Avant de commencer j'aimerai tout d'abord vous dire qu'il est possible d'arrêter une session. En arrêtant la session tout ce qui était sauvegarder sera bien entendu perdu. C'est un peu comme quand on se déconnecte d'un site, il nous redemandera de nous reloguer.

Il suffit de faire un « **session_unset()** » ou **« unset($_SESSION['nomDeLaVariable']) »** qu'on verra un peu plus tard**.**

(Exos Bonus)

0) **(Difficile) Améliorez le code du jeu du hasard. Je veux qu'à chaque fois que vous introduisez un numéro il continue de vous dire si c'est plus grand ou plus petit etc. Mais je veux que ce numéro aléatoire soit garder en mémoire tant que nous l'avons pas trouver. Si on le trouve, il donnera bien entendu le message comme quoi on l'a trouvé mais il faudra aussi qu'il le libère de la mémoire. Que quand on relance jeu du hasard il recrée un nouveau nombre et ainsi de suite.**

Pour commencer nous allons créer un **dossier** « **functions** » là où se trouve nos fichiers. Dans ce dossier on va créer un fichier qui s'appelle « **authentification.php** ». Dans ce fichier PHP on va créer une fonction qui va s'appeler « **is_connected** », elle ne recevra rien en paramètre et retournera un booléen.

authentification.php U

functions > authentification.php > ...

```php
<?php
    function is_connected() : bool {
        return true;
    }
?>
```

On va **modifier** le contenu de la fonction de manière à vérifier que l'utilisateur est connecté, on pourra se baser sur le principe des sessions que nous venons de voir.

```php
return !empty($_SESSION['connected']);
```

Si ce n'est pas vide, ça va me renvoyer true, si c'est vide ça va me renvoyer false.

Maintenant on peut utiliser cette fonction la dans un fichier qu'on va appeler « **dashboard.php** »(pas dans functions). Dashboard en anglais veut dire tableau de bord. En gros c'est une représentation visuelle des informations importantes. On aurait pu l'appeler bien entendu comme on le souhaitait.

On va créer un onglet dashboard qui exécutera le fichier.

(Exos)

1) **Créez l'onglet dashboard (basez-vous sur les autres onglets).**

Dans ce fichier on va tester d'appeler la fonction « is_connected » et vérifié grâce aux sessions si ça fonctionne bien.

```
dashboard.php > ...
1   <?php
2       $nav = "DashBoard";
3       $title = "Dashboard";
4       session_start();
5       require "functions/authentification.php";
6       var_dump(is_connected());
```

Ici on voit bien que j'importe bien le fichier **authentification.php** qui se trouve dans le dossier **functions**. On oublie pas de lancer la session avant. Puis on fait un **var_dump** pour vérifier que vaut l'appelle de la fonction **is_connected().**

Si on clique sur l'onglet dashboard de notre site. On verra bien que le booléen est à false. Ça veut dire qu'on n'est pas connecté.

Si durant la session on donne la valeur de 1 à notre variable globale. Rajoutez cette ligne juste au-dessus du require.

```
$_SESSION['connected']=1;
```

Il retournera désormais true sur notre site. Vous auriez pu aussi lui donner la valeur true directement.

Voilà une manière de savoir comment on pourra définir si un utilisateur est connecté.

Vous remarquez qu'à chaque fois, je démarre la session. Il y a moyen de vérifier si une session est démarrée ou pas. C'est en utilisant la fonction **« session_status »,** elle se trouve bien entendu dans la documentation officielle sur php.net. Elle permet de renvoyer le statut courant de la session. Vous avez 3 constantes qui représente les différentes valeur :

Valeurs de retour

- **PHP_SESSION_DISABLED** si les sessions sont désactivées.
- **PHP_SESSION_NONE** si les sessions sont activées, mais qu'aucune n'existe.
- **PHP_SESSION_ACTIVE** si les sessions sont activées, et qu'une existe.

On peut donc faire quelques choses comme cela dans le fichier authentification.php :

```php
function is_connected() : bool {
    if (session_status() === PHP_SESSION_NONE){
        session_start();
    }
    return !empty($_SESSION['connected']);
}
```

En gros, si la session n'est pas lancée, il fera un session_start().

On peut maintenant supprimer la ligne ou on lançait la session dans notre fichier **dashboard.php**.

Si votre session est toujours active il suffit de faire un session_unset() ou un **unset($_SESSION['connected'])** après le var_dump().

Maintenant il faut empêcher l'utilisateur d'avoir accès à cette page si il n'est pas connecté, et le rediriger vers une page qu'on créera qui s'appellera **« login.php ».**

Pour rediriger un utilisateur vers une autre page il suffit de mettre une entête location, elle permet d'indiquer vers quelle page doit être rediriger l'utilisateur.

