

Projet n°1

Bornes de recharge véhicules électriques

RT1 HUFFMAN

SAE 203














Sommaire

1. Présentation du projet et de l'arborescence des fichiers.....	2
2. Page affichage.....	3
3. Page ajout.....	4
4. Page suppression.....	6
5. Page recherches.....	8

1. *Présentation du projet et de l'arborescence des fichiers*

Ce projet permet de gérer l'ensemble des bornes de recharge pour les véhicules électriques en France. La base de données contient des données relatives à la localisation et aux caractéristiques techniques des bornes de recharge. Nous avons réalisé 4 pages pour le site web présentant la base de données, une page affichage, d'ajout, de suppression et de recherches.

Toutes les pages html et php sont au même niveau à la racine du projet. Il y a également deux dossiers : BDD pour la base de données qui contient le fichier sql et "BaseBootstrap". Le fichier "BaseBootstrap" contient les fichiers nécessaires au bon fonctionnement de Bootstrap, un dossier "css" avec les feuilles de style Bootstrap ainsi que les feuilles de style que nous avons créé "afficherAmenageurs.css" et "ajouterAmenageur.css". Au même niveau que le dossier "css" nous avons les dossiers : "fonts" pour les polices, "img" pour les images et "js" pour les scripts.

 BDD	fichier bdd	2 weeks ago
 BaseBootstrap	Update afficherAmenageurs.css	last week
 Presentation_sae203.pdf	Add files via upload	7 hours ago
 README.md	Update README.md	2 weeks ago
 SAE203_TSANGUE_GRANVISIR_CLERC.pdf	Add files via upload	10 hours ago
 afficherAmenageurs.php	Update afficherAmenageurs.php	last week
 ajouterAmenageur.html	Update ajouterAmenageur.html	last week
 ajouterAmenageur.php	Update ajouterAmenageur.php	last week
 recherches.php	Update recherches.php	10 hours ago
 supprimerAmenageurs.html	Update supprimerAmenageurs.html	last week
 supprimerAmenageurs.php	Update supprimerAmenageurs.php	last week

2. Page affichage

La page `afficherAmenageurs.php` contient le code html et php pour afficher le contenu de la table "amenageurs" de notre base de données.

```
<head>
  <title>Affichage de l'Amenageur</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <link rel="stylesheet" href="BaseBootstrap/css/afficherAmenageurs.css">
```

Nous avons utilisé des classes bootstrap et une feuille de style css pour la mise en page de tout le site web.

```
<?php

try {
    // Récupération des données
    $servername = "127.0.0.1";
    $dbname = "BUTRT1_lg409538";
    $username = "lg409538";
    $userpassword = "MDP_lg409538";
    //echo 'ca ma';
    $lienBDD = new PDO("mysql:host=$servername;dbname=$dbname", "$username", "$userpassword");
    $lienBDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    //echo "Acces BDD réaliser";
```

Avant de pouvoir afficher le contenu de la base de données, on commence par se connecter à la base de données.

```
//Prepare la requête
$requeteSQL = $lienBDD->prepare("SELECT siren, nom, contact FROM amenageurs");
$requeteSQL->execute();
// Affichage des résultats dans un tableau
$tab = $requeteSQL->fetchAll(PDO::FETCH_ASSOC);
```

On prépare ensuite la requête avant de l'exécuter.

```
if ($tab) {
    foreach ($tab as $row) {
        echo "<tr><td>" . $row["siren"] . "</td><td>" . $row["nom"] . "</td><td>" . $row["contact"] . "</td></tr>";
    }
}
else {
    echo "<tr><td colspan='3'>0 résultats</td></tr>";
```

On affiche ensuite le résultat sous forme de tableau à l'aide d'une boucle for.

