

Documentation PT4

Partie 1: Introduction

Le projet tuteuré 4 est un projet de développement informatique.

Mon projet consiste avec d'autres collègues de réaliser une application web de gestion des absences.

Je vais développer la partie "saisie des absences". Le but est que l'utilisateur puisse choisir les étudiants qui était absent un jour précis et pouvoir inscrire leurs absences dans une base de données.

J'ai à disposition un cahier des charges fournis par le professeur pour savoir exactement les fonctionnalités que l'application devra comprendre. De plus je dispose également de la structure de la base de données (tables, champs).

Partie 2: Scénarii

L'interface doit ressembler au maximum au cahier des charges.

L'application se lance avec le fichier **index.php**.

Le dossier **app/** contient les fichiers utiles pour l'**application principale**,

Le dossier **Bootstrap/** contient les fichiers **Bootstrap**,

Le dossier **jQuery/** contient le framework **jQuery** ainsi que l'extension **jQueryUI**,

le dossier **css/** et le dossier **js/** contiennent respectivement **les fichiers css** et **les fichiers javascript**,

le fichier **connexion.php** est le fichier qui contient la **connexion PDO à la base de données**.

Sur la partie gauche de la page ce sont **les fonctionnalités supplémentaires** elles proviennent du répertoire **app+/** nous y reviendront un peu plus loin.

Sur la partie centrale:

- **3 boutons radios** pour sélectionner **une filière (Info, Mmi, Geii)**
- Une fois que l'utilisateur a coché une des 3 radios, un nouveau block s'affiche avec 2 autres boutons radios pour choisir **la promotion (1ere année ou 2ème année)**
- Une fois que l'utilisateur a coché une des 2 radios pour la promotion un nouveau block apparait pour cette fois-ci pour choisir **le groupe** recherché(**CM, TD, TP**) toujours avec des boutons radios, pour que le choix soit cohérent:
 - Si on sélectionne TD1 alors ce sera TP1 et TP2 d'affiché
 - Si on sélectionne TD2 alors ce sera TP3 et TP4 d'affiché
- Un fois que l'utilisateur a coché le CM, un TD et un TP, le block des fonctionnalités supplémentaires est remplacé par un **tableau qui affiche les élèves correspondant à la filière, la promo et au groupe choisi**, dans ce tableau se trouve également **2 colonnes de checkbox**, une pour **les retards** et une pour **les absences**, si l'élève a été absent on coche la checkbox dans la ligne de l'élève absent de la colonne absence pareil pour les retards.

- De plus un block s'affiche avec un **select** qui contient **toutes les matières du DUT**.
- Dès que l'utilisateur sélectionne une matière, un **nouveau select** apparaît avec les **professeurs exerçant cette matière**, il y a également une **checkbox** pour savoir si **une évaluation a eu lieu le jour de l'absence**, il y aussi un **calendrier (datepicker)** qui s'affiche pour définir le jour ou l'élève a été absent.
- Lorsqu'on choisi un jour du calendrier un block à droite de la page apparaît qui **affiche les étudiants déjà absents ce jour** qui appartiennent à **la filière et aux groupe TD et TP sélectionné**, Nous avons également un block qui apparaît avec **les horaires potentielles d'absences** dans un **select**.
- L'utilisateur peut maintenant sélectionné **plusieurs horaires d'un coup en maintenant la touche CTRL**, si par exemple l'élève a été absent toute la journée.
- Dès que l'utilisateur a choisi un horaire il peut **valider les absences** avec le bouton qui s'affiche juste en dessous des horaires, si l'utilisateur clique sur le bouton **une alerte lui demandera de confirmer l'enregistrement**.

Lorsque l'utilisateur a cliqué sur le bouton, on l'envoie sur une nouvelle page nommé **ajout_absence.php**. Il y a deux cas de figures pour cette page:

- Si l'utilisateur n'a coché aucune checkbox de retard ou d'absence alors ce script affichera simplement une alerte qui lui précisera qu'il n'a pas coché de case d'absence ou de retard et il sera renvoyer à la page `index.php`,

- Sinon cette page montrera les enregistrements que l'utilisateur vient d'effectuer, cela lui affichera dans un tableau:

- **le module** que l'utilisateur a sélectionné
- **le professeur** que l'utilisateur a sélectionné
- **la date** que l'utilisateur a sélectionné
- **les horaires** que l'utilisateur a sélectionné
- **le nom et prénom des étudiants absents** que l'utilisateur a sélectionné
- **le nom et prénom des étudiants en retard** que l'utilisateur a sélectionné,

Voilà pour l'application globale cependant pour pouvoir utiliser pleinement cette application **il faut avoir beaucoup de données de bases: le étudiants, les professeurs, les modules**. C'est pourquoi sur la page `index.php` nous avons une **partie intitulé fonctionnalités supplémentaires**.

Projet Tuteuré 4

Chaque lien nous dirige vers un fichier **modif_etudiant.php**, **modif_enseignants.php** ou **modif_modules.php**.

modif_etudiant.php:

- Choix de la filière, l'utilisateur choisi entre les **3 filières proposés** grâce à des radios buttons.
- Une fois que l'utilisateur à choisi une filière, on lui demande alors **la promotion (année 1 ou année 2)**
- Une fois la promotion choisi, on affiche dans **un select tous les étudiants** correspondant à la filière et à la promotion et on affiche aussi **un lien pour supprimer l'étudiant qui sera sélectionné**.
- Il y a aussi un lien pour **créer un étudiant** se lien redirige vers un fichier avec un formulaire qui demande **un nom, un prénom, un numéro étudiant, une adresse email, la filière, la promotion, le groupe de l'étudiant**.

modif_enseignants.php

- Sur cette page on affiche dans un **sélect** immédiatement **tous les professeurs**.
- lorsque l'utilisateur sélectionne un professeur on affiche **les modules que le professeur sélectionné exerce** et aussi un lien pour supprimer le professeur qui sera sélectionné.
- Il y a aussi un lien pour **créer un enseignant** se lien redirige vers un fichier avec un formulaire qui demande **un nom, un prénom et un code ADE**.

modif_modules.php

- Sur cette page on affiche dans un **sélect** immédiatement **tous les modules existants**
- lorsque l'utilisateur sélectionne un module on **affiche le professeur qui est en charge de ce module** et aussi un lien pour supprimer le module qui sera sélectionné.
- Il y a aussi un lien pour **créer un module** se lien redirige vers un fichier avec un formulaire qui demande **un code PPN, un nom de module, un code enseignant, un code Scodoc, une unité d'enseignement(UE), le nombres d'heures en CM, le nombres d'heures en TD et le nombres d'heures en TP**.

Partie 3: Fonctionnement du programme

1)Base de données

Pour que cette application fonctionne correctement, il nous faut au départ une base de données structurée, voici les tables avec lesquelles je vais travailler:

- **absence** (cette table permet de stocker les absences)
- **enseignant** (cette table contient tous les enseignants)
- **étudiant** (cette table contient tous les étudiants)
- **groupe** (cette table contient les groupes)
- **module** (cette table stocke toutes les matières)

Projet Tuteuré 4

Ces tables étaient dans le cahier des charges avec les champs ainsi que leurs types.

Cependant j'ai rajouté **deux données importantes** pour simplifier l'application:

- l'ajout de la nouvelle table nommée **retard** cette table possède exactement les **mêmes champs que la table absence** sauf qu'elle **sauvegarde les retards**.
- l'ajout d'un champ de nom '**etufiliere**' (**varchar 5**) dans la table **etudiant** qui permet de connaître **la filière de l'étudiant (mmi, info ou geii)** qui permet ainsi de les trier plus facilement.

II) Technologies utilisés

Pour la base de données nous avons utilisés **phpMyAdmin** comme application de gestion de base de données car il utilise le **SGBD MySQL**, qui est celui que nous avons vu en cours.

Comme demandé dans le cahier des charges nous avons utilisé le **langage PHP**, pour moderniser le site et le rendre dynamique nous avons utilisés les **CSS de Bootstrap**, le **framework jQuery** ainsi que **jQueryUI**.

J'utilise l'éditeur de texte **Sublime Text 2**.

Pour le débogage j'utilise le module **Firebug** de Mozilla.

III) Fonctionnement du programme

index.php

Cette page représente la page d'accueil de l'application chaque page consultable par l'utilisateur possède un lien vers index.php

Cette page est la page la plus dynamique du site elle est liée par le framework jQuery et une page js nommée js/index.js

- Lorsque l'utilisateur clique sur une filière on affiche le block 'promotion', c'est le script index.js qui détecte à chaque fois que ce soit un choix de radios buttons ou une sélection dans un sélect on affiche le block suivant.

- Dans le block de sélection de groupe (CM,TD,TP), pour que cela soit cohérent :
 - si on choisit la radio TD1 on peut choisir uniquement TP1 et TP2, à l'inverse
 - si on choisit la radio TD2 on peut choisir uniquement TP3 et TP4

Pour que cela fonctionne on lance une fonction change dans la div contenant les radios TD1 et TD2, cette fonction change se déclenche à chaque fois qu'on change la radio sélectionné. Dans cette fonction change, j'effectue un callback d'une fonction nommée **affichageGroupeTP()** qui elle teste quelle radio entre TD1 et TD2 est cochée et affiche en conséquence dans la div des TP les radios correspondantes.

- Une fois que le groupe (CM, TD, TP) est sélectionné la fonction **affichage_etudiant()** est appelée, qui fait une requête ajax et envoie au script nommé **/app/liste_etudiants.php** la filière, la promo, le groupe CM, le groupe TD, le groupe TP qui ont été sélectionnés. Ce script lui effectue une requête qui **récupère tous les étudiants dans un tableau** qui appartient à la filière, à la promo et au groupe sélectionné, dans **ce tableau on ne conserve uniquement le nom et prénom** de l'étudiant et **on encode le tableau en JSON**.

On affiche les données du JSON dans un tableau qui sera affiché à gauche de la page index.php

Projet Tuteuré 4

- Une fois les groupes sélectionnées on a l'affichage dans un sélect de tous les modules existant grâce au script **app/matieres.php** qui lance une requête qui récupère le nom et le code de chaque matière. **la value d'une matière dans le sélect correspond à son code ppn.**

- Lorsqu'on sélectionne une matière, une **requête ajax** est faite qui envoie au script **app/professeurs.php** uniquement l'id du module. Ce script effectue une requête qui récupère d'abord **l'id du professeur dans la table module** et ensuite **récupère le nom et le prénom du professeurs** grâce à l'id récupéré dans la **table enseignant** on stocke l'enseignant dans un tableau qu'on encode en **json**. On affiche ensuite ce professeur dans **le sélect en dessous avec comme value l'id du professeur.**

- Un calendrier s'affiche a présent il s'agit d'un **datepicker**. le datepicker est disponible grâce au package jQueryUI.

- Le datepicker est créer grâce au fichier **js/datepickerfr.js** où on peut fixer nos paramètres par exemple le format de la date, la langue. Par défaut le datepicker ne stocke pas la date que l'on sélectionne, du coup lorsqu'on choisit une date du datepicker j'ai créer un **champ caché qui conservera la date sélectionnée.**

- De surcroît lorsqu'on clique sur une date, on appelle la fonction **affichage_absent()** qui effectue une requête ajax et envoie au script **app/liste_absents.php** :

- **la date** contenue dans le champ caché
- **la filière** qu'on a choisi
- **la promotion** choisi,
- **le groupe TP et TD**

Ce script effectue 4 requêtes grâce à une boucle foreach qui contient les horaires de cours 8h30-10h30, 10h30-12h30, 14h00-16h00, 16h00-18h00.

- Chaque requête va récupérer dans **la table etudiant le nom et prénom de chaque étudiant** qui se trouve dans la **table absence** à la **date indiqué**, à l'**horaire indiqué** qui appartiennent également à **la filière sélectionné**, à **la bonne promo** et au **groupe TD et TP sélectionné.**

Tout ces noms sont stockés dans un tableau **\$absent**, si aucun étudiant lors d'un créneau n'a été absent alors on met dans **\$absent** un **"/"**.

Puis on encode le tableau en **JSON**.

Enfin dès que la requête ajax est finie, on affiche tout dans un tableau html ayant pour id **#table_absent**.

- On a maintenant la possibilité de choisir les créneaux des absences qui se situe dans un sélect

- Dès qu'un créneau aura été sélectionner le bouton valider permettra de soumettre le formulaire

- Lorsqu'on appuie sur valider nous sommes rediriger vers le fichier **ajout_absence.php**
Ce fichier teste si des checkbox ont été coché dans le tableau de gauche,
si aