# TP 6 Javascript

***Vous devrez déposer l’ensemble des fichiers par SFTP dans le dossier G?/www/TP5. Date limite vendredi 16/10/2020 à 8h00.***

***Penser à renommer aussi script.js***

*Il arrive que quand vous modifiez un fichier javascript ou css le navigateur utilise la version qu’il a en cache et ne regarde pas si vous avez fait des modifications. Il faut alors effacer la cache dans Paramètre>effacer les données de navigation>Images et fichiers caches.*

***Mettre tous les fichiers du TP précédent dans le dossier TP6***

1 Premiers pas en javascript : Sur la page listPatient.php, nous souhaitons que lorsque l’on clique sur le bouton Ajouter un patient, le formulaire d’ajout apparaisse en dessous (avec une animation de déroulement) au lieu de l’afficher dans une nouvelle page.

1. Copier le code HTML du formulaire en dessous du bouton. Enlever tout PHP de ce formulaire pour avoir tous les champs vide. Les textes doivent correspondre au texte du formulaire d’ajout (et non du formulaire d’édition). Idem pour le action de la balise form
2. Vérifier qu’il apparait bien et que vous pouvez toujours ajouter un patient dans la base
3. Ajouter à la balise form l’id FormAddPatient
4. Dans le fichier CSS ajouter :

**#FormAddPatient** **{**

**display:none;**

**}**

Pour le faire disparaitre quand on charge la page

1. Remplacer le code du bouton ajouter un patient (en dessous du tableau et non celui en dessous du formulaire) par

<button class=**"btn btn-primary"** id=**"BtnAddPatient"**>**Ajouter un patient**</button>

1. Dans le head de listpatient ajouter.

<script src=**"js/jquery-3.5.1.min.js"**></script>

<script src=**"js/script.js"**></script>

Télécharger jQuery pour le mettre dans le sous dossier js et créer le fichier script.js dans ce même dossier.

1. Dans ce fichier js, écrire

$**().**ready**(function()** **{**

$**(**'#BtnAddPatient'**).**click**(function()** **{**

$**(**'#FormAddPatient'**).**slideDown**();**

**});**

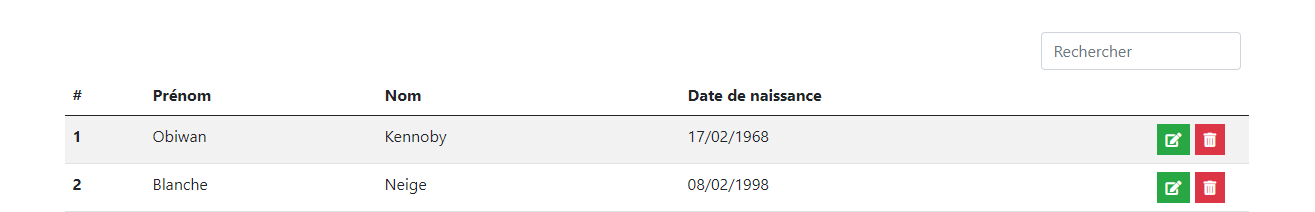
**});**

1. Tester que si on appuie sur le bouton, le formulaire apparait en se déroulant.

2 Champs rechercher :

Toujours depuis le fichier listpatients.php, Ajouter en haut de la page un champ texte « rechercher » (mis en page avec bootstrap). Mettre ce champ rechercher dans un div de largeur 200px aligné à droite (en mettant margin-left à auto en CSS).

Résultat attendu :



L’objectif va être maintenant qu’à chaque chiffre/lettre taper dans rechercher, seuls les patients qui ont un prénom un nom ou un id contenant ces chiffres/lettres apparaissent. Il faudra aussi afficher sous le champ rechercher le nombre de résultat correspondant à la recherche.

1. Dans le fichier script.js Ajouter un écouteur d’évènement sur le « keyup » du champ rechercher.
2. Quand cet événement keyup se produit, afficher le texte saisi dans un alert pour vérifier que ça marche. L’enlever après
3. Ajouter une commande en jQuery pour régler le CSS de tous les tr à l’intérieur de la balise tbody en display none
4. Ajouter une commande en jQuery pour régler le CSS de tous les tr à l’intérieur de la balise tbody contenant le texte rechercher en display : table-row (utiliser le selecteur contains <https://api.jquery.com/contains-selector/> )
5. Ajouter un div en html sous le champ rechercher qui contiendra le texte « (n correspondances) » où n correspond au nombre de résultat correspondant à la recherche (utiliser $(…..).length pour compter les éléments). Faire un test si n >=2 mettre un s à « correspondance(s) »
6. Sauvegarder le fichier .php en listpatientssansajax.php pour avoir une sauvegarde.

