

IT103 - Projet programmation web

2015-2016

rgiraud@labri.fr

Introduction

Ce projet de programmation web IT103 s'effectue par binôme. Il a pour but de vous faire développer une application web utilisant l'architecture dynamique côté serveur.

Vous serez notés sur toutes les composantes de l'application :

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

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

- HTML5
- CSS 3
- MySQL
- PHP 5, fonctions uniquement, pas de mécanismes objets

Déroulement

Les groupes sont constitués une fois pour toutes au début de la première séance. Le reste de la 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 a bien compris le sujet et pour répondre aux questions techniques qui auront été soulevées par une première confrontation au sujet.

Rendu

Concernant le rendu, l'archive du projet contenant tous les fichiers est à uploader sur moodle (cours IT103). Le code source du projet est demandé. Celui-ci sera situé dans un répertoire *src*. Pas de rapport pour ce projet, seulement un fichier au format texte brut, nommé *architecture.txt*, décrivant en quelques lignes 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 descriptions doivent être courtes et indiquer clairement quel est le rôle du fichier. Enfin, vous devez fournir un document en texte brut nommé *installation.txt* qui explique la procédure d'installation, si nécessaire, de votre application web (initialisation de la base de données, etc.).

Le tout est à compresser dans une archive nommée *it103-2015-gN.zip* avec *N* étant le numéro de votre groupe attribué sur moodle. Veillez à bien indenter vos fichiers de code PHP, CSS, HTML, etc. Les groupes ne respectant pas ces règles ou celles de la section déploiement seront pénalisés lors de la notation. Le projet est à uploader sur moodle au plus tard le 20/05/2016.

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

Sujet

Vous devez développer une application/un site de paris sportifs en ligne. Le but est d'avoir un site fonctionnel, avec gestion évolutive des cotes en fonction du nombre de paris et qui simule tout le côté gestion de l'argent.

Comme pour les autres sites de paris en ligne, l'utilisateur doit créer un compte, et s'authentifier à chaque connexion. Il peut alors créditer son compte (cette partie sera simulée, l'utilisateur pourra choisir de créditer son compte par un simple formulaire), et ensuite parier sur les paris disponibles.

Vous pouvez simuler quelques paris de base, pour des sports comme le foot, le tennis, ou autres. Avec des résultats possibles comme Victoire Equipe 1 - Victoire Equipe 2, ou plus élaborés en rajoutant la possibilité de match nul ou des options.

Les fonctionnalités sont simples à imaginer, l'objectif est d'avoir un site fonctionnel le plus élaboré qui soit en termes de fonctionnalités mais surtout robuste au niveau de la gestion de la base de données (inscription, déconnexion, mots de passe cryptés). Une interface user-friendly sera également appréciée lors de la notation.

Fonctionnalités

Écran d'accueil/connexion

L'écran d'accueil est un simple écran présentant le site, son fonctionnement, ses règles, les liens vers les différents menus, etc, soyez inventifs. Celui-ci peut suggérer à l'utilisateur de se connecter. Pour cela, ce dernier 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 de son profil. Sinon, un message d'erreur doit expliquer pourquoi la tentative de connexion a échoué.

Écran d'inscription

L'écran d'inscription 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, un pseudo, une adresse. Une fois l'utilisateur inscrit, il doit être redirigé vers l'écran de connexion. Attention, il ne peut y avoir qu'un seul utilisateur ayant une adresse mail donnée. Les vérifications sur les champs du formulaire doivent être effectuées côté client et côté serveur.

L'utilisation d'un script Javascript est possible et recommandée (mais facultative car non vu en cours) pour vérifier avant l'envoi, l'état du formulaire.

Vérifications. Les vérifications à effectuer peuvent être les suivantes : le mot de passe fait au moins 6 caractères, les noms font plus de deux caractères, le département est valide, le mail est dans un format correct (login@domaine.extension), et toutes les données demandées ont été saisies. Si les données saisies par l'utilisateur ne sont pas correctes, elles peuvent être affichées sur fond rouge. Dans ce cas, des messages, également en rouge, peuvent expliquer pourquoi les données ne sont pas correctes.

