

Projet Intranet - Mission 8 : Authentification des utilisateurs

Objectif :

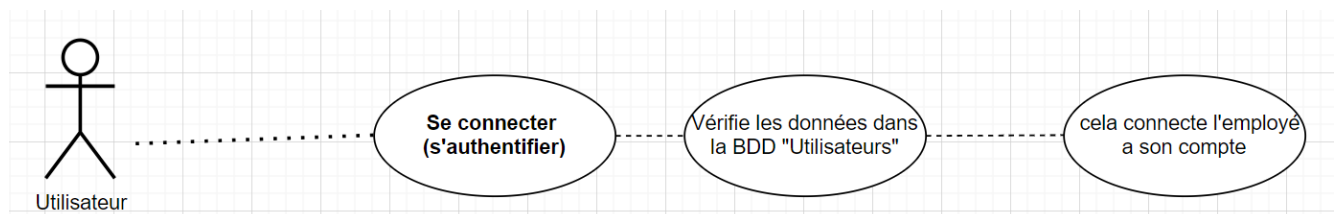
Table des matières

1.	Exercice 1 :	1
----	--------------------	---

1. Exercice 1 :

Question 1) Mettez à jour votre représentation UML pour faire apparaître le cas d'utilisation "Authentification".

Voici ci-dessous la partie de la représentation UML qui concerne l'authentification :



Nous pouvons voir que pour un utilisateur lambda, Le système de connexion vérifie les identifiants et mots de passes de l'utilisateur dans la base de données avant que celui-ci ne lui accorde l'accès à son compte. Ce processus affecte tous les utilisateurs, y compris les administrateurs.

Réalisez un formulaire de connexion et son fonctionnement avec php. Le mot de passe doit être haché en php puis comparé avec celui de la base de données.

Bien réalisé comme on peut le voir ci-dessous (code html), avec mes différents champs avec et le mot de passe :

```
<div class="inscription-box">
  <h2>Connexion</h2>

  <?php if (!empty($message)): ?>
    <p style="color: red; text-align: center;"><?= htmlspecialchars($message) ?></p>
  <?php endif; ?>
  <form method="post" action="connexionUtilisateur.php">
    <div class="container">
      <div class="left-side">
        <h1 class="texte-en-gras">Bienvenue</h1>
        <p class="policeMonospace">Sur le site de l'entreprise</p>
      </div>
      <div class="right-side">
        <div class="input-group">
          <h2 class="texte-en-gras">Se connecter</h2>
          <input type="text" id="username" name="username" placeholder="Identifiant" required>
        </div>
        <div class="input-group">
          <input type="password" id="password" name="mdp" placeholder="Mot de passe" required>
        </div>
        <br>
        <button type="submit">Se connecter</button>
        <p>Vous n'avez pas de compte ? <a href="inscriptionUtilisateur.php" class="login-link">Inscrivez-vous</a></p>
      </div>
    </div>
  </form>
</div>
```

et le code php qui permet de faire fonctionner le code pour pouvoir se connecter :

```
<?php
// Démarrage de la session
session_start();

// Paramètres de connexion
$host = 'localhost';
$port = 3307;
$dbname = 'company_project_intranet';
$username = 'root';
$password = '';

try {
    // Connexion à la base de données avec PDO
    $dsn = "mysql:host=$host;port=$port;dbname=$dbname;charset=utf8mb4";
    $pdo = new PDO($dsn, $username, $password, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);
} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
}

// Variable pour stocker les messages d'erreur
$message = "";

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    if (!empty($_POST['username']) && !empty($_POST['mdp'])) {
        $login = htmlspecialchars(trim($_POST['username']));
        $mdp = $_POST['mdp'];

        try {
            // Vérification de la base de données pour l'utilisateur
            $requete = $pdo->prepare("SELECT login, password, codeGrp FROM utilisateurs WHERE login = :login");
            $requete->bindParam(':login', $login);
            $requete->execute();
            $resultat = $requete->fetch();
        }
    }
}
```

