

INFO0074-1

# Informatique pour sciences économiques et de gestion

*MySQL-PHP*

Eddy Heins (eddy.heins@uliege.be)

Élise Vandomme (elise.vandomme@uliege.be)

# Objectifs

- Maîtriser MySQL avec PHP
- Réaliser une petite application Professeur et Cours. Elle sert de base pour votre projet.
- Gestion des droits d'accès avec login et signout
  - Backend administrateur
  - Frontend professeur

# Interfacer PHP et MySQL

Pour relier

- une application développée en PHP
- une base de données MySQL

On utilisera l'extension « mysqli » (*i* pour *improved*) incluse dans PHP depuis la version 5.

Elle permet d'exécuter sur la base de données, des requêtes écrites dans un fichier.php.

Par exemple :

```
$sql = "SELECT * FROM Professeur";  
$result = mysqli_query($conn,$sql);
```

# Connexion à la DB depuis PHP

- Pour manipuler la base de données (modifier ou extraire des données), il faut s'y connecter depuis le fichier PHP. Le code pour établir la connexion est

```
mysqli_connect($servername, $username, $password,$dbname);
```

- Les 4 arguments sont dans l'ordre :
  - Le nom du serveur : php2.hec.ulg.ac.be
  - Le nom d'utilisateur : s123456 (votre identifiant s)
  - Le mot de passe : votre mot de passe pour la DB (identique à celui de FileZilla)
  - Le nom de la base de données : mschyns\_s123456 (à remplacer par votre s)
- Ce code renvoie
  - Vrai : si la connexion avec la base de données a pu être établie
  - Faux : sinon

# LAB 0 – Connexion à la DB depuis PHP

Dans le fichier lab0.php, établissez une connexion avec votre base de données.

- Dans le fichier PHP, entre les balises `<?php` et `?>`, définissez les 4 variables :

```
$servername = "php2.hec.ulg.ac.be";  
$username = "votre_identifiant_s";  
$password = "votre_mot_de_passe_FileZilla";  
$dbname = "mschyns_votre_identifiant_s";
```

- Stockez le résultat (Vrai/Faux) de la commande de connexion dans une variable :

```
$conn = mysqli_connect($servername, $username, $password,$dbname);
```

- Vérifiez l'état de la connexion et affichez un message d'information :

```
if (!$conn) {  
    die("Connexion non réussie : " . mysqli_connect_error());  
}  
echo "Connexion réussie";
```

# Connexion à la DB depuis PHP

- Si le programme PHP est connecté au serveur MySQL, on peut y faire des requêtes MySQL.

```
$sql = "SELECT * FROM Professeur";  
$result = mysqli_query($conn,$sql);  
ou
```

```
$result = mysqli_query($conn,"SELECT * FROM Professeur");
```

- Le résultat de cette requête est stocké dans la variable \$result sous forme d'une liste dans le cas du SELECT ou booléen pour UPDATE, INSERT, DELETE...  
➔ Il faut traiter \$result (le résultat) selon le type de votre requête.
- Une bonne pratique est de rompre la connexion lorsqu'on n'a plus besoin de la base de données :

```
mysqli_close($conn);
```

# Afficher le résultat d'un SELECT

- On peut lire le résultat enregistrement par enregistrement avec la fonction `mysqli_fetch_array($result)`. Le résultat de cette fonction est une liste s'il y a un ou des enregistrements. Quand il n'y a pas d'enregistrement, il retourne `false`.
- Pour lire et afficher le résultat de la requête précédente, on peut utiliser une boucle *while* comme ce qui suit:

```
while($row = mysqli_fetch_array($result)){  
    echo $row['matricule']." ";  
    echo $row['nom']." ";  
    echo $row['prenom']." ";  
    echo $row['email']." ";  
    echo $row['GSM']." ";  
    echo $row['UER']." ";  
    echo "<br>";  
}
```

Nom d'une colonne de  
la table « Professeur »

# LAB 1 – Afficher enregistrements avec SELECT

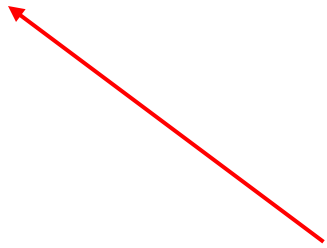
Dans le répertoire LAB1, modifier le fichier « list\_prof.php » pour afficher tous les enregistrements de la table Professeur :

