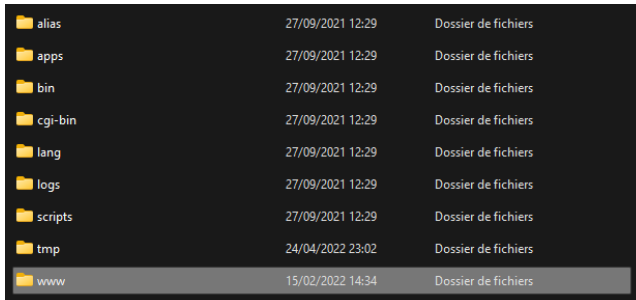
Télécharger wampserver sur le site officiel : <http://www.wampserver.com/fr/>

Lancer le programme d'installation et suivre les étapes comme ci-dessous

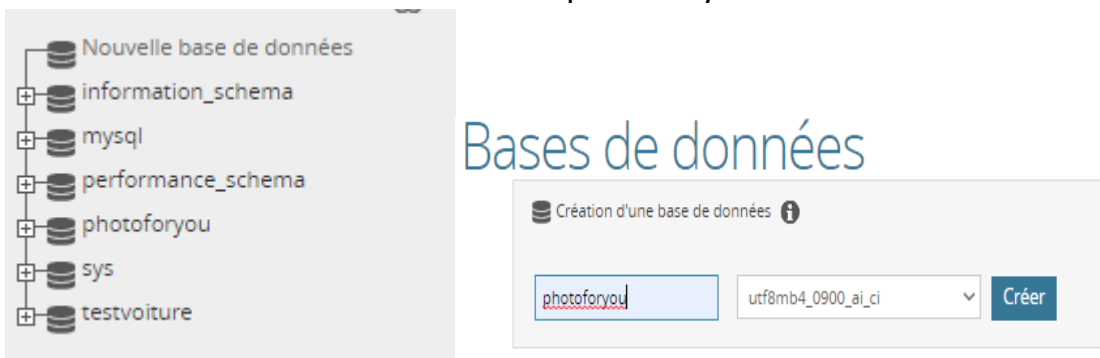
- Accepter les termes du contrat de License et Suivant
- Accepter les prérequis de Wampserver, avec les paquetages Visual Studio 9, 10, 11, 13, 14, 15 et Suivant
- Sélectionner l'emplacement de wamp dans le disque C:/ et Suivant
- Installer les dernières versions de PHP et la dernière version de MySQL
- Suivant sur le dossier du menu démarrer
- Installer pour lancer l'installation
- A la fin de l'installation, il vous sera demandé quel navigateur vous souhaitez utiliser, choisissez celui par défaut ou changez-le puis Suivant
- Terminer pour finaliser l'installation
- Lancer Wamp en tant qu'administrateur et aller à l'url <http://localhost> pour arriver sur l'interface wamp.

Installation du site et de la BDD :

- Aller à l'emplacement d'installation de wamp et ouvrir le dossier /www



- Dans ce dossier créer un dossier PhotoForYou et y mettre l'archive comprenant le site (Lien en annexe)
- Dans PhotoForYou, récupérer photoforyou.sql et le garder de côté
- Lancer phpmyadmin à l'url <http://localhost/phpmyadmin>
- Se connecter avec root / *pas de mot de passe*
- Créer une nouvelle base de données photoforyou



- Après l'avoir créé, aller dans Importer et importer dans cette BDD vide le fichier photoforyou.sql

Importation dans la base de données « photoforyou »

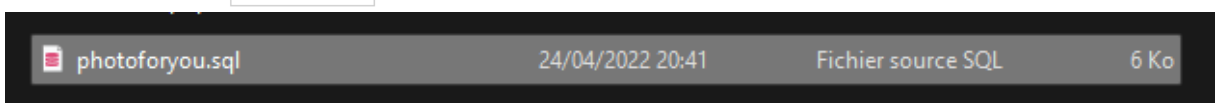
FICHIER À IMPORTER :

Le fichier peut être compressé (gzip, bzip2, zip) ou non.
Le nom du fichier compressé doit se terminer par .[format].[compression]. Exemple : .sql.zip