AT Professionnel – Projet Intranet

```
36      $resultat = $requete->fetch();
37
38      if ($resultat) {
39          // Hachage du mot de passe entré en SHA-256 avant la comparaison
40          $mdp_hash = hash('sha256', trim($mdp)); // Hachage du mot de passe utilisateur
41
42          // Comparaison des hachages
43          if ($mdp_hash === $resultat['password']) {
44              // Mot de passe correct
45              $_SESSION['login'] = $resultat['login'];
46              $_SESSION['role'] = $resultat['codeGrp']; // Récupère le groupe utilisateur
47
48              // Redirection vers la page d'accueil
49              header("Location: index.php");
50              exit;
51          } else {
52              $message = "Identifiant ou mot de passe incorrect.";
53          }
54      } else {
55          $message = "Identifiant non trouvé.";
56      }
57      } catch (PDOException $e) {
58          $message = "Erreur lors de la connexion : " . $e->getMessage();
59      }
60  } else {
61      $message = "Veuillez remplir tous les champs.";
62  }
63  }
64
65  // Afficher les messages d'erreur
66  if ($message) {
67      echo $message;
68  }
69  ?>
```

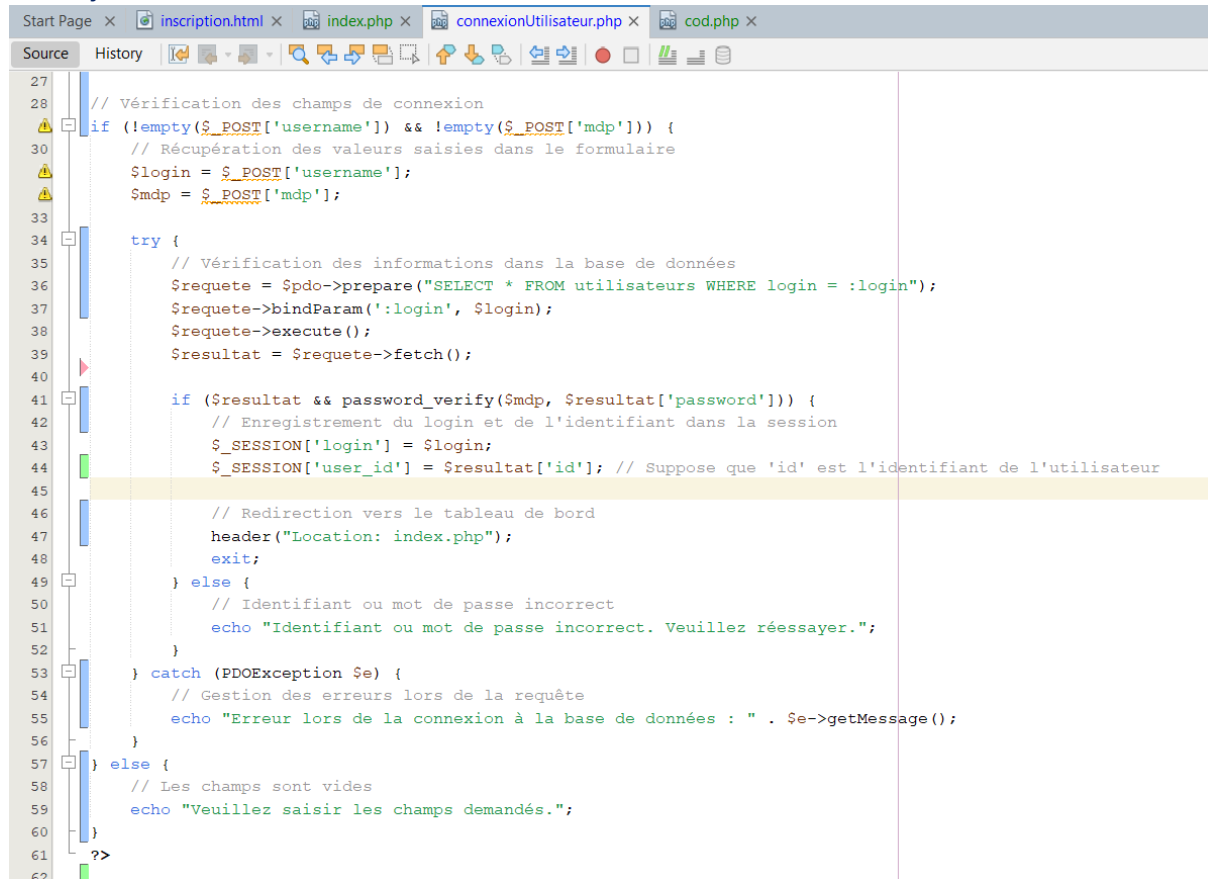
Ce script PHP permet de gérer l'authentification d'un utilisateur en vérifiant ses identifiants dans une base de données. Il commence par initialiser une variable `$message` pour stocker d'éventuels messages d'erreur. Ensuite, il vérifie si la requête HTTP est de type POST, ce qui signifie que le formulaire de connexion a été soumis. Si les champs username et mdp ne sont pas vides, il récupère et nettoie la valeur du champ username en utilisant `htmlspecialchars` et `trim`, tandis que le mot de passe est simplement stocké dans la variable `$mdp`.

