Projet 2020 : création d'une application participative "petites annonces"

Vous travaillerez **en binôme ou en trinôme**.
Durée : 15 h de TP prévues du 4 novembre au 9 décembre.
À rendre sous forme d'archive (fichier zip, tar.gz ou autre) **par email** :
mathias.chouet@gmail.com , **le 11 décembre** à 18h au plus tard.
Lors de la dernière séance de TP (9 décembre) vous devrez me **présenter en 10mn** maximum votre projet, même s'il n'est pas terminé (j'en tiendrai compte).

Énoncé

Ce projet consiste à construire une application Web qui gère des petites annonces.
Une **petite annonce** est caractérisée par :

- un titre
- une description, c'est à dire le texte de l'annonce
- une catégorie (immobilier, vêtements, loutres…)
- le nom ou pseudo du vendeur
- un prix
- une photo
- le point de rendez-vous pour retirer l'objet (avec des coordonnées GPS)
- une date d'ajout

(la structure de la base de données vous sera donnée plus bas)

L'application permettra de **lister** les annonces existantes et **rechercher** parmi elles, **saisir** de nouvelles annonces, **supprimer** des annonces existantes.

Les données seront stockées dans une **base de données** SQLite.
La suppression d'annonces nécessitera une **authentification** (login, mot de passe).
L'application devra avoir un **style** et une **ergonomie** agréables, et sera **dynamique**.
Les échanges seront effectués à travers des **Webservices**, en utilisant le format **JSON**.
L'application devra être consultable sur un appareil mobile (design adapté aux petits écrans).

Avant de commencer

Déroulement du développement

Vous êtes libres de l'organisation, de la présentation et du style : des consignes de base sont données mais vous pouvez changer ce qui vous plaît : soyez créatifs !

1 - Squelette de l'application Web (4 points)

Organisation du code

Votre application sera composée de plusieurs pages (HTML, PHP), plusieurs Webservices (PHP), du code côté client (Javascript) et du style (CSS).

Prévoyez dès le début d'organiser proprement votre répertoire de travail : créez un répertoire pour le projet puis créez des sous-répertoires si besoin. Vous nommerez vos fichiers de telle sorte qu'il soit facile de comprendre leur rôle (ex: il vaut mieux nommer votre webservice de liste des annonces "`liste_annonces.php`" que "`projet_1.php`").

Vous n'oublierez pas d'indenter (décaler à droite avec des tabulations ou des espaces) les blocs de votre code pour faciliter la lisibilité et éviter les erreurs.

Première page

a) Créez une page d'accueil en HTML, avec un entête et un pied de page. L'entête contiendra le titre de l'application, et un logo de votre choix. Le pied de page contiendra vos noms, la date de dernière mise à jour du site. Ajoutez d'autres informations si vous le souhaitez.
Le centre de la page contiendra un texte de bienvenue qui présente brièvement l'application.
Remplissez la partie `<head>` comme vous l'avez appris : titre, codage de caractères etc.

b) Ajoutez un menu à votre page, en ligne en haut de la page. Ce menu servira à passer d'une page à l'autre. Pour l'instant il n'y a qu'une page, mais vous en ajouterez par la suite. Vous enrichirez au fur et à mesure le menu par de nouveaux liens (un par page).

c) Incluez dans le `<head>` de la page un fichier Javascript, que vous remplirez plus tard.

d) Incluez aussi un fichier CSS, et créez un style basique pour votre page. Par exemple: couleur de fond, bordures sur l'entête et le pied de page, padding et/ou marges pour aérer la page, couleur et police du titre. Agrémentez votre style autant que possible, mais arrangez-vous pour que votre page ne fasse pas mal aux yeux !

2 - Base de données (2 points)

a) Téléchargez le fichier "`bdd.php`" dans votre dossier de projet. Créez un nouveau fichier vide "`annonces.db`" puis donnez les droits adéquats à l'aide de la commande suivante :
`chmod 777 . && chmod 777 annonces.db`

"`bdd.php`" contient deux fonctions toutes prêtes pour initialiser la base de données : l'une crée une table pour stocker les petites annonces, et l'autre la remplit avec des données d'exemple.

b) Créez un fichier PHP pour amorcer votre base de données : il incluera “`bdd.php`”, et appellera successivement les deux fonctions.

