Exercice

L'Authentification

Consignes de l'Exercice

Exercice : Création d'un système d'authentification basique en PHP

Supposons que vous développiez un site web simple nécessitant une authentification pour accéder à certaines pages. Il faut d'abord créer une base de données MySQL contenant une table "utilisateurs" avec les champs suivants : `id`, `nom_utilisateur`, `email` et `mot_de_passe`.

Votre tâche ensuite est de créer un système d'authentification basique en PHP en suivant les étapes suivantes :

1. Créez une page d'accueil (`index.php`) qui permet aux utilisateurs de s'inscrire ou de se connecter.
2. Créez une page d'inscription (`inscription.php`) avec un formulaire permettant aux utilisateurs de saisir leur nom d'utilisateur, leur adresse e-mail et leur mot de passe. Assurez-vous de valider les données côté serveur pour éviter les entrées incorrectes.
3. Créez un script PHP (`traitement_inscription.php`) pour gérer l'inscription des utilisateurs. Ce script devrait vérifier que le nom d'utilisateur ou l'adresse e-mail n'est pas déjà utilisé, hacher le mot de passe, puis insérer l'utilisateur dans la base de données.
4. Créez une page de connexion (`connexion.php`) avec un formulaire permettant aux utilisateurs de saisir leur nom d'utilisateur (ou adresse e-mail) et leur mot de passe.
5. Créez un script PHP (`traitement_connexion.php`) pour gérer la connexion des utilisateurs. Ce script devrait vérifier les informations d'identification dans la base de données et créer une session pour l'utilisateur si les informations sont correctes.
6. Créez une page sécurisée (`tableau_de_bord.php`) accessible uniquement aux utilisateurs connectés. Assurez-vous que cette page vérifie la session pour s'assurer que l'utilisateur est authentifié avant de lui permettre d'accéder au contenu.
7. Créez un bouton de déconnexion sur la page de tableau de bord (`tableau_de_bord.php`) qui permet aux utilisateurs de se déconnecter et de détruire leur session.

Consignes supplémentaires :

- Utilisez PDO pour interagir avec la base de données et assurez-vous d'éviter les vulnérabilités telles que les injections SQL.
- Assurez-vous que les mots de passe sont stockés de manière sécurisée en hachant avec `password_hash()` lors de l'inscription et en vérifiant avec `password_verify()` lors de la connexion.
- Gérez les erreurs et les messages d'information de manière appropriée pour guider l'utilisateur tout au long du processus d'authentification.
- Validez les données du formulaire côté serveur pour garantir que les utilisateurs saisissent des données valides et sécurisées.
- Utilisez des sessions pour suivre l'état de l'authentification de l'utilisateur.