

Projet Intranet

Objectif :

Table des matières

1. Exercice 1 :	2
-----------------------	---

1.Exercice 1 :

Mettez à jour vos représentations UML.

Adaptez la la base de données et intégrez le code PHP. Vous n'êtes pas obligé d'utiliser AJAX pour les notifications mais c'est toujours mieux.

J'ai d'abord fait un script envoie une requête AJAX à "notifications.php" pour récupérer les notifications sous format JSON. Si la requête réussit, il affiche les notifications dans un conteneur HTML, en formatant les données avec des éléments comme le type de notification, le message et la date. Si une erreur ou un message est renvoyé, il est affiché en rouge. Si la requête échoue, un message d'erreur est affiché à la place des notifications. La fonction setInterval permet de rafraîchir les notifications toutes les 10 secondes sans recharger la page.

```
<script>
function chargerNotifications() {
    $.ajax({
        url: "notifications.php",
        type: "GET",
        dataType: "json",
        success: function (data) {
            let html = "";

            if (data.error) {
                html = "<p style='color:red;'>" + data.error + "</p>";
            } else if (data.message) {
                html = "<p>" + data.message + "</p>";
            } else {
                data.forEach(notif => {
                    html += `<div class="notification-item">
                        <strong>${notif.type_notification.toUpperCase()}</strong>: ${notif.message} <br>
                        <small>De: ${notif.utilisateur}</small> - <small>${notif.date_notification}</small>
                    </div>`;
                });
            }

            $("#notifications-container").html(html);
        },
        error: function () {
            $("#notifications-container").html("<p style='color:red;'>Erreur lors de la récupération des notifications.</p>");
        }
    });
}

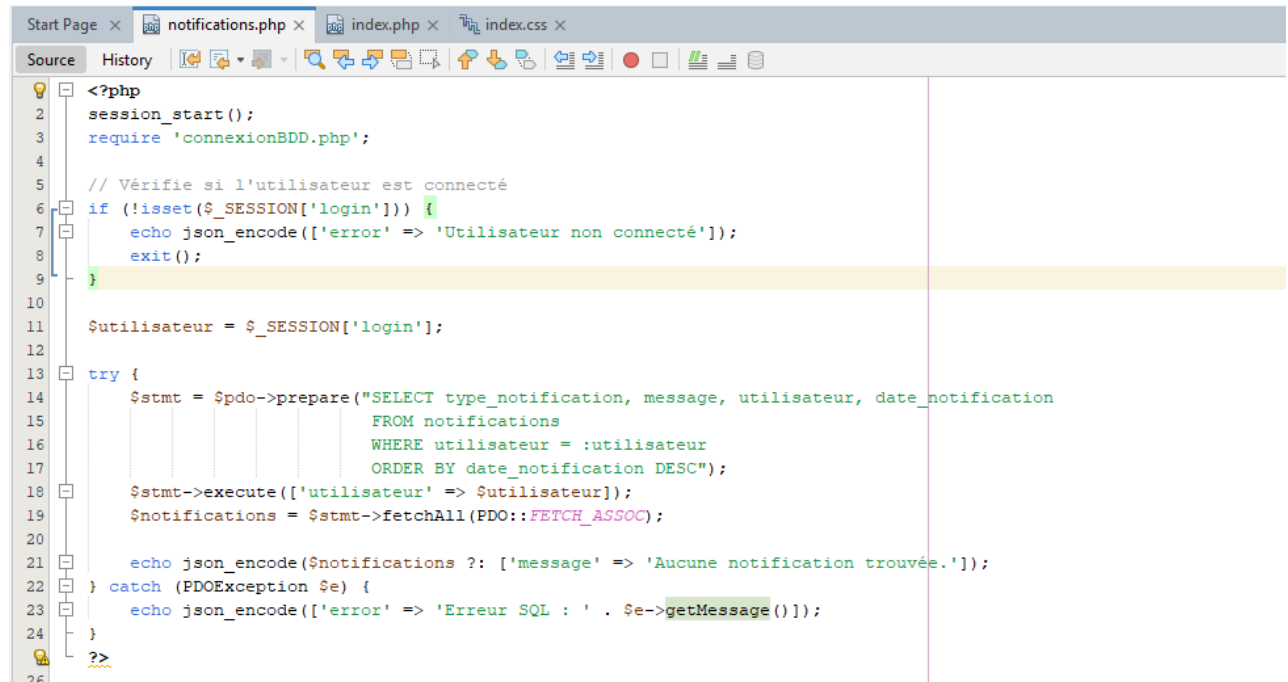
$(document).ready(function () {
    chargerNotifications();
    setInterval(chargerNotifications, 10000); // Rafraichit toutes les 10 secondes
});
</script>
```

Dans ce bloc :

```
<!-- Notifications -->
<div class="dashboard-item-6">
  <article class="dashboard-notifications">
    <div class="item-titre-container">
      <div class="item-shadow"></div>
      <h1 class="item-titre">Notifications</h1>
    </div>

    <!-- Conteneur pour les notifications -->
    <div id="notifications-container">
      <p>Chargement des notifications...</p>
    </div>
  </article>
</div>
```

Et grâce à cette fonction php qui va permettre de récupérer les notifications :



```
<?php
2 session_start();
3 require 'connexionBDD.php';
4
5 // Vérifie si l'utilisateur est connecté
6 if (!isset($_SESSION['login'])) {
7     echo json_encode(['error' => 'Utilisateur non connecté']);
8     exit();
9 }
10
11 $utilisateur = $_SESSION['login'];
12
13 try {
14     $stmt = $pdo->prepare("SELECT type_notification, message, utilisateur, date_notification
15                           FROM notifications
16                           WHERE utilisateur = :utilisateur
17                           ORDER BY date_notification DESC");
18     $stmt->execute(['utilisateur' => $utilisateur]);
19     $notifications = $stmt->fetchAll(PDO::FETCH_ASSOC);
20
21     echo json_encode($notifications ?: ['message' => 'Aucune notification trouvée.']);
22 } catch (PDOException $e) {
23     echo json_encode(['error' => 'Erreur SQL : ' . $e->getMessage()]);
24 }
25
26
```

Ce code PHP commence par démarrer une session et inclure la connexion à la base de données. Il vérifie ensuite si l'utilisateur est connecté via la session. Si l'utilisateur n'est pas connecté, il renvoie une erreur JSON. Si l'utilisateur est connecté, il prépare une requête SQL pour récupérer les notifications associées à cet utilisateur, triées par date de manière décroissante. Les notifications sont ensuite récupérées et envoyées au format JSON. Si aucune notification n'est trouvée, un message indiquant qu'il n'y a pas de notifications est envoyé. En cas d'erreur SQL, un message d'erreur détaillé est renvoyé.

Et voici le visuel actuel qui peut évoluer (css pas encore touché) :

