

# Site location voitures

## 1 Le concept

Site du type Le Bon Coin pour les voitures.

Un propriétaire peut vendre ou mettre son véhicule en location

Un utilisateur la louer ou l'acheter

## 2 Les Interfaces disponibles

### Barre de navigation avec deux onglets

Mettre une annonce (propriétaire)

Voir toutes les annonces (utilisateur)

### Accueil avec toutes les annonces de voitures

Seules les dix dernières annonces seront visibles (les dix dernières)

- titre (marque)
- description
- code postal
- ville
- type (location ou vente)
- tarif selon le type

### Onglet Ajouter une annonce

J'accède a un formulaire, avec différents champs a renseigner (tous obligatoires)

- Titre
- description
- code postal
- ville
- tarif
- type
- bouton de validation du formulaire

## Onglet Voir toutes les annonces

Même affichage qu'en accueil, sauf que tous les véhicules seront affichés, et que sera précisé si le véhicule est encore disponible ou pas

Si un véhicule me plaît, je peux accéder à sa fiche individuelle avec un bouton cliquable

## Onglet Fiche individuelle

En accédant à cette fiche individuelle, je peux laisser mes coordonnées au propriétaire pour qu'il me recontacte

## 3 Création de la BDD

Je vais avoir besoin d'une base de données pour enregistrer les différents véhicules, afficher les infos en Front, ainsi que les coordonnées des utilisateurs/particuliers qui veulent louer ou acheter

Voici la table avec les différents champs en BDD (9 champs)

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id 🗝️	int(11)			Non	Aucun(e)		AUTO_INCREMENT	🔧 Modifier 🛑 Supprimer ▼ Plus
2	title	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
3	description	text	utf8mb4_general_ci		Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
4	code_postal	int(5)			Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
5	city	varchar(70)	utf8mb4_general_ci		Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
6	price	int(11)			Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
7	reservation_message	text	utf8mb4_general_ci		Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
8	type	enum('location', 'vente')	utf8mb4_general_ci		Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus
9	created_at	datetime			Non	Aucun(e)			🔧 Modifier 🛑 Supprimer ▼ Plus

## 4 Maquette

La maquette fournie est prête à l'emploi (header et footer dans le dossier inc)

## 5 Connexion à la BDD

Dans le dossier inc créer un fichier init.php

Voici le code pour se connecter à la bdd

Dans ce même fichier j'initialise sans leur donner une valeur, deux variables `$erreur` et `$content` qui me serviront dans mon projet

```
<?php

$pdo = new PDO('mysql:host=localhost;dbname=location_voiture', 'root', '', array(
    PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING,
    PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8'
));

$erreur = '';
$content = '';
```

J'inclurai ce fichier init.php dans mon header pour récupérer la connexion à la BDD ainsi que les variables initialisées partout dans mon projet

```
<?php require_once('inc/init.php') ?>

<!DOCTYPE html>
<html lang="fr">
<head>
```

## 6 Formulaire d'ajout

C'est la première étape de notre construction de site. Il nous faut alimenter la BDD qui va à son tour alimenter le site

Je vais dans mon fichier add.php pour mettre en place le processus de validation du formulaire

En premier, je vérifie que je reçois bien qlq chose dans mon formulaire avec if(\$\_POST)

```
if($_POST){
}
}
```

Je vais à présent vérifier dans mon traitement PHP que l'on ne m'envoie pas dans le formulaire autre chose que ce que je veux recevoir. Je vais procéder à cette vérification champs par champs

Celui pour le titre (la marque de la voiture)

```
if(!isset($_POST['title']) || iconv_strlen($_POST['title']) < 3 ||
iconv_strlen($_POST['title']) > 20){
    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format titre !
</div>';
}
```

Si la donnée n'a pas été envoyée, ou si la longueur est inférieure à 3 ou supérieure à 20, je ne valide pas ce champs

La description

```
if(!isset($_POST['description']) || iconv_strlen($_POST['description']) <
3 || iconv_strlen($_POST['description']) > 200){
    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format
description !</div>';
}
```

Même restriction, sauf que l'on peut poster jusqu'à 200 caractères

Le code postal

```
if(!isset($_POST['code_postal']) || !preg_match("#^[0-9]{5}$#",
$_POST['code_postal'])){
```

```

    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format
code_postal !</div>';
}

```

je fais la vérification avec un preg\_match qui me permet de contrôler les caractères que je tolère et leur nombre

La ville

```

if(!isset($_POST['city']) || iconv_strlen($_POST['city']) < 3 ||
iconv_strlen($_POST['city']) > 30){
    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format ville !
</div>';
}

```

Je ne veux pas en dessous de 3 et pas au-dessus de 30 caractères

Le prix

```

if(!isset($_POST['price']) || !preg_match("#^[0-9]{1,7}$#",
$_POST['price'])){
    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format prix !
</div>';
}

```

Pareil que pour le code postal, j'utilise un preg\_match

Le type

```

if(!isset($_POST['type']) || $_POST['type'] != "location" && $_POST['type'] !=
"vente"){
    $erreur .= '<div class="alert alert-danger" role="alert">Erreur format type !
</div>';
}

```

Je vérifie cette fois que je reçois bien ou location ou vente avec \$\_POST, sinon, message d'erreur

Si tout s'est bien passé, si tous les champs ont été correctement remplis

Cela veut dire que je n'ai reçu aucun message d'erreur et cela veut surtout dire que la variable \$erreur n'a reçu aucun contenu.