Il suffit de faire un **header('Location : cheminDeRedirection'),** dans notre cas ici on va le rediriger vers la page de login.

Dans notre fichier **« dashboard.php »** on va mettre ceci :

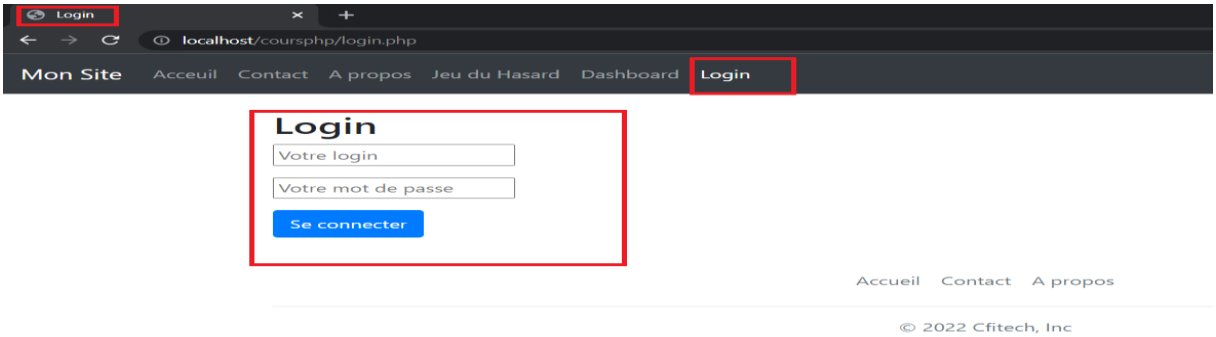
```php
dashboard.php > ...
<?php
    $nav = "DashBoard";
    $title = "Dashboard";
    require "header.php";
    require "functions/authentification.php";
    var_dump(is_connected());
    header('Location: coursphp/login.php');

    require "footer.php";
```

Maintenant il nous reste à créer notre fichier login.php.

(Exos)

2) **Créer la page login, un formulaire avec 2 variables « pseudo » et « password » :**



Maintenant qu'on a notre page login, si on clique sur l'onglet dashboard, il nous redirigera sur la page login pour devoir se loguer.

Dans le code on va essayer de récupérer ce qu'on écrit dans notre formulaire de connexion. On va créer un compte avec comme pseudo « votre prénom » et comme mot de passe « 12345 ». On devra vérifier qu'on a bien introduit ces deux valeurs pour pouvoir être connecté.

Faites ces modification dans votre fichier login.php :

```php
<?php
    $nav = "login";
    $title = "Login";
    $erreur = null;
    if (!empty($_POST['pseudo']) || !empty($_POST['password'])){
        if ($_POST['pseudo'] === "" && $_POST['password'] === "12345"){

        }else {
            $erreur = "Identifiants incorrects !";
        }
    }
    require "header.php";
    if ($erreur) : ?>
      <div class="alert alert-danger">
        <?php echo $erreur ?>
      </div>
    <?php endif;
?>
```

Ici je n'ai pas encore définis ce qu'on doit faire quand on se connectera correctement.

Je crée une variable $erreur qui va me servir de retourner une erreur si on a pas introduit les bons identifiants. Je vérifie ensuite si dans notre formulaire on a bien écrit quelques choses dans pseudo et dans password. Si oui je vérifie ensuite que ce qu'il y a dedans sont bien mon prénom et mon password. Sinon je donne à notre variable erreur un message.

Ensuite je vérifie si la variable erreur est vide, si elle est toujours à null, c'est qu'on aura pas eu d'erreur donc il n'entrera pas dans le IF. Par contre si on a eu une erreur de pseudo ou de password, il aura une valeur donc il entrera dans le IF et affichera l'erreur sous forme d'alerte grâce à la class alert alert-danger.

Allez maintenant sur votre site et introduisez un mauvais mot de passe ou un mauvais pseudo et vous devriez avoir normalement cette alerte avec ce message ci :

Il nous reste plus qu'à gérer maintenant notre fameuse session connected, qui doit recevoir vrai, quand on se connecte avec le bon pseudo et le bon mot de passe.

Pour le connecter je vais démarrer la session et donner à notre variable globale 1 mais aussi rediriger notre utilisateur vers la page DashBoard :

```
4   $erreur = null;
5   if (!empty($_POST['pseudo']) || !empty($_POST['password'])){
6       if ($_POST['pseudo'] === "Julien" && $_POST['password'] === "12345"){
7           session_start();
8           $_SESSION['connected'] = 1;
9           header("Location: dashboard.php");
10          //exit();
```

Maintenant vous pouvez essayé de vous connectez via login, si vous tapez bien votre pseudo et mot de passe, vous serez redirigé vers dashboard.

