|  |
| --- |
| FUTURE LINK |
| Dossier de Programmation |
| [Sous-titre du document] |

|  |
| --- |
| BLANC Jérémie, CORTIAL Nicolas, EMERY Antoine  08/06/2021 |

Table des matières

# I – Programmation php/HTML/CSS

Ouiiiiiiiiiiiiiiiiiii

## I.1 – Fonctionnalités pour l’étudiant

Pour permettre un affichage optimisé et le dynamisme du site web, nous avons utilisé le langage PHP pour faire le lien entre la base de données et l’interface utilisateur codée en HTML et CSS.

### I.1.a - affichage de la liste

Premièrement, nous devons distinguer selon ce que souhaitait faire l’utilisateur en venant sur la page quelle requête exécuter. Dans le cas où celui-ci aura effectué une recherche par mots-clés, une requête pouvant traiter ce mot-clé sera exécutée, sans quoi une simple requête de récupération sans critères prendra le relais.

**if**(isset($\_POST['rechercher']))

{

$reqListeStages = $linkpdo->prepare("SELECT stage.intitule AS intitule, entreprise.nom\_entreprise AS nom\_entreprise, stage.id\_stage AS id\_stage FROM stage, entreprise WHERE stage.id\_entreprise = entreprise.id\_entreprise AND entreprise.valider = 1 AND (entreprise.nom\_entreprise LIKE CONCAT('%',:recherche,'%') OR stage.competence1 LIKE CONCAT('%',:recherche,'%') OR stage.competence2 LIKE CONCAT('%',:recherche,'%') OR stage.competence3 LIKE CONCAT('%',:recherche,'%') OR stage.intitule LIKE CONCAT('%',:recherche,'%'))");

$reqListeStages->execute(**array**("recherche" => $\_POST['recherche']));

}

**else**

{

$reqListeStages = $linkpdo->prepare("SELECT stage.intitule AS intitule, entreprise.nom\_entreprise AS nom\_entreprise, stage.id\_stage AS id\_stage FROM stage, entreprise WHERE stage.id\_entreprise = entreprise.id\_entreprise AND entreprise.valider = 1");

$reqListeStages->execute();

}

Ensuite, il nous suffit d’afficher une élément de liste par ligne de la requête, on affichera ici l’intitulé du stage et l’entreprise, ainsi qu’un bouton permettant de consulter l’annonce.

<?php

**while**($data = $reqListeStages->fetch())

{

**echo** "<div class=**\"**elementListeStage**\"**><h3>".$data['intitule']."</h3><br/>Entreprise : ".$data['nom\_entreprise']."<br/>

   <form class=**\"**formVoirAnnonce pure-form**\"** method = **\"**POST**\"** action = **\"**ListeStagesEtudiant.php**\"**>

   <input type=**\"**hidden**\"** value = **\"**".$data['id\_stage']."**\"**name=**\"**id\_stage\_affiche**\"**/>

   <button class = **\"**boutonVoirAnnonce pure-button**\"** name = **\"**boutonVoirAnnonce**\"** type = **\"**submit**\"**> Voir l'annonce </button>

   </form> <br/>

   </div>";

}

?>

### I.1.b - affichage de l’annonce

Pour cette fonctionnalité, on effectue d’abord une requête récupérant l’ensemble des informations à afficher, ce qui nous permettra ensuite de concaténer le rendu final rempli de celles-ci avant de l’afficher.

$rendu = "";

**if**(isset($\_POST['boutonVoirAnnonce']))

{

//requête

$reqAffichageStage = $linkpdo->prepare("SELECT stage.intitule AS intitule, stage.description\_stage AS description\_stage, representant.nom\_representant AS nom\_representant, representant.prenom\_representant AS prenom\_representant, stage.competence1 AS competence1, stage.competence2 AS competence2, stage.competence3 AS competence3, stage.id\_stage as id\_stage, representant.id\_representant AS id\_representant FROM stage, representant, entreprise WHERE stage.id\_stage = :stage AND stage.id\_representant = representant.id\_representant");

$reqAffichageStage->execute(**array**("stage" => (int)$\_POST['id\_stage\_affiche']));

$dataAffichage = $reqAffichageStage->fetch();

$id\_representant = $dataAffichage['id\_representant'];

//concaténation du rendu

$rendu = "<span><h2>".$dataAffichage['intitule']."</h2><br/><br/><br/></span>";

$rendu = $rendu."<h3>Description</h3>".$dataAffichage['description\_stage']."<br/><br/><br/><h3>Representant</h3>".$dataAffichage['nom\_representant']." ".$dataAffichage['prenom\_representant']."<br/><br/><br/><h3>Compétences requises</h3>".$dataAffichage['competence1']." ".$dataAffichage['competence2']." ".$dataAffichage['competence3']."<br/><br/>

<br>

<form class=**\"**formReserverRDV pure-form**\"** method = **\"**POST**\"** action = **\"**ListeStagesEtudiant.php**\"**>