Parcourir les fichiers : Aucun fichier choisi (Taille maximale : 128Mio)


Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier :




- Après avoir installé la base de données, il faut créer un utilisateur pour la base de données qui aura seulement les droits de SELECT, UPDATE et DELETE (le nom de l'utilisateur et le mot de passe se trouve dans le fichier include/access_gestuser.php, dans PhotoForYou)


Nouvel utilisateur


 [Ajouter un compte d'utilisateur](#)


Informations pour la connexion

Nom d'utilisateur : Saisir une valeur: 

⚠ Un compte existe déjà avec le même nom d'utilisateur, mais éventuellement un nom d'hôte différent.

Nom d'hôte : Local 

Mot de passe : Saisir une valeur:  Strength: Très faible

Saisir à nouveau : 

Privileges globaux ☒ Tout cocher

NB : les noms de privileges sont exprimes en anglais.

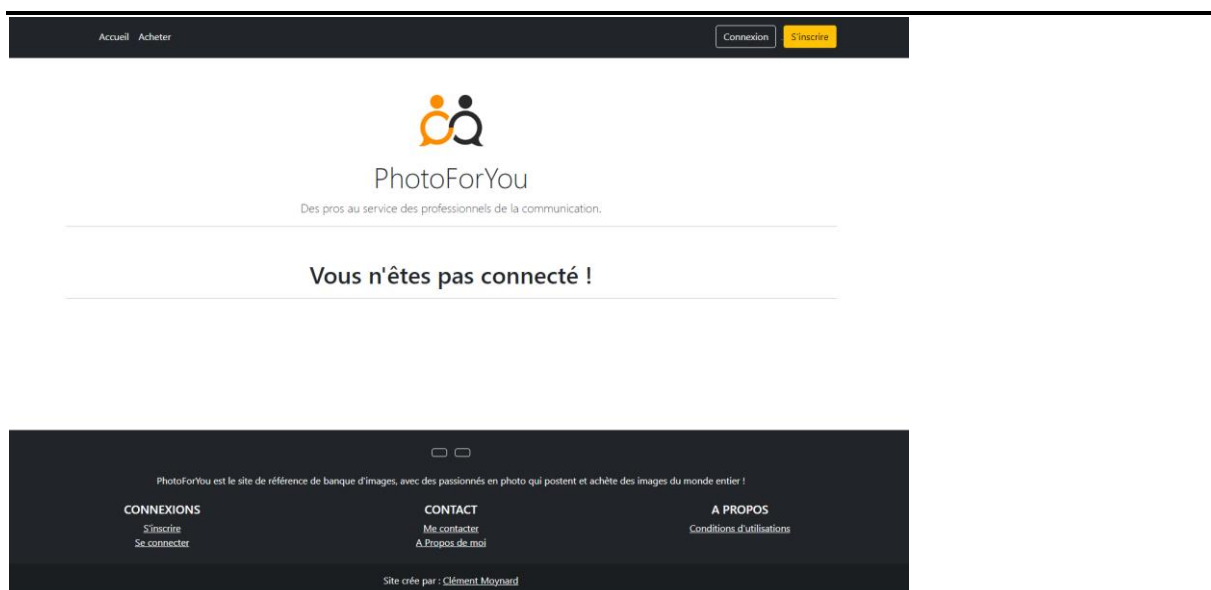
| | | | |
|--|---|---|---|
| <input checked="" type="checkbox"/> Données | <input type="checkbox"/> Structure | <input type="checkbox"/> Administration | Limites de ressources |
| <input checked="" type="checkbox"/> SELECT <input type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input type="checkbox"/> FILE | <input type="checkbox"/> CREATE <input type="checkbox"/> ALTER <input type="checkbox"/> INDEX <input type="checkbox"/> DROP <input type="checkbox"/> CREATE TEMPORARY TABLES <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> CREATE ROUTINE <input type="checkbox"/> ALTER ROUTINE <input type="checkbox"/> EXECUTE <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> EVENT <input type="checkbox"/> TRIGGER | <input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES <input type="checkbox"/> REPLICATION CLIENT <input type="checkbox"/> REPLICATION SLAVE <input type="checkbox"/> CREATE USER | <small>NB : une valeur de 0 (zéro) lève la limite.</small> MAX QUERIES PER HOUR <input type="text" value="0"/> MAX UPDATES PER HOUR <input type="text" value="0"/> MAX CONNECTIONS PER HOUR <input type="text" value="0"/> MAX USER_CONNECTIONS <input type="text" value="0"/> SSL <input checked="" type="radio"/> REQUIRE NONE <input type="radio"/> REQUIRE SSL <input type="radio"/> REQUIRE X509 <input type="radio"/> SPECIFIED REQUIRE CIPHER <input type="text"/> REQUIRE ISSUER <input type="text"/> REQUIRE SUBJECT <input type="text"/> |

