**Manuel technique "Bibliothèque Sterling"**

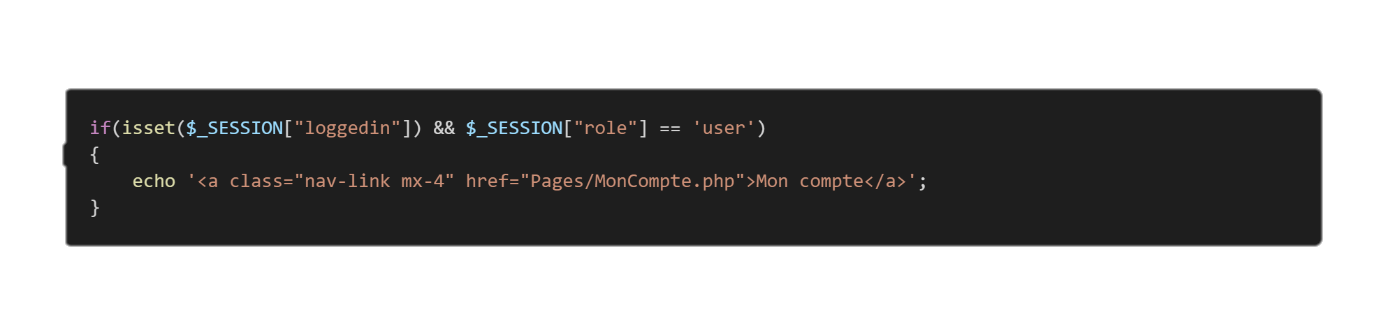
**Généralités** (index.php)

Le frontend du site est codé en HTML5, CSS3 et Bootstrap 4.3.1 ainsi qu'en JavaScript. La partie backend est créée à partir de PHP 7.4.26. Les données du site, notamment les informations sur les comptes, c’est-à-dire leur nom d’utilisateur, leur mot de passe haché, leur rôle et leur statut (actif ou suspendu), ainsi que les informations relatives aux livres et aux emprunts, sont stockées dans une base de données MySQL (version 8.0.27) gérée par phpMyAdmin 5.1.1 et accessible avec le nom d'utilisateur "root" et le mot de passe "AT2jgTMEx-cHeIT9". L'environnement du développement est WampServer 3.2.6. Le versioning du site est fait via GitHub.

Pour la connexion à la base de données, les variables ainsi que le PHP Data Object sont définies dans un fichier de configuration (config.php).



Selon le rôle de l'utilisateur, la barre de navigation affiche des éléments différents. Pour cela, le site regarde la variable globale dans laquelle a été stocké le rôle après la connexion de l'utilisateur. Pour pouvoir accéder à la variable globale *$\_SESSION*, il faut d'abord lancer une session avec *session\_start()*.



Exemple d'un élément de la barre de navigation qui dépend du rôle

L'affichage des livres sur la page index se fait avec une boucle qui parcourt la base de données. Le site vérifie le rôle de l'utilisateur et si celui-ci est connecté, puis affiche des options différentes selon les cas.



La barre de recherche fonctionne avec un script en jQuery 3.3.1. Le script parcourt la liste des livres et cache ceux qui ne contiennent pas le terme recherché dans le titre ou le nom d'auteur.



**Fonctionnalités**

**Créer un compte** (register.php)

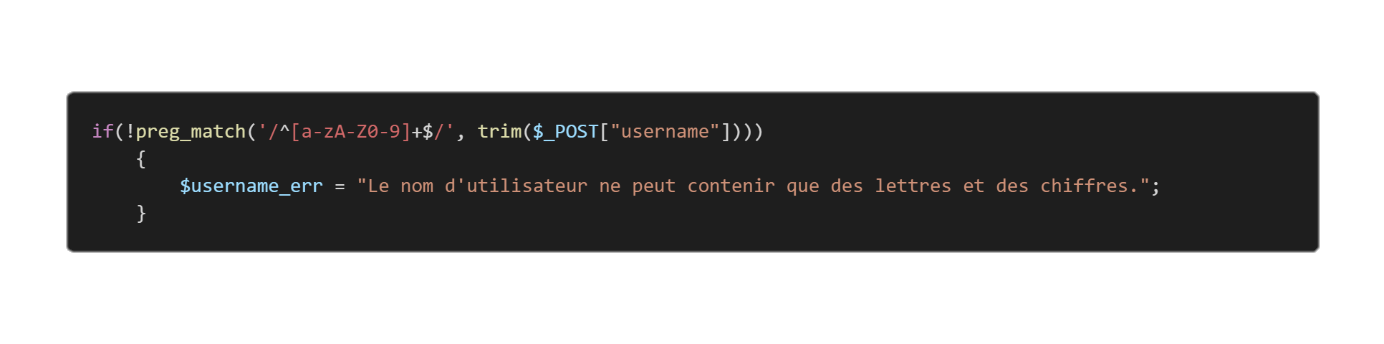
Avant de pouvoir accéder au formulaire de création de compte, le site vérifie si l'utilisateur n'est pas déjà connecté et, si nécessaire, fait une redirection vers la page de connexion.