Une requête préparée est ensuite exécutée pour rechercher l'utilisateur dans la base de données en fonction de son login. Si l'utilisateur est trouvé, le mot de passe saisi est haché avec l'algorithme SHA-256, puis comparé avec celui stocké dans la base de données. Si les hachages correspondent, la connexion est validée et des informations de session sont enregistrées, notamment le login et le rôle de l'utilisateur (`codeGrp`). L'utilisateur est alors redirigé vers la page `index.php`.

Si le mot de passe ne correspond pas, un message d'erreur indiquant que les identifiants sont incorrects est affiché. Si aucun utilisateur correspondant n'est trouvé, un message précisant que l'identifiant est inconnu est retourné. En cas d'erreur avec la base de données, un message détaillant l'erreur est affiché. Enfin, si l'un des champs est vide, un message demandant de remplir tous les champs est généré. Le script affiche ensuite le message d'erreur, s'il y en a un.

Ajoutez une redirection vers le tableau de bord après l'authentification de l'utilisateur.

Bien ajouté une redirection vers le tableau de bord



```
27 // Vérification des champs de connexion
28 if (!empty($_POST['username']) && !empty($_POST['mdp'])) {
29     // Récupération des valeurs saisies dans le formulaire
30     $login = $_POST['username'];
31     $mdp = $_POST['mdp'];
32
33     try {
34         // Vérification des informations dans la base de données
35         $requete = $pdo->prepare("SELECT * FROM utilisateurs WHERE login = :login");
36         $requete->bindParam(':login', $login);
37         $requete->execute();
38         $resultat = $requete->fetch();
39
40         if ($resultat && password_verify($mdp, $resultat['password'])) {
41             // Enregistrement du login et de l'identifiant dans la session
42             $_SESSION['login'] = $login;
43             $_SESSION['user_id'] = $resultat['id']; // Suppose que 'id' est l'identifiant de l'utilisateur
44
45             // Redirection vers le tableau de bord
46             header("Location: index.php");
47             exit;
48         } else {
49             // Identifiant ou mot de passe incorrect
50             echo "Identifiant ou mot de passe incorrect. Veuillez réessayer.";
51         }
52     } catch (PDOException $e) {
53         // Gestion des erreurs lors de la requête
54         echo "Erreur lors de la connexion à la base de données : " . $e->getMessage();
55     }
56 } else {
57     // Les champs sont vides
58     echo "Veuillez saisir les champs demandés.";
59 }
60
61 ?>
```

Figure 1 : Fichier connexionUtilisateur avec redirection

Ce code commence par vérifier si les champs "username" et "mdp" (mot de passe) ont été remplis dans le formulaire. Si ces champs ne sont pas vides, il récupère les valeurs saisies pour le nom d'utilisateur et le mot de passe. Ensuite, une requête est préparée pour interroger la base de données à la recherche d'un utilisateur ayant ce nom d'utilisateur. La requête est exécutée et le résultat est récupéré. Si un utilisateur correspondant est trouvé et que le mot de passe saisi correspond à celui stocké dans la base de données (grâce à la fonction `password_verify`), les informations de connexion sont enregistrées dans la session (avec le nom d'utilisateur et l'identifiant de l'utilisateur), et l'utilisateur est redirigé vers la page `index.php`. Si les identifiants sont incorrects, un message d'erreur est affiché. Si la requête échoue, une exception est capturée, et un message d'erreur est affiché concernant la connexion à la base de données. Enfin, si les champs du formulaire sont vides, un message demandant de remplir les champs est affiché.

