

Projet : site internet "Readers Community"

Ghizlane Elachouri & Oualid Hamza

I. Brief:

Présentation du site web :

Readers Community est un site d'échange de livres d'occasion entre particulier.

Mode d'emploi :

Chaque utilisateur en arrivant sur le site web soit s'inscrire soit se connecter. Une fois il est connecté, il peut :

- 1/ **afficher** la liste de ses livres disponibles à l'échange.
- 2/ **afficher** les détails d'un livre (titre, l'auteur, résumé, catégorie, image)
- 3/ **ajouter** des livres qu'il possède en remplissant un formulaire d'information sur les livres qu'il souhaite proposer (le titre, l'auteur, catégorie, image).
- 4/ **modifier** les détails sur les livres disponibles.
- 5/ **supprimer** un livre.
- 6/ commander des livres,

II. Conception:

1. Création de la base de données

La base de données est nommée *readerdb* et comporte 3 tables:

1) table user: 4 colonnes (id, int PK AI), (name, varchar(50)), (email, varchar(50)), (password, varchar(50))

2) table book: 8 colonnes (id, int PK AI), (title, varchar(50)), (author, varchar(50)), ((category, varchar(50)), (summry, text), (comment, text), (image, text) et (owner, varchar(50))

3) table categorie : (id, int PK AI), ((category, varchar(50)),

2. Les différentes pages : (Modèle MVC)

- *connexion.php* : la page d'accueil avec une présentation du site et comportant deux liens (connexion ou inscription)

Pour la connexion on a un formulaire en mode POST avec 2 champs de type text : email, mot de passe et un bouton submit

L'attribut action du formulaire redirige vers l'adresse *view_books.php*

Pour l'inscription on a un formulaire en mode POST avec 3 champs de type text : nom, email et mot de passe et un bouton submit

L'attribut action du formulaire redirige vers la page de connexion *connexion.php*

- page web *view_books.php* permet
- affiche le formulaire
- affiche les messages d'alerte
- affiche les messages de déconnexion
- analyse des variables de type post
- Poo

class user

methode constructeur pour se connecter à la base

attribut private *mysqli* pour contenir la variable de connexion à la base

une méthode *verify_user* qui établit une requete SELECT pour obtenir les informations user depuis la base de données.

Si c'est vide, alors user non reconnu

une méthode *add_user* qui établit une requete INSERT (formulaire) les informations user vers la base de données.

Si c'est vide, alors user non ajouter

une méthode redirect vers home_page.php ou vers admin.php

une méthode logout...

Class Book

méthode constructeur pour se connecter à la base

attribut private mysqli pour contenir la variable de connexion à la base

une méthode read_book qui établit une requête SELECT dans la base de données.

une méthode add_book qui établit une requête INSERT dans la base de données.

une méthode update_book qui établit une requête UPDATE dans la base de données.

une méthode delete_book qui établit une requête DELETE dans la base de données.

process

2. affichage des articles

2.1 liste des articles

PHP se connecte à MySQL avec la fonction mysqli().

la requête SQL SELECT est effectuée

les articles sont affichés au moyen d'une boucle while

ils contiennent le name et le slug pour créer le lien qui affichera le contenu

dans la même page.

2.2 suppression des articles

un lien après le nom de l'article contient le id de l'article et permet d'accéder à la page delete.php.

Dans cette page un GET recupere le id. Cette variable est ensuite utilisée pour la clause WHERE de la requête de suppression.