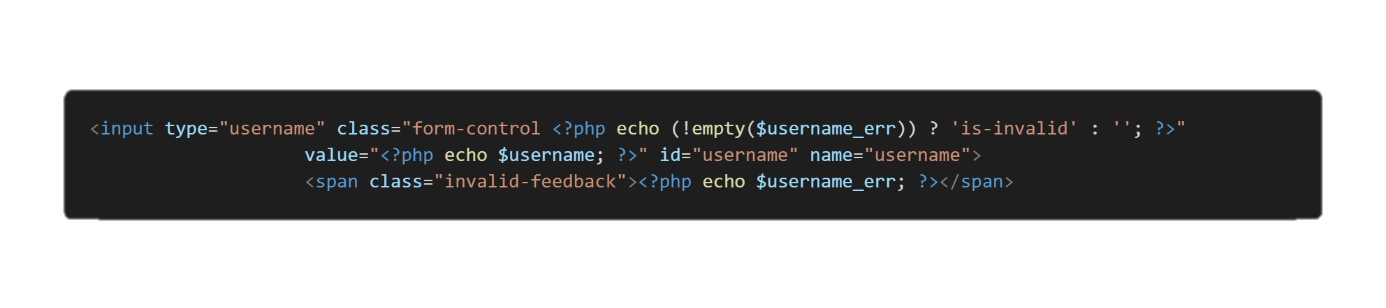
Les données du formulaire sont envoyées à un fichier PHP afin d'être traitées. Le code PHP se situe dans le même fichier que le code qui déclenche l'envoi des données.

****

Lors du traitement, le code affiche une erreur si un des champs est vide, si le nom d'utilisateur contient autre chose des chiffres et des lettres, si le nom d'utilisateur existe déjà, si le mot de passe contient moins que 8 caractères et si le mot de passe de confirmation ne correspond pas au mot de passe. Pour cela, il vérifie chaque cas et stocke des messages d'erreur dans les variables prévues à cet effet. Les variables non-vides sont par la suite affichées à l'écran et mises en forme avec des classes Bootstrap.

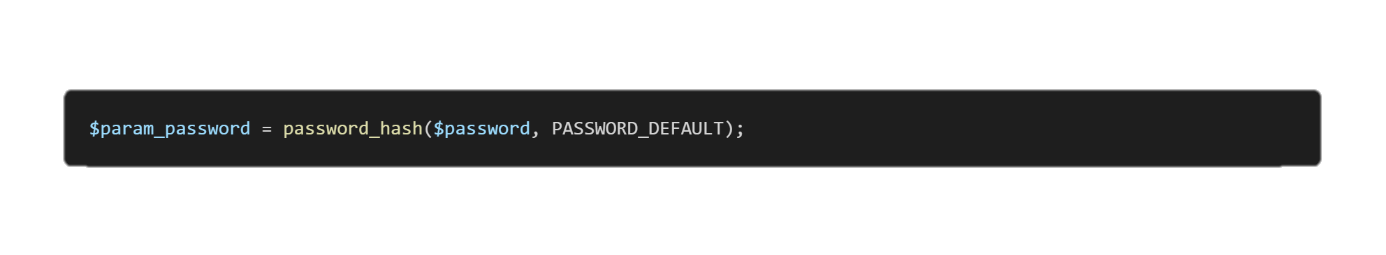


Exemple d'une erreur

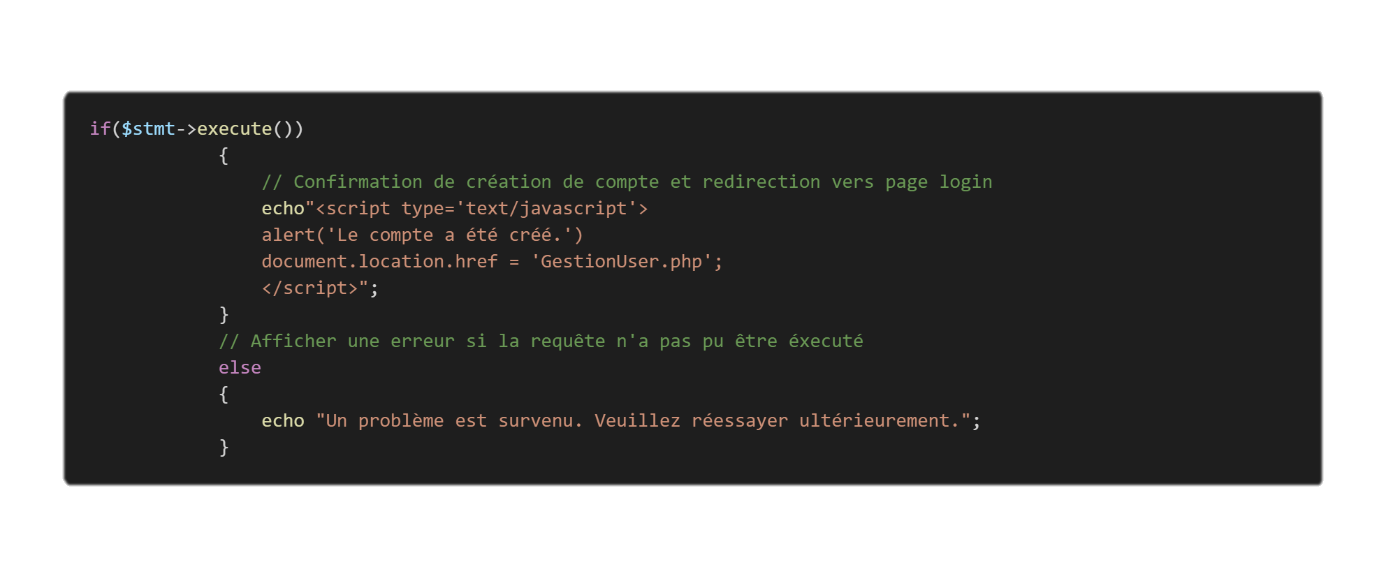


Exemple d'affichage d'une erreur

Si toutes les variables prévues pour les messages d'erreur sont vides, une requête "insert" est préparée afin d'ajouter les données des champs dans la base de données. Le mot de passe est stocké de façon hachée.



La communication avec la base de données se fait avec un PHP Data Object (PDO). Dans le cas où la requête n'a pas pu s'exécuter, une erreur s'affiche, sinon une confirmation de création de compte est affichée dans une fenêtre popup utilisant JavaScript et l'utilisateur est redirigé vers la page de connexion (login.php).



**Se connecter** (login.php)

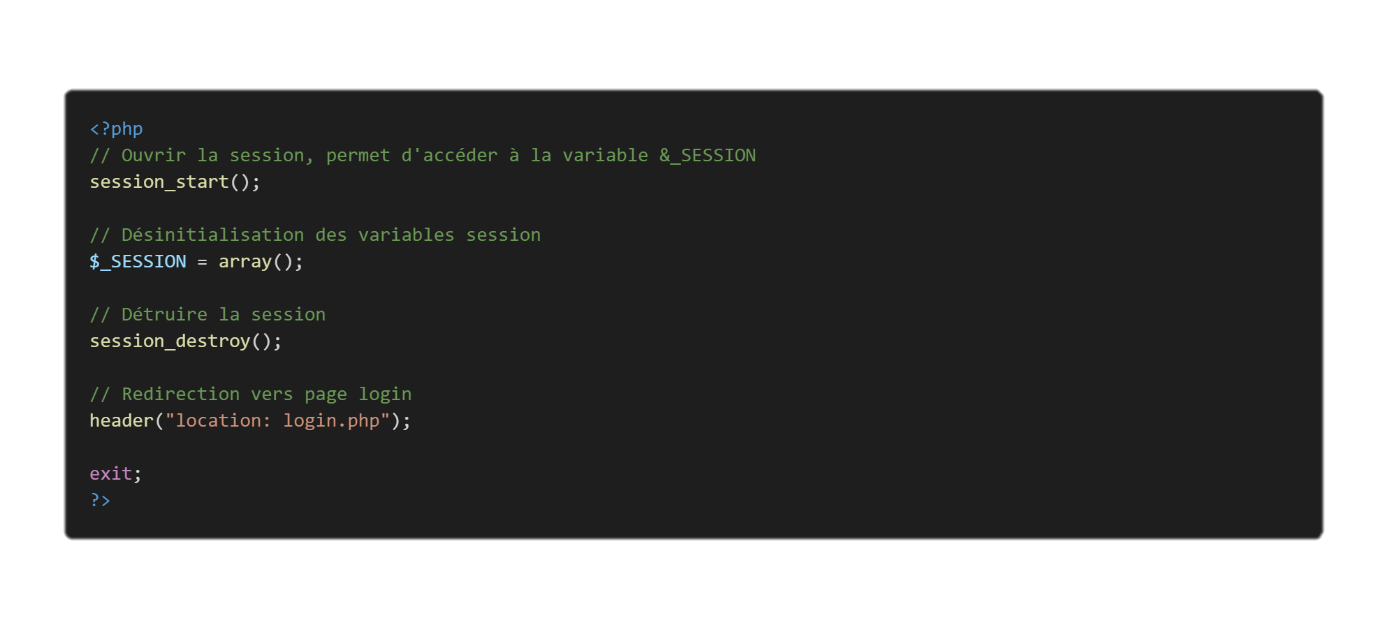
L'option de connexion n'apparait que lorsque l'utilisateur est déconnecté. Le formulaire envoie les données au code PHP dans le même fichier. Si les champs sont vides ou si le mot de passe ou le nom d'utilisateur sont invalides, une erreur est affichée (cf "***Créer un compte"***). Pour vérifier si le nom d'utilisateur correspond à un compte, on déclenche une requête "select" et on regarde le nombre de lignes affectées. Si celui-ci est supérieur à 0, le nom d'utilisateur existe dans la base de données.



