

## Itération 5 recherche par type de cuisine

**User story** : en tant qu'utilisateur (connecté ou pas), je dois pouvoir rechercher un restaurant par type de cuisine, afin de mieux cibler ma recherche.

**Durée maximum** : 1 séance.

**Date d'exigibilité** : 24/10/2022 à 14h

Les utilisateurs doivent pouvoir rechercher les restaurants par type de cuisine dans la recherche simple ou par multicritères



Cette nouvelle version de l'application devra être déployée sur le serveur web de production du BTS SIO Joliverie.

la recherche par type est bien affichée sur la page d'accueil du site



La recherche par type est réalisée grâce à des checkbox ce qui permet de choisir plusieurs type (le résultat applique un "et/ou")

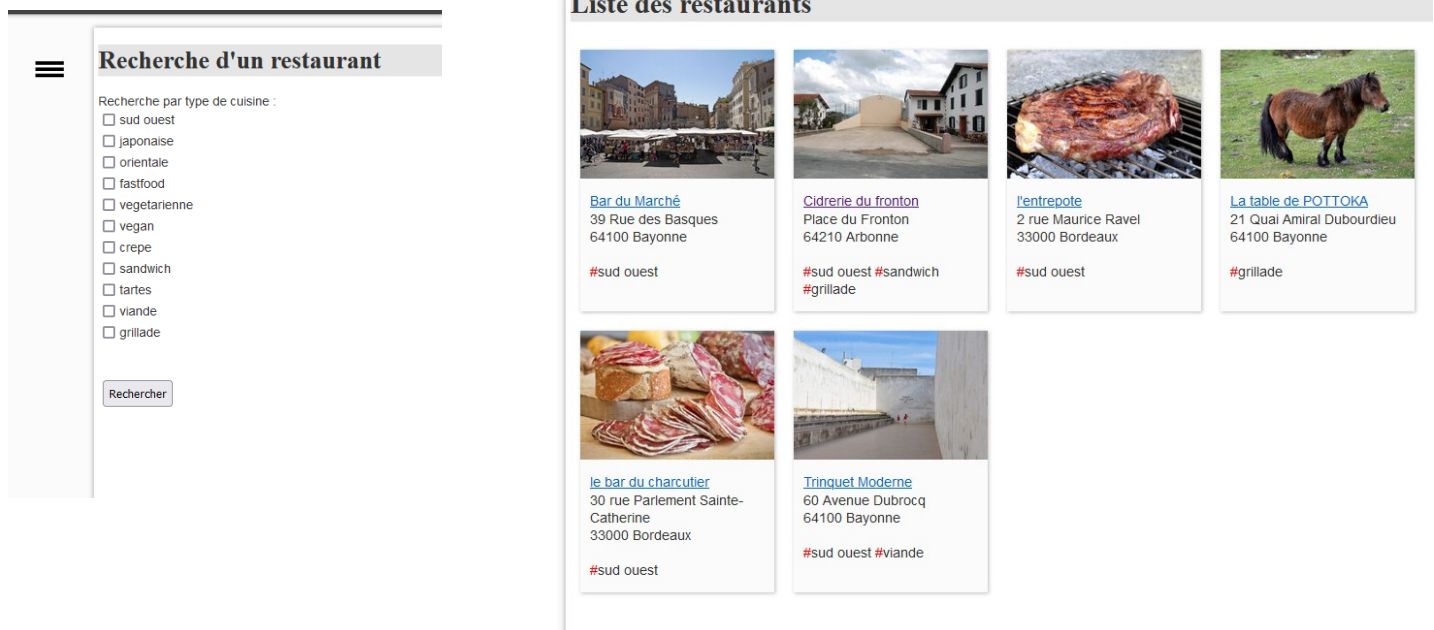
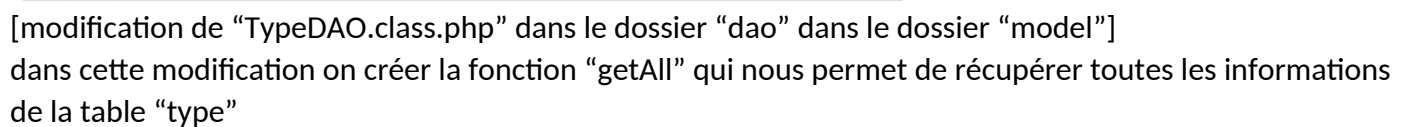
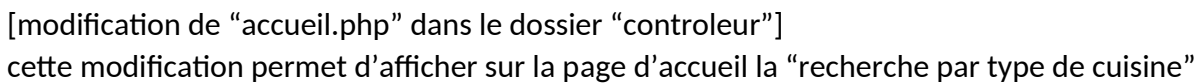