- Etape 1 : Faire la requête pour sélectionner tous les professeurs de la table Professeur. Le résultat est sauvegardé dans la variable \$result
- Etape 2 :
  - Utiliser une boucle *while* pour afficher le résultat
  - Utiliser une instruction conditionnelle pour afficher un message d'information s'il n'y a pas de résultat.
- Etape 3 : Se déconnecter du serveur MySQL



# LAB 1 – Afficher enregistrements avec SELECT

- Etape 1 : Faire la requête SELECT pour sélectionner tous les professeurs de la table Professeur. Le résultat est sauvegardé dans la variable \$result :



Création de requête stockée dans une variable

- Autre solution possible :

# LAB 1 – Afficher enregistrements avec SELECT

- Etape 2 : afficher les résultats (condition et boucle while)

- Etape 3 : Se déconnecter du serveur MySQL

# Remarque sur \$\_SERVER

- Dans le LAB précédent, les informations de connexion n'étaient pas explicitement données. Par exemple : il n'y a pas « php2.hec.ulg.ac.be » codé en dur.

```
$servername = $_SERVER['SERVER_NAME'];  
$username = $_SERVER['DB_USER'];  
$password = $_SERVER['DB_PASS'];  
$dbname="mschyns_{$username}";  
$conn=mysqli_connect($servername, $username, $password,$dbname);
```

- Nous avons utilisé des variables **superglobales**, des variables internes qui sont toujours disponibles, quel que soit le contexte : \$\_SERVER['DB\_USER'], ...
- \$\_SERVER est un tableau contenant des informations générées par le serveur. Les entrées dépendent donc du serveur utilisé.
- NB : Nous avons déjà rencontré des variables superglobales avant, lorsque nous récupérions les informations d'un formulaire : \$\_POST[ ], \$\_GET[ ]

# Menu & style

- Afin que notre application CoursProf soit plus conviviale, nous aimerions avoir un menu de navigation



Matricule	Nom	Prénom	Email	GSM	UER
u200214	Dubois	Alain	alain.dubois@uliege.be	0477123456	Operations
u203658	Dopagne	Helene	helene.dopagne@uliege.be	0477123457	Langues

- Ainsi qu'un affichage des enregistrements sous la forme d'une table.

# LAB 2 – Ajouter le menu et style

- Utilisez le répertoire LAB2 contenant le fichier de style CSS. Modifier list\_prof.php pour y ajouter le menu : Professeur, Cours.

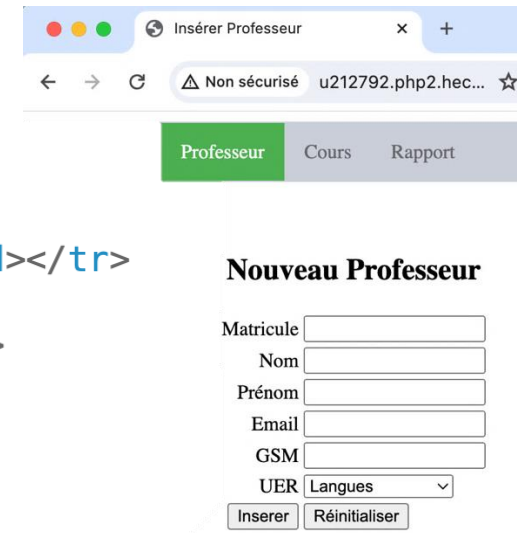
# LAB 3 – Afficher enregistrements avec SELECT (CSS)

- Utilisez le répertoire LAB3. Modifiez list\_prof.php pour afficher les enregistrements sous forme d'une table.