Ensuite, on vérifie si le mot de passe entré par l'utilisateur correspond au mot de passe haché et si le compte n'est pas suspendu, ce qui empêcherait la connexion. Sinon, la variable *$\_SESSION["loggedin"]*prend la valeur "vrai" et le compte considéré comme connecté et l'utilisateur est envoyé sur la page de bienvenue (bienvenu.php).

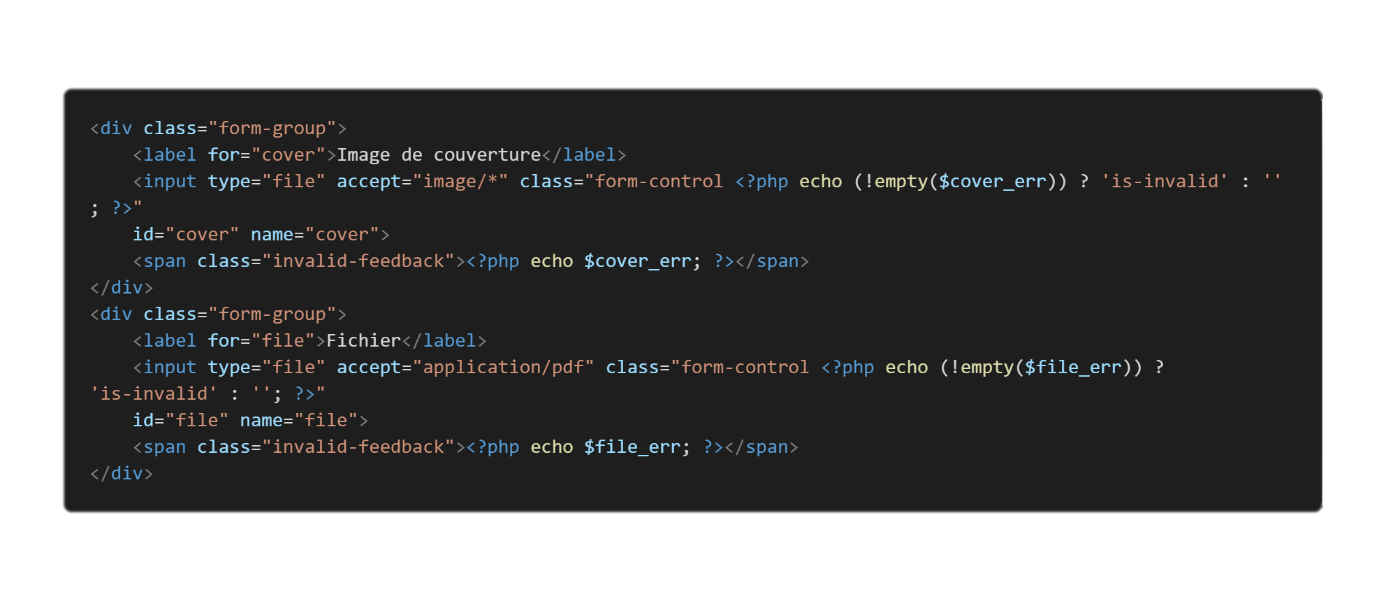


Pour se déconnecter (logout.php), la variable *$\_SESSION* est désinitialisée et la session détruite. Ensuite, il y a une redirection vers la page de connexion (login.php).

