

TP : Application de Gestion de Bibliothèque

Objectif

Dans ce TP, vous allez apprendre à manipuler le DOM en JavaScript pour afficher dynamiquement des livres sous forme de cartes, gérer les erreurs de saisie et implémenter un système de recherche. Vous allez également ajouter des fonctionnalités supplémentaires pour améliorer l'expérience utilisateur.

Guide pour la Réalisation de l'Application de Gestion de Bibliothèque

Vous allez créer une application web permettant d'ajouter des livres à une bibliothèque et de les rechercher. L'interface se base sur le fichier `bibliotheque.html` fourni, qui comporte un formulaire pour ajouter un livre et une section pour afficher les livres existants.

1. Comprendre le Fichier HTML

Le formulaire d'ajout de livre :

Vous trouverez plusieurs champs :

- **Titre** : `<input>` avec `<p id="titre-error">` pour afficher un message d'erreur si nécessaire.
- **Auteur** : `<input>` avec `<p id="auteur-error">`.
- **Année de publication** : `<input type="number">` avec `<p id="annee-error">`.
- **URL de la couverture** : `<input>` avec `<p id="couverture-error">`.

La section de recherche :

Un champ `#filtre` permet de rechercher les livres par titre.

La zone d'affichage des livres :

Une `<div id="livres">` où seront ajoutées les cartes de livre créées.

2. Ajouter un Livre

a. Intercepter le formulaire

- Écoutez l'événement `submit` sur le formulaire identifié par `id="formulaire"`.
- Utilisez `e.preventDefault()` pour éviter que la page ne se recharge lors de la soumission.

b. Récupérer et Valider les Données

Pour chaque champ, récupérez la valeur saisie et effectuez une validation simple :

- **Titre & Auteur** : Vérifiez que le champ n'est pas vide, sinon affichez un message dans `<p id="titre-error">` ou `<p id="auteur-error">`.
- **Année de publication** : Vérifiez que l'année est un nombre valide (par exemple, plus grand que 0). Si l'année est invalide, affichez un message dans `<p id="annee-error">`.
- **Couverture** : Vérifiez que l'URL de la couverture est renseignée. Si vide, affichez un message dans `<p id="couverture-error">`. Assurez-vous de vider le message d'erreur si l'utilisateur corrige sa saisie.

c. Créer et Afficher la Carte

Si toutes les validations sont correctes :

- Créez dynamiquement une carte (par exemple, un `<div>` avec des classes Bootstrap) contenant :
 - Le titre et l'auteur (`<h5 class="card-title">`).
 - L'année de publication (`<p class="card-text">`).
 - L'image de couverture (``).
 - Un bouton de suppression (`<button class="btn btn-danger">`).
- Ajoutez la carte dans `<div id="livres">`.

3. Recherche de Livre

Ajoutez un champ de recherche permettant de filtrer les livres affichés.

- L'utilisateur tape un titre, seules les cartes correspondantes doivent rester visibles.
- Utilisez `toLowerCase()` pour rendre la recherche insensible à la casse.
- Masquez les cartes qui ne correspondent pas et affichez celles qui conviennent.

4. Supprimer un Livre

- Pour chaque carte créée, ajoutez un bouton « Supprimer ».
- Lorsque l'utilisateur clique sur ce bouton, retirez la carte correspondante du DOM.

5. Trier par Année de Publication

- Implémentez une fonctionnalité pour trier les livres selon leur année de publication (du plus ancien au plus récent, ou l'inverse).
- Ajoutez un bouton ou un menu déroulant pour activer cette fonctionnalité.

6. Modifier un Livre

- Ajoutez une fonctionnalité permettant de modifier les informations d'un livre après sa création.
- Lorsqu'un utilisateur clique sur une carte, pré-remplissez le formulaire d'ajout avec les informations du livre et permettez à l'utilisateur de les modifier. Lors de la validation, mettez à jour la carte affichée.

7. Afficher un Nombre Total de Livres

- Ajoutez une fonctionnalité pour afficher le nombre total de livres présents dans la bibliothèque en temps réel. Ce nombre doit être mis à jour chaque fois qu'un livre est ajouté ou supprimé.

8. Sauvegarder et Restaurer les Données

- Implémentez une fonctionnalité permettant de sauvegarder les livres dans le stockage local du navigateur (`localStorage`). Lors du rechargement de la page, les livres sauvegardés doivent être restaurés automatiquement.

9. Afficher un Message d'Alerte

- Ajoutez un message d'alerte qui s'affiche si l'utilisateur tente d'ajouter un livre sans remplir tous les champs correctement. Ce message disparaît une fois que l'utilisateur a corrigé la saisie.

10. (Optionnel) Pagination

- Si la bibliothèque contient un grand nombre de livres, ajoutez un système de pagination pour afficher les livres par groupes (par exemple, 5 livres par page). Les utilisateurs peuvent naviguer entre les pages pour voir d'autres livres.

Points Importants

- Utilisez `document.createElement()` pour créer chaque élément.
- Utilisez `appendChild()` pour ajouter les éléments dans la bonne structure.
- Pensez à écouter l'événement `click` sur le bouton pour supprimer la carte du DOM.
- Ajoutez un `EventListener` sur le champ de recherche pour filtrer dynamiquement les cartes.

Conseils

- N'oubliez pas `trim()` ! Il permet de supprimer les espaces inutiles lorsque vous récupérez les valeurs du formulaire.
- Testez par étapes. Commencez par créer un seul livre avant de généraliser.
- Vérifiez la console (`console.log()`) pour déboguer vos erreurs.
- Commentez votre code pour expliquer chaque étape.

À vous de jouer ! 🚀