

Rapport

## TP2 DE PROGRAMMATION WEB

le 5 mars 2019,

Lauren SILVA ROLAN SAMPAIO,  
Maj. Image, Son & Machine Learning  
[lauren.rolan@ecole.ensicaen.fr](mailto:lauren.rolan@ecole.ensicaen.fr)

Tuteur école : Myriam  
MOKHTARI-BRUN





# TABLE DES MATIÈRES

---

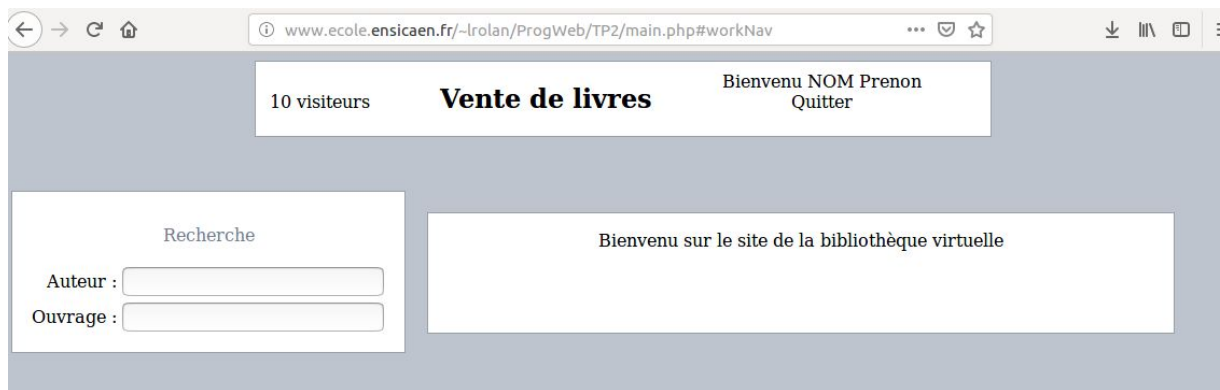
La page principale	4
Recherche des auteurs	4
<b>RECHERCHE D'OUVRAGES PAR TITRE</b>	<b>5</b>
<b>RECHERCHE D'OUVRAGES PAR AUTEUR</b>	<b>6</b>
<b>AMÉLIORATIONS</b>	<b>6</b>

# ANALYSE DU CODE

*Vérification des fonctionnalités.*

## 1. La page principale

La page principale, donnée par l'archive *main.php* peut être visualisée dans la Figure 1.



*Figure 1 Page principale*

Le *nav* de recherche a été modifié de façon à ne pas être une liste sans ordre, mais un tableau; cela a permis d'aligner les labels et les inputs. De plus, l'attribut *display* a été mis à *inline-block* de façon à encapsuler le contenu.

Le script *counter.php* donne une heure pour le cookie *visite*, et réécrit le document texte *counter.txt* avec la nouvelle valeur à chaque fois elle est mis à jour.

## 2. Recherche des auteurs

La recherche et affichage des auteurs est réalisée en trois étapes : (i) la fonction AJAX est appelée par l'événement *keyup* dans l'input du auteur (marqué dans le code avec l'ID *authorInput*, (ii) le programme *recherche\_auteur.php* est appelé, et son retour (en format JSON) est utilisé dans (iii) la fonction Javascript *affiche\_auteurs()*. De cette façon, on obtient les résultats comme dans la Figure 2.



Figure 2 Résultat de la recherche par nom de l'auteur

La partie base de données dans le côté serveur est restreinte à la ligne SQL envoyé au serveur postgres :

```
SELECT * FROM auteurs WHERE nom LIKE '%$match%' OR prenom LIKE '%$match%'
```

Après l'exécution de cette ligne, on fait un *fetchAll* avec la classe Auteur (définie dans le programme PHP) comme paramètre. Le retour est le tableau d'objets du type auteur encodé en format JSON.

### 3. Recherche d'ouvrages par titre

La recherche et affichage des ouvrages est, comme antérieurement dit, réalisée en trois étapes : (i) la fonction AJAX est appelée par l'événement *keyup* dans l'input de l'ouvrage (marqué dans le code avec l'ID *workInput*, (ii) le programme *recherche\_ouvrages\_titre.php* est appelé, et son retour (en format JSON) est utilisé dans (iii) la fonction Javascript *affiche\_ouvrages()*. De cette façon, on obtient les résultats comme dans la Figure 3.