Je procède donc à l'envoi du formulaire. Je code, puis les explications

```

if(empty($erreur)){
    // ajout
    $ajoutVehicule = $pdo->prepare("INSERT INTO vehicule (title, description,
code_postal, city, price, type, created_at) VALUES
(:title, :description, :code_postal, :city, :price, :type, NOW()) ");
    $ajoutVehicule->bindValue(':title', $_POST['title'], PDO::PARAM_STR);
    $ajoutVehicule->bindValue(':description', $_POST['description'], PDO::PARAM_STR);
    $ajoutVehicule->bindValue(':code_postal', $_POST['code_postal'], PDO::PARAM_INT);
}

```

```

    $ajoutVehicule->bindValue(':city', $_POST['city'], PDO::PARAM_STR);
    $ajoutVehicule->bindValue(':price', $_POST['price'], PDO::PARAM_INT);
    $ajoutVehicule->bindValue(':type', $_POST['type'], PDO::PARAM_STR);
    $ajoutVehicule->execute();

    $content .= '<div class="alert alert-success alert-dismissible fade show mt-5"
role="alert">
        <strong>Félicitations !</strong> Ajout du véhicule réussie !
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>';
    // ajout
}

```

La méthode **prepare** permet une meilleure sécurisation lors de l'envoi des données en bdd. Elle est plus longue qu'avec la méthode **query**, mais celle ci est à réserver à la requête SELECT

Voir chapitre [requêtes préparées](#)

Une fois terminé mes bindValue ( en vérifiant si c'est un STR ou un INT que je veux envoyer en BDD) j'exécute ma requête (obligatoire avec **prepare**)

## 7 Affichage des données en Front

Je vais à présent récupérer mes insertions en BDD vers l'accueil et la page qui affiche tous mes véhicules

D'abord sur l'accueil dans index.php

En traitement php dans le haut de mon fichier je vais faire une requête pour sélectionner toutes les données concernant les véhicules

```

$afficheVéhicules = $pdo->query("SELECT * FROM vehicule ORDER BY created_at DESC
LIMIT 0, 10");

```

Et je vais ensuite me servir de cette requête pour récupérer chaque valeur individuellement dans les emplacements de mon tableau

Je vais faire une boucle while autour des balises **<tbody>**

```

<tbody>
    <?php while ($vehicule = $afficheVéhicules->fetch(PDO::FETCH_ASSOC)) : ?>

```

Et je la referme immédiatement juste avant la balise fermante de **tbody**

```

    <?php endwhile; ?>
</tbody>

```

Voici le reste de la syntaxe à l'intérieur de `tbody`

```
<tr>
  <td>
    <strong><?= strtoupper($vehicule['title']) ?></strong>
    <p>
      <small>
        <?= $vehicule['description'] ?>
      </small>
    </p>
  </td>
  <td>
    <?= $vehicule['code_postal'] ?> <?= $vehicule['city'] ?>
  </td>
  <td>
    <span class="badge bg-primary"><?= $vehicule['type'] ?></span>
    <span class="badge bg-secondary"><?= $vehicule['price'], "€" ?></span>
  </td>
</tr>
```

Je fais la même chose dans `list.php`, avec qlq petites différences

Le même traitement PHP en haut de fichier, sauf que je ne limite pas à 10. Je veux toutes les afficher

```
$afficheVéhicules = $pdo->query("SELECT * FROM vehicule ORDER BY created_at DESC");
```

Je récupère la même `while` et `endwhile` que dans l'index et je les positionne dans `<tbody>`

Le premier `<td>` est aussi similaire

```
<td>
  <strong><?= strtoupper($vehicule['title']) ?></strong>
  <p>
    <small>
      <?= $vehicule['description'] ?>
    </small>
  </p>
</td>
```

Dans le second `<td>` en revanche je vais avoir une différence

```
<td>
  <?= $vehicule['code_postal'] ?> <?= $vehicule['city'] ?>
  <?php if(!empty($vehicule['reservation_message'])): ?>
  <span class="badge bg-success">Ce bien a déjà été réservé !</span>
  <?php endif ?>
</td>
```

La première ligne sera similaire mais juste après je vais coder une condition dans le cas où l'indice `reservation_message` aurait reçu une valeur en BDD

S'il est vide, en revanche le message ne s'affichera pas

Le <td> suivant ne change pas

```
<td>
  <span class="badge bg-danger"><?= $vehicule['type'] ?></span>
  <span class="badge bg-warning"><?= $vehicule['price'], "€" ?></span>
</td>
```

Il me reste un dernier traitement à faire sur cette page pour le <td> qui suit

```
<td>
  <a href="show.php?id=<?= $vehicule['id'] ?>" class="btn btn-danger">Voir
l'annonce</a>
</td>
```

Je vais faire transiter avec la méthode GET l'id du véhicule qui m'intéresse dans l'attribut href