

Compte rendu Geoworld

Sommaire

Partie 1 : élèves : Consulter les informations	1
Partie 2 : Enseignant : Mettre à jour les données	6
Partie 3 : Admin : Gérer les roles des utilisateurs	8

Antoine BALCEROWIAK

Luca Kiusi

lien github: <https://github.com/Antoine-Balcerowiak/Projet-Geoworld>

Avant de commencer le compte rendu voici le code sql afin d'intégrer ma base de données utilisateurs

```
INSERT INTO `utilisateur` (`id`, `nom`, `prenom`, `login`, `password`, `role`) VALUES
(1, 'balcerowiak', 'Antoine', 'abalce', 'sio', 'admin'),
(4, 'kyks', 'mbappe', 'mbapp', 'sio', 'prof'),
(5, 'messi', 'lionel', 'messi', 'sio', 'eleve'),
(6, 'bon', 'jean', 'jeanbon', 'mdp', 'eleve');
```

Partie 1 : élèves : Consulter les informations

Tout d'abord il faut pouvoir se connecter ou s'inscrire sur le site

authentification.php :

```
<form action="login.php" method="post">
Votre login : <input type="text" name="login"><br />
Votre mot de passe : <input type="password" name="password"><br />
<a href="inscription.php" >Inscription</a><br>
<input type="submit" value="Connexion">

</form>
```

Se programme permet de récupérer les données login et password entré au clavier par l'utilisateur et ensuite une fois valider de renvoyer vers login.php afin de vérifier si le login et le password sont valide

```

<?php
require_once('inc/manager-db.php');
if (isset($_POST['login']) && isset($_POST['password']) && !empty($_POST['login'])&&
!empty($_POST['password'])) {
    $result = getAuthentication($_POST['login'],$_POST['password']);
    print_r($result);
    if($result){
        session_start ();
        $_SESSION['nom'] = $result->nom;
        $_SESSION['identifiant'] = $result->id;
        $_SESSION['role'] = $result->role;
        header ('location: index.php');
    }
    else{
        header ('location: authentication.php');
    }
}
else {
    header ('location: authentication.php');
}
?>

```

Ici si le login et le password sont valide on récupère le nom,id et le role de l'utilisateur et ensuite on le renvoie vers index.php mais si les variables sont vides ou incorrects on renvoie vers authentication.php, donc si on n'a pas de compte on peut s'en crée un en cliquant sur inscription qui envoie sur inscription.php

```

<div class="form-group col-md-6 col-sm-6">
<input type="text" class="form-control" name="nom" required placeholder="Nom" />
</div>
<div class="form-group col-md-6 col-sm-6">
<input type="text" class="form-control" name="prenom" required placeholder="Prénom" />
</div>
</div>
<div class="row">
<div class="form-group col-md-6 col-sm-6">
<input type="text" class="form-control" name="login" required placeholder="login" />
</div>
<div class="form-group col-md-6 col-sm-6">
<input type="text" class="form-control" name="password" required placeholder="password" />
</div>
</div>
<input type="hidden" name="role" value="eleve">
<div class="col-md-12 col-sm-12 text-center">
<button type="submit" class="btn-contact">Terminer</button>
</div>

```

Donc ici on doit mettre notre nom,prénom,login et le password afin qu'il soit enregistrer dans la base de données grace à Donne.php qui récupère les données tapé au clavier pour ensuite les insérer à la base de données

```
<?php

echo 'nom : ' . $_GET['nom']. '<br>';
echo 'prénom : ' . $_GET['prenom']. '<br>';
echo 'login : ' . $_GET['login']. '<br>';
echo 'password : ' . $_GET['password']. '<br>';
echo 'role : ' . $_GET['role']. '<br>';
?>
<a href="listeuti.php" >Liste utilisateur</a>

<?php

    require_once("inc/connect-db.php");

    $nom = $_GET['nom'];
    $prenom = $_GET['prenom'];
    $login = $_GET['login'];
    $password = $_GET['password'];
    $role = $_GET['role'];

    $sql="INSERT INTO utilisateur ( nom, prenom,login,password,role)VALUES ( :nom,
:prenom, :login, :password, :role)";
    try {
        $query= $pdo -> prepare($sql);
        $query->bindParam(':nom' , $nom, PDO::PARAM_STR);
        $query->bindParam(':prenom' , $prenom, PDO::PARAM_STR);
        $query->bindParam(':login' , $login, PDO::PARAM_STR);
        $query->bindParam(':password' , $password, PDO::PARAM_STR);
        $query->bindParam(':role' , $role, PDO::PARAM_STR);
        $query->execute();

        header ("location:listeuti.php");
    }
    catch (PDOException $e){
        echo 'erreur : ' . $e->getMessage();
    }
?>
```

pour pouvoir se connecter Une fois connecter on peut ensuite consulter les informations sur le site index.php

```

if (isset($_GET['continent']) && !empty($_GET['continent'])) {
    $continent = $_GET['continent'];
} else {
    $continent = 'Africa';
}

