

## TP 3 : Création de la vue en liste des recettes

- Créer les recettes suivantes :
  - Titre : Treipaïs  
Description : Le « Treipaïs » est une spécialité pâtissière du Limousin créé par un groupement de pâtisseries locaux. Il est composé de trois textures et de trois parfums : un fond de feuilletine, une mousse aux châtaignes, une mousse au chocolat noir, un biscuit à la noisette et un glaçage au chocolat, surmonté d'un « marron » glacé et de deux feuilles vertes en pâte d'amandes.  
Auteur : [contact@lacoza.fr](mailto:contact@lacoza.fr)
  - Titre : Confit de canard  
Description : La viande est cuite pendant au moins 2 h dans la graisse chaude, entre 70 et 85 °C, puis mise en bocaux et recouverte de graisse, de telle sorte que l'air ne puisse pas entrer en contact avec elle et la détériorer.  
Auteur : [contact@lacoza.fr](mailto:contact@lacoza.fr)
  - Titre : Soupe au pistou  
Description : La soupe au pistou (sopa au pístu, soupo au pístou) est une soupe aux légumes d'été, avec des pâtes, servie avec du pistou, un mélange d'ail, d'huile d'olive et de basilic haché. Le terme pistou désigne, en provençal, le pilon du mortier qui sert à faire la préparation, et non pas le basilic qui se dit baseli.  
Auteur : [contact@lacoza.fr](mailto:contact@lacoza.fr)
  - Titre : Tapenade  
Description : La tapenade (tapenada, tapenado) est une recette de cuisine provençale, mise au point en 1880 par un chef-cuisinier de Marseille. Elle est principalement constituée d'olives broyées, d'anchois, de thon mariné, d'herbes et de câpres (tapena en occitan, d'où son nom).  
Auteur : [contact@lacoza.fr](mailto:contact@lacoza.fr)
- Vérifier la présence des données dans la base de données
- Créer le menu **Recettes** entre Accueil et Contact
- Créer le contrôleur **listerController.php** qui permet de lister toutes les recettes

```
<?php

// préparation de la requête d'insertion dans la base de données

/** @var PDO $pdo */
$requete = $pdo->prepare("SELECT * FROM recettes");

// exécution de la requête et récupération des données
$requete->execute();
$recipes = $requete->fetchAll(PDO::FETCH_ASSOC);

require_once(__DIR__.DIRECTORY_SEPARATOR.'..'.DIRECTORY_SEPARATOR.'Views'.
DIRECTORY_SEPARATOR.'liste.php');
```

- Créer la vue **liste.php** qui affiche les liste des recettes

```
<body>

<h1>Recettes</h1>

<div>
    <?php foreach ($recipes as $recipe) : ?>
        <div>
            <h2><?php echo $recipe['titre']; ?></h2>
            <p><?php echo $recipe['description']; ?></p>
            <a href="mailto:<?php echo $recipe['auteur']; ?>"><?php echo
$recipe['auteur']; ?></a>
        </div>
    <?php endforeach; ?>
</div>

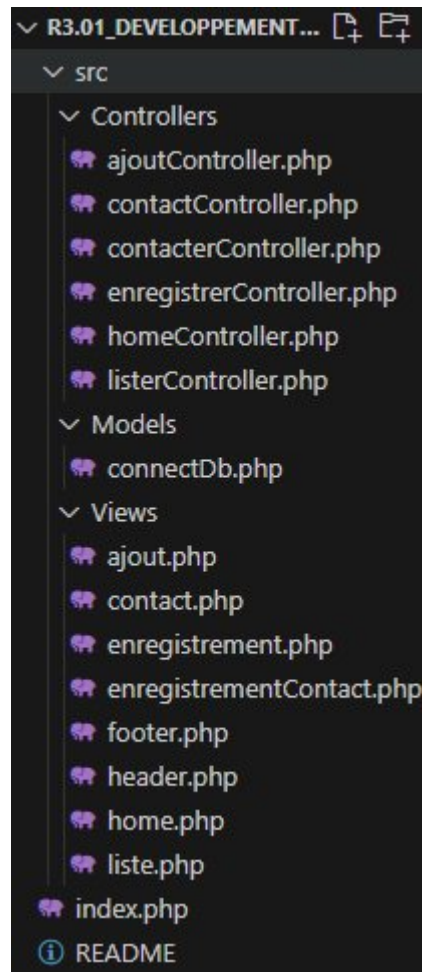
<a href="?c=home" class="btn btn-primary">Retour à l'accueil</a>

</body>
```

- On va améliorer le visuel de la liste des recettes grâce à des « cards » Bootstrap : <https://getbootstrap.com/docs/5.3/components/card/>

```
<div class="row">
    <!-- Boucle permettant de lister les recettes -->
    <?php foreach ($recipes as $recipe) : ?>
        <div class="col-4 p-2">
            <!-- Utilisation des Cards Bootstrap -->
            <div class="card">
                <div class="card-body">
                    <h2 class="card-title"><?php echo
$recipe['titre']; ?></h2>
                    <p class="card-text"><?php echo
$recipe['description']; ?></p>
                    Auteur : <a href="mailto:<?php echo
$recipe['auteur']; ?>"><?php echo $recipe['auteur']; ?></a>
                </div>
            </div>
        </div>
    <?php endforeach; ?>
</div>
```

- Le projet s'agrandit et le nombre de fichiers aussi :



*Liste des fichiers du projet*

- Pour avoir une meilleure visibilité des éléments du projet, nous allons regrouper certains éléments :
  - Les vues :
    - Nous allons organiser les vues par « entité » ou type d'éléments, dans notre cas, les vues associées aux recettes, aux contacts notamment
    - Une fois les dossiers créés, déplacer les fichiers et modifier les fichiers impactés
  - Les contrôleurs :
    - Nous allons organiser les contrôleurs en les transformant en classes, ainsi tout ce qui concerne les recettes sera dans une classe `RecetteController.php`, et de même pour `ContactController.php`
    - Créer le fichier **RecetteController.php** avec 3 fonctions : ajouter, enregistrer et lister

```

<?php

class RecetteController {

    // Fonction permettant d'ajouter une nouvelle recette
    function ajouter() {
    }

    // Fonction permettant d'enregistrer une nouvelle recette
    function enregistrer() {
    }

    // Fonction permettant de lister les recettes
    function lister() {
    }

}

```

- Nous allons maintenant déplacer le code des fichiers correspondants dans cette classe

```

<?php

class RecetteController {

    // Fonction permettant d'ajouter une nouvelle recette
    function ajouter() {
        require_once __DIR__ . DIRECTORY_SEPARATOR . '..' .
        DIRECTORY_SEPARATOR . 'Views' . DIRECTORY_SEPARATOR . 'Recette' .
        DIRECTORY_SEPARATOR . 'ajout.php';
    }
}

```

- Faire de même avec les autres fonctions (attention : il y a un piège)
- Supprimer les fichiers **ajoutController.php**, **enregistrerController.php**, et **listerController.php**
- Ensuite, modifier le fichier **index.php** pour importer cette classe, nous allons le faire au début

```
<?php

// import de la classe RecetteController
require_once(__DIR__.DIRECTORY_SEPARATOR.'src'.DIRECTORY_SEPARATOR.'Contro
llers'.DIRECTORY_SEPARATOR.'RecetteController.php');

// connexion à la base de données
require_once(__DIR__.DIRECTORY_SEPARATOR.'src'.DIRECTORY_SEPARATOR.'Models
'.DIRECTORY_SEPARATOR.'connectDb.php');
```

- Puis modifier la structure de contrôle **switch** au niveau du cas **ajout** pour commencer :

```
case 'ajout':
    $recetteController = new RecetteController();
    $recetteController->ajouter();
    break;
```

- Faire de même pour les cas **enregistrer** et **lister**
- Créer le fichier **ContactController.php** et effectuer les mêmes opérations