(Exos)

3) **Maintenant qu'on a vu comment se connecter, essayez de personnaliser le message avec votre pseudo dans votre dashboard :**

4) **Créer un onglet «Logout», et faites en sorte que quand on clique dessus il nous déconnecte et nous renvoie vers la page d'accueil.**
5) **Je veux aussi maintenant que quand on est connecté et qu'on clique sur l'onglet login, on soit redirigé vers la page dashboard.**

On va un peu améliorer le code pour avoir tous plus ou moins la même chose pour le login et le logout.

```php
<ul class="navbar-nav">
   <li class="nav-item <?php if ($nav === "login"): ?> active <?php endif ?>">
        <a class="nav-link" href="login">Login</a>
   </li>
   <li class="nav-item">
        <a class="nav-link" href="logout">Logout</a>
   </li>
</ul>
```

On met nos deux onglet en-dehors de **<ul class= "navbar-nav mr-auto">….**

On va modifier le code pour faire en sorte que quand on est pas connecté, on ne voit pas logout et quand on est connecté on ne voit pas login. Mais avant de faire ça nous allons d'abord ajouter dans notre fichier header.php un **require_once**, rappelez-vous il permet d'inclure un fichier qu'une seule fois. Ici on a besoin de charger qu'une seule fois les fonctions.

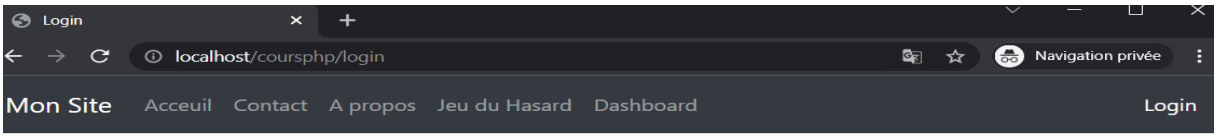
```php
require_once "functions/authentification.php";
```

Vous le mettez à la toute première ligne de votre fichier « header.php ». Vous pourrez maintenant effacer tous les **« require "functions/authentification.php"; »** que vous trouverez dans vos fichiers**.** Par contre maintenant qu'il est dans le header, il faut que les « **require "header.php";** » soient juste après les déclarations de variables.

```php
login.php > ...
1   <?php
2       $nav = "login";
3       $title = "Login";
4       $erreur = null;
5       require "header.php";
6       if (is_connected()){
```

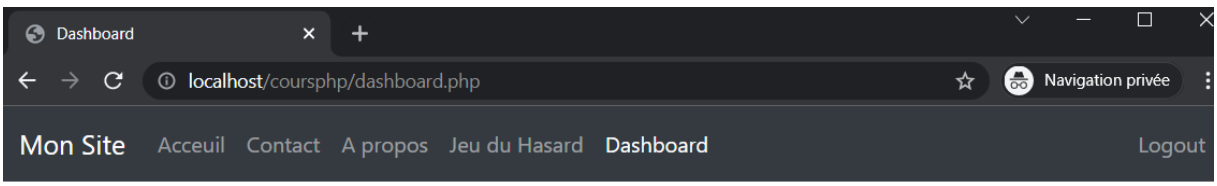
Par exemple ici dans le fichier **« login.php »** j'ai remonté le « require » juste après mes déclarations de variables.

Maintenant nous pouvons implémenter notre site pour que quand nous sommes pas connecté on voit l'onglet Login.



Login

Par contre quand je suis connecté l'onglet login disparait mais on voit l'onglet Logout.



Bienvenue Julien dans votre DashBoard

(Exos)

6) **Essayez donc de faire ce que je vous ai montré juste au-dessus, c'est-à-dire que quand je me connecte je vois l'onglet Logout par contre quand je ne suis pas connecté je vois l'onglet Login.**

Nous avons terminé ce chapitre sur les sessions. Vous pouvez continuer à vous exercer avec les sessions chez vous sur votre site, créer des onglets, des formulaires, des scripts etc. Avant de faire le lien entre les Bases de données et le site web, nous allons voir l'un des plus gros chapitre dans les langages de programmations, c'est-à-dire l'orienté objet. On a besoin de connaitre le concept d'objet et de classe pour pouvoir connecter le site à la base de données. Vous verrez que d'une certaine manière on a déjà vu ce système d'objet en base de données avec nos tables qui sont comme des classes en programmations.