   <input type=**\"**hidden**\"** value = **\"**".$dataAffichage['id\_stage']."**\"**name=**\"**id\_stage\_reserve**\"**/>

   <input type=**\"**hidden**\"** value = **\"**".$dataAffichage['id\_representant']."**\"**name=**\"**id\_representant\_stage\_reserve**\"**/>

   <button class = **\"**boutonReserverRDV pure-button**\"** name = **\"**boutonReserverRDV**\"** type = **\"**submit**\"**> Réserver un entretien </button>

   </form>";

}

/

\*

\* Suite de la page

\*

\*

/

//affichage du rendu

     <?php echo $rendu; ?>

### I.1.c - réserver l’entretien et gestion des exceptions

// On vérifie si le bouton d’ajout a été cliqué

**if**(isset($\_POST['boutonReserverRDV']))

{

//on prépare la requête d’insertion de créneau

$reqReservationRDV = $linkpdo->prepare("INSERT INTO creneaux(id\_etudiant, id\_representant, id\_stage) VALUES (:etudiant, :representant, :stage)");

//On essaye d'exécuter la requête avec les informations fournies. Si l'étudiant possède déjà 3 créneaux, ou a déjà réservé un entretien pour  ce même stage, on récupère l’exception déclenchée par le trigger côté base de données.

**try**

{

$reqReservationRDV->execute(**array**("etudiant" => $\_SESSION['id'], "representant" => $\_POST['id\_representant\_stage\_reserve'],"stage" => $\_POST['id\_stage\_reserve']));

$message = "Entretien réservé";

**echo** '<script type="text/javascript">window.alert("'.$message.'");</script>';

header("Refresh:0");

}

**catch** (Exception $e)

{

$message = $e->getMessage();

$message = substr($message, 40, strlen($message)-40);

//On affiche le message d’exception personnalisée renvoyé par la BDD

**echo** '<script type="text/javascript">window.alert("'.$message.'");</script>';

header("Refresh:0");

}

}

L’affichage de la photo de profil de l'étudiant est permis via une rubrique dans la base de données, qui contient le chemin d’accès vers la photo stockée dans le système de fichiers, sous /upload/etu\_photos/\*nom de la photo\* (idem pour les CV)

Les uploads, pour commencer, sont gérés par un fichier dédié “upload.php”, qui se charge d’analyser quel type de fichier a été transféré et par quel type d’utilisateur, afin de le placer dans le dossier adéquat, en ayant au préalable supprimer l’ancien fichier.

$file\_type = $\_FILES['fileToUpload']['type'];

**if**($file\_type == "application/pdf")

{

*//supprimer le fichier existant*

*//requete*

  $reqSupprCV = $linkpdo->prepare("SELECT etudiant.lien\_cv\_etudiant from etudiant WHERE id\_etudiant = :id\_etudiant");

$reqSupprCV->execute(**array**("id\_etudiant" => $\_SESSION['id']));

**while**($data = $reqSupprCV->fetch())

    {

$file\_pointer = $data['lien\_cv\_etudiant'];

*// unlink() pour supprimer le fichier*

**if** (!unlink($file\_pointer))

{

**echo** "mise à jour impossible";

}

**else**

{

**echo** "misa à jour du CV...";

}

$target\_dir = "./uploads/etu\_CV/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$imageFileType = strtolower(pathinfo($target\_file,PATHINFO\_EXTENSION));

}

  $reqMAJCV = $linkpdo->prepare("UPDATE etudiant SET lien\_cv\_etudiant = :lien\_cv\_etudiant WHERE id\_etudiant = :id\_etudiant");

}

On contrôle ensuite par une succession de condition si le fichier est intègre et qu’on peut le stocker (variable $uploadOk), avant de finaliser l’upload comme suit :

*// si une erreur a été rencontrée*

**if** ($uploadOk == 0)

{

**echo** "Sorry, your file was not uploaded.";

}

*// si tout est ok, on essaye d'uploader le fichier*

**else**

{

*//upload du nouveau fichier*

**if** (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file))

   {

**echo** "The file".htmlspecialchars(basename($\_FILES["fileToUpload"]["name"])). " has been uploaded.";

   }

**else**

   {

**echo** "Sorry, there was an error uploading your file.";

   }

**if**($\_SESSION['utilisateur'] == 'etudiant')

    {

   header("Location:affichageProfilEtudiant.php");

    }

**else**

    {

       header("Location:profil\_entreprise.php");

    }

}

Finalement, de simples champs d’entrées sont déclarés dans le code HTML pour permettre le passage d’un fichier de moins de 50ko :

<form action=\"upload.php**\"** method=**\"**post**\"** enctype=**\"**multipart/form-data**\"**>

  Selectionnez la photo (moins de 500 ko) :<br/>

  <input class = **\"**pure-button**\"** type=**\"**file**\"** name=**\"**fileToUpload**\"**><br/>