# LAB 4 – Ajouter enregistrement avec INSERT

- Dans le répertoire LAB4, modifier le fichier « insert\_prof.php » pour qu'il permette d'ajouter un professeur dans la table Professeur.
- Le fichier « insert\_prof.php » contient déjà un formulaire :

```
<!-- Formulaire pour ajouter un professeur-->
<form name="prof_insert" action="insert_prof.php" method="post">
  <table align="center" border=0>
    <tr><td align=right>Matricule </td><td><input type="text" name="matricule"></td></tr>
    <tr><td align=right>Nom </td><td><input type="text" name="nom"></td></tr>
    <tr><td align=right>Prénom </td><td><input type="text" name="prenom"></td></tr>
    <tr><td align=right>Email </td><td><input type="text" name="email"></td></tr>
    <tr><td align=right>GSM </td><td><input type="text" name="GSM"></td></tr>
    <tr><td align=right>UER </td><td><select name="UER">
      <option value="Langues">Langues</option>
      <option value="Economie">Economie</option>
      <option value="Finances et droit">Finances et droit</option>
      <option value="Operation">Operations</option>
      <option value="Management">Management</option>
    </select></td></tr>
    <tr><td align=right><input type="submit" name="sub_btn" value="Inserer"></td><td><input type="reset"
      name="res_btn" value="Réinitialiser"></td></tr>
  </table>
</form>
```



Insérer Professeur

Non sécurisé u212792.php2.hec...

Professeur Cours Rapport

**Nouveau Professeur**

Matricule

Nom

Prénom

Email

GSM

UER

- Il faut maintenant récupérer les données du formulaire et les traiter.

# LAB 4 – Ajouter enregistrement avec INSERT

- C'est le même fichier insert\_prof.php qui traitera les données.

Rappel : `<!-- Formulaire pour ajouter un professeur-->`  
`<form name="prof_insert" action="insert_prof.php" method="post">`

➔ Penser à utiliser une instruction conditionnelle avec *isset* pour ne pas avoir d'erreur lors du premier accès à la page insert\_prof.php

```
<?php
//Test si l'utilisateur a cliqué sur le bouton Insérer (au moins 1x)
if (isset($_POST['sub_btn'])){
    //Etape 1: Récupérer les données du formulaire

    //Etape 2: Vérifier la qualité des données fournies

    //Etape 3: Connecter au serveur MySQL

    //Etape 4: lancer la requête INSERT

    //Etape 5: déconnexion du serveur

}
?>
```

- Les étapes 1 à 5 ne sont exécutées que si l'utilisateur a cliqué au moins une fois sur le bouton « Insérer ».



# LAB 4 – Ajouter enregistrement avec INSERT

- Etape 1 : Récupérer les données du formulaire avec la méthode POST

# LAB 4 – Ajouter enregistrement avec INSERT

- Etape 2 : Vérifier la qualité des données fournies par l'utilisateur
- A minima, il faut que le matricule, nom, prénom et email soient complétés (donc différents d'une chaîne de caractères vides).  
NB : On pourrait inclure des tests de qualité plus poussés à cette étape comme
  - vérifier la forme du matricule (u suivi de 6 chiffres);
  - vérifier que l'adresse email se termine bien par uliege.be; etc
- Les étapes 3 à 5 ne sont exécutées que si le contrôle qualité donne satisfaction.

# LAB 4 – Ajouter enregistrement avec INSERT

- Etape 3 : Connexion au serveur MySQL (comme dans list\_prof.php)

# LAB 4 – Ajouter enregistrement avec INSERT

- Etape 4 : Requête INSERT INTO en affichant un message d'information
- Etape 5 : Déconnexion du serveur

# Utiliser le même code dans plusieurs pages

- Le code du menu de navigation a été recopié dans le fichier `insert_prof.php`, depuis `list_prof.php`.

Professeur

Cours

Rapport

```
<div class="menu"> <!-- début menu -->
  <ul class="menu">
    <li class="menu"><a class="active" href="list_prof.php">Professeur</a></li>
    <li class="menu"><a href="list_cours.php">Cours</a></li>
    <li class="menu"><a href="reporting.php">Rapport</a></li>
  </ul>
  <br><br>
</div> <!-- fin menu-->
```

- En termes de maintenance, plus il y aura de pages dans l'application, plus il y aura d'endroits à modifier en cas de changement de menu.  
➔ Ce n'est pas pratique.
- L'instruction *include* (ou *require*) permet d'inclure et d'exécuter le contenu du fichier spécifié. Par exemple, si nous écrivons le code du menu de navigation dans `menu.php` :

```
<div class="contenant"> <!-- début contenant -->
  <?php include 'menu.php'; ?>
</div> <!-- début contenu -->
```

# PHP – include/require

- L'instruction *include* (ou *require*) permet d'inclure et d'exécuter le contenu du fichier spécifié (HTML ou PHP).  
➔ Il permet la réutilisation de certaines parties du code.

