Développement Web Back End 2A INFO

TP1 : Mise en place du site « Vente de livres »

NB : Utiliser le document sur foad Prérequis_TP1.pdf

1. Introduction

Le but du TP est de réaliser les bases d'un site marchand de vente de livres. On utilisera pour cela la base de données livres (Fig. 2). La première étape (bien modeste) va consister à créer un compteur du nombre de visites et à créer des requêtes SQL permettant de consulter la BDD de livres.

2. Création du compteur (fichier)

On va implémenter le compteur à l'aide d'un fichier texte qui contient le nombre de visites. À chaque chargement de la page, un code PHP charge ce fichier, incrémente le compteur et sauve le résultat dans ce même fichier. Créer une page PHP qui implémente ce mécanisme et affiche le nombre total de visites de la page.

3. Utilisation des compteurs (cookie)

Le problème de ce type de méthode est qu'une personne rechargeant la même page 10 fois va incrémenter artificiellement le compteur. Pour remédier à cela, implémenter un mécanisme qui positionne un cookie d'une durée de vie d'une heure chez l'utilisateur. Le compteur ne doit être incrémenté que si le cookie n'est pas positionné (auquel cas on le positionne). Implémenter ce mécanisme sous forme d'une fonction qui incrémente (éventuellement) la valeur du compteur dans le fichier et renvoie celui-ci.

4. Consultation du catalogue

1. Créer une page d'accueil (Fig. 1) qui comporte une bannière supérieure, un menu à gauche et une partie principale à droite. Pour cela, utiliser <header>, <section>, <nav>, , ... ainsi qu' une feuille de styles. Détails :

   (a) **La bannière supérieure :** codée par <*header*> contenant 3 <*section*> :
   La 1ère <*section*> contient le nombre de visites, la <*section*> du milieu contient le titre du site et la <*section*> de droite contient 2 lignes pour « *Bienvenue Nom Prénom* » et « *Quitter* » (pour l'instant, on affichera uniquement ces textes). Donner un attribut *id* pour chaque section.

   (b) **Le menu :** codé par <*nav*> contenant une liste de balises <*input*> *:*
   Chaque <*input*> sera associée à un événement *keyup* permettant d'exécuter une fonction JavaScript (AJAX) lorsqu'une touche du clavier est relâchée. Dans le TP2, la fonction recherchera des informations dans la BDD *livres* selon la saisie. Les informations s'afficheront dans la partie principale à droite.

   (c) **La partie principale :** codée par <*section*> composée d'un paragraphe <*p*> pour « *Bienvenu sur le site de la bibliothèque virtuelle*. » et de 2 <*div*> avec attribut *id* pour chacune. Dans le TP2, ces 2 <*div*> permettront d'afficher simultanément 2 listes de données (auteurs et ouvrages), résultats des recherches faites par l'internaute.
   Dans ce TP1, faire uniquement des petits tests pour afficher dans les div ce que vous aurez saisi dans le menu.

Fig. 1 – Un exemple de présentation du site en cours de réalisation en TP2

2. Implémenter en SQL les recherches suivantes et les tester sous la BDD *livres* (postgresql)
   a) recherche d'auteurs connaissant une partie du nom,
   b) recherche d'ouvrages connaissant une partie du titre,
   c) recherche d'ouvrages connaissant le code d'un auteur,
   d) recherche d'exemplaires connaissant le code d'un ouvrage (Nom de l'éditeur et Prix).

3. Une fois familiarisé avec la BDD *livres*, intégrer la 1ère requête SQL dans un programme *requete.php* à tester indépendamment du site web. Utiliser *PDO* (*PHP Data Objects)* pour accéder à la BDD depuis PHP, exécuter la requête (*query*(), *fetch*(), ...) et afficher le résultat sous forme d'un tableau HTML. Un fois familiarisé avec *PDO*, passer au TP2.