3 Supposons maintenant que le nombre de patient est très grand. Il n’est pas possible de charger la liste complète de tous les patients à chaque ouverture de page. Nous devons donc faire des pages numérotées. De plus lorsqu’on tape une lettre on redemande au serveur de rechercher dans la base pour nous renvoyer les patients correspondants.

* Créer un fichier rowspatient.php qui renvoi la liste des patients sans l’entête du HTML mais avec les <tr><td>. *(Même code que listpatients.php sans tout le HTML avant et après el foreach.)*

*Exemple de résultat attendu :*

<tr onclick=**"location.href='listedossiers.php?id=1';"**>

<th scope=**"row"**>**1**</th>

<td>**Obiwan**</td>

<td>**Kennoby**</td>

<td>**17/02/1968**</td>

<td>

<a href=**"formpatient.php?id=1"** class=**"btn btn-success btn-sm rounded-0"** type=**"button"** data-toggle=**"tooltip"** data-placement=**"top"** title=**"Edit"**><i class=**"fa fa-edit"**></i></a>

<a href=**"deletepatient.php?id=1"** class=**"btn btn-danger btn-sm rounded-0"** type=**"button"** data-toggle=**"tooltip"** data-placement=**"top"** title=**"Delete"**><i class=**"fa fa-trash"**></i></a>

</td>

</tr>

<tr onclick=**"location.href='listedossiers.php?id=2';"**>

<th scope=**"row"**>**2**</th>

<td>**Blanche**</td>

<td>**Neige**</td>

<td>**08/02/1998**</td>

<td>

<a href=**"formpatient.php?id=2"** class=**"btn btn-success btn-sm rounded-0"** type=**"button"** data-toggle=**"tooltip"** data-placement=**"top"** title=**"Edit"**><i class=**"fa fa-edit"**></i></a>

<a href=**"deletepatient.php?id=2"** class=**"btn btn-danger btn-sm rounded-0"** type=**"button"** data-toggle=**"tooltip"** data-placement=**"top"** title=**"Delete"**><i class=**"fa fa-trash"**></i></a>

</td>

</tr>

* Modifier la requête en « SELECT \* FROM patients LIMIT :n , 3 » pour n’obtenir que 3 patients en commençant par le $n+1 ème. (S’assurer qu’il y a au moins 7 patients dans la table. Ici nous mettons 3 pour éviter d’avoir à créer 50 patients mais en production nous mettrions entre 10 et 500)

Nous aurons :n qui vaudra ($\_GET['page']-1)\*3 ; (le numéro de la page doit être passé en argument du fichier par la méthode GET)

* Modifier le fichier qui affiche la liste des patients :
  + Plutôt que de faire la requête dans ce fichier et de faire le foreach pour afficher les lignes, écrire

$\_GET['page']=1 ;

include 'rowspatient.php';

* + Ajouter en dessous du tableau :

<div class=**"text-center"**>

<button type=**"button"** class=**"btn btn-light"** id=**"previous"**>**Précédente**</button>

<b>*&nbsp;* **Page** <span id=**"numPage"**>**1**</span>*&nbsp;*</b>

<button type=**"button"** class=**"btn btn-light"** id=**"next"**>**Suivante**</button>

</div>

* Dans le fichier script.js
  + Créer une fonction updatetable :

**function** updatetable**()** **{**

$**.**get**(**'listpatient.php",**{** page**:** $**(**'#numpage'**).**html**()** **}, function(** msg **)** **{**

$**('**tbody'**).**html**(**msg**);**

**});**

**}**

* + Ajouter les écouteurs d’événement pour que lorsque l’on appuie sur suivant on on modifie #numPage et on lance la fonction updatetable. De même pour le bouton précédent. *(Il faudra utiliser la fonction parseInt)*
* Dans rowspatient.php modifier la requête en

SELECT \* FROM patient WHERE LOWER(nom) LIKE '%:recherche%' OR LOWER( prenom ) LIKE '%:recherche%' OR CAST(id as CHAR) LIKE '%:recherche%' LIMIT :n , 3

(On regarde si l’un des champ contient :recherche ; on met tout en minuscule pour être insensible à la casse. On convertit id en chaine de caractere.)

Nous aurons

:recherche qui vaudra isset($\_GET['recherche'])?strtolower($\_GET['recherche']):"";

Si $\_GET['recherche'] existe on le met à ($\_GET['recherche']) en minuscule sinon on le met à ""

* Modifier script.js pour ajouter ce paramètre $\_GET ['recherche'] . A chaque keyup lancer updatetable et remettre le numéro de page à 1.