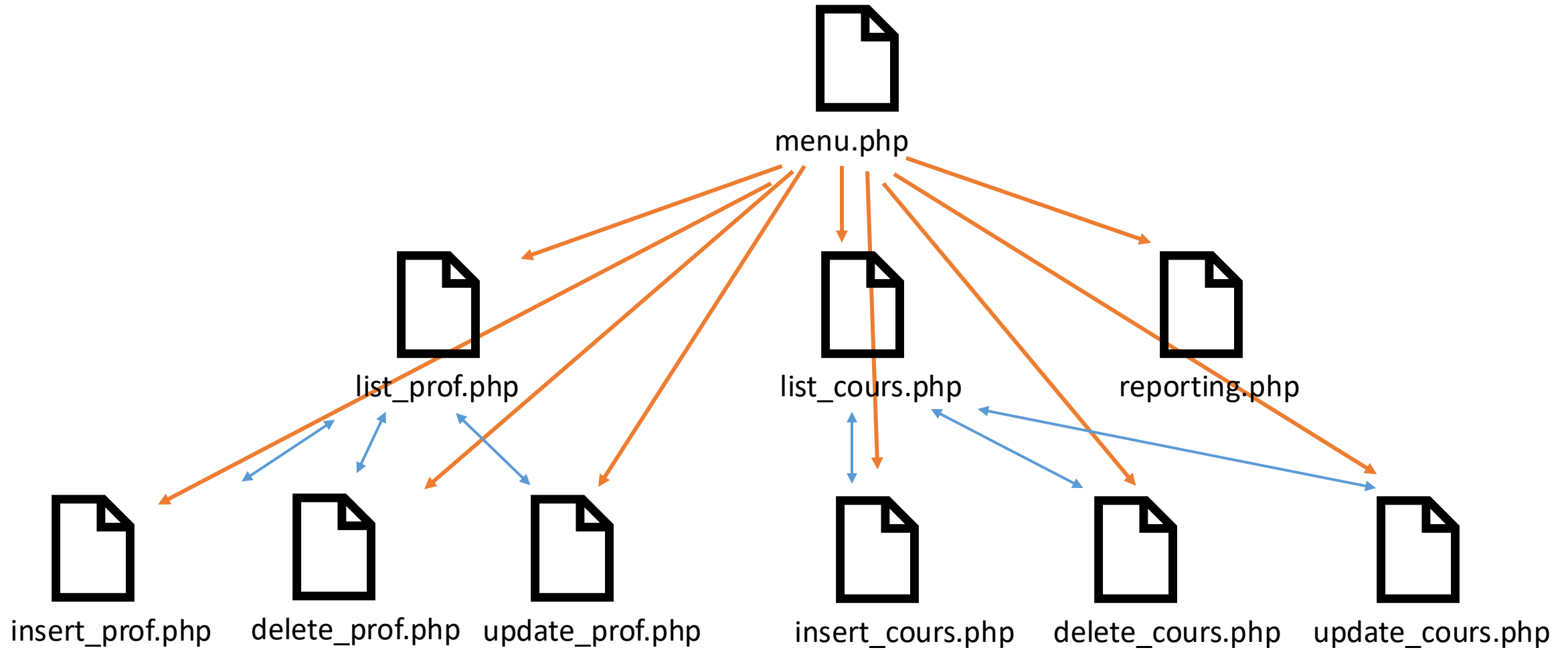
- Le fonctionnement d'*include* et *require* est similaire sauf quand il y a une erreur.

Dans le cas d'un fichier introuvable,

- *require* affiche des erreurs et arrête le programme complètement
  - *include* affiche des erreurs et le programme continue à s'exécuter normalement.
- Si on veut que PHP vérifie si le fichier a déjà été inclus, et si c'est le cas, ne l'inclut pas une deuxième fois, il faut utiliser
    - *require\_once*
    - *include\_once*

# LAB 5 – Include/Require

- L'application CoursProf que nous développons aura la structure suivante :



# LAB 5 – Menu.php

- Dans le répertoire du LAB 5, dans les fichiers list\_prof.php et insert\_prof.php, remplacez le code du menu de navigation par

```
<?php include 'menu.php'; ?>
```

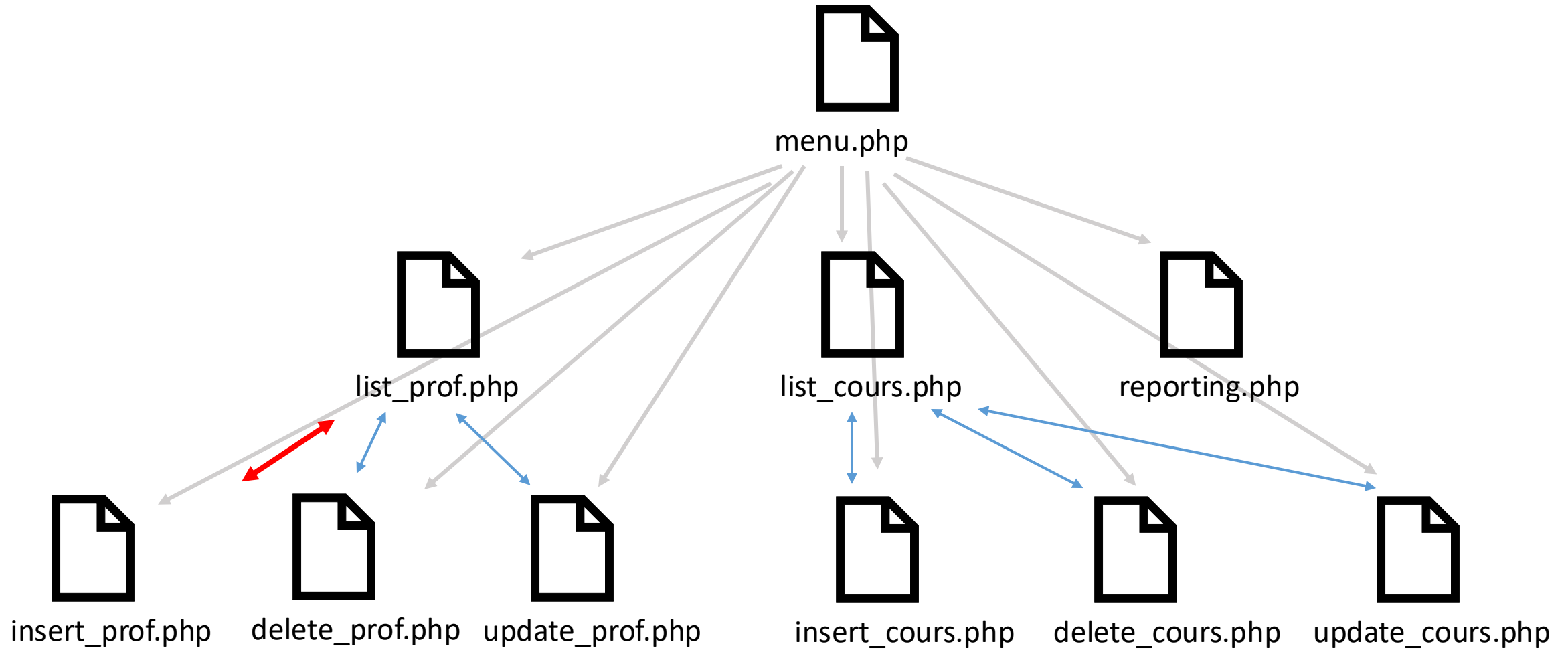
- Modifiez le fichier menu.php afin qu'il soit dynamique. Pour cela utilisez PHP et la variable superglobale \$\_SERVER['PHP\_SELF'] qui renvoie le nom de la page avec son chemin. L'instruction basename permet de ne garder que le nom de la page.

```
menu.php x
1 <div class="menu"> <!-- début menu -->
2 <ul class="menu"> <!-- Menu de navigation -->
3 <li class="menu"><a
4 <?php if (basename($_SERVER['PHP_SELF']) == "list_prof.php"){echo 'class="active"';}?>
5 href="list_prof.php">Professeur</a></li>
6 <li class="menu"><a
7 <?php if (basename($_SERVER['PHP_SELF']) == "list_cours.php"){echo 'class="active"';}?>
8 href="list_cours.php">Cours</a></li>
9 <li class="menu"><a
10 <?php if (basename($_SERVER['PHP_SELF']) == "reporting.php"){echo 'class="active"';}?>
11 href="reporting.php">Rapport</a></li>
12 </ul>
13 <br><br>
14 </div> <!-- fin menu-->
```