Note : ces fonctions ne doivent normalement être exécutées qu’une seule fois. En cas de problème, une fonction est prévue pour vider la table.

Par la suite vous réutiliserez “`bdd.php`” dans vos Webservices afin de vous connecter à la base de données et y exécuter des requêtes.

Note: pour visualiser plus facilement le contenu de votre base de données, vous pouvez utiliser https://sqliteonline.com (File > Open DB)

3 - Affichage des données en liste (3 points)

a) Créez une nouvelle page pour **visualiser la liste** des petites annonces. Elle devra se connecter à la base de données pour obtenir les données des annonces. Les annonces seront affichée avec toutes leurs caractéristiques et leur photo, les unes au dessus des autres.

b) Ajoutez un lien vers cette page dans le menu, et assurez-vous que l’on puisse passer d’une page à l’autre (page d’accueil et liste des annonces) à l’aide du menu.

Note : si plusieurs de vos pages contiennent des parties communes (un menu, un entête ou un pied de page par exemple !), vous pouvez utiliser le mécanisme “include” de PHP afin d’éviter de répéter le contenu commun dans chaque page.

4 - Liste dynamique, recherche (3 points)

Liste dynamique

a) Créez une nouvelle page et ajoutez-la au menu. Cette page affichera également la liste des annonces, mais cette fois **de façon dynamique en utilisant AJAX** (Javascript + Webservices), comme indiqué ci-dessous.

b) Créez un Webservice pour lire les données depuis la base de données. Il fonctionnera avec la même requête que celle utilisée dans la page de liste précédente, mais les données devront maintenant être renvoyées au format JSON, au lieu de les afficher directement à l’écran.

c) Placez un bouton “afficher les données” dans la page. Un appui sur ce bouton devra appeler le Webservice et afficher les données dans la page, avec le même rendu (toutes les caractéristiques plus l’image) que dans la page statique réalisée au point 3.

Recherche d'annonces

d) Modifiez le Webservice pour permettre de faire une recherche parmi les annonces. Pour cela, ajoutez **un paramètre** au service, qui **modifiera la requête SQL** et ne renverra que les données dont un des champs correspond au paramètre.
Pour la requête SQL avec filtrage, inspirez-vous de ce tutoriel : http://sql.sh/cours/where/like
Pour utiliser plusieurs clauses, inspirez-vous de ceci : http://sql.sh/cours/where/and-or

Par exemple, si le paramètre vaut "lou", le service renverra l'annonce publiée par "Louise Duschmoll", mais aussi celle dont le titre est "Lot de 3 loutres".

e) Ajoutez un champ de texte à côté du bouton "afficher les données". Lorsqu'on appuie sur le bouton, la valeur du champ de texte doit être passée au service comme paramètre de recherche, et seuls les résultats correspondant à la recherche doivent être affichés.

5 - Ajout et suppression de données (3 points)

Ajout d'une petite annonce

a) Créez un Webservice qui permette d'ajouter une nouvelle annonce dans la base de données. Pour la requête d'insertion, inspirez-vous des exemples présents dans "`bdd.php`"
Note : le champ "id" prendra automatiquement la valeur numérique suivante (auto_increment), et le champ "date" prendra automatiquement le *timestamp* actuel, si vous ne mentionnez pas ces colonnes après le nom de la table, dans la requête `INSERT`.

b) Sur la même page que la liste dynamique, créez un formulaire HTML pour ajouter une annonce. Pour gérer l'image, utilisez un simple champ de texte dans lequel on saisira une URL. La validation de ce formulaire devra appeler le Webservice d'ajout, et actualiser la liste des annonces afin d'afficher les données qu'on vient d'ajouter.

Suppression d'une annonce

c) Créez un Webservice qui permette de supprimer une annonce de la base de données, en prenant en paramètre son identifiant ("`id`").
Pour la requête SQL de suppression, inspirez-vous de ce tutoriel (ignorez le paragraphe "Attention") : http://sql.sh/cours/delete

d) Modifiez la liste dynamique pour afficher à côté de chaque annonce un bouton "supprimer". Un appui sur ce bouton devra appeler le Webservice de suppression, et actualiser la liste des annonces à l'écran. Vous pourrez utiliser la fonction Javascript "`confirm()`" pour demander confirmation avant de supprimer.