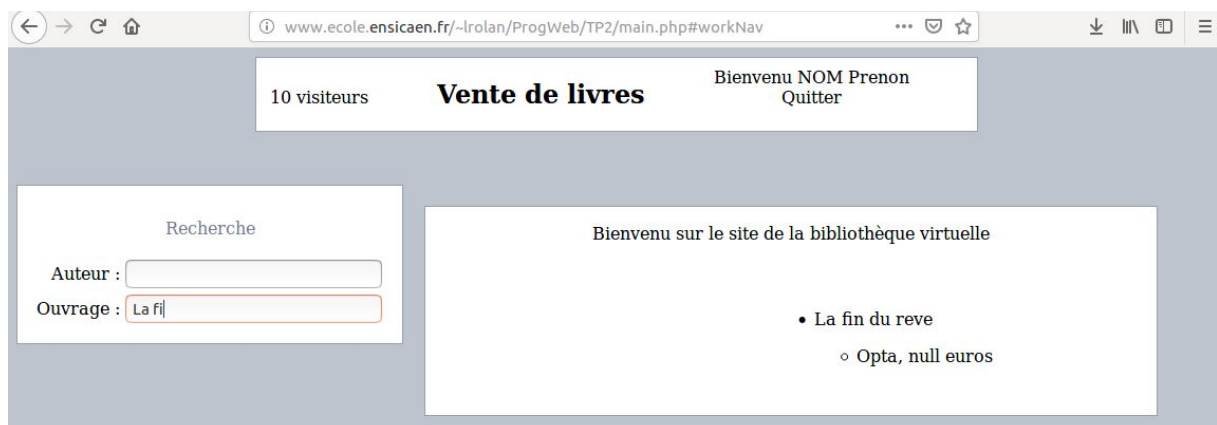


Figure 3 Résultat de la recherche par nom de l'ouvrage

La partie base de données dans l'archive PHP, cette fois, est plus complexe, parce que il faut accéder à plusieurs tables. On cherche dans la table *ouvrage* par le nom du livre; en trouvant un tableau de correspondances, une deuxième fonction est appelée. Cette fonction cherche donc les exemplaires correspondantes à chaque livre dans la table *exemplaire*, et après cela, cherche par le nom de l'éditeur dans la table *editeurs*. Une fois ces deux tables sont examinées, un objet *exemplaire* est créé et ajouté dans un tableau à être renvoyé.

Finalement, un objet *book* est formé dans la première fonction, transformé en un objet JSON et envoyé vers la côté client.

## 4. Recherche d'ouvrages par auteur

Cette fonction unit les fonctionnalités précédentes. On ajoute un link sur le nom de l'auteur, et l'action *onclick* appelle la fonction *AJAX recherche\_ouvrages\_auteur()* qui, par sa fois, appelle le programme PHP *recherche\_ouvrage\_auteur*. Ce programme accède aux tables *ecrit\_par*, qui retourne le code des ouvrages écrits par cet auteur.

Une fois qu'on a ces ouvrages, il suffit de refaire la recherche comme dans le programme *recherche\_ouvrages\_titre*, avec une fonction spécifique pour les exemplaires.

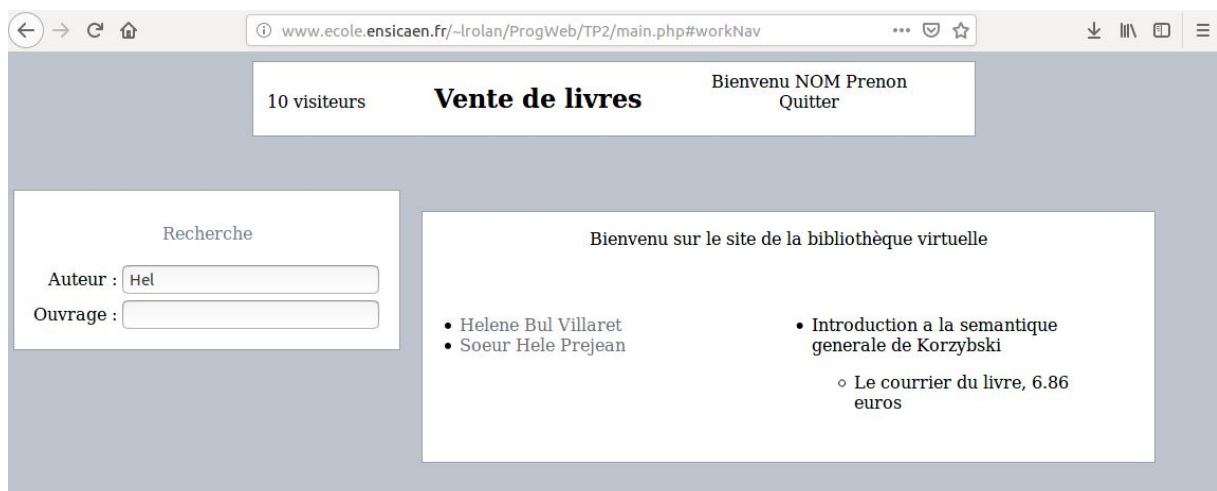


Figure 4 Résultat de la recherche d'une ouvrage par nom de l'auteur

## 5. Améliorations

On remarque une grande quantité de code redondant, qui peut être simplifié. Le PDO peut être une entité isolée, aussi comme la fonction *searchExemplaires* dans le PHP.

De plus, le *design* de la page pouvait être améliorée par l'utilisation du *framework* Bootstrap.