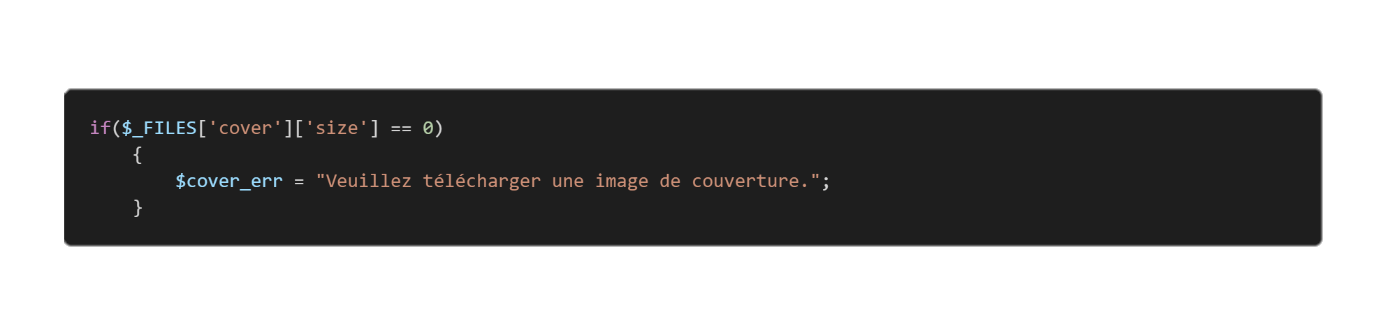


**Ajouter un livre (admin)** (AjoutLivre.php)

Le formulaire d’ajout de livre appelle le code PHP situé dans le même fichier pour traiter les données. Des erreurs sont affichées si un des champs est vide ou si aucun fichier n'a été sélectionné, de la même façon que le formulaire de création de compte. Le champ "Image de couverture" n'accepte que des fichiers image et le champ "Fichier" n'accepte que des fichiers en format PDF.



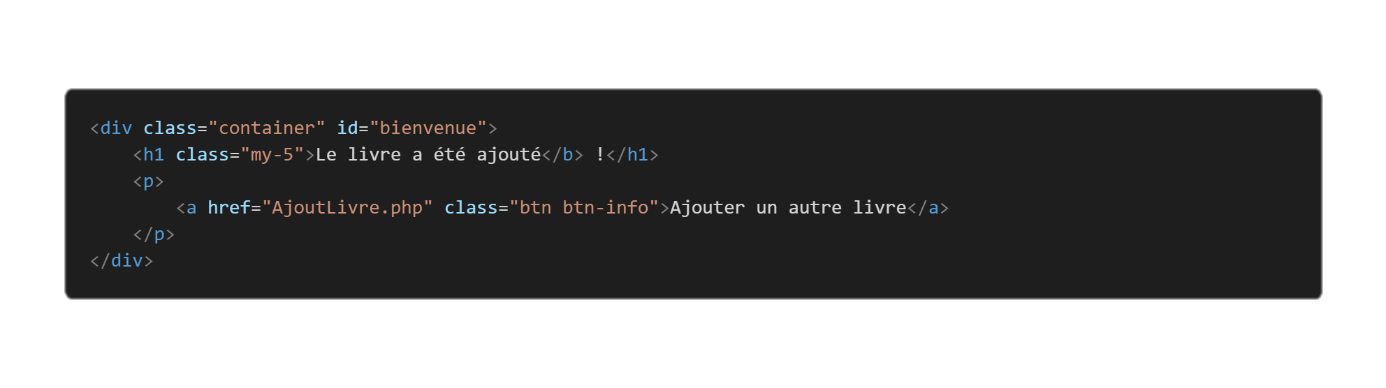
Afin de vérifier que l'utilisateur a bien choisi un fichier, on vérifie la taille du tableau dans lequel celui-ci est stocké.