Note: vous devrez stocker quelque part l'id de l'annonce pour pouvoir demander sa suppression - une bonne idée est d'utiliser les attributs HTML "`data`" (voir TP 5)

6 - Authentification (3 points)

Nous voulons maintenant ajouter un système d'authentification pour les utilisateurs de l'application. Un utilisateur sera identifié par un nom d'utilisateur (login) et un mot de passe. L'objectif est de n'autoriser l'ajout et la suppression de données qu'aux utilisateur authentifiés.

Authentification

a) Créez un Webservice d'authentification. Ce service recevra deux paramètres (nom d'utilisateur et mot de passe) et devra contrôler la validité du couple login / mot de passe. Il devra renvoyer :

- le nom d'utilisateur si la personne est connue et a fourni le bon mot de passe
- un code d'erreur HTTP 401 dans le cas contraire

Vous utiliserez le mécanisme des sessions PHP pour définir si un utilisateur est connecté ou non. Si un utilisateur a fourni un couple login / mot de passe correct, une session sera créée.

Les utilisateurs et leurs mots de passe seront stockés dans la base de données, vous devrez pour cela créer une nouvelle table. Inutile de permettre de les modifier, on considère qu'ils ne changent pas.

b) Créez un formulaire d'authentification avec deux champs (nom d'utilisateur, mot de passe) et un bouton pour se connecter. Ce formulaire devra être accessible sur toutes les pages (dans le menu par exemple, ou dans un coin en haut de la page).
L'appui sur le bouton de connexion devra appeler le Webservice d'identification, et en fonction de sa réponse modifier l'affichage : si l'utilisateur est correctement identifié, on affichera un message de bienvenue (ex: Bonjour Camille) et un bouton pour se déconnecter. Sinon, on affichera une erreur et on proposera à nouveau de se connecter.

Lorsqu'on change de page, l'état de connexion devra être conservé. Pour cela, on utilisera les sessions PHP pour déterminer si l'utilisateur est connecté ou non.

Restriction d'accès pour la suppression

c) Toujours en utilisant les sessions PHP, modifiez le Webservice de suppression pour qu'il ne soit accessible qu'aux utilisateurs authentifiés. Dans le cas contraire, il devra maintenant renvoyer une erreur.
Dans la liste dynamique, lors d'un appui sur le bouton de suppression d'une annonce, vous devrez afficher une erreur si l'utilisateur n'est pas authentifié.

Note: on pourrait simplement masquer les boutons de suppression de l'interface, mais cela n'empêcherait pas un vilain hacker d'appeler le Webservice directement !

7 - Style et ergonomie (2 points)

2 points seront accordés au style et à l'ergonomie de votre application. Vos pages devront être agréables à lire et utiliser l'espace correctement. Un utilisateur doit pouvoir comprendre au premier coup d'œil comment utiliser votre application et trouver toutes les informations dont il a besoin. Par exemple, si la police est trop petite, s'il faut faire défiler la page pour trouver le login ou le menu, ce n'est pas très pratique...
L'ergonomie devra être satisfaisante : un minimum de clics pour obtenir le résultat désiré, des formulaires clairs, des boutons et des messages au titre explicite, etc.

Ce n'est **pas obligatoire** pour avoir tous les points, mais vous pouvez utiliser un framework CSS pour agrémenter votre style. Parmi les plus connus, il y a Bootstrap (facile à utiliser) ou Materialize (plus difficile), mais il en existe beaucoup d'autres.
Pour utiliser un de ces ces frameworks, téléchargez le(s) fichier(s) CSS et incluez-le(s) à vos pages comme vous incluriez n'importe quel autre fichier CSS.
L'ajout de classes CSS aux éléments de votre page permet de bénéficier de l'un ou l'autre des styles prévus par les frameworks (consultez la documentation).

Attention: un framework CSS mal utilisé peut rendre votre site moins beau !

8 - Navigation mobile (1 point)

Cette application doit pouvoir être visible sur un appareil mobile, tel qu'un téléphone ou une tablette. Ces appareils ont un petit écran, souvent orienté verticalement.

Modifiez le style si nécessaire pour que l'application s'affiche correctement sur un petit écran vertical. Vous utiliserez pour cela une ou plusieurs **media-query CSS**. Pour vérifier le rendu, réduisez la fenêtre de votre navigateur et assurez-vous que les pages restent lisibles.