

Applications Web - Mini projet

Objectif

L'objectif de ce mini projet est de mettre en pratique les concepts étudiés en cours :

1. Description de documents avec XHTML
2. Présentation des documents avec CSS
3. Construction de documents interactifs avec Javascript
4. Production automatique (simple) de documents avec PHP
5. Traitement des formulaires

Méthode d'évaluation

L'objectif n'est pas forcément d'obtenir l'application la plus complète, mais de montrer que vous maîtrisez les techniques de base.

Le projet est à rendre par mail à l'adresse (bruno@univ-tln.fr) avec comme sujet `projetI44-VotreNom`. Le message contiendra une archive au format zip (`I44-votreNom.zip`). Cette archive comportera le site web complet et un court document texte (`avancement.txt`) qui reprendra les points du sujet et qui précisera votre avancement. Vous présenterez les résultats de votre projet lors de la dernière séance de TP.

Ceux qui ne peuvent pas avoir accès à une machine avec php doivent me contacter.

La date limite de retour est le 1 mai 2006.

Sujet

Le sujet de ce projet est la gestion simple d'une **petite** classe composée d'élèves et de leurs notes dans différentes matières.

On suppose que chaque élève possède un identifiant unique (un entier) et qu'il est décrit par les informations suivantes : Un nom, un prénom, une date de naissance, une adresse email et l'adresse de sa page web personnelle. La liste des élèves peut donc être représentée par un fichier formaté de la forme suivante :

```
1:Durant:Pierre:20001013:pierre.durant@monecole.fr:pdurant.monecole.fr
2:Dupond:Jean:20011109:jean.dupont@monecole.fr:jdupont.monecole.fr
...
```

De plus, on associe aussi à chaque élève une photographie qui sera stockée dans un fichier `code_de_l_eleve.png` dans le répertoire adapté.

On suppose que chaque matière possède un code unique et un nom (F pour Français, A pour Anglais, M pour Mathématique, SP pour Sciences physiques, SV pour Sciences de la vie). Les matières peuvent donc être décrites dans un fichier formaté de la façon suivante :

```
F:Français
A:Anglais
...
```

Les élèves passent un ou plusieurs contrôles par matière. Ils ont donc une ou plusieurs notes pour chacune. Les notes peuvent être représentées dans un fichier formaté de la façon suivante : (code de l'élève:code de la matière:note)

```
1:F:12
2:F:13
1:A:8
2:A:11
```

Les enseignants de chaque matière donne une appréciation à chaque élève. Chaque matière à son propre répertoire (F, A, M SP, SV) et dans celui-ci l'appréciation d'un élève est dans un fichier dont le nom est `code_de_l_eleve.txt`.

1 Les documents XHTML (obligatoire)

Vous pourrez commencer par construire les modèles de documents XHTML suivants :

1. Un document qui présente la fiche d'un élève dans un fichier XHTML (**dont le nom est de la forme `code.xhtml`**). Ce document affiche toutes les informations dont vous disposez, y compris les notes, les moyennes (que vous calculerez à la main pour le moment), la photographie et l'appréciation pour chaque matière. Vous utiliserez pour cela la richesse de `xhtml` (listes, tableaux, ...) et les structures sémantiquement neutres `div` et `span` associées aux attributs `id` et `class`.
2. un document `codeMatiere.xhtml` qui présente les informations concernant une matière : son nom la liste des notes des élèves dans cette matière et la moyenne.
3. un document `liste.xhtml` qui présente la liste des élèves et qui permet d'accéder à la description détaillée de chacun.
4. un document `saisie.xhtml` qui contient deux formulaires : le premier permet de saisir les informations concernant un élève, le second d'ajouter une note à un élève dans une matière. Dans un premier temps, ces formulaires n'auront aucun effet.

Vos documents seront conformes à la recommandation XHTML et ils comporteront **toutes les métadonnées utiles**.

2 La présentation en CSS (obligatoire)

Vous réaliserez la présentation des documents précédents en utilisant CSS. Vous veillerez en particulier :

1. à faire un bon usage des attributs `id` et `class`.
2. à utiliser des sélecteurs aussi précis que possible.
3. à utiliser les différents algorithmes de placement.

3 L'utilisation de javascript

3.1 Des documents dynamiques (obligatoire)

Vous utiliserez le langage javascript pour rendre dynamique vos documents. En particulier, la liste des élèves devra mettre en valeur l'élève survolé et sa photographie (et seulement celle là) sera affichée.

Vous insérerez un menu dans la page décrivant un élève permettant d'accéder dans l'ordre des codes au premier ou au dernier élève, au suivant ou au précédent. Ce menu sera réalisé avec javascript et CSS, vous expliquerez son fonctionnement en quelques lignes dans le compte rendu.

Vos utiliserez javascript pour rendre votre formulaire plus réactif, les champs seront vérifiés par le client avec l'envoi.

3.2 Un formulaire réactif et dynamique (optionnel)

Dans le formulaire de saisie des notes, il sera possible d'ajouter dynamiquement un champ au formulaire pour saisir autant de notes que l'on veut à l'aide d'un bouton " ajouter une note ".

4 La construction des documents en PHP

Vous utiliserez la documentation de php <http://www.php.net/docs.php>.

Les listes des élèves, des matières et des notes sont décrites dans les fichiers textes présentés précédemment (l'an prochain vous apprendrez à utiliser une base de données, à l'heure actuelle les modifications ne devront être faites que par une personne à la fois).

4.1 La liste des élèves

Ecrire un document php (`liste.php`) qui produit automatiquement le document XHTML qui affiche la liste des élèves (cf. question 1).

Cette page pourra calculer automatiquement des informations supplémentaires : nombre d'élèves, moyenne des notes, ...

4.2 La description d'un élèves

Ecrire un document php qui prend en paramètre le code d'un élève et qui affiche les informations détaillées sur celui-ci (cf question 2).

5 Le formulaire d'ajout d'un film

Ecrivez les documents php `ajoutEleve.php` et `ajoutNote.php` qui permet de traiter les données saisies dans les formulaires précédents.