

Rapport de projet web

Mahydine LAZZOULI 204

Yoan ESTRADE 201

Hachimi BOUIZEGARENE 201

Imane BENYETTOU 201

Présentation du projet	2
Fonctionnalités et implémentations	2
Création d'un espace/inscription	2
Connexion et déconnexion	4
Fonctionnalités de la carte	6
Bilan et améliorations	9

Présentation du projet

Le site Soccer Search permet de pouvoir trouver des clubs de football en fonction de son niveau, sa tranche d'âge et sa localisation. En entrant ces informations, les clubs correspondant à ces critères s'affichent sur la carte via des marqueurs, et permettent de candidater directement depuis la carte via une fiche club.

Il permet aussi d'avoir des espaces dédiés pour les candidats, qui peuvent visualiser les clubs auxquels ils ont candidaté, et suivre l'évolution de leur candidature, mais aussi un espace recruteur/club, ayant accès à tous les joueurs ayant envoyé leur candidature. Ces espaces peuvent être créés via un formulaire de création de compte accessible sur la navbar/header.

Fonctionnalités et implémentations

Création d'un espace/inscription

La création d'un espace candidat ou recruteur est réalisée via la page d'inscription

[Trouver un club](#)[se connecter](#)

Nom d'utilisateur

Mot de passe

Validation de Mot de passe

Crée compte

Si vous avez déjà un compte : [Se connecter](#)

Dans le cas où l'utilisateur possède déjà un compte, il à la possibilité d'être redirigé vers la page de connexion, où il pourra accéder à son espace avec son pseudo et mot de passe habituels.

```
<form method="POST">
  <label class="username">Nom d'utilisateur</label>
  <input type="text" name="username" placeholder="Nom d'utilisateur">
  <label class="password">Mot de passe</label>
  <input type="password" name="password" placeholder="Mot de passe">
  <label class="password">Validation de Mot de passe</label>
  <input type="password" name="passwordconfirm" placeholder="Mot de passe">
  <label class="validate">Crée compte</label>
  <div class="submit"><div id="txt">Soumettre</div></div>
  <label class="err"><?php if(isset($errorMsg)){echo ' <span class="errorMsg">'. $errorMsg.' </span>';}else{} ?></label>
  <label class="creat">Si vous avez déjà un compte : <a href="connexion.php">Se connecter</a></label>
</form>
<?php require 'ressources/footer.php'; ?>
```

Le traitement du formulaire une fois complété et validé par l'utilisateur est réalisé dans un fichier différent inscription-traitement.php :

```
if(isset($_POST['validate'])){ //si bouton valider est cliqué
  if(!empty($_POST['username']) && !empty($_POST['password']) && !empty($_POST['passwordconfirm'])){ //si rien n'est vide
    if ctype_alnum(str_replace(' ', '', $_POST['username'])){ //si aucun caractère interdit
      if($_POST['password'] == $_POST['passwordconfirm']){ //si les mdp sont pareils
        if(strlen($_POST['username']) >= 3){ //si username > 3
          if(strlen($_POST['password']) >= 8){ //si mdp > 8
            if($_POST['username'] != $_POST['password']){// si le mdp et l'username sont different

              $username = htmlspecialchars($_POST['username']);
              $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

              //requette de recherche d'occurence du nom d'utilisateur
              $userAlreadyExists = $bdd->prepare('SELECT username from utilisateurs where username = :username');
              $userAlreadyExists->bindParam(':usernameToSearch', $username);
              $userAlreadyExists->execute();

              if($userAlreadyExists->rowCount() == 0){ //si le pseudo n'a pas été trouvé

                //requette d'insertion d'utilisateur en bdd
                $insertUser = $bdd->prepare('INSERT INTO utilisateurs(username, userPassword) VALUES(:username, :password)');
                $insertUser->bindParam(':username', $username);
                $insertUser->bindParam(':userPassword', $password);
                $insertUser->execute();

                //requette de récupération de l'id de l'utilisateur
                $getUserInfos = $bdd->prepare('SELECT idUser FROM utilisateurs WHERE username = :username');
                $getUserInfos->bindParam(':username', $username);
                $getUserInfos->execute();
                $userInfos = $getUserInfos->fetch();

                //initialise les variables de session
                $_SESSION['logged'] = true;
                $_SESSION['idUser'] = $userInfos['idUser'];
                $_SESSION['username'] = $username;

                header('Location: index.php');
                exit();
              }else{
                $errorMsg = 'Le nom d\'utilisateur est déjà utilisé';
              }
            }
          }
        }
      }
    }
  }
}
```

Une fois le formulaire rempli, le fichier effectue les vérifications nécessaires concernant la conformité du mot de passe, l'existence ou non de l'utilisateur dans la base de données. Dans le cas où tout fonctionne, une nouvelle session est créée pour l'utilisateur, et ses informations seront stockées dans la base de données dédiée.

```
    }else{
        $errorMsg = 'Le mot de passe ne peut pas être identique au nom d\'utilisateur';
    }
    }else{
        $errorMsg = 'Le mot de passe doit contenir au moins 8 caractères';
    }
    }else{
        $errorMsg = 'Le nom d\'utilisateur doit contenir au moins 3 caractères';
    }
    }else{
        $errorMsg = 'Les mots de passe sont différents';
    }
    }else{
        $errorMsg = 'Les seuls caractères spéciaux autorisés sont (- , _ , .)';
    }
    }else{
        $errorMsg = 'Veuillez compléter tout les champs';
    }
}
```

Connexion et déconnexion

Trouver un club



se connecter

Nom d'utilisateur

root

Mot de passe

....

Envoyer

Soumettre

Si vous n'avez pas de compte : [Crée un compte](#)

Les fonctions de connexion et déconnexion sont implémentées dans le fichier navbar.php, utilisé sur toutes les pages du site

```
<div id="nav_bar">
    <?php
        if(isset($_SESSION['logged']) && $_SESSION['logged']){
            echo '<a href="mescandidatures.php"> Espace Club <br>';
            echo isset($_SESSION['username'])?$_SESSION['username']:'';
            echo '<a/>';
        }else{
            echo '<a href="index.php">Trouver un club<a/>';
        }
    ?>

    
    <?php
        if(isset($_SESSION['logged']) && $_SESSION['logged'])
            echo '<a href="actions_php/deconnexion.php">déconnexion<a/>';
        else
            echo '<a href="connexion.php">se connecter<a/>';
    ?>
</div>
```

La page de connexion 'connexion.php' présente les champs de connexion classiques et propose à l'utilisateur de pouvoir créer un compte s'il ne l'a pas encore fait.

```
1 <?php
2 $css = '<link rel="stylesheet" href="css/style.css">
3       <link rel="stylesheet" href="css/login.css">';
4 if(isset($_SESSION['logged']) && $_SESSION['logged'])
5     header('Location: index.php');
6 $pageName = 'LogIn - SoccerSearch';
7 require 'ressources/header.php';
8 require 'actions_php/connexion_traitement.php';
9 ?>
10 <div id="content">
11     <?php require 'ressources/navbar.php'; ?>
12 </div>
13 <form method="POST">
14     <label class="username">Nom d'utilisateur</label>
15     <input type="text" name="username" placeholder="Nom d'utilisateur">
16     <label class="password">Mot de passe</label>
17     <input type="password" name="password" placeholder="Mot de passe">
18     <label class="validate">Envoyer</label>
19     <div class="submit"><div id="txt"><input type="submit" name="validate"></div></div>
20     <label class="err"><?php if(isset($errorMsg)){echo '<span class="errorMsg">'.$errorMsg.'</span>';} ?></label>
21     <label class="creat">Si vous n'avez pas de compte : <a href="inscription.php"> Crée un compte</a></label>
22 </form>
23 <?php require 'ressources/footer.php'; ?>
```

Le traitement du formulaire une fois complété et validé par l'utilisateur est réalisé dans un fichier différent 'connexion-traitement.php' :

```

if(isset($_POST['validate'])){//si bouton valider est cliqué
    if(!empty($_POST['username']) && !empty($_POST['password'])){ //si rien n'est vide

        $username = htmlspecialchars($_POST['username']);
        $password = htmlspecialchars($_POST['password']);

        $userExist = $bdd->prepare('SELECT * FROM utilisateurs WHERE username = :username'); //prepare la requette get info
        $userExist->bindParam(':username', $username); //donne le paramètre username a la requette pour chercher
        $userExist->execute();

        if($userExist->rowCount() > 0){ //si un utilisateur avec cet username est trouvé alors
            $userInfos = $userExist->fetch(); //converti le resultat de requette en tableau exploitable en php
            if(password_verify($password, $userInfos['userPassword'])){
                $_SESSION['userInfos'] = $userInfos;
                $_SESSION['logged'] = true;
                $_SESSION['idUser'] = $userInfos['idUser'];
                $_SESSION['username'] = $userInfos['username'];
                header('Location: index.php');
                exit();
            }else{
                $errorMsg = 'Mot de passe incorrect';
            }
        }else{
            $errorMsg = 'Aucun compte associé à ce nom d\'utilisateur';
        }
    }else{
        $errorMsg = 'Veuillez compléter tout les champs';
    }
}

```

Celui-ci recherche dans la base si l'utilisateur existe déjà, si le mot de passe correspond bien à celui défini lors de l'inscription. Si le nom d'utilisateur entré est introuvable, un message d'erreur est renvoyé à l'utilisateur.

Fonctionnalités de la carte

La carte est intégrée au site via l'API OpenStreetMap et MapTiler.

```

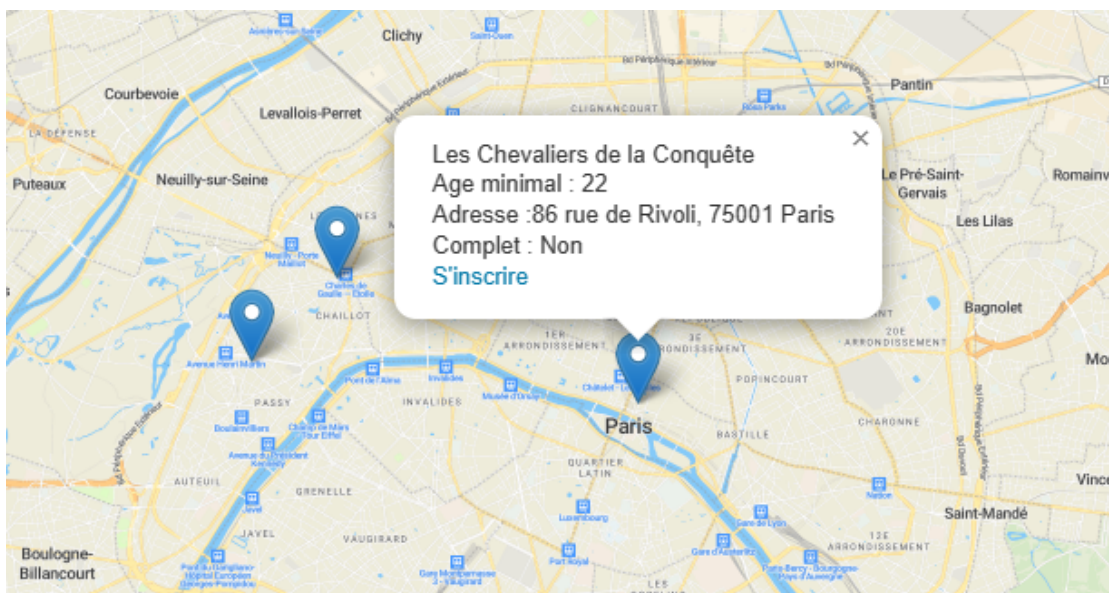
// Afficher la MAP
var map = L.map('mapid').setView([48.88, 2.33], 10.5);
L.tileLayer('https://api.maptiler.com/maps/streets-v2/{z}/{x}/{y}.png?key=ZXWma8gDUYjZyNg2h6Y5', {attribution : '<a href="https://www.maptiler.com/copyright/" target="_blank">©copy; MapTiler</a> <a href="https://www.openstreetmap.org/copyright" target="_blank">©copy; OpenStreetMap contributors</a>',}).addTo(map);
function ajaxPlacerMarqueurs(ageMin, ageMax){
    <?php
    echo 'let url; let data;';
    foreach($clubs as $club): ?>
        if(ageMax > <?php echo $club['age'] ?> && <?php echo $club['age'] ?> > ageMin){
    <?php
        echo 'url = "https://us1.locationiq.com/v1/search";';
        data = "key=pk-f6fa35e331b50cb08b98353c9410c66e&q=" . $club['Adresse'] . ' &format=json";';
        echo '$.ajax({';
            type: "GET",
            dataType: "JSON",
            url: url,
            data:data,
            error: function(){console.log("pb in .ajax")},
            success: function(result){';
                echo 'Marqueur_bdd= L.marker([result[0].lat, result[0].lon]).addTo(map).bindPopup("';
                echo $club['NomClub'] . ' <br> Age minimal : ' .
                $club['age'] . ' <br> Adresse : ' .
                $club['Adresse'] . ' <br> Complet : ' .
                if($club['Nbr_adh_max'] - $club['Nbr_adh'] <= 0){
                    echo'Oui <br>';
                }
                else{
                    echo'Non <br>';
                }
                echo '<a onclick=hrefCandidature(\"' . str_replace(" ", "&nbsp;", $club['NomClub']) . '\", ' . $club
                ['IdClub'] . ');>S\'inscrire</a>\"';
            echo '));'; ?>
        }
    <?php

```

La recherche du club se fait de manière très simple, avec un champ de recherche pour la ville, et un sélecteur pour la tranche d'âge, montrant ainsi à l'utilisateur les clubs recrutant pour sa propre tranche d'âge. Les clubs s'affichent sous forme de marqueurs bleus.



Si l'utilisateur clique sur l'un des marqueurs, une carte s'affiche en pop-up, indiquant les informations clés du club : son nom, son adresse, l'âge minimal requis, et s'il est complet ou non, ainsi qu'un bouton d'inscription direct pour y déposer sa candidature.



Les candidatures une fois remplies sont traitées par le fichier 'candidature-traitement.php'.

Le fichier s'assure d'abord qu'aucun champ n'est vide, et stocke la candidature dans la base de données dédiée aux candidatures.

```
$NomClub = $_GET['NomClub'];
$IdClub = $_GET['IdClub'];
$IdUtil = $_SESSION['IdUser'];

if(isset($_POST['validate'])){//si bouton valider est cliqué
    if( !empty($_POST['prenom']) &&
        !empty($_POST['nom']) &&
        !empty($_POST['email']) &&
        !empty($_POST['phonenumber']) &&
        !empty($_POST['age'])
    ){ //si rien n'est vide

        $prenom = htmlspecialchars($_POST['prenom']);
        $nom = htmlspecialchars($_POST['nom']);
        $email = htmlspecialchars($_POST['email']);
        $phonenumber = htmlspecialchars($_POST['phonenumber']);
        $age = htmlspecialchars($_POST['age']);

        $demandeExist = $bdd->prepare('SELECT * FROM candidature WHERE IdUtil = :IdUtil And IdClub = :IdClub'); //prepare la requette
        $demandeExist->bindParam(':IdUtil', $IdUtil); //donne le paramètre IdUtil à la requette pour chercher
        $demandeExist->bindParam(':IdClub', $IdClub); //donne le paramètre IdClub à la requette pour chercher
        $demandeExist->execute();

        if($demandeExist->rowCount() == 0){ //si un aucune demande avec ces deux infos sont trouvées alors

            $sql = "INSERT INTO candidature (IdUtil, IdClub, Nom, Prénom, email, numTel, age)
                VALUES (:IdUtil, :IdClub, :nom, :prenom, :email, :phonenumber, :age)";

            $trt = $bdd->prepare($sql);
            $trt->execute(array(':IdUtil' => $IdUtil, ':IdClub' => $IdClub, ':nom' => $nom, ':prenom' => $prenom, ':email' => $email,
                                ':phonenumber' => $phonenumber, ':age' => $age));

            header('Location: candidatureconfirmee.php');
            exit();

        }else{
            $errorMsg = 'Demande pour ce club déjà réalisée';
        }
    }else{
        $errorMsg = 'Veuillez compléter tout les champs';
    }
}
```

Les candidatures des joueurs sont accessibles via l'espace candidat, et offre l'option de retirer sa/ses candidature(s) à un ou plusieurs clubs, qui sera supprimé de la base de données.

```
$getCandidature = $bdd->prepare('SELECT `IdClub`, `Nom`, `Prénom`, `email`, `numTel`, `age` FROM `candidature` WHERE IdUtil = :IdUtil');
$getCandidature->bindParam(':IdUtil', $_SESSION['IdUser']); //donne le paramètre IdUtil à la requette pour chercher
$getCandidature->execute();

$lesCandidature = $getCandidature->fetchAll();

// Définition des dimensions du tableau
$lignes = count($lesCandidature);
$colonnes = count($lesCandidature[0])/2;

$getCandidateClubs = $bdd->prepare('SELECT NomClub FROM clubfoot WHERE IdClub IN (SELECT IdClub FROM candidature)');
$getCandidateClubs->execute();
$getCandidateClubs = $getCandidateClubs->fetchAll();

// Génération du tableau
$columns = array('Club', 'Nom', 'Prénom', 'Email', 'Numéro de téléphone', 'Age renseigné', 'Annuler');

echo '<table>';
echo '<tr>';
foreach ($columns as $column) { //crée les titres de colonnes
    echo '<th>' . $column . '</th>';
}
echo '</tr>';

$row_colors = array('#f6f6f6', '#ffffff'); //permettra d'alterner les couleurs de chaque lignes avec $i et un modulo
$i = 0;

for ($i = 0; $i < $lignes; $i++) {
    $color = $row_colors[$i % 2];
    echo '<tr style="background-color: ' . $color . '">';
    echo '<td>' . $getCandidateClubs[$i][0] . '</td>';
    for ($c = 0; $c < $colonnes-1; $c++) {
        echo '<td>' . $lesCandidature[$i][$c+1] . '</td>';
    }
    echo '<td>' . '<p onclick="deleteCandidature(' . $c . ')">Supprimer</p>' . '</td>';
    echo '</tr>';
    $i++;
}
echo '</table>';
```


Bilan et améliorations

Concernant les améliorations possibles au niveau de la carte, nous aurions pu implémenter une fonctionnalité prenant en compte l'adresse personnelle du candidat, et créant un itinéraire complet pour se rendre au club sélectionné. Le visuel de la carte pourrait également être amélioré, avec des marqueurs plus personnalisés et de meilleures cartes pop-up, avec des photos par exemple.