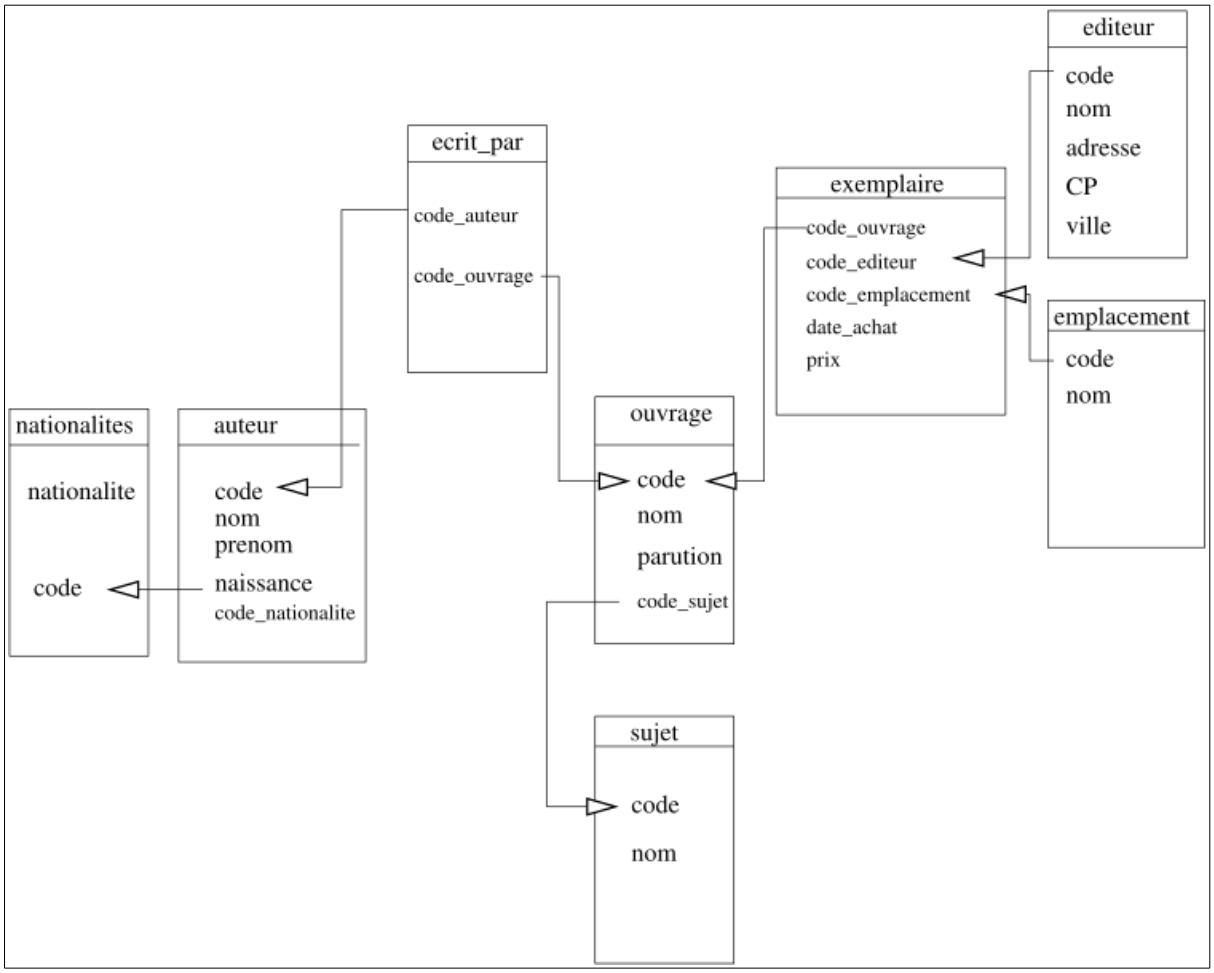

Fig. 2 – MCD de la base de données *livres*. Les clés étrangères et primaires sont représentées par les flèches qui vont de la clé étrangère à la clé primaire.

NB : Attention, dans la table exemplaire, le champs code (le code de l'exemplaire qui est la clef primaire) n'apparaît pas mais vous devez considérer qu'il existe.

1. Introduction

L'objet de ce TP est d'intégrer les requêtes développées en TP1 (recherche d'auteurs par nom, recherche d'ouvrages par titre, etc.) dans un programme PHP. Le but est d'exécuter ce programme à travers une fonction JavaScript (AJAX) afin d'afficher les résultats dynamiquement et sans quitter la page web courante.

2. Recherche d'auteurs par nom

Rappel : dans le TP1, la balise <*section*> centrale (où est affiché "Bienvenue sur le site …") a été partagée en une <*div*> gauche et une <*div*> droite (selon le fichier CSS).