Stockez dans une variable de session, l'identifiant de l'utilisateur.

Fait dans la question précédente, donc voici la ligne concernée :

`$_SESSION['user_id'] = $resultat['id'];` // Suppose que 'id' est l'identifiant de l'utilisateur

AT Professionnel – Projet Intranet

Dans cette ligne, la valeur de l'identifiant de l'utilisateur (\$resultat['id'], qui est récupéré depuis la base de données) est stockée dans la variable de session \$_SESSION['user_id']. Cela permet de garder trace de l'utilisateur connecté pendant sa navigation sur le site, car la variable de session sera disponible sur toutes les pages où la session est active.

```
<!-- Texte à droite du profil -->
<div class="header-item-2">
<div class="texte_profil">
  <?php if (isset($_SESSION['login'])): ?>
    <!-- Si l'utilisateur est connecté, afficher son identifiant et le bouton de déconnexion -->
    <h2>Bienvenue, <?php echo $_SESSION['login']; ?> !</h2>
    <form action="deconnexion.php" method="post">
      <button type="submit">Se déconnecter</button>
    </form>
  <?php else: ?>
    <!-- Si l'utilisateur n'est pas connecté, afficher un lien pour se connecter -->
    <a href=" ../php/connexionUtilisateur.php"><p>Connectez-vous</p></a>
  <?php endif; ?>
</div>
```

Affichez cet identifiant sur le tableau de bord.

Donc au départ on peut bien se connecter :

Connexion

Bienvenue
Sur le site de l'entreprise
Se connecter

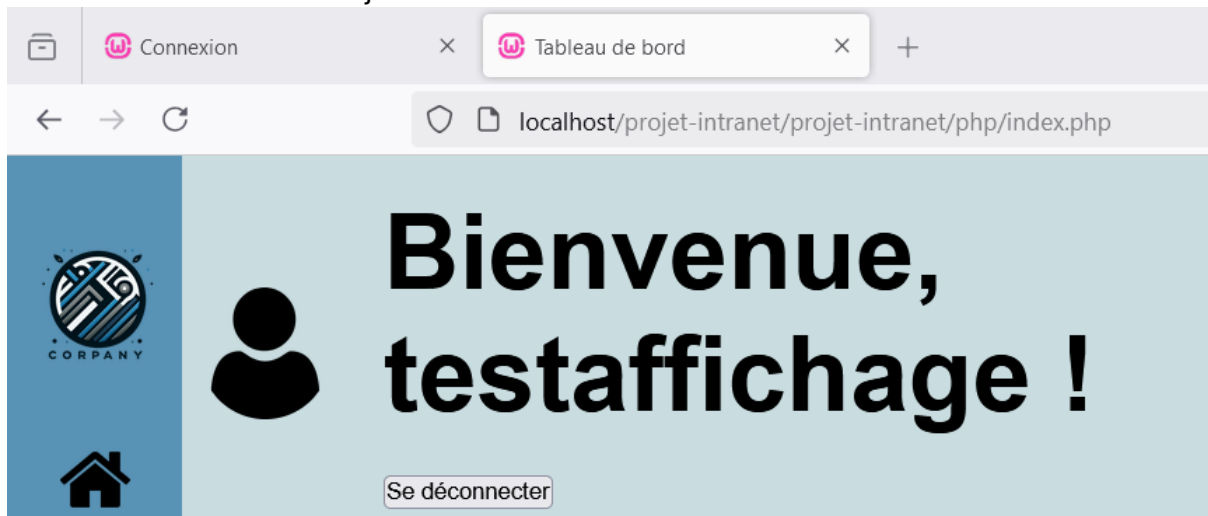
testaffichage
.....

