Amélioration du projet MARISOL : Sécurisation

Dans un premier temps, j'ai observé que certaines requêtes SQL dans le code utilisaient la clause SELECT *. Cela peut poser des risques de sécurité et des problèmes de performance. Les risques potentiels incluent :

- **Exposition des données sensibles** : Le fait de sélectionner toutes les colonnes peut entraîner la récupération et l'exposition de données confidentielles, comme les mots de passe ou les informations personnelles.
- **Injections SQL** : L'utilisation non contrôlée des entrées utilisateur dans les requêtes peut permettre à des attaquants de manipuler les requêtes SQL pour exécuter des commandes malveillantes.
- **Impact sur les performances** : Sélectionner toutes les colonnes augmente la charge sur le serveur de base de données, surtout si certaines colonnes ne sont pas nécessaires à l'opération.
- **Non-conformité aux Règles de Sécurité (RGPD)** : Selon le RGPD, seules les données strictement nécessaires doivent être collectées et traitées, ce qui n'est pas respecté avec un SELECT*.

Pour y remédier, j'ai spécifié explicitement les attributs nécessaires dans chaque requête SQL, plutôt que d'utiliser SELECT*. Cela limite la quantité de données récupérées et améliore la sécurité.

```
// Requête SQL pour sélectionner tous les élèves
$requete = "SELECT * FROM eleve";
// Préparation de la requête
$stmtement = $connexion->prepare($requete);
// Exécution de la requête
$stmtement->execute();
// Récupération des résultats de la requête sous forme de tableau associatif
$eleves = $stmtement->fetchAll(PDO::FETCH_ASSOC);

// Affichage des élèves s'ils existent
if ($eleves) {
    echo "<table>";
    echo "<tr><th>Code</th><th>Nom</th><th>Prénom</th><th>Date de naissance</th><th>Adresse</th><th>Code postal</th><th>Ville</th><th>Téléphone</th></tr>";
    foreach ($eleves as $eleve) {
        echo "<tr>";
        echo "<td>" . $eleve['CODE_ELEVE'] . "</td>";
        echo "<td>" . $eleve['NOM_ELEVE'] . "</td>";
        echo "<td>" . $eleve['PRENOM_ELEVE'] . "</td>";
        echo "<td>" . $eleve['DATE_NAISS'] . "</td>";
        echo "<td>" . $eleve['ADR1_ELEVE'] . " " . $eleve['ADR2_ELEVE'] . "</td>";
        echo "<td>" . $eleve['CP_ELEVE'] . "</td>";
        echo "<td>" . $eleve['VILLE_ELEVE'] . "</td>";
        echo "<td>" . $eleve['TEL_ELEVE'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "Aucun élève trouvé.";
}
```

J'ai donc spécifié les attributs dans le SELECT :

```
// Requête SQL pour sélectionner tous les élèves
$requete = "SELECT CODE_ELEVE, NOM_ELEVE, PRENOM_ELEVE, DATE_NAISS, ADR1_ELEVE, ADR2_ELEVE, CP_ELEVE, VILLE_ELEVE, TEL_ELEVE FROM eleve";
// Préparation de la requête
$stmtement = $connexion->prepare($requete);
// Exécution de la requête
$stmtement->execute();
// Récupération des résultats de la requête sous forme de tableau associatif
$eleves = $stmtement->fetchAll(PDO::FETCH_ASSOC);

// Affichage des élèves s'ils existent
if ($eleves) {
    echo "<table>";
    echo "<tr><th>Code</th><th>Nom</th><th>Prénom</th><th>Date de naissance</th><th>Adresse</th><th>Code postal</th><th>Ville</th><th>Téléphone</th></tr>";
    foreach ($eleves as $eleve) {
        echo "<tr>";
        echo "<td>" . $eleve['CODE_ELEVE'] . "</td>";
        echo "<td>" . $eleve['NOM_ELEVE'] . "</td>";
        echo "<td>" . $eleve['PRENOM_ELEVE'] . "</td>";
        echo "<td>" . $eleve['DATE_NAISS'] . "</td>";
        echo "<td>" . $eleve['ADR1_ELEVE'] . " " . $eleve['ADR2_ELEVE'] . "</td>";
        echo "<td>" . $eleve['CP_ELEVE'] . "</td>";
        echo "<td>" . $eleve['VILLE_ELEVE'] . "</td>";
        echo "<td>" . $eleve['TEL_ELEVE'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "Aucun élève trouvé.";
}
}
```

➔ Je l'ai fait pour toutes.

Ensuite, j'ai remarqué que le site permettait à un utilisateur non authentifié d'accéder à certaines sections sans s'identifier. Cela permet à des attaquants ou des utilisateurs non autorisés. J'ai donc mis en place une authentification obligatoire.

Voici la requête de création de la table "utilisateurs" :

```
1 CREATE TABLE utilisateurs (
2     id_utilisateur INT AUTO_INCREMENT PRIMARY KEY,
3     nom_utilisateur VARCHAR(50) NOT NULL UNIQUE,
4     mot_de_passe VARCHAR(255) NOT NULL,
5     cree_le TIMESTAMP DEFAULT CURRENT_TIMESTAMP
6 );
7
```

➔ Il y a une colonne « cree_le » pour assurer la traçabilité de la date de création de l'utilisateur.

Les étapes de l'authentification :

1. Fichier securite.php :

➤ **Objectif :** Ce fichier contient des fonctions pour sécuriser les entrées utilisateur et valider les mots de passe.

2. Fichier enregistrer.php :

- **Objectif :** Ce fichier gère l'inscription des utilisateurs dans la base de données. Il vérifie que les mots de passe sont valides et que le nom d'utilisateur n'est pas déjà utilisé. Ensuite, il stocke le mot de passe haché avec SHA-256 pour assurer sa sécurité.

3. Fichier login.php :

- **Objectif :** Ce fichier gère la connexion des utilisateurs. Il vérifie les informations d'identification (nom d'utilisateur et mot de passe) en les comparant avec celles de la base de données. Si les informations sont correctes, une session est créée pour l'utilisateur, lui permettant d'accéder aux sections protégées du site.

4. Fichier formulaire-auth.php :

- **Objectif :** Ce fichier fournit une interface utilisateur pour la connexion. Il affiche un formulaire de connexion qui envoie les données à login.php pour authentification.

5. Fichier inscription.php :

- **Objectif :** Ce fichier fournit présente une interface utilisateur pour l'inscription. Les utilisateurs peuvent créer un compte en fournissant un nom d'utilisateur et un mot de passe, qui sont ensuite traités par enregistrer.php.

Inscription

Nom d'utilisateur :

Mot de passe :

Confirmer le mot de passe :

Déjà un compte ? [Connectez-vous ici](#)

Connexion

Nom d'utilisateur :

Mot de passe :

Pas encore inscrit ? [Inscrivez-vous ici](#)

En parallèle, pour simplifier la gestion des enregistrements et améliorer la performance, j'ai défini les clés primaires dans la base de données comme AUTO_INCREMENT, ce qui permet de générer automatiquement des identifiants uniques pour les utilisateurs.

Structure

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	A_I	Con
CODE_ELEVE	INT		Aucun(e)			☐	☐	☐

Transaction ONLINE

Structure

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	A_I	Commentaires	Virtualité
CODE_ELEVE	INT		Aucun(e)			☐	☒		

Transaction ONLINE