Si un fichier a été sélectionné, il est copié vers un dossier qui porte son nom (il est créé s'il n'existe pas). Si un fichier avec le même nom existe déjà, un chiffre ajouté à la fin du nom est incrémenté. Afin d'assurer le bon fonctionnement du code sous Windows et Linux, les répertoires du chemin vers le fichier sont séparés par un *DIRECTORY\_SEPARATOR* qui prend la valeur du séparateur de répertoire par défaut du système.

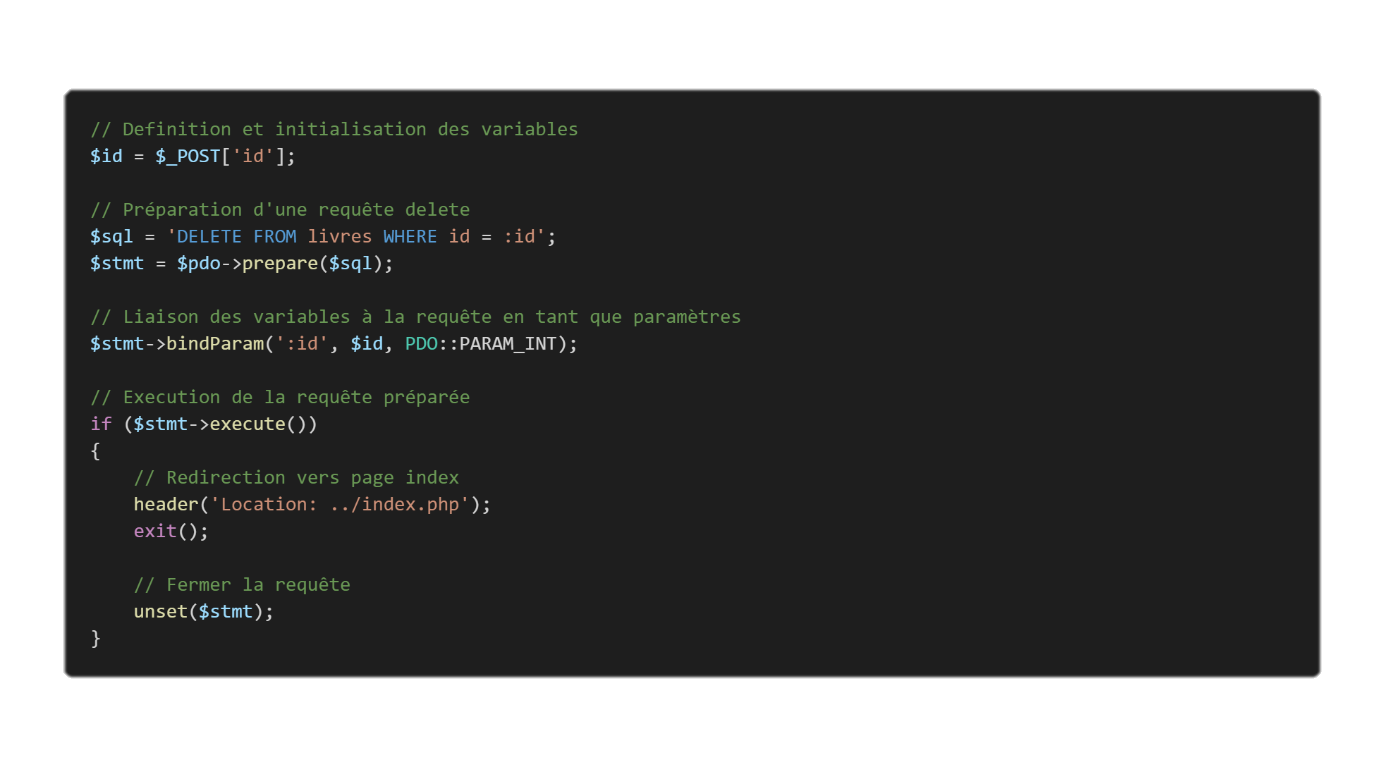


S'il n'y a pas de message d'erreur, une requête est déclenchée pour insérer le libellé, le nom d'auteur et le chemin vers l'image de couverture et le fichier PDF. Si la requête aboutit, l'utilisateur est redirigé vers une page de confirmation qui permet d'ajouter un autre livre (success.php).



**Supprimer un livre (admin)** (SuppressionLivre.php)

Si le compte possède le rôle "admin", les livres peuvent être supprimés depuis la page d'accueil (index.php). L'*id* du livre sélectionné est envoyé à un fichier PHP (SuppressionLivre.php) qui supprime de la base de données le livre correspondant.