- Quand tout est fini, lancer le site à l'url <http://localhost/PhotoForYou>

Présentation du site

Exemples de pages

Page d'accueil visiteur



Page d'inscription

FORMULAIRE D'INSCRIPTION :

Votre Nom

Votre prénom

Votre email

Mot de passe (8 caractères MIN et 20 MAX)

Confirmez votre mot de passe

☒ Client ☐ Photographe

Date de naissance

Choisir un fichier

Vous avez déjà un compte ? [Connectez-vous !](#)

Page profil de photographe (sans images en vente)

Votre profil de photographe



Bonjour Chovanec !

Nom : Chovanec

Prénom : Jeremy

Courriel : grosbg@gmail.com

Vos crédit(s) : 3954

Rajouter des crédits.

! Supprimer votre compte !

Page de vente de photos

Ajouter une photo dans la galerie

Vous touchez 80% sur chaque photo vendu !

Nom de la Photo :

Description de la photo :

Prix de vente de la photo :

Choisir un fichier

Aucun fichier choisi

Ajouter votre photo !

Annuler la vente

Page d'achats de photos

Parcourir la galerie

Connecté en tant que Chovanec Jeremy

Vos crédits(s) : 3954



Vallée Rocheuse

Des rochers qui font plaisir #moumou.

Acheter pour 30 crédits



Daniel Craig

Le célèbre acteur qui incarne James Bond dans les films du même nom.

Acheter pour 75 crédits



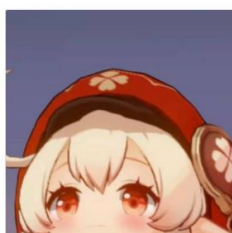
Doigby

Le streamer qui a été le GOAT durant le ZEVENT 2021

Acheter pour 400 crédits



...



Dodoko

Un personnage du jeu Genshin Impact

Acheter pour 800 crédits



Terry

Un homme extrêmement beau, donc qui vaut très chère

Acheter pour 9000 crédits

Page de profil client (avec photos achetées)

Votre profil de client



Bonjour MONROCQ !

Nom : MONROCQ

Prénom : Ugo

Courriel : aaaa@aa.fr

Vos crédit(s) : 0

Rajouter des crédits.

Photos achetées :



Page profil photographe avec aperçu des photos en ventes et déjà vendus

Photos en vente :



Photos vendus :



Page admin gestion des utilisateurs

Gestion Utilisateur

| Id | Nom | Prenom | Mail | Grade | Credits | Etat | Supprimer |
|----|----------|----------|------------------|-------------|---------|-------|-----------|
| 5 | Chovanec | Jeremy | grosbg@gmail.com | photographe | 3954 | valid | Supprimer |
| 10 | Evin | Baptiste | baptbg@gmail.com | photographe | 4610 | valid | Supprimer |
| 12 | MONROCQ | Ugo | aaaa@aa.fr | client | 0 | banni | Supprimer |