1) Créer et tester un programme *recherche_auteurs.php* qui recherche les auteurs dont le nom contient le paramètre *debnom*. Ce fichier renvoie un document JSON structuré de la façon suivante :

[{"code":266,"nom":"Ambrose","prenom":"David"},
{"code":182,"nom":"Anderson","prenom":"Kevin J."},
…
]

Utiliser :
- *header("Content-Type: application/json; charset=UTF-8")* ;
- PDO (*PHP Data Objects)* pour accéder à la base de données depuis PHP (*recommandé*)
- *query()* pour exécuter la requête SQL retournant les auteurs à partir de la BDD
- *fetchAll()* pour récupérer les données des auteurs à partir de la BDD
- *json_encode()* pour encoder les données en format JSON affichées avec *echo* : JSON PHP

2) Sur événement *keyup* de la 1ère balise <*input*> du menu, appeler une fonction AJAX *recherche_auteurs()* qui lance l'exécution du programme *recherche_auteurs.php* paramétré par *debnom* dont la valeur provient de la <*input*>. Les données au format JSON retournées par *recherche_auteurs.php* seront *"parsées"* dans *la fonction callback affiche_auteurs(…)* pour obtenir un tableau (*array*) d'auteurs. Celui-ci permettra de créer une liste ordonnée d'auteurs comme suit :

```
<ol>
  <li><a href="#" onclick="recherche_ouvrages_auteur(code)">Nom Prénom</a></li>
  ...
</ol>
```

Cette liste devra être attachée à la <*div*> gauche (retrouvée à partir de son *id*).

- Dans un 1er temps, afficher uniquement le *Nom Prénom* de chaque auteur (sans les liens).
- La fonction *AJAX recherche_ouvrages_auteur(code)* devra être définie en paragraphe « *4. Recherche d'ouvrages par auteur* ».

Utiliser :
- JavaScript : *getElementById, innerHTML* et le (+) pour générer les listes de liens ; JSON HTML
- HTML : attributs événements *onclick, keyup, ….*
- AJAX : *XMLHttpRequest*, *responseText*, *onreadystatechange*, *open*(), *send*() … ; AJAX
- *JSON : JSON.parse(req.responseText)* où *req* est l'objet de type *XMLHttpRequest* ; JSON Parse

3. Recherche d'ouvrages par titre

1) Créer et tester un programme *recherche_ouvrages_titre.php* qui recherche les ouvrages (et les exemplaires associés) dont le titre contient le paramètre *debtitre*. Ce fichier renvoie un document JSON structuré de la façon suivante :

```
[{"code":392,"nom":"Dangereuses visions I",
  "exemplaires":[{"nom":"J'ai lu","code":404,"prix":"0.50"}]
 },
 {"code":397,"nom":"Dangereuses visions II",
  "exemplaires":[{"nom":"J'ai lu","code":409,"prix":"0.50"}]
 },
 ...
]
```

Utiliser : les mêmes références que celles proposées dans la question **2. Recherche par auteur**

2) Sur événement *keyup* de la 2ème balise <*input*> du menu, appeler une fonction AJAX *recherche_ouvrages_titre()* qui lance le programme *recherche_ouvrages_titre.php* paramétré par *debtitre* dont la valeur provient de la <*input*>. Les données au format JSON retournées par *recherche_ouvrages_titre.php seront "parsées"* dans la fonction *callback affiche_ouvrages(…)* pour obtenir un tableau (*array*) d'ouvrages . Celui-ci permettra de créer une liste ordonnée d'ouvrages et, pour chacun, une liste d'exemplaires comme suit :

```
<ol>
 <li> titre de l'ouvrage
  <ul>
      <li>Nom editeur1, prix1 euros</li>
      <li>Nom editeur2, prix2 euros</li>
      ...
  </ul>
 </li>
  …
</ol>
```

Cette liste devra être attachée à la <*div*> située à droite (retrouvée à partir de son *id*).

Dans un 1er temps, afficher uniquement le titre de chaque ouvrage (sans les exemplaires).
Dans un 2nd temps, définir une fonction JavaScript *affiche_exemplaires(exemplaires)* qui affiche dans la liste <*ul*> les exemplaires de l'ouvrage.
Utiliser : les mêmes références que celles proposées dans la question **2. Recherche par auteur**

4. Recherche d'ouvrages par auteur

Dans la question 2) du paragraphe « **2. Recherche d'auteurs par nom »,** la fonction AJAX *recherche_ouvrages_auteur(code)* lance le programme *recherche_ouvrages_auteur.php* paramétré par *code* de l'auteur. Les données au format *JSON* retournées par *recherche_ouvrages_auteur.php* seront *"parsées"* dans la fonction *callback affiche_ouvrages(…)* pour obtenir un tableau (*array*) d'ouvrages. Celui-ci permettra de créer la liste des ouvrages de l'auteur de la même forme que le document vu à la « question *2) du paragraphe 3. Recherche d'ouvrages »*.
Cette liste devra être attachée à la <*div*> située droite (retrouvée à partir de son *id*).

NB : faire en sorte que les affichages des listes ne se chevauchent pas.