# LAB 5 – Menu.php

- L'application CoursProf que nous développons aura la structure suivante :



- Ajouter le lien entre `insert_prof` et `list_prof` dans le fichier `menu.php`

# LAB 5 – Menu.php

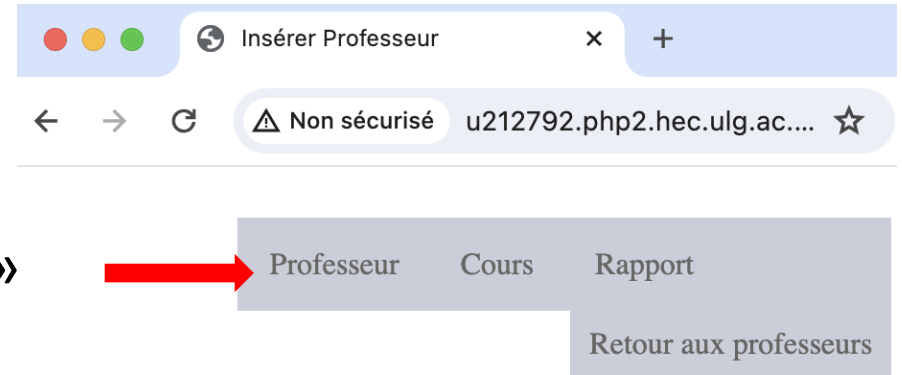
- Ajouter le lien entre « insert\_prof » et « list\_prof » dans le fichier « menu.php ».
  - Si la page actuelle est « list\_prof.php »,  
=> il faut pouvoir atteindre « insert\_prof.php »  
en cliquant sur « Insérer un professeur »
  - Si la page actuelle est « insert\_prof.php »,  
=> il faut pouvoir atteindre « list\_prof.php »  
en cliquant sur « Retour aux professeurs »

```
<ul class="menu_s"> <!-- Menu d'opérations -->
  <?php
    if (basename($_SERVER['PHP_SELF']) == "list_prof.php"){
      echo '<li class="menu_s"><a href="insert_prof.php">Insérer un professeur</a></li>';
    }
    elseif (basename($_SERVER['PHP_SELF']) == "insert_prof.php"){
      echo '<li class="menu_s"><a href="list_prof.php">Retour aux professeurs</a></li>';
    }
  ?>
</ul>
```

- Le style de la classe *menu\_s* est déjà défini dans le fichier css.

# LAB 5 – Menu.php

- Le rendu visuel de cette modification laisse à désirer :
- On aimerait que l'item « Professeur » du menu soit actif même lorsqu'on est sur la page « insert\_prof.php »
- Modifiez le code php du menu de navigation pour cela :