Exemple de code

Code de la fonction de bannissement et suppression Partie Admin

```

if (isset($_GET["id"])) {

    if (isset($_GET["etat"])) { //on vérifie que l'utilisateur a bien toute les données

        $id = $_GET["id"];
        $etat = $_GET["etat"];
        if($etat == 'valid'){
            // Valid devient banni
            $req = $db->prepare("UPDATE photoforyou.users SET etat='banni' WHERE id = :id");
            $req->bindValue(':id', $id, PDO::PARAM_INT);
            $req->execute();
        }

        if($etat == 'banni'){
            // Banni devient valid
            $req = $db->prepare("UPDATE photoforyou.users SET etat='valid' WHERE id = :id");
            $req->bindValue(':id', $id, PDO::PARAM_INT);
            $req->execute();
        }

        if ($etat == 'suppr') { //si son etat est suppr

            $requete = 'SELECT nomPhoto FROM photoforyou.galerie where id_vendeur = :id';
            $instruction = $db->prepare($requete);
            $instruction->bindParam(':id', $id, PDO::PARAM_STR);
            $instruction->execute();
            $result = $instruction->fetchAll();
            foreach ($result as $ligne) {
                unlink("images/galerie/".$ligne['nomPhoto']); //on supprime les photos en local
            }

            $req = $db->prepare("DELETE FROM photoforyou.galerie WHERE id_vendeur = :id");
            $req->bindValue(':id', $id, PDO::PARAM_INT);
            $req->execute(); //on supprime les photos de la bdd

            $req = $db->prepare("DELETE FROM photoforyou.users WHERE id = :id");
            $req->bindValue(':id', $id, PDO::PARAM_INT);
            $req->execute(); //on supprime l'utilisateur de la bdd
        }
        header("location:gestionUtilisateur.php");
    }
}

```

Code des fonctions d'affichage des photos en vente et vendus photographe

```
if ($_SESSION['type'] == 'photographe') {

    $query = "SELECT * from photoforyou.galerie where id_vendeur = " . $_SESSION['idUtilisateur'] . ";";
    $requete = $db->query($query);
    $result = $requete->fetchAll();
    $photos_vendus = [];

    if (!empty($result)) { //si le photographe a des photos en vente
        echo '<hr class="my-4">';
        echo '<p class="lead">Photos en vente : </p>';
        echo '<div class="row row-cols-6">';

        foreach ($result as $ligne) { //pour chaque photo mise en vente par le photographe
            if ($ligne['id_acheteur'] == NULL) {
                echo '<div class="col">';
                echo "&<img src='images/galerie/" . htmlentities($ligne['nomPhoto']) . "' id='photo' class='img-responsive float-left h-100 w-100' >" ; //affichage de la photo
                echo '</div>';
            } else {
                array_push($photos_vendus, $ligne); //on stocke les photos déjà vendues dans un tableau
            }
        } echo '</div>';
    }
    if (!empty($photos_vendus)) { //si le photographe a des photos vendues

        echo '<hr class="my-4">';
        echo '<p class="lead">Photos vendus : </p>';
        echo '<div class="row row-cols-6">';

        foreach ($photos_vendus as $ligne) { //pour chaque photo vendue par le photographe
            echo '<div class="col">';
            echo "&<img src='images/galerie/" . htmlentities($ligne['nomPhoto']) . "' id='photo' class='img-responsive float-left h-100 w-100' >" ; //affichage de la photo
            echo '</div>';
        } echo '</div>';
    }
}
```

Code de la fonction d'affichage des photos achetées par le client

```
} else {

    $query = "SELECT * from photoforyou.galerie where id_acheteur = " . $_SESSION['idUtilisateur'] . ";";
    $requete = $db->query($query);
    $result = $requete->fetchAll();

    if (!empty($result)) { //si le visiteur a des photos achetées
        echo '<hr class="my-4">';
        echo '<p class="lead">Photos achetées : </p>';
        echo '<div class="row row-cols-6">';

        $query = "SELECT * from photoforyou.galerie where id_acheteur = " . $_SESSION['idUtilisateur'] . ";";
        $requete = $db->query($query);
        $result = $requete->fetchAll();

        foreach ($result as $ligne) { //pour chaque photo achetée par le visiteur
            echo '<div class="col">';
            echo "&<img src='images/galerie/" . htmlentities($ligne['nomPhoto']) . "' id='photo' class='img-responsive float-left h-100 w-100' >" ; //affichage de la photo
            echo '</div>';
        }
    }
}
```