Écran de profil

Cet écran permet à un utilisateur de visualiser ses informations et éventuellement de les modifier. Pseudo, adresse, rappel des paris en cours, des crédits restants, possibilité d'annulation de pari, etc. Bien sûr la modification des données ne doit être rendue possible que par l'utilisateur lui-même.

Écran des paris

Lorsque l'on clique sur un pari, on doit être transféré vers une page dédiée proposant les options possibles. Un utilisateur doit pouvoir choisir un résultat, par exemple Victoire Equipe 1. Un champ est alors disponible pour entrer le montant que l'on souhaite parier sur ce résultat. Avant de valider le pari, il faut vérifier que l'utilisateur a assez de crédit, et qu'il n'a pas déjà parié sur cet évènement.

Le pari validé, les crédits de l'utilisateur diminuent, le pari apparait dans l'écran de profil de l'utilisateur et la cote est mise à jour (cf. Gestion des cotes).

Fin d'un pari. La fin d'un évènement sportif et du pari associé sera simplifiée en proposant un bouton "Fin du pari" sur la page du pari. En cliquant sur ce bouton, on considère alors l'évènement terminé. On peut alors lire le résultat préalablement écrit dans la base de données lors de la création du pari et ensuite enclencher les paiements éventuels aux joueurs qui ont parié sur le bon résultat.

Gestion des cotes

Comme pour les vrais sites de paris sportifs, on souhaite actualiser les cotes en fonction des tendances. Par exemple, pour le match de foot Equipe 1 (E1) - Equipe 2 (E2). Si on initialise à :

E1 gagne = 2.00 (en pariant 1€ on gagne 2€, soit 1€ de bénéfice)

Match nul = 3.00

E2 = 4.00

Et que 10 personnes parient sur E1, alors la cote doit descendre et les autres doivent remonter, puisque la tendance semble indiquer que les gens croient en une victoire de l'Equipe 1. Il convient alors de borner les cotes de façon à ne pas descendre en dessous de 1.01 (ce qui n'aurait pas d'intérêt) et en faisant attention à ce que toutes les cotes prises individuellement ne soient pas au dessus du nombre de résultats possibles pour un évènement (si 3 cotes au dessus de 3 pour un évènement, on parie sur les 3 et on est sûr de gagner de l'argent). Un modèle d'évolution proposé :

$$Cote_{E1} = 1 + \frac{x_0}{(N * K_{E1})^n} \quad (1)$$

avec x_0 la cote initiale au dessus de 1 pour le E1, soit 1, K_{E1} le ratio de parieurs ayant choisi ce résultat, N , le nombre de résultats possibles (ici 3), et $n \geq 1$ un facteur d'évolution du système. En gros, si tout le monde a parié équitablement (initialisé comme ça), on a $N * K_{E1} = 1$ et $Cote_{E1} = 1 + x_0$. Si tout le monde parie sur E1, $K_{E1} \rightarrow 1$ et donc $Cote_{E1} = 1 + x_0/N^n$ et donc diminue. Enfin si $K_{E1} \rightarrow 0$, la cote augmente (on peut la majorer).

Pour chaque pari, il faut donc stocker le nombre de parieurs sur chaque résultat et à chaque fois qu'un pari est effectué on doit noter une évolution de la cote.

Fonctionnalités bonus

A vous d'être imaginatifs, inspirez-vous des autres sites. Chaque fonctionnalité en plus sera prise en compte et appréciée selon sa difficulté.