Figure 1: ici on a sélectionné "sud ouest" et "grillade"

sur la capturé d'écran ci-dessus on peut voir que la recherche nous rend les restaurants :

- avec "grillade"
- avec "sud ouest"
- avec "grillade" ET "sud ouest"



```

modele/dao/RestoDAO.class.php
+20 -0 View file @9f330749

... @@ -252,6 +252,26 @@ class RestoDAO {
252     }
253     return $lesObjets;
254 }

255 +
256 + public static function getRestoByType(string $types): array {
257 +     $lesObjets = array();
258 +     try {
259 +         $requete = "SELECT * FROM resto INNER JOIN typeparresto ON resto.idR = typeparresto.idR INNER
JOIN type ON typeparresto.idTC = type.idTC WHERE type.libelleTC IN ('".$types."' ) GROUP BY nomR;";
260 +         $stmt = Bdd::getConnexion()->prepare($requete);
261 +         $ok = $stmt->execute();
262 +         // attention, $ok = true pour un select ne retournant aucune ligne
263 +         if ($ok) {
264 +             // Pour chaque enregistrement
265 +             while ($enreg = $stmt->fetch(PDO::FETCH_ASSOC)) {
266 +                 //Instancier un nouveau restaurant et l'ajouter à la liste
267 +                 $lesObjets[] = self::enregistrementVersObjet($enreg);
268 +             }
269 +         }
270 +     } catch (PDOException $e) {
271 +         throw new Exception("Erreur dans la méthode " . $e->getMessage());
272 +     }
273 +     return $lesObjets;
274 + }

255 275
256 276
257 277
...

```

[modification de “RestoDAO.class.php” dans le dossier “dao” dans le dossier “model”]

cette modification nous permet de créer la fonction “getRestoByType”, cette fonction nous retourne toute les information des restaurants qui ont le bon paramètre (ici le type de cuisine) après qu’on l’ai renseigné dans la recherche

```

vue/vueRechercheResto.php
+9 -1 View file @9f330749

... @@ -14,6 +14,8 @@
14  /** @var string $cpR nom du restaurant recherché */
15  /** @var string $voieAdrR nom du restaurant recherché */
16

17 + use modele\dao\TypeDAO;
18 +

17 19 ?>
18 20
19 21 <h1>Recherche d'un restaurant</h1>
... @@ -38,7 +40,13 @@
38 40 break;
39 41 case "type":
40 42 ?>
41 - <br /><?php
43 + Recherche par type de cuisine : <br />
44 + <?php
45 + foreach (TypeDAO::getAll() as $unType) {
46 +     echo "<input type='checkbox' name='".$str_replace(' ', '_', strval($unType))."'
value='1'>".$unType."<br />";
47 + }
48 + ?>
49 + <?php

42 50 break;
43 51 case "multi":
44 52 ?>

...

```

[modification de “vueRechercheResto.php” dans le dossier “vue”]

cette modification permet de, pour chaque type dans la table « Type » lui associer une checkbox dans la recherche, ici on a ajouter un str qui remplace les espaces par des « \_ » dans le programme

[modification de "rechercheResto.php" dans le dossier "controleur"]

ici, la premiere modification permet d'afficher la recherche par type via si on a cliquer sur un autre type de recherche

la deuxieme modification permet au programme de lire les differents types et d'ensuite, les ajouter a la liste des checkbox

la troisieme modification sert a transformer la liste en « str »

5 of 5