Code de la fonction d'achat de photo sur la galerie

```

if (isset($_POST['acheter']))
{
    $creditAchat = intval($_POST['nbCredit']);
    $idAchat = $_POST['idAchat'];
    $requete = "SELECT * from photoforyou.galerie where idPhoto = :idPhoto";
    $instructions = $db->prepare($requete);
    $instructions->bindParam(':idPhoto', $idAchat, PDO::PARAM_INT);
    $instructions->execute();
    $num = $instructions->fetchAll();
    try {
        if ($_SESSION['credit'] >= $creditAchat AND count($num) > 0)
        { //on vérifie que l'utilisateur a bien toutes les données

            $instruction = $db->prepare("UPDATE users SET credit = credit - :credit WHERE id = " . $_SESSION['idUtilisateur'] . ";");
            $instruction->bindParam(':credit', $creditAchat, PDO::PARAM_INT);
            $instruction->execute();

            $instruction = $db->prepare("UPDATE galerie SET id_acheteur = :id_acheteur WHERE idPhoto = :idAchat;");
            $instruction->bindParam(':id_acheteur', $_SESSION['idUtilisateur'], PDO::PARAM_INT);
            $instruction->bindParam(':idAchat', $idAchat, PDO::PARAM_INT);
            $instruction->execute(); //on met à jour la photo achetée en bdd pour l'enlever de la galerie et l'afficher sur le profil

            $query = "SELECT * from photoforyou.users where id = " . $_SESSION['idUtilisateur'] . ";";
            $requete = $db->query($query);
            $result = $requete->fetch();
            $_SESSION['credit'] = htmlentities($result['credit']); //on met à jour le nombre de crédits de l'utilisateur
            unset($result);
            echo "<script>";
            alert("Votre achat a été effectué !");
            location.href="profil.php";
            </script>;
        }
        else
        {
            echo "<script>";
            alert("Vous n'avez pas assez de crédits. Vous pouvez en rajouter sur votre profil !");
            location.href="galerie.php";
            </script>;
        }
    } catch (PDOException $e) {
        echo "<div>Erreur : </div> . $e->getMessage();";
        var_dump($_POST);
    }
}

```

Code de la fonction d'affichage des photos dans le catalogue

```

<div class="container">
    <div class="jumbotron">
        <h1 class="display-4">Parcourir la galerie</h1>
        <?php echo "<p class='lead'>Connecté en tant que '$_SESSION['nomUtilisateur']' . '$_SESSION['prenomUtilisateur']' . '</p>?";
        <?php echo "<p class='lead'>Vos crédits(s) : " . $_SESSION['credit'] . "</p>?";
        <hr class="my-4">
    </div>
</div>

<div class="container">
    <div class="row row-cols-1 row-cols-md-3">
        <?php
            $query = "SELECT * from photoforyou.galerie where id_acheteur IS NULL;";
            $requete = $db->query($query);
            $result = $requete->fetchAll();
            foreach ($result as $ligne) { //affichage des photos non achetées
                $galPhoto = "images/galerie/" . htmlentities($ligne['nomPhoto']); //pour chaque élément dans le tableau, on récupère chaque info et on récupère la photo dans le dossier d'image

                echo //on génère la card de la photo, avec toute les infos de cette dernière dans la bdd
                "<div class='col-sm-3 mb-3'>
                    <form action='achat.php' method='POST'>
                        <div class='card'>
                            <input type='number' name='nbCredit' value='$_.$ligne['prixCard']' style='display: none;'/>
                            <input type='hidden' name='idAchat' value='$_.$ligne['idPhoto']' />
                            <img src='$_.$galPhoto' class='card-img-top' alt='$_.$ligne['idPhoto']' />
                            <div class='card-body'>
                                <h5 class='card-title'>$_.$ligne['nomCard']</h5>
                                <p class='card-text'>
                                    $_.$ligne['descCard']
                                </p>
                                <button type='submit' value='acheter' name='acheter' class='btn btn-success'>Acheter pour " . $_.$ligne['prixCard'] . " crédits</button>
                            </div>
                        </div>
                    </form>
                </div>;
            }
        <?php
    </div>
</div>

```

Difficultés rencontrées

La difficulté principale a été de créer un site avec un semblant de responsive, ce qui n'est pas le cas de toutes les pages, l'utilisation de Bootstrap pour la mise en page a été d'une grande aide, surtout pour la partie esthétique.