**Gérer les comptes (admin)** (GestionUser.php)

L'affichage d'une liste des comptes existants se fait via une boucle dans une table (cf ***"Généralités"***). La barre de recherche fonctionne de la même façon que vu auparavant.

Créer des comptes (CreationUser.php)

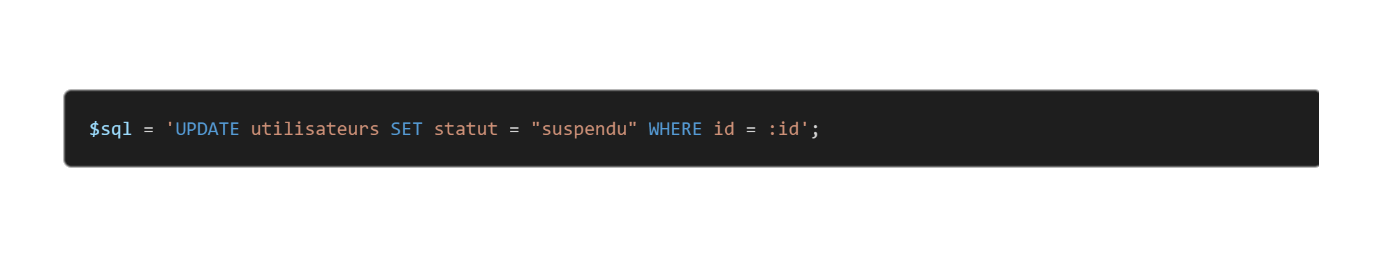
Le formulaire de création de compte destiné aux comptes "admin" se comporte de manière quasi identique au formulaire de création de compte pour les utilisateurs sans compte, mais son code PHP envoie sur la page de gestion des utilisateurs et non sur la page de connexion (cf ***"Créer un compte"***).

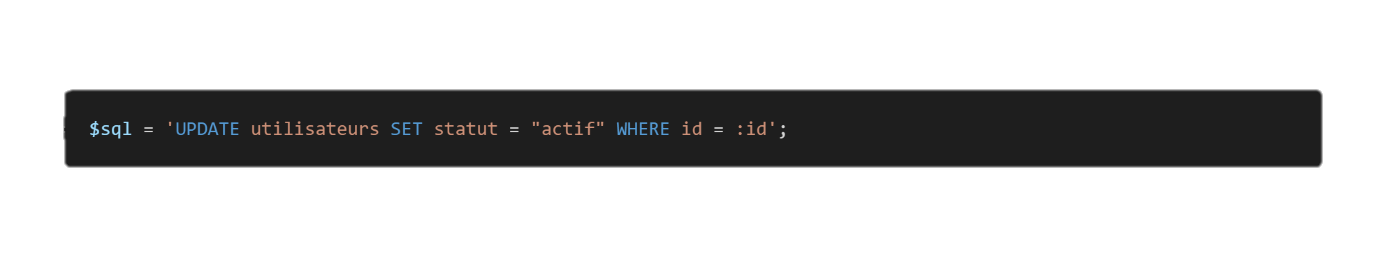
Supprimer des comptes (supprimer.php)

Le formulaire de suppression de compte se comporte de manière quasi identique au formulaire de suppression de livre (cf ***"Supprimer un livre"***).

Suspendre des comptes (suspension.php)

Si le compte choisi a le statut "actif", une requête "update" change celui-ci en "suspendu". Le compte ne peut par la suite plus se connecter (cf ***"Se connecter"***). Dans le cas contraire, le statut est changé en "actif" et le compte peut à nouveau se connecter.





Modifier des comptes (modification.php)

