

Objectifs:

- Le but de ce workshop est de traiter l'authentification via la variable globale SESSION.
- Les fichiers source de ce workshop sont disponibles sur GitHub :

<https://github.com/ghadabenkhalifa/AtelierSession>

Partie 1: Introduction au session

1. Définition

La variable globale `$_SESSION` est dédiée pour gérer des sessions utilisateur. A cet effet, nous pouvons gérer facilement les utilisateurs qui doivent accéder à un contenu nécessitant une identification. Dans ce qui suit nous allons traiter un exemple simple dans le but de manipuler les sessions.

2. Création de la session

La première étape est d'utiliser la fonction `session_start()` qui indique au serveur d'activer la gestion des sessions, ce qui induit la possibilité de manipuler la variable globale `$_SESSION`.

```
< ?php session_start() ; ?>
```

Cette variable présente un tableau associatif temporaire dans lequel nous pouvons stocker les informations désirées afin de gérer l'accès au contenu. L'id d'une session est généré automatiquement et envoyé via un header, pour être stocké dans un cookie. La fonction `session_id()` permet de récupérer et modifier l'id de la session.

```
< ?php session_id() ; ?>
```

3. Mettre fin à la session

En cas de déconnexion, la fonction `session_destroy()` sera utilisée pour fermer la session du visiteur. Cette fonction est automatiquement appelée lorsque le visiteur ne charge plus de page de votre site pendant plusieurs minutes (c'est le timeout), mais vous pouvez aussi créer une page « Déconnexion » si le visiteur souhaite se déconnecter manuellement.

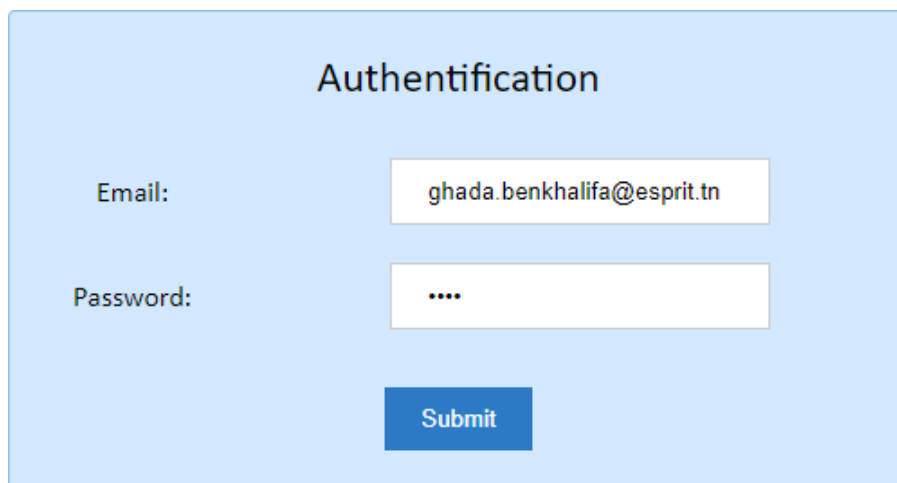
Partie 2: Exemple applicatif

Dans cette partie, nous allons utiliser le code source de l'atelier 7. En se basant sur les sessions (vues dans la partie 1), l'objectif de cet exemple est de créer un système de gestion des droits d'accès basé sur le rôle de l'utilisateur.

Étape 1: Créer un nouveau fichier intitulé « *connexion.php* » qui contient un formulaire d'authentification, assure la connexion avec un couple email et mot de passe et redirige vers une page « *ProfilUser.php* ».

L'utilisateur sera connecté si le couple « email » et « mot de passe » introduits sont existants dans la table « utilisateur »

Le fichier « *ProfilUser.php* » est à créer, son contenu sera détaillé dans une étape suivante.



The image shows a web form titled "Authentification" on a light blue background. It contains two input fields: "Email:" with the value "ghada.benkhalifa@esprit.tn" and "Password:" with masked characters "....". Below these fields is a blue "Submit" button.

NB : Pour bien respecter l'architecture MVC, l'étape de connexion est à faire dans le Contrôleur « *UtilisateurC.php* » :

```
function connexionUser($email,$password){  
    $sql="SELECT * FROM Utilisateur WHERE Email='". $email . "' and Password = '". $password."'";  
    $db = config::getConnexion();  
    try{  
        $query=$db->prepare($sql);  
        $query->execute();  
        $count=$query->rowCount();  
        if($count==0) {  
            $message = "pseudo ou le mot de passe est incorrect";  
        } else {  
            $x=$query->fetch();  
            $message = $x['role'];  
        }  
    }  
    catch (Exception $e){  
        $message= " ".$e->getMessage();  
    }  
    return $message;  
}
```

Étape 2 : Dans cette étape, le fichier « **connexion.php** » sera modifié comme suit :

1. Démarrage d'une session
2. Stockage de valeurs relatives à l'utilisateur connecté (Email).

```
<?php
session_start();

include_once '../Model/Utilisateur.php';
include_once '../Controller/UtilisateurC.php';
$message="";
$userC = new UtilisateurC();
if (isset($_POST["email"]) &&
    isset($_POST["password"])) {
    if (!empty($_POST["email"]) &&
        !empty($_POST["password"]))
    {
        $message=$userC->connexionUser($_POST["email"],$_POST["password"]);

        $_SESSION['e'] = $_POST["email");// on stocke dans le tableau une colonne ayant comme nom "e",
        // avec l'email à l'intérieur

        if($message!='pseudo ou le mot de passe est incorrect'){
            header('Location:ProfilUser.php');}
        else{
            $message='pseudo ou le mot de passe est incorrect';
        }
    }
    else
        $message = "Missing information";
}
?>
```

Étape 3: Editer les fichiers « *ProfilUser.php* » pour afficher un message comme suit :

« Bienvenue Utilisateur **Email** »

Avec **Email** est l'adresse mail stockée dans la session

Et ajouter un bouton « **Déconnecter** » qui redirige vers la page « *deconnexion.php* »

<div>Déconnecter</div>
<div>Bienvenue Utilisateur ghada.benkhalifa@esprit.tn</div>

Figure : Page « *ProfilUser.php* »

Étape 4: Créer un fichier « *deconnexion.php* » qui permet de fermer la session, et affiche un message de confirmation de la déconnexion.

«*deconnexion.php*»

```
<?php
session_start();
session_destroy();
echo 'Vous êtes déconnecté. <a href="./connexion.php">Se connecter ?</a>';
```