MATUCHET Louis louis.matuchet9@etu.univ-lorraine.fr
HUBLAU Alexandre alexandre.hublau9@etu.univ-lorraine.fr

L3 Informatique
2019-2020

Programmation web

Lien du site: https://boissons.alexandre-hublau.com

Analyse

Le stockage des données et leur organisation

Il y a dans notre projet deux types d'éléments à stocker et chacun a un système différent :

- Les données (recettes dans Donnees.php) : On utilisera uniquement le fichier susnommé, nous récupérons les recettes provenant de ce fichier que nous stockons dans un array (tableau).
- Les utilisateurs (connexion, inscription) : Ils seront stockés dans la base de données après inscription. C'est également ce moyen là qui permet de sauvegarder les recettes des utilisateurs.

La gestion/vérification du formulaire

Pour ce qui est de la gestion de la vérification du formulaire d'inscription nous utilisons des expressions régulières que l'on peut retrouver dans le fichier UtilisateursController.
Pour la connexion de l'utilisateur (après inscription), on fait juste une requête sur la base de données pour voir s'il existe bien et s'il correspond à la bonne personne.

La gestion du panier

Il y a deux cas pour la gestion du panier :

- L'utilisateur est connecté : les ajouts des recettes au panier sont faites sur la base de données relativement à l'utilisateur connecté.
- L'utilisateur n'est pas connecté : les recettes sont ajoutés dans le panier mais seulement dans la session. S'il quitte et relance le site il ne verra plus les recettes ajoutées auparavant.

La création du compte

Un utilisateur peut créer un compte. Ses données sont sauvegardés en base de données. Cependant, le **mot de passe est chiffré** avant d'être sauvegardé !

Le traitement de la connexion utilisateur

L'utilisateur entre ses informations. Nous exécutons la requête préparée pour voir si l'utilisateur existe :
SELECT id,pseudo,mot_de_passe FROM utilisateurs WHERE pseudo=? LIMIT 1

Si l'utilisateur existe et que son mot de passe chiffré est vérifié par rapport au mot de passe entré, alors il est connecté. Nous le sauvegardons alors dans une session. Sinon nous envoyons un message d'erreur.

Système de recherche

Le système de recherche se passe en deux temps :

Une page PHP s'occupe de filtrer les recettes. Nous utilisons array_filter() pour filtrer les données du tableau des recettes grâce à un système de mots clés. Puis nous renvoyons dans le format JSON.

Exemple: "recherche=Pastis,Pas de malibu" renvoie les cocktails qui contiennent du pastis mais pas de malibu.

Une seconde page est utilisée pour effectuer les requêtes AJAX. En visitant la page de recherche en PHP, nous arrivons à charger correctement les cocktails et les afficher à l'utilisateur.

Mode d'emploi pour les utilisateurs

Il y a 2 types d'utilisateurs pour notre application : l'utilisateur connecté et celui qui ne l'est pas. Un utilisateur non connecté aura les mêmes permissions qu'un utilisateur connecté en ce qui concerne l'ajout de recettes.
Il pourra s'il le souhaite parcourir les recettes qui existent sur le site, utiliser le moteur de recherche de recettes et même ajouter des recettes dans son panier.

Une différence avec un utilisateur connecté (et donc inscrit) est que ce panier n'est pas éternel et disparaîtra quand l'utilisateur quittera le site. Un utilisateur connecté pourra donc retrouver les recettes précédemment ajoutées s'il revient sur le site plus tard et se connecte.
Aussi, un utilisateur connecté pourra accéder à la page pour modifier ses informations personnelles.

Mise en ligne + lien

La mise en ligne a été faite et on peut retrouver le projet sous le lien:
https://boissons.alexandre-hublau.com

Description du zip / arborescence

- dossier "src" avec toutes les sources php.
  - dossier "config" avec les configurations (informations concernant la BDD par exemple)
  - dossier "modeles" qui contient les informations de l'applications
  - dossier "vues" qui contient le code HTML
  - dossier "controlleurs" qui contient la logique de notre application
- dossier "public" avec l'"index.php", les fichiers css, javascript ainsi que toutes les images

La description de l'interface et les fonctionnalités implantées

Toutes les fonctionnalités demandées dans le sujet ont été implémentées.

Les messages flash

Les messages flash sont des messages qui apparaissent uniquement pendant le chargement d'une seule page. Pour afficher des messages tel que les messages d'erreur ou de succès, nous les avons implémenté. Ils sont sauvegardés dans la session de l'utilisateur.

Autoloader

On a utiliser un autoloader (une fonction du psr-4) pour ne pas à faire 50 *require()* et *include()* dans chaque classe. La fonction est utilisée ici: https://www.php-fig.org/psr/psr-4/examples/

Environnement utilisé

Le projet fonctionne sous WAMP 3.2.0.
Nous avons codé le projet avec PHP 7.2.

En revanche, l'environnement de production (https://boissons.alexandre-hublau.com) tourne sous NGINX et une base de données MariaDB (MariaDB 15.1 & NGINX 1.14.0).
Le système d'exploitation est Ubuntu Server 19.10.

En cas de difficulté ou pour avoir plus d'information, nous pouvez vous envoyez un mail:

louis.matuchet9@etu.univ-lorraine.fr
alexandre.hublau9@etu.univ-lorraine.fr