```

Cette partie permet de récupérer le continent qu'on choisit dans le dropdown grace au header.php et aussi si aucun continent n'est sélectionné d'afficher Africa pour avoir un continent de base

header.php

```

require_once "inc/manager-db.php";
$listeContinents=getNomContinents();
?>
<?php foreach($listeContinents as $cont): ?>

    <a class="dropdown-item" href="index.php?continent=<?php echo"$cont-
>Continent"; ?>"><?php echo "$cont->Continent"; ?></a>

<?php endforeach ?>

```

Donc ici on récupère les continents qui sont dans la base de données grace a la fonction getNomContinents() et qui les affichent dans le dropdown

manager-db.php

```

function getNomContinents()
{
    global $pdo;
    $query = 'SELECT DISTINCT (Continent) FROM Country;';
    return $pdo->query($query)->fetchAll();
}

```

Une fois le continent choisit-on affiche le nom des pays leur population, la capitale et leurs langues parlées grace a l'index.php

```

<table class="table">
<tr>
<th>Nom</th>
<th>Population</th>
<th>Capital</th>
<th>Langue parlées</th>
</tr>

```

Donc ici on fait l'affichage du tableau

```

$desPays = getCountriesByContinent($continent);
<?php foreach ($desPays as $pays ) :?>
<tr>
<td> <?php echo $pays->Name; ?></td>
<td> <?php echo $pays->Population; ?></td><?php

```

Donc ici on affiche le nom et la population du pays grace a la fonction `getCountriesByContinent($continent);`

```

function getCountriesByContinent($continent)
{
    global $pdo;
    $query = 'SELECT * FROM Country WHERE Continent = :cont;';
    $prep = $pdo->prepare($query);
    $prep->bindValue(':cont', $continent, PDO::PARAM_STR);
    $prep->execute();
    return $prep->fetchAll();
}

```

```

$capital = $pays->Capital;
$cap = getCapital($capital);
<?php if(empty($cap)): ?>
    <td> <?php echo "vide"; ?></td>
<?php else:?>
    <td> <?php echo $cap[0]->Name; ?></td>
<?php endif; ?>

```

Ici on affiche la capitale du pays grace a la fonction `getCapital($capital)` pour cette fonction on récupère le numéro de la capitale par rapport au pays puis on l'envoie dans la fonction qui récupère le numéro de la capitale et nous renvoie le nom de la capitale et si la fonction nous renvoie aucune valeur on écrit vide

```

function getCapital($idcap)
{
    global $pdo;
    $query = 'SELECT Name FROM city WHERE id= :capital;';
    $prep=$pdo->prepare($query);
    $prep->bindValue(':capital', $idcap, PDO::PARAM_INT);
    $prep->execute();
    return $prep->fetchAll();
}

```

Ensuite on affiche les langues parlées par pays

```
<?php $nompays=$pays->Name;
$language=getLanguage($nompays);
foreach ($language as $langue) :?>
    <a> <?php echo $langue->Name; ?> </a>
    <a> <?php echo "-"; ?> </a>
<?php endforeach; ?>
<?php if(empty($language)):?>
    <a> <?php echo"vide"; ?> </a>
<?php endif; ?>
```

Donc ici on affiche les langues parlées grace a la fonction getLanguage(\$nompays); pour cette fonction on lui envoie le nom du pays afin qu'il nous renvoie les langues parlées grace a une jointure entre trois tables (language, countrylanguage et country) et si la fonction nous renvoie rien on écrit vide

```
function getLanguage($pays)
{
    global $pdo;
    $query = 'SELECT language.Name FROM language, countrylanguage,
country WHERE country.id=countrylanguage.idCountry AND
countrylanguage.idLanguage=language.id AND country.Name=:pays;';
    $prep=$pdo->prepare($query);
    $prep->bindValue(':pays', $pays, PDO::PARAM_STR);
    $prep->execute();
    return $prep->fetchAll();
}
```