Se connecter

Vous n'avez pas de compte ? **Inscrivez-vous**

Figure 2 : Tentative de connexion

Et quand on se connecte cela renvoie bien vers le tableau de bord et voici le résultat :



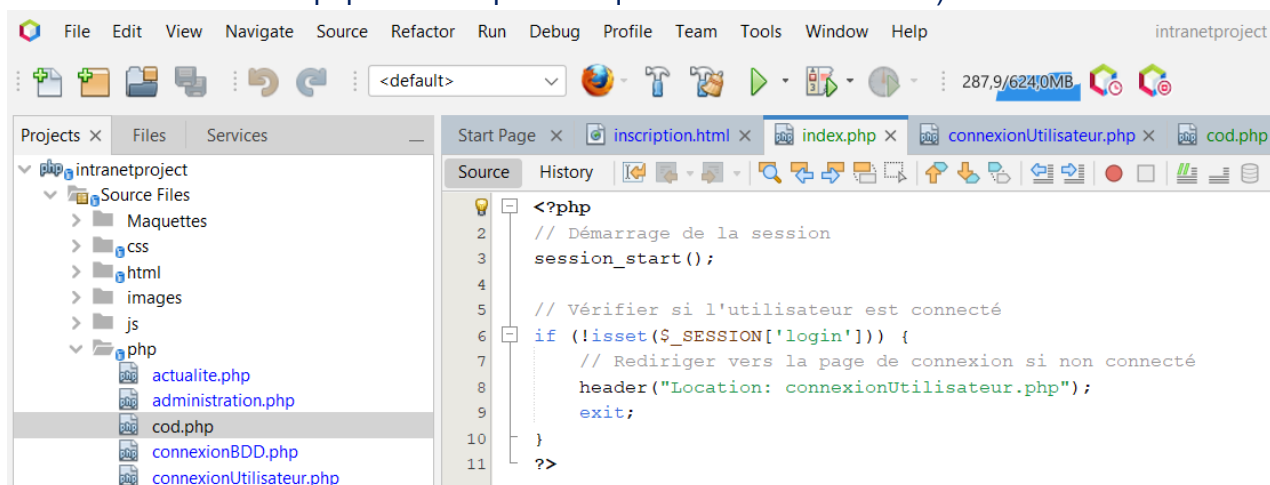
Le css est encore à régler mais la fonctionnalité marche !

Protégez le tableau de bord pour qu'il ne soit pas accessible par un utilisateur non authentifié.

On a bien protégé le tableau de bord comme on peut le voir ci-dessous si l'utilisateur n'est pas connecté cela redirigera vers la page connexionUtilisateur.php :

```
Start Page x inscription.html x index.php x connexionUtilisateur.php x
Source History
<?php
2 // Démarrage de la session
3 session_start();
4
5 // Vérifier si l'utilisateur est connecté
6 if (!isset($_SESSION['login'])) {
7     // Rediriger vers la page de connexion si non connecté
8     header("Location: connexionUtilisateur.php");
9     exit;
10 }
11 ?>
```

Test , lorsque l'on veut démarrer le projet et qu'on appuie sur la flèche verte pour démarrer (qui se lance normalement sur index.php automatiquement qui est le tableau de bord) :



AT Professionnel – Projet Intranet

Ici on va vérifier grâce à ce script PHP qui commence par démarrer une session avec la fonction `session_start()`, ce qui permet de récupérer ou de créer des variables de session. Ensuite, il vérifie si une variable de session appelée `login` est définie. Si cette variable n'existe pas, cela signifie que l'utilisateur n'est pas connecté. Dans ce cas, le script redirige l'utilisateur vers la page de connexion (`connexionUtilisateur.php`) grâce à la fonction `header("Location: ...")` et termine l'exécution du script avec `exit` pour éviter que d'autres parties du code ne soient exécutées.

On appuie sur run et cela nous affiche cette page :

localhost/projet-intranet/projet-intranet/php/connexionUtilisateur.php

[Connectez-vous](#)

Rechercher...

Connexion

Bienvenue

Sur le site de l'entreprise

Se connecter

Identifiant

Mot de passe

Se connecter

Vous n'avez pas de compte ? [Inscrivez-vous](#)

Donc cela nous demande bien de nous connecter, donc c'est protéger.