

Projet programmation web

2011-2012 - falleri@labri.fr

Introduction

Le projet de programmation web doit être réalisé par **binômes**. Il a pour but de vous faire développer une application web utilisant l'architecture *page générées - client riche*. Vous serez notés sur les trois parties de l'application, c'est-à-dire :

- Qualité du code XHTML/CSS (sur 6)
- Qualité de l'apparence visuelle, simplicité d'utilisation (sur 4)
- Qualité du code PHP/SQL, qualité de la structure de la base de données (sur 6)
- Qualité du code JavaScript (sur 4)

Voici la liste des technologies qui doivent être autorisées dans le projet :

- XHTML 1.1
- CSS 2
- SQLite 2
- PHP 5, fonctions **uniquement**, pas de mécanismes objets
- JavaScript, librairie JQuery hautement recommandée

Enfin faites particulièrement **attention à la copie**, j'utilise un logiciel qui permet de détecter les copier-coller automatiquement, tout projet suspect sera étudié avec la plus grande attention...

Concernant le rendu, vous devez rendre le code source du projet, situé dans un répertoire *src* ainsi qu'un rapport au format texte brut, nommé *architecture.txt*, décrivant l'intégralité des fichiers contenus dans votre application, selon la forme suivante :

```
...
src/fichier1.php : description du fichier 1.
src/fichier2.js : description du fichier 2.
...
```

Les description doivent être courtes et indiquer clairement quelles sont les fonctionnalités du fichier. Enfin vous devez fournir un document en texte brut nommé *installation.txt* qui doit expliquer la procédure d'installation, si nécessaire, de votre application web (initialisation des bases de données, etc...). Vous devez compresser le tout dans une archive nommée *it103-2012-gN.zip* avec *N* étant le numéro de votre groupe. Enfin, les fichiers Javascript, PHP, CSS et XHTML doivent être correctement indentés. Les notes des groupes ne respectant pas ces règles se verront amputées de deux points.

Le projet est à rendre par mail au plus tard le 20/05/2012.

Sujet

Vous devez développer un gestionnaire de post-its multi utilisateurs. Le principe de cette application est le suivant : les utilisateurs doivent tout d'abord s'inscrire ; ensuite ils peuvent se connecter sur l'application ; après connexion, l'application permet de visualiser les différents post-its publiés par l'utilisateur, ainsi que les différents post-its qu'on lui a partagés. Il est possible de cliquer sur chaque post-it afin de voir son contenu.

Fonctionnalités

Écran de connexion

L'écran de connexion permet à un utilisateur de se connecter. Pour cela il doit fournir un mail ainsi qu'un mot de passe. L'écran de connexion doit fournir un lien vers l'écran d'inscription. Si l'utilisateur arrive à se connecter, il doit être redirigé vers l'écran d'accueil. Sinon, un message d'erreur doit expliquer pourquoi la tentative de connexion à échouée.

Écran d'inscription

L'écran de connexion permet à un utilisateur de s'inscrire. Pour cela il doit fournir un mail, un mot de passe qui doit être saisi deux fois pour éviter les erreurs. Il doit aussi fournir un nom, un prénom, une date de naissance. Une fois l'utilisateur inscrit, il doit être redirigé vers l'écran de connexion. Un script Javascript doit vérifier avant l'envoi du formulaire que le mot de passe fait au moins 6 caractères, que le mail est dans un format correct (login@domaine.extension), et que la date de naissance est dans le format correct (AAAAMMJJ). En outre, toutes les données demandées doivent être saisies. Si les données saisies par l'utilisateur ne sont pas correctes, elles doivent être affichées sur fond rouge. Dans ce cas des messages, également en rouge, doivent expliquer pourquoi les données ne sont pas correctes. Attention, il ne peut y avoir qu'un seul utilisateur ayant une adresse mail donnée.

Écran d'accueil

L'écran d'accueil permet à un utilisateur connecté de voir la liste des post-its qui l'intéresse. Il est séparé en deux zones : la zone des post-its dont il est propriétaire (possédés), et la zone des post-its qu'on lui a partagé (partagés). Enfin l'écran d'accueil doit fournir un lien vers l'écran d'ajout de post-its, et un lien qui permet de se déconnecter et de retourner sur l'écran de connexion.