Partie 2 : Enseignant : Mettre à jour les données

Ici pour l'enseignant il a les meme droits que l'utilisateur sauf qu'il peut mettre à jour les données des pays

Donc pour que l'enseignant dont le nom du role est prof est accès à la mise a jour des données on utilise

```
<?php if($_SESSION['role']=='prof' ): ?>
```

Donc pour que l'enseignant voit l'update mais pas l'utilisateur dans le tableau de l'index.php on va rajouter un if

```
<?php if($_SESSION['role']=='prof' ): ?>
    <th>Update</th>
<?php endif; ?>
```

et ensuite pour le lien on va mettre un if aussi

```
<?php if($_SESSION['role']=='prof' ): ?>
    <td> <a href="updateCoun.php?id=<?php echo $pays->id; ?>" >Update</a></td>
<?php endif; ?>
```

Donc quand on clique sur le lien on arrive sur updateCoun.php en récupérant l'id du pays qui le met dans le lien

```
<?php if($_SESSION['role']=='prof' ): ?>
<?php
require_once("inc/connect-db.php");

$id = $_GET['id'] ;
$pays = getpays($id);

$capital = $pays->Capital;
$cap = getCapital($capital);

$nompays=$pays->Name;
$language=getLanguage($nompays);?>

<form action="updateCountry.php" method="get" >
<fieldset>
<legend> <i>Pays</i></legend>
Nom :
<input type="text" name="Name" required value="<?php echo $pays->Name; ?>" /> <br />
Population :
<input type="text" name="Population" required value="<?php echo $pays->Population; ?>"
/> <br />
Capitale :
<input type="text" name="Capital" required value="<?php echo $cap[0]->Name; ?>" /> <br
/>
    <fieldset>
        <input type="submit" value="mettre à jour" />
        <input type="reset" value="Effacer" />
    </fieldset>
<?php endif; ?>
```

Donc ici tout d'abord on récupère l'id qu'il y a dans le lien afin de savoir de quel pays on doit récupérer les données qu'on récupère grâce aux memes fonctions qu'on a utilisées dans l'index.php afin de les afficher dans un formulaire

Ensuite quand on met à jour les données on clique sur le bouton "mettre à jour" qui nous renvoie sur updateCountry.php qui met à jour les données de la base par rapport aux modifications effectuer

```
<?php
require_once('updateCoun.php');
require_once("inc/connect-db.php");

$Name = $_GET['Name'];
$Population = $_GET['Population'];
$Capital = $_GET['Capital'];
$Langue = $_GET['Langue'];

$sql = "UPDATE country set Name=:Name,Population=:Population where id=:id";
$sql2 = "UPDATE city set Name=:Name where idcountry=:id";
try {

    $statement = $pdo->prepare($sql);
    $statement->bindParam(':id', $id, PDO::PARAM_INT);
    $statement->bindParam(':Name', $Name, PDO::PARAM_STR);
    $statement->bindParam(':Population', $Population, PDO::PARAM_STR);
    $stat = $pdo->prepare($sql2);
    $stat->bindParam(':id', $id, PDO::PARAM_INT);
    $stat->bindParam(':Name', $Capital, PDO::PARAM_STR);

    $statement->execute();
    $stat->execute();

    header("Location:index.php");
}
catch(PDOException $e){
    echo 'Erreur : '.$e->getMessage();
}
?>
<br>
<a href="index.php" >liste</a>
```

Partie 3 : Admin : Gérer les roles des utilisateurs

L'admin possède les memes droits que l'utilisateur et l'enseignant sauf qu'il a accès à la liste des utilisateur et peut la modifier donc au

```
<?php if($_SESSION['role']=='prof' ): ?>
```

on rajoute le role admin


```
<?php if($_SESSION['role']=='admin' || $_SESSION['role']=='prof' ): ?>
```