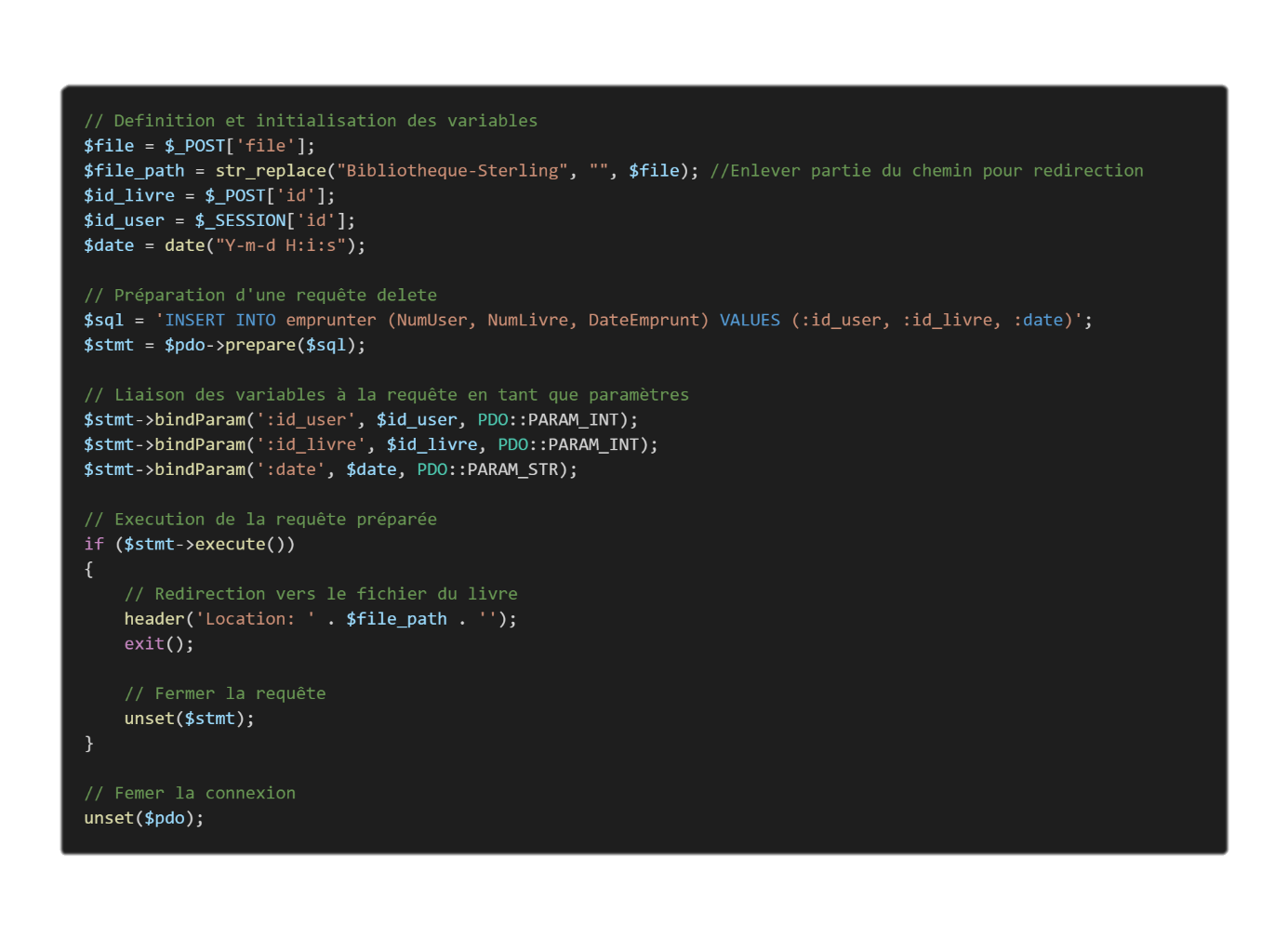
Après envoi du formulaire, les données sont envoyées à un fichier PHP (maj.php) qui met à jour la base de données en remplaçant les informations du compte en question avec les données du formulaire, en vérifiant que le mot de passe respecte des contraintes de longueur.



Exemple de mise-à-jour du rôle d'un compte

**Télécharger des livres (user)** (emprunt.php)

Si le compte possède le rôle "user", les livres peuvent être téléchargés sur la page d'accueil (index.php). Le bouton "Emprunter" envoie l'*id* du livre concerné ainsi que l'*id* de l'utilisateur à un fichier PHP qui exécute une requête "insert" pour ajouter les données reçues ainsi que la date actuelle en format "Y-m-d H:i:s" dans la table "Emprunter" de la base de données. Ensemble, ces trois données constituent la clé primaire afin de permettre à un utilisateur d'emprunter le même livre plusieurs fois. Par la suite, le fichier du livre est ouvert dans le navigateur et peut être téléchargé.



**Gérer son compte (user)** (MonCompte.php)

Le changement de mot de passe se fait via un fichier PHP situé dans le même fichier que le code de la page. On vérifie d'abord que les champs ne sont pas vides et que le nouveau mot de passe est assez long et on affiche une erreur le cas échéant (cf ***"Créer un compte"***). Si tous les messages d'erreur sont vides, une requête "update" remplace l'ancien mot de passe par le nouveau mot de passe qui a d'abord été haché.

