Lycée TOUCHARD WASHINGTON

Ciel 1

Web Avancé

TD n°2

Cookies et sessions avec PHP

Gestion des cookies:

Les questions suivantes ne peuvent pas être faites si le TD1 n'a pas été entièrement terminé.

Il est possible d'avoir la liste des cookies via javascript avec : window.document.cookie

1. Créez une page **show_cookie.html** qui affichera les cookies créer précédemment dans la div ayant pour id "***cook***" (on donne le code de la page, il vous faut donner le code de **show_cookie.js** en jquery).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Afficher les cookies</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="js/libs/jquery/jquery.min.js" type="text/javascript"></script>
    <script src="js/show_cookie.js" type="text/javascript"></script>
  </head>
  <body>
    <div id="cook">les cookies</div>
  </body>
</html>
```

On souhaite compléter notre site en stockant les login et mot de passe de la page d'inscription.

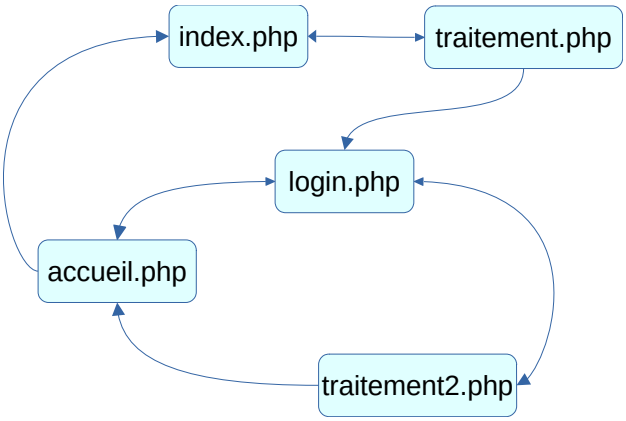
Pour cela, on va dans un premier temps, utiliser un système de cookies.

L'idée est d'avoir une variable cookie qui contiendra l'ensemble des couples login/mot de passe saisi par les utilisateurs qui se sont inscrits via le formulaire.

2. Ajouter les attributs suivants au formulaire pour les champs input de login, du mot de passe et de la confirmation du mot de passe :

```
pattern="^[a-zA-Z]+$" title="Que des lettres"
```

L'architecture finale aura l'aspect suivant:



En php , pour se rediriger vers une page on utilise la fonction header. Par exemple pour charger la page "**ici.php**", on utilisera le code suivant:

```
<?php
header("Location: ici.php");
exit();
```

3. Modifiez votre script **traitement.php** afin qu'il fasse une redirection vers **index.php** (votre formulaire) si les informations reçues ne sont pas correctes (mot de passe et confirmation ne sont pas identiques), et qu'il fasse une redirection vers le script **login.php** sinon.
4. Testez si la redirection se fait comme prévu.

Il faut maintenant créer notre variable "cookie" que l'on nommera ***logmdp*** qui contiendra les login et les mot de passe.

Afin d'avoir les couples login et mot de passe facilement exploitable les couples seront de la forme login/motdepasse et ses derniers seront séparées par "|".

Le contenu de logmdp ressemblera donc à l'exemple ci-dessous:

toto/titi|rbidochon/toto|rgalopin/tutu

Ainsi, on a toto qui a pour mot de passe titi, rbidochon qui a pour mot de passe toto et rgalopin qui a pour mot de passe tutu.

Le code suivant permet de générer la variable cookie ***logmdp***. On suppose que le login et le mot de passe ont été préalablement stocké dans les variables ***$login*** et ***$mdp***.

```
$logmdp = filter_input(INPUT_COOKIE, "logmdp");
if ($logmdp) {
    setcookie("logmdp", $logmdp . "|" . $login . "/" . $mdp);
} else {
    setcookie("logmdp", $login . "/" . $mdp);
}
```

5. Complétez votre script **traitement.php** pour qu'il génère le cookie ***logmdp***.
6. Vérifiez que le cookie est bien généré à l'aide de votre script **show_cookie.html**.

7. Codez la page **login.php** qui demandera un login et un mot de passe et enverra ses derniers vers un script **traitement2.php**.

Voici une partie du code de **traitement2.php**

```php
<?php
$logmdp = filter_input(INPUT_COOKIE, "logmdp");
if ($logmdp) {
   $tabLogin = explode("|", $logmdp);
} else {
   $tabLogin = [];
}
if (filter_input(INPUT_SERVER, 'REQUEST_METHOD') === 'POST') {
        $login = // récupération du login venant de login.php
        $mdp = // récupération du mot de passe venant de login.php
        $couple = $login . "/" . $mdp;
}
if (in_array($couple, $tabLogin)) {
 // générer un cookie nommé logok et prenant comme valeur true
// redirection vers accueil.php
} else {
 // générer un cookie nommé logok et prenant comme valeur false
// redirection vers login.php
}
```

8. Que fait la fonction ***explode*** ? Quel sera le résultat attendu dans notre cas ?
9. Que fait la fonction ***in_array*** ? Quel sera le résultat attendu dans notre cas ?
10. Complétez le script **traitement2.php**.

La page **accueil.php** reprend le code de la page **show_cookie.html** avec en plus un lien vers la page **login.php** et un second lien vers la page **index.php** (page d'inscription).

11. Codez la page **accueil.php.**
12. Modifiez la page **login.php** afin que si la variable cookie "***logok***" est vraie, il y ai une redirection vers la page **accueil.php.**

Gestion des sessions:

On veut faire la même chose que précédemment, mais en utilisant des variables de session.

Il sera également possible de se déconnecter.

13. Quelle fonction devez-vous impérativement appeler au début de chaque script afin de pouvoir utiliser les variables de session ?

Contrairement aux variables de type "cookie" qui ne pouvaient contenir que des chaines de caractères, il est possible d'avoir des variables de session de type tableau.

On peut donc modifier la génération de la variable logmdp comme suit:

```
if (!isset($_SESSION['logmdp'])) {
      $_SESSION['logmdp'] = [];
   }
   $_SESSION['logmdp'][] = $login."/".$mdp;
```

Les couples logins/mot de passe seront ainsi stockés sous la forme:

```
$_SESSION['logmdp'][0]="toto/titi"
$_SESSION['logmdp'][1]="rbidochon/toto"
$_SESSION['logmdp'][2]="rgalopin/tutu"
```

14. Recréer un projet, mais en utilisant les sessions à la place des cookies.

Dans le script **traitement2.php** il n'est plus besoin de séparer les login/mot de passe pour générer un nouveau tableau. Vous pouvez utiliser directement la variable de session logmdp dans l'appel de la fonction ***in_array***.

15. Modifier le script **accueil.php** afin d'ajouter un bouton deconnexion qui fera appel à un script **deconnexion.php** et qui détruira la variable de session ***logok*** et redirigera vers la page **accueil.php**.