Zone des post-its possédés

Cette zone affiche la liste des titres de post-its dont l'utilisateur est propriétaire, ainsi que la date à laquelle ils ont été postés. Ils sont classés par dates d'ajouts (le plus récent en premier). Un clic sur le titre doit amener à l'écran de visualisation de post-its. En outre, à coté de chaque post-its, il doit y avoir un lien qui permet de supprimer le post-it et un lien qui permet d'éditer le post-it.

Zone des post-its partagés

Cette zone affiche la liste des titres de post-its partagés à l'utilisateur, ainsi que la date à laquelle ils ont été postés et le propriétaire des post-its. Ils sont classés par dates d'ajouts (le plus récent en premier). Un clic sur le titre doit amener à l'écran de visualisation de post-its.

Écran de visualisation des post-its

Cet écran doit afficher le titre du post-it, la date d'ajout, le propriétaire et la liste des utilisateurs avec lesquels il est partagé. Quand l'utilisateur courant est le propriétaire, il doit y avoir en outre un lien qui permet d'éditer le post-it et un lien qui permet de supprimer le post-it.

Écran d'ajout/édition des post-its

Cet écran doit afficher un formulaire permettant de saisir le titre du post-it (maximum 150 caractères), et la liste des utilisateurs avec lesquels il est partagé. Le propriétaire ainsi que la date d'ajout (ou de dernière mise à jour si le formulaire est en mode édition) seront automatiquement calculés en PHP. En mode édition, les champs permettant de saisir le titre, contenu et utilisateurs doivent être pré-remplis. Une attention particulière sera apportée à la manière dont vous gérez le partage avec les autres utilisateurs. En effet, il ne faut pas présager que celui qui saisit le post-it connaît les mails des utilisateurs avec lesquels il veut partager le post-it. Il sera donc nécessaire d'utiliser un mécanisme d'auto-complétion ou des cases à cocher (ou tout autre mécanisme provenant de votre imagination débordante). En outre, il ne doit pas être possible de déposer un post-it qui n'a pas de titre ou de contenu, ou un titre trop long. Si c'est le cas la données manquante doit être affichée sur fond rouge et un message d'erreur également en rouge doit être affiché.

Indications

Bien évidemment, le contenu variable du site (utilisateurs, post-its) devra être stocké dans une base de données de type SQLite. Réfléchissez bien à la structure de votre base, elle est plus compliquée que celle abordée dans les TP. Pour créer la structure initiale de la base, je vous conseille de rédiger un script *install.php* destiné à être exécuté une seule fois puis supprimé. Si vous utilisez cette technique, n'oubliez pas de le documenter dans le fichier *installation.txt*. Bien entendu, il est **interdit** de stocker le mot de passe des utilisateurs en clair dans la base de données. Vous devez donc trouver un mécanisme qui évite cela. Utilisez pour cela les [fonctions de hachage](#).

Toute la partie serveur (réception des données provenant des formulaires, écritures et lectures dans la base) doit se faire en PHP. Pour être à même de gérer le mécanisme de connexion (qui nécessite de pouvoir savoir à la demande d'une page si on est déjà connecté ou non pour afficher l'écran de connexion ou l'écran d'accueil) vous devez utiliser les [sessions PHP](#), qui n'ont pas été abordées en cours. Vous devrez donc vous auto-former sur ce sujet.

Les pages web doivent être en XHTML et ne pas renvoyer d'erreurs quand elles sont validées par l'[outil du W3C](#). CSS2 doit être utilisé le plus possible et doit aussi ne pas renvoyer d'erreur après validation pour l'[outil du W3C](#).

Enfin, les formulaires devront être validés en utilisant Javascript. Vous avez le droit de vous appuyer sur la bibliothèque [jQuery](#) (non abordée en cours, donc auto-formation) pour simplifier l'écriture du code Javascript.

Pour vérifier la correction des données rentrées par l'utilisateur (est-ce que c'est bien une adresse mail, ...), je vous conseille d'utiliser les [expressions régulières](#). Elles sont utilisables autant en Javascript que en PHP. En outre vous **devez** impérativement utiliser Javascript pour la vérification des données quand c'est possible.

Pour ceux qui veulent développer sous Windows, je recommande l'utilisation de [XAMPP](#) qui permet d'installer un serveur web avec PHP en local.

Déroulement

Le projet sera distribué au cours de la première séance de TP. Les groupes seront constitués en début de séance. Le reste de séance sera consacré à la clarification du sujet et au début de l'implémentation. Une deuxième séance sera ensuite effectuée pour vérifier que tout le monde à bien compris le sujet et pour répondre aux questions techniques qui auront été soulevées par une analyse plus en profondeur du sujet.