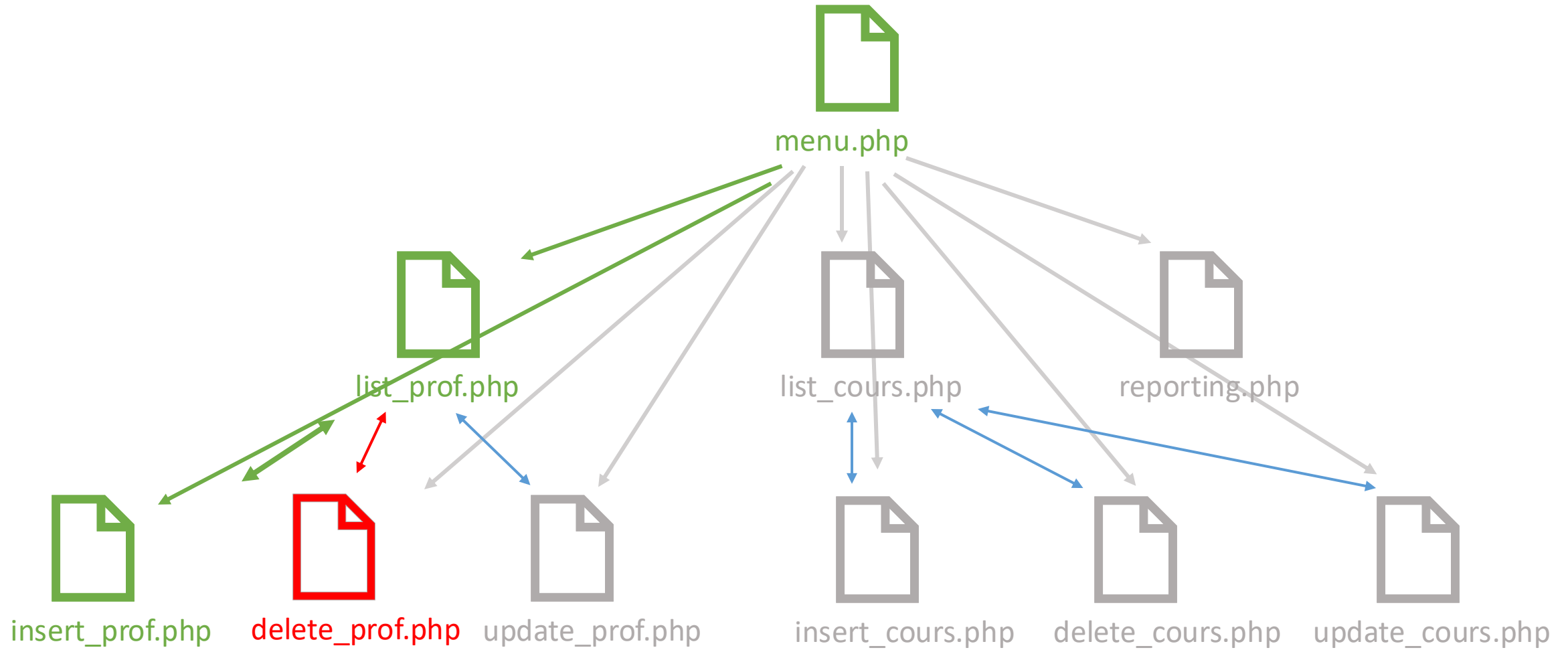
```
menu.php
1 <?php $page = basename($_SERVER['PHP_SELF']);?>
2 <div class="menu"> <!-- début menu -->
3 <ul class="menu"> <!-- Menu de navigation -->
4 <li class="menu"><a
5 <?php if ($page == "list_prof.php" || $page == "insert_prof.php"){
6 <echo 'class="active"';}?>
7 <href="list_prof.php">Professeur</a></li>
8 <li class="menu"><a
9 <?php if ($page == "list_cours.php"){echo 'class="active"';}?><href="list_cours.php">
10 Cours</a></li>
11 <li class="menu"><a
12 <?php if ($page == "reporting.php"){echo 'class="active"';}?><href="reporting.php">
13 Rapport</a></li>
14 </ul>
```

## Nouveau Professeur

Matricule	<input type="text"/>
Nom	<input type="text"/>
Prénom	<input type="text"/>
Email	<input type="text"/>
GSM	<input type="text"/>
UER	<input type="text" value="Langues"/>
<input type="button" value="Insérer"/> <input type="button" value="Réinitialiser"/>	

# Application CoursProf

- Récapitulatif des développements réalisés (en vert) :



- Ajoutons la possibilité de supprimer un professeur.

# Table des matières

- [Introduction PHP-MySQL & mysqli](#)
- [SELECT](#)
- [Superglobales : \\$ \\_SERVER](#)
- [INSERT INTO](#)
- [Require/include](#)
- [DELETE FROM](#)
- [UPDATE](#)
- [Rapports](#)
- [Exercices](#)
- [Backend application](#)
- [Superglobales : \\$ \\_SESSIONS](#)
- [Frontend application](#)
- [Remarques sur le projet](#)