Donc pour accéder a la liste des utilisateurs on va utiliser que

```
<?php if($_SESSION['role']=='prof' ): ?>
```

listeuti.php

```
<?php if($_SESSION['role']=='admin'): ?>
<h1>Liste utilisateurs</h1>
<table border=2>
  <tr>
    <th>id</th>
    <th>nom</th>
    <th>prenom</th>
    <th>login</th>
    <th>password</th>
    <th>role</th>
    <th>update</th>
    <th>delete</th>
  </tr>

  <?php foreach ($listeUti as $utilisateur ) :?>
  <tr>
    <td><?php echo $utilisateur->id; ?></td>
    <td><?php echo $utilisateur->nom; ?></td>
    <td><?php echo $utilisateur->prenom; ?></td>
    <td><?php echo $utilisateur->login; ?></td>
    <td><?php echo $utilisateur->password; ?></td>
    <td><?php echo $utilisateur->role; ?></td>
    <td> <a href="updateUti.php?id=<?php echo $utilisateur->id; ?>" >update</a></td>
    <td> <a href="delete.php?id=<?php echo $utilisateur->id; ?>" >delete</a></td>
  </tr>
  <?php endforeach; ?>
<?php
require_once 'javascripts.php';

?>
<?php endif; ?>
```

Donc ici on crée le tableau et ensuite on intègre les valeurs liées aux utilisateurs et on rajoute un lien update ou delete afin de soit modifier ou supprimer un utilisateur

delete.php

```

<?php
session_start ();
if($_SESSION['role']=='admin' ): ?>
<?php
require_once('inc/connect-db.php');

if ( isset($_GET['id']) && !empty($_GET['id'])){
$id = $_GET['id'] ;
}
$query = "delete from utilisateur where id = :id ";
try {
    $prep = $pdo->prepare($query);
    $prep->bindParam(':id', $id, PDO::PARAM_INT);
    $prep->execute();

    header("Location:listeuti.php");
}
catch ( Exception $e ) {
    die ("erreur dans la requete ".$e->getMessage());
}
?>
<?php endif ?>

```

donc ici quand on clique sur delete on récupère l'id de l'utilisateur qui est dans le lien pour ensuite le supprimer de la base de données

si on clique sur update on arrive sur updateUti.php

```

<?php if($_SESSION['role'] == 'admin'): ?>
<?php
require_once("inc/connect-db.php");

//on récupère et on vérifie que l'id figure dans l'URL
if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $uti = getUti($id);
} else {
    // Redirect if id is not provided
    header('location: some_other_page.php');
    exit(); // Add exit() to stop the execution of the script
}
?>
<form action="update.php" method="get">
    <fieldset>
        <legend><i>Utilisateur</i></legend>
        Nom :
        <input type="text" name="nom" required value="<?php echo $uti->nom; ?>" /> <br
    />
        Prénom :
        <input type="text" name="prenom" required value="<?php echo $uti->prenom; ?>"
    /> <br />
        login :
        <input type="text" name="login" value="<?php echo $uti->login; ?>" /> <br />
        password :
        <input type="text" name="password" value="<?php echo $uti->password; ?>" /> <br
    />
        role :
        <select name="role">
            <option value="eleve" <?php if ($uti->role == 'eleve') {echo 'selected';}
?>>élève</option>
            <option value="prof" <?php if ($uti->role == 'prof') {echo 'selected';}
?>>prof</option>
            <option value="admin" <?php if ($uti->role == 'admin') {echo 'selected';}
?>>admin</option>
        </select>
        <input type="hidden" name="id" value="<?php echo $uti->id ?>" />
    </fieldset>
    <input type="submit" value="mettre à jour" />
    <input type="reset" value="Effacer" />
</form>
<?php endif; ?>

```

qui permet de récupérer les données de l'utilisateur par rapport a son id pour ensuite pouvoir les modifier et une fois modifier on arrive sur update.php

```

if($_SESSION['role']=='admin'): ?>
<?php
// à faire sur chaque donnée reçue
$nom = $_GET['nom'];
$prenom = $_GET['prenom'];
$login = $_GET['login'];
$password = $_GET['password'];
$role = $_GET['role'];

// on rédige la requête SQL
$sql = "UPDATE utilisateur set
nom=:nom,prenom=:prenom,login=:login,password=:password,role=:role
where id=:id";
try {
//on prépare la requête avec les données reçues
$stmt = $pdo->prepare($sql);
$stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
$stmt->bindParam(':prenom', $prenom, PDO::PARAM_STR);
$stmt->bindParam(':login', $login, PDO::PARAM_STR);
$stmt->bindParam(':password', $password, PDO::PARAM_STR);
$stmt->bindParam(':role', $role, PDO::PARAM_STR);
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
//On renvoie vers la liste des salariés
header("Location:listeuti.php");
}
catch(PDOException $e){
echo 'Erreur : '.$e->getMessage();
}
?>
<br>
<a href="listeuti.php" >liste</a>
<?php endif; ?>

```

Il permet de mettre à jour la base de données par rapport aux modifications effectuées