Les pages d'affichages des photos sur les profils ont été particulièrement dures à coder, car ma connaissance PHP n'était pas assez poussée à l'époque où je l'ai codé, mais elle a été retravaillée et est beaucoup plus intuitive maintenant.

Lier les photos de la galerie avec les Clients et les Photographes ont été assez corsés, sans aucune utilisation de clé primaires et secondaires.

Le système de catégorie n'a pas été fait, car je n'y arrivais absolument pas, cette tâche est à prévoir.

Changements à prévoir

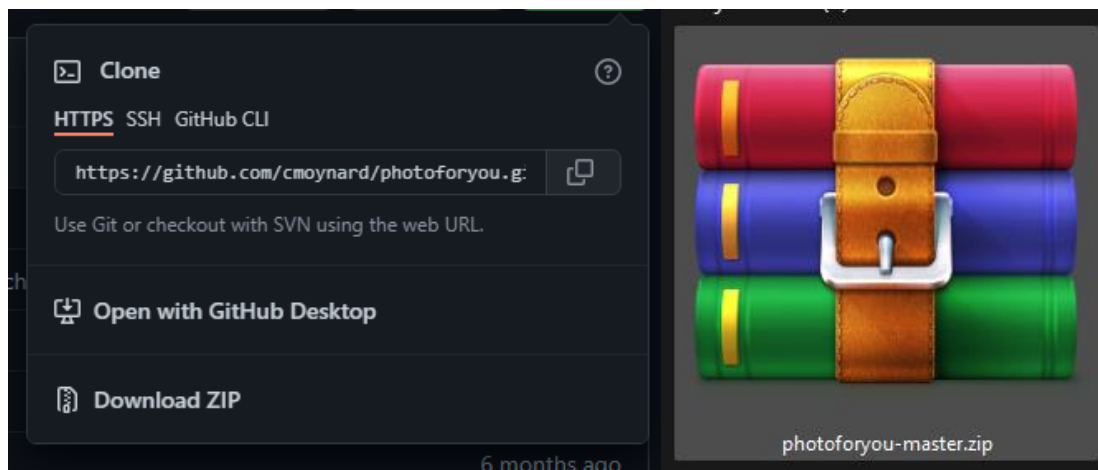
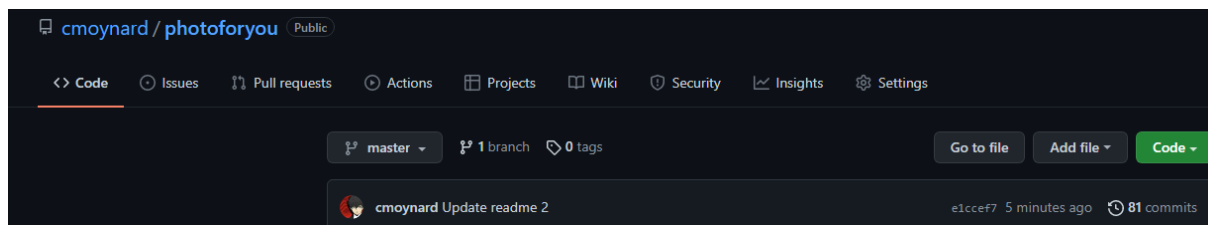
- Système de catégorie pour les photos
- Créditer le photographe quand une de ses photos est achetée
- Description détaillée de l'image avant achat
- Système de panier
- Système de vérification en attente pour un compte photographe, validé ou non par un admin.

Annexe

Github du projet

>> [<< https://github.com/cmoynard/photoforyou](https://github.com/cmoynard/photoforyou)

Pour télécharger le site via ce lien, cliquer sur Code en vert, puis sur Download ZIP :



Le site doit être ensuite extrais du fichier compressé.