Liste non exhaustive :

- Fonction de recherche des paris par mots-clés, par catégories de sport (+++)
- Enregistrer des paris en favoris. (++)
- Upload de photo pour le profil de l'utilisateur (++)
- Liens utiles (aide dépendance au jeu, etc). (+)
- Utilisation de JavaScript pour la vérification des données, champs, etc. (++++)

Indications

Base de données. Bien évidemment, le contenu variable du site (utilisateurs, paris) devra être stocké dans une base de données de type MySQL. Réfléchissez bien en amont à la manière de structurer votre base de données. Celle-ci est en effet plus complexe que celle abordée dans les TP. A l'ENSEIRB, vous avez la possibilité de développer du code PHP et MySQL depuis votre page vvpedago. Votre base de données peut être gérée depuis *phpmyadmin*. A noter qu'une seule base au nom de votre login est disponible et que les droits d'accès ne sont pas paramétrables.

Pour vous y connecter depuis un script php (utilisation de *mysqli*),

```
$mysqli = mysqli_connect('localhost','login','password','login');
if(mysqli_connect_errno($mysqli)){
    echo "Echec lors de la connexion à MySQL : " . mysqli_connect_error();
}
```

avec password, le mot de passe que vous avez reçu par mail le jour de la création de votre page web. Pour récupérer la base de données, si vous travaillez via *phpmyadmin* sur les machines de l'école, vous devez exporter votre base. Pour cela, placez-vous sur localhost et suivez la procédure d'exportation pour récupérer votre *localhost.sql* qui sera à renommer avant rendu (voir section Déploiement).

Bien entendu, le mot de passe des utilisateurs ne doit jamais être stocké en clair dans la base de données. Vous devez donc trouver un mécanisme qui évite cela (fonctions de hachage).

Sessions PHP. 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. Les pages web doivent être en HTML, avec utilisation du CSS2, et ne pas renvoyer d'erreurs quand elles sont validées par l'outil du W3C. Enfin, les formulaires peuvent 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 le bon formalisme des données rentrées par l'utilisateur (est-ce que c'est bien une adresse mail, etc.), je vous conseille d'utiliser les expressions régulières. Elles sont utilisables autant en Javascript qu'en PHP. Les vérifications sur les champs du formulaire doivent se faire évidemment côté client et côté serveur afin d'éviter des problèmes de sécurité. En outre vous pouvez utiliser Javascript pour la vérification des données quand c'est possible. L'utilisation de JavaScript pour la vérification est ici optionnelle, mais en pratique pour vos futures applications, TOUJOURS effectuer la vérification les données! (Autant vous former lors de ce projet).

Bootstrap. Pour ce qui est du design, il est possible d'utiliser un template bootstrap ou autre.

Déploiement

L'archive du projet contenant tous les fichiers est à uploader sur moodle (cours IT103). Avant de rendre votre projet, que vous ayez développé sur vos machines ou celles de l'ENSEIRB, vous devez modifier les droits d'accès à votre base de données MySQL (sinon sanction!).

Votre fichier est donc à renommer : *gN.sql* et la base accédée dans ce fichier doit se nommer *gN*.

Pensez également à modifier dans tous vos fichiers, vos procédures de connexions à la base! Effectivement, lorsque je vais récupérer vos bases, que j'importe sur ma machine, les connexions doivent se faire automatiquement via, `mysqli_connect('localhost','root','gN')`.

Pensez à fournir une base de test suffisante, avec plusieurs paris disponibles, afin que les fonctionnalités puissent être directement testées. A la décompression de l'archive et après import de votre base, un double-clic sur votre page d'accueil (.php) doit permettre de tout tester. Cet aspect sera grandement pris en compte dans le barème de notation.

Conseils

La programmation web demandée est assez vaste mais relativement facile, il s'agit donc beaucoup de s'autoformer. Le travail demandé n'est pas difficile, il suffit d'aller chercher l'information, de trouver des exemples. La question que vous vous posez a forcément déjà été posée en ligne, alors utilisez Google!

Un trou sur l'écriture de telle ou telle fonction : <http://www.labri.fr/perso/falleri/perso/ens/it103/> pour une réponse rapide.

Pour ceux qui veulent développer sur leur machine, je recommande l'utilisation de XAMPP qui permet d'installer un serveur web avec PHP en local avec tout ce qu'il faut! <https://www.apachefriends.org/fr/index.html>