



PROSIT : C'EST PARTI MON COOKIE



TABLE DES MATIERES :

MOTS CLES :

- Wish-list
- Injection sql
- CRUD
- Couche d'abstraction
- MCD
- Cookies
- Matrices des droits d'accès (Cahier de charge)
- Pilotes et administrateur

Wish-list : Liste des fonctionnalités, besoins ou améliorations souhaitées pour un projet, un logiciel ou une application. Elle sert souvent à définir les priorités et évolue en fonction des besoins des utilisateurs ou des clients.

- **Injection SQL** : Type d'attaque qui exploite une faille de sécurité dans une application en insérant du code SQL malveillant dans une requête. Cela peut permettre d'accéder à des bases de données, d'exfiltrer des informations sensibles ou de modifier des données.
- **CRUD** : Acronyme pour **Create, Read, Update, Delete**. Il représente les quatre opérations de base pour manipuler des données dans une base de données.
- **Couche d'abstraction** : Niveau intermédiaire entre une application et une ressource sous-jacente (comme une base de données ou un matériel). Elle permet de simplifier l'accès aux ressources en masquant les détails d'implémentation et en fournissant une interface standard.
- **MCD (Modèle Conceptuel de Données)** : Schéma représentant les entités, leurs attributs et les relations entre elles dans une base de données. Il est utilisé en conception de bases de données pour structurer les informations avant leur implémentation.
- **Cookies** : Petits fichiers stockés par un navigateur web qui contiennent des informations sur la session d'un utilisateur (ex : préférences, identifiants de session). Ils peuvent être utilisés pour la personnalisation, le suivi des utilisateurs ou l'authentification.
- **Matrice des droits d'accès (Cahier des charges)** : Tableau définissant les autorisations des utilisateurs sur un système ou une application. Chaque ligne représente un utilisateur ou un rôle, et chaque colonne une ressource ou une action autorisée (lecture, écriture, suppression, etc.).
- **Pilotes et administrateur** :

- **Pilotes (drivers)** : Logiciels permettant au système d'exploitation de communiquer avec un matériel (ex : carte graphique, imprimante).
- **Administrateur** : Utilisateur ayant des privilèges élevés pour gérer un système informatique, configurer des paramètres, installer des logiciels ou gérer les comptes utilisateurs.

CONTEXTE :

L'équipe Web4All va finaliser son site web avec l'interaction de la base de données

PROBLEME :

Problèmes :

Comment réaliser les interactions entre la base de données et le site web

CONTRAINTES :

- Faire attention aux entrées saisie par les utilisateurs
- Faire une couche d'abstraction
- Faire l'authentification et restriction d'accès
- CRUD

LIVRABLE :

- Suite du code avec la base de données

GENERALISATION :

Développement Web

PISTE DE SOLUTION :

- CRUD
- Token de connexion

PLAN D'ACTION :

- Mettre en place les CRUD et les différentes entités
- Couche d'abstraction
- Dynamiser le formulaire de saisie
- Authentification et restriction des accès
- Ajouter fonctionnalités: Page de création d'une offre de stage

Cours (+ développement mobile)

NOTIONS DE COURS :

SOLUTION

Cours sur l'utilisation de PDO, l'authentification, les sessions et les bases du développement mobile

1. Introduction

Ce cours couvre les bases de PDO pour interagir avec une base de données en PHP, la gestion de l'authentification avec les cookies et les sessions, ainsi qu'une introduction aux concepts du développement mobile.

2. Utilisation de PDO (PHP Data Objects)

2.1 Qu'est-ce que PDO ?

PDO (PHP Data Objects) est une interface d'accès aux bases de données en PHP qui offre une abstraction entre le code et le type de base de données utilisé.

2.2 Connexion à une base de données avec PDO

```
<?php
$dsn = "mysql:host=localhost;dbname=ma_base;charset=utf8";
$user = "root";
$password = "";

try {
    $pdo = new PDO($dsn, $user, $password, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);
    echo "Connexion réussie !";
} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
}
```

```
}  
?>
```

2.3 Exécuter des requêtes SQL avec PDO

Requête simple (sans paramètres)

```
$sql = "SELECT * FROM utilisateurs";  
$stmt = $pdo->query($sql);  
$result = $stmt->fetchAll();
```

Requête préparée (avec paramètres)

```
$sql = "SELECT * FROM utilisateurs WHERE email = :email";  
$stmt = $pdo->prepare($sql);  
$stmt->execute(['email' => 'test@example.com']);  
$result = $stmt->fetch();
```

Insertion de données

```
$sql = "INSERT INTO utilisateurs (nom, email, password) VALUES (:nom,  
:email, :password)";  
$stmt = $pdo->prepare($sql);  
$stmt->execute([  
    'nom' => 'Dupont',  
    'email' => 'dupont@example.com',  
    'password' => password_hash('mon_mot_de_passe', PASSWORD_DEFAULT)  
]);
```

3. Mécanismes d'authentification en PHP

3.1 Stockage sécurisé des mots de passe

PHP propose la fonction `password_hash()` pour hacher un mot de passe :

```
$hash = password_hash("monMotDePasse", PASSWORD_DEFAULT);
```

Pour vérifier un mot de passe :

```
if (password_verify("monMotDePasse", $hash)) {  
    echo "Mot de passe correct !";  
} else {  
    echo "Mot de passe incorrect !";  
}
```

3.2 Authentification avec sessions et cookies

Connexion de l'utilisateur

```
session_start();  
$_SESSION['user_id'] = $utilisateur['id'];  
setcookie("user", $utilisateur['id'], time() + 3600, "/", "", false,  
true);
```

Vérification de l'authentification

```
if (isset($_SESSION['user_id']) || isset($_COOKIE['user'])) {  
    echo "Utilisateur connecté";  
} else {  
    echo "Veuillez vous connecter.";  
}
```

Déconnexion de l'utilisateur

```
session_start();  
session_destroy();  
setcookie("user", "", time() - 3600, "/");
```

4. Manipulation des sessions et cookies en PHP

4.1 Démarrer une session

```
session_start();
```

4.2 Définir une variable de session

```
$_SESSION['nom'] = "Dupont";
```

4.3 Récupérer une variable de session

```
echo $_SESSION['nom'];
```

4.4 Supprimer une variable de session

```
unset($_SESSION['nom']);
```

4.5 Détruire une session

```
session_destroy();
```

4.6 Créer un cookie

```
setcookie("user", "Dupont", time() + 3600, "/", "", false, true);
```

4.7 Lire un cookie

```
echo $_COOKIE['user'];
```

4.8 Supprimer un cookie

```
setcookie("user", "", time() - 3600, "/");
```

5. Introduction au développement mobile

5.1 Différents types d'applications mobiles

- **Applications natives** : Développées pour une plateforme spécifique (iOS, Android) en utilisant Swift, Kotlin ou Java.
- **Applications web mobiles** : Applications accessibles via un navigateur web et adaptées aux mobiles.

- **Applications hybrides** : Développées avec des technologies web (HTML, CSS, JavaScript) et encapsulées dans une application native (via Cordova, React Native, Flutter).