  <button class = **\"**boutonUploadImage pure-button**\"**   type = **\"**submit**\"** name=**\"**submit**\"**>Uploader</button>

</form>

<form action=**\"**upload.php**\"** method=**\"**post**\"** enctype=**\"**multipart/form-data**\"**>

  Selectionnez le CV (moins de 500 ko) :<br/>

  <input class = **\"**pure-button**\"** type=**\"**file**\"** name=**\"**fileToUpload**\"**><br/>

  <button class = **\"**boutonUploadImage pure-button**\"**   type = **\"**submit**\"** name=**\"**submit**\"**>Uploader</button>

</form>

<br/><br/><br/>

<form action=**\"**affichageProfilEtudiant.php**\"** method=**\"**post**\"**><br/>

  <button class = **\"**boutonUploadImage pure-button**\"**   type = **\"**submit**\"** name=**\"**annuler**\"**>Annuler</button>

</form>

L’affichage des rendez-vous se fait en deux temps : une première partie en haut de page prépare et exécute la requête de récupération des informations des différents entretiens via l’identifiant de session :

$reqRDV = $linkpdo->prepare("

SELECT entreprise.nom\_entreprise AS nom\_entreprise,

representant.nom\_representant AS nom\_representant,

representant.prenom\_representant AS prenom\_representant,

stage.intitule AS intitule,

creneaux.date\_horaire AS date\_horaire,

representant.id\_representant AS id\_representant,

stage.id\_stage AS id\_stage

FROM creneaux, entreprise, representant, stage

WHERE creneaux.id\_etudiant = :etudiant AND

representant.id\_entreprise = entreprise.id\_entreprise AND

creneaux.id\_stage = stage.id\_stage AND

creneaux.id\_representant = representant.id\_representant");

$reqRDV->execute(**array**("etudiant" => $\_SESSION['id']));

Ensuite, pour chaque élément retourné dans le résultat de la requête, on affiche les informations du créneau dans une <div> qui contiendra également les boutons d’annulation et de consultation de l’annonce.

**while**($data = $reqRDV->fetch())

{

**echo** "<div class = **\"**creneauEtudiant**\"**> <h3> Entreprise : ".$data['nom\_entreprise']."</h3><br/> <b> Intitulé : </b> ".$data['intitule']."<br/> <b> Répresentant : </b> ".$data['nom\_representant']." ".$data['prenom\_representant']."<br/> <b> Date et heure : </b>".$data['date\_horaire']."<br/>

<form class=**\"**formVoirAnnonce pure-form**\"** method = **\"**POST**\"** action = **\"**ListeStagesEtudiant.php**\"**> <input type=**\"**hidden**\"** value = **\"**".$data['id\_stage']."**\"** name=**\"**id\_stage\_affiche**\"** /><button class = **\"**boutonVoirAnnonce pure-button**\"** name = **\"**boutonVoirAnnonce**\"** type = **\"**submit**\"**> Voir l'annonce </button> </form>"."

<form class=**\"**pure-form**\"** method = **\"**POST**\"** action = **\"**ConsultationMenu.php**\"**>

<input type=**\"**hidden**\"** value = **\"**".$data['id\_representant']."**\"** name=**\"**id\_representant**\"** />

<input type=**\"**hidden**\"** value = **\"**".$data['id\_stage']."**\"** name=**\"**id\_stage\_affiche**\"** />

<input type=**\"**hidden**\"** value = **\"**".$data['date\_horaire']."**\"** name=**\"**date\_horaire\_creneau\_suppr**\"** />

<button class = **\"**boutonAnnulerCreneau pure-button**\"** name = **\"**boutonAnnulerCreneau**\"** type = **\"**submit**\"**> Annuler le rendez-vous </button>

</form>

</div>";

}

Le bouton d’annulation fait appel à une procédure SQL qui se charge de retirer l’élément de la table (voir la section sur les triggers).

**if**(isset($\_POST["boutonAnnulerCreneau"]))

{

$reqSuppr = $linkpdo->prepare("CALL supressionCreneau(:etudiant, :representant, :stage, :date\_heure);");

$reqSuppr->execute(**array**("etudiant" => $\_SESSION['id'], "representant" => $\_POST['id\_representant'], "stage" => $\_POST['id\_stage\_affiche'], "date\_heure" => $\_POST['date\_horaire\_creneau\_suppr']));

header("Refresh:0");

}

Le bouton de consultation de l’annonce est une simple redirection vers la liste des stages avec l’identifiant du stage passé en paramètres par un champ hidden dans le formulaire.

ENTREPRISE

Profil

Le profil de l’entreprise contient son logo, affiché de la même manière que la photo des étudiants, son nom, sa présentation, et la liste de ses représentants.

Des boutons de modification et/ou d’ajout/suppression permettent de traiter les champs correspondants, et sont affichés de manière tout à fait classique.

Mes offres de stage

Cette liste des stage pour l’entreprise a été codée comme l’as pu être la liste des stages proposés pour l’étudiant plus haut.

Planning entreprise