Développement Web Back End
TP3 : Achat de livres

1. Introduction

On veut réaliser un panier qui va coder l'ensemble des achats d'une personne. Ce panier sera stocké en mémoire chargé à partir de celle-ci à chaque nouvelle connexion et sauvegardé dans la base de données à chaque déconnexion.

2. Modification de la base

Créer trois tables *clients*, *panier* et *commande*. La table client contient, le nom, le prénom, l'adresse (adresse,CP, ville, pays) et la date d'inscription. La table *panier* contient des triplés (code client, code exemplaire, quantite). La table *commande* contient l'ensemble des articles qui ont été commandés. Elle sera composée de quintuplés (code client, code exemplaire, quantite, prix, date). Utiliser :

- https://stph.scenari-community.org/bdd/sql1.pdf
- https://www.w3schools.com/sql/default.asp

3. Enregistrement d'un client

1. Dans un fichier *inscription.sql*, définir en PL/pgSQL une fonction *inscription* qui effectue l'insertion dans la base. Cette fonction retourne le code client si l'inscription s'est bien passée, ou 0 s'il existe une personne de même nom, prénom et adresse dans la base. Exécuter cette fonction sous postgresql. Utiliser : COURS_EXEMPLES_PLPGSQL_4p.pdf

2. Créer un programme *inscription.php* qui appelle la fonction précédente et renvoie son résultat (« no » ou *codeclient*).

3. Définir une fonction AJAX *enregistrement()* qui lance le programme *inscription.php* en utilisant les données saisies par le client ; selon le retour de *inscription.php :*
— En cas de succès :
- la fonction positionne un cookie (se terminant en 2050) contenant le code client de l'utilisateur
- la fonction redirige l'utilisateur sur la page d'accueil.

— En cas d'échec :
- la fonction affiche un message approprié dans une balise *div* située sous le formulaire d'inscription.

Utiliser : le champ *responseText* et la propriété *innerHTML*.

4. Dans la section principale, créer un formulaire d'enregistrement d'un client, permettant de renseigner tous les champs requis. Sur clic du bouton ≪ *Envoyer* ≫, la fonction AJAX *enregistrement()* est lancée.

Par la suite, le formulaire ne devra apparaître que sur clic sur le lien « *inscription* » (voir paragraphe 4.).
Utiliser : https://www.w3schools.com/php/php_forms.asp

Développement Web Back End
TP3 : Achat de livres

4. Affichage du nom de l'utilisateur et ouverture de session

Dans *index.php*, ajoute*r en 1ère ligne session_start()* afin d'ouvrir une session. Modifiez la bannière selon :
1. Si l'utilisateur ne possède pas de cookie, afficher dans la balise supérieure un lien « *inscription* ».
2. Si l'utilisateur possède un cookie client et si le nom et le prénom ne font pas partie des variables de sessions, effectuer une requête sur la base pour récupérer son nom et son prénom. Le nom, le prénom et le code client doivent être stockés comme variables de sessions. Modifier enfin la *div* correspondant à la bannière supérieure pour afficher "Bienvenu Prenom Nom".
Utiliser :

- https://www.w3schools.com/php/php_sessions.asp
- https://www.w3schools.com/js/js_cookies.asp

5. Remplissage du Panier

Modifier l'affichage des exemplaires du TP précédent pour afficher un lien [ajouter au panier] à coté de chaque exemplaire. Ce lien appelle une fonction AJAX *ajouter_panier(code_exemplaire)* qui appelle elle-même un programme *ajouter_panier.php.* Ce programme insère dans la table panier l'exemplaire commandé par le client. La fonction *callback* lancera une alerte "*L'article a été ajouté au panier*".

6. Consultation du panier

Ajouter dans la bannière, à droite, un lien Consulter le panier. Sur clic sur ce lien, le contenu du panier s'affiche dans la partie principale :
— une liste contenant : le titre de l'ouvrage, le nom de l'éditeur, la quantité, le prix.
— Le prix total.
— un bouton ≪ Commander ≫
— un bouton ≪ Fermer ≫ .

Sur clic du bouton ≪ *Commander* ≫, transférer les quintuplés (code client, code exemplaire, quantite, prix, date) contenu dans la table *panier* vers la table *commande*.

Sur clic du bouton ≪ *Fermer* ≫, le contenu du panier disparaît.

7. Vider le panier

Ajouter dans la bannière, à droite, un lien Vider le panier. Ce lien appelle une fonction AJAX *vider_panier()* qui appelle elle-même un programme *vider_panier.php.* Ce programme supprime dans la table panier les lignes de commandes du client. La fonction *callback* lancera une alerte "*Le panier a été vidé*".