3. Page ajout

Le code est séparé dans deux pages. La page ajouterAmenageur.html contient le formulaire pour ajouter une entrée dans la base de données.

```
<div class="col-md-8 text-left">
  <h1>Amenageur</h1>
  <section class="col-md-7">
    <form action="ajouterAmenageur.php" method="get">
      <fieldset>
        <legend>Ajouter Amenageur</legend>
        <label for="siren"> N° siren : </label>
        <input type="text" name="siren" id="siren"> <br>
        <label for="nom"> Nom : </label>
        <input type="text" name="nom" id="nom"><br>
        <label for="contact"> Contact : </label>
        <input type="text" name="contact" id="contact">
      </fieldset>
      <input type="submit" value="Ajouter">
    </form>
```

L'attribut action indique l'emplacement de la page ajouterAmenageur.php où se trouve le code php pour utiliser les données récoltées par le formulaire et l'ajouter dans la base de données. On utilise la méthode get car le formulaire ne modifie pas l'état du site.

```
<?php
// récupération des données du formulaire
$siren=$_GET["siren"];
$nom=$_GET["nom"];
$contact=$_GET["contact"];
```

On commence par récupérer les données du formulaire et on les stocke dans des variables.

```
try {
  // Récupération des données
  $servername = "127.0.0.1";
  $dbname = "BUTRT1_lg409538";
  $username = "lg409538";
  $userpassword = "MDP_lg409538";

  $lienBDD = new PDO("mysql:host=$servername;dbname=$dbname", "$username", "$userpassword");
  $lienBDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Puis on se connecte à la base de données.

```
// Preparation de la requête  
$requeteSQL = $lienBDD->prepare("INSERT INTO amenageurs (siren, nom, contact) VALUES (:siren, :nom, :contact)");
```

Ensuite, on prépare la requête SQL d'ajout.

```
// Liaison des données avec la requête SQL  
$requeteSQL->bindParam(":siren",$siren);  
$requeteSQL->bindParam(":nom",$nom);  
$requeteSQL->bindParam(":contact",$contact);
```

Après, on lie les données avec la requête SQL.

```
// Exécution de la requête SQL  
$requeteSQL->execute();  
echo "<p class='panel panel-success'>Votre aménageur a bien été ajouté à la base de donnée !</p>";
```

Ensuite, on exécute la requête et on affiche un message sur la page pour indiquer à l'utilisateur que l'ajout a été réalisé avec succès.

```
catch (PDOException $e)  
{  
    // traitement d'erreurs  
    die ("Erreur : ".$e->getMessage());  
}
```

Enfin, on affiche un message d'erreur en cas d'échec.

4. Page suppression

Comme pour l'ajout, nous avons séparé le code dans deux pages. La page `supprimerAmenageur.html` contient le formulaire pour supprimer une entrée dans la base de données.

```
<div class="col-sm-8 text-left">
  <h1>Amenageur</h1>
  <section class="col-md-7">
    <form action="supprimerAmenageurs.php" method="get">
      <fieldset>
        <legend>Supprimer Amenageur</legend>
        <label for="siren"> N° siren : </label>
        <input type="text" name="siren" id="siren"> <br>
        <label for="nom"> Nom : </label>
        <input type="text" name="nom" id="nom"><br>
        <label for="contact"> Contact : </label>
        <input type="text" name="contact" id="contact">
      </fieldset>
      <input type="submit" value="Supprimer">
    </form>
```

L'attribut `action` indique l'emplacement de la page `supprimerAmenageur.php` où se trouve le code php pour utiliser les données récoltées par le formulaire et les supprimer de la base de données. On utilise la méthode `get` car le formulaire ne modifie pas l'état du site.

```
<?php
// récupération des données du formulaire
$siren=$_GET["siren"];
$nom=$_GET["nom"];
$contact=$_GET["contact"];
//echo "<p>siren : $siren</p>";
//echo "<p>nom : $nom</p>";
//echo "<p>contact : $contact</p>";

try {
    // Récupération des données
    $servername = "127.0.0.1";
    $dbname = "BUTRT1_lg409538";
    $username = "lg409538";
    $userpassword = "MDP_lg409538";

    $lienBDD = new PDO("mysql:host=$servername;dbname=$dbname", "$username", "$userpassword");
    $lienBDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

D'abord, on récupère les données du formulaire et on les stock dans des variables, puis on réalise l'accès à la base de données.

```
// Suppression de donne de la table ameneurs
$requeteSQL = $lienBDD->prepare("DELETE FROM ameneurs
                                WHERE siren=:siren
                                AND nom=:nom
                                AND contact=:contact");
```

Ensuite, on prépare la requête SQL de suppression.

```
// Liaison des données avec la requête SQL
$requeteSQL->bindParam(":siren",$siren);
$requeteSQL->bindParam(":nom",$nom);
$requeteSQL->bindParam(":contact",$contact);

// Exécution de la requête SQL
$requeteSQL->execute();
echo "<p class='panel panel-success'>Votre aménageur a bien été supprimé à la base de données !</p>";
```

Après, on lie les données à la requête SQL avant de l'exécuter puis d'afficher sur la page que la suppression a bien été effectuée.

```
catch (PDOException $e)
{
    // traitement d'erreurs
    die ("Erreur : ".$e->getMessage());
}
?>
```

On finit par afficher un message d'erreur en cas d'échec.

5. Page recherches

La page recherche.php permet à l'utilisateur de chercher une ville et d'afficher toutes ses bornes de recharges sous forme d'un tableau.

```
<form method="POST" action="">
  <div class="form-group">
    <label for="commune">Choisissez votre commune</label>
    <select id="commune" name="commune" class="form-control" onchange="this.form.submit()">
      <option value="">Sélectionner une commune</option>
    <?php
      // Connexion à la base de données
      $servername = "127.0.0.1";
      $dbname = "BUTRT1_lg409538";
      $username = "lg409538";
      $userpassword = "MDP_lg409538";
```

On crée un formulaire qui affichera un menu déroulant avec toutes les villes qui ont des bornes de recharge. On se connecte à la base de données.

```
try {
  $lienBDD = new PDO("mysql:host=$servername;dbname=$dbname", $username, $userpassword);
  $lienBDD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  // Obtenir la liste des communes
  $requeteSQL = $lienBDD->prepare("SELECT DISTINCT s.commune FROM stations s ORDER BY s.commune;");
  $requeteSQL->execute();
  $communes = $requeteSQL->fetchAll(PDO::FETCH_ASSOC);
```

On prépare la requête SQL de avant de l'exécuter.

```
// Initialiser la liste des opérateurs
if (isset($_POST['commune']) && $_POST['commune'] !== '') {
  $commune = $_POST['commune'];
  // Obtenir la liste des opérateurs pour la commune sélectionnée
  $requeteSQL = $lienBDD->prepare("SELECT DISTINCT o.nom FROM operateurs o
                                INNER JOIN stations s
                                ON o.id = s.operateur_id
                                WHERE s.commune = :commune
                                ORDER BY o.nom");

  $requeteSQL->bindParam(':commune', $commune, PDO::PARAM_STR);
  $requeteSQL->execute();
  $operateurs = $requeteSQL->fetchAll(PDO::FETCH_ASSOC);
```

Ensuite, on initialise la liste des opérateurs, avec une requête SQL on obtient la liste des opérateurs de la commune choisie par l'utilisateur, puis on lie les données à la requête.


```
<?php
foreach ($communes as $donnee):
?>
    <option value="<?= $donnee['commune'] ?>" <?= (isset($commune) && $commune === $donnee['commune']) ? 'selected' : '' ?>>
        <?= $donnee['commune'] ?>
    </option>
<?php endforeach; ?>
</select>
```

Après, on génère des options dans l'élément select du formulaire pour le menu déroulant.

```
<?php if (!empty($operateurs)): ?>
    <h2>Liste des opérateurs pour <?= $commune ?></h2>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Nom de l'opérateur</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($operateurs as $operateur): ?>
                <tr>
                    <td><?= $operateur['nom'] ?></td>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
<?php elseif (isset($commune)): ?>
    <p>Aucun opérateur trouvé pour la commune sélectionnée.</p>
<?php endif; ?>
```

Enfin, on affiche la liste des opérateurs sous forme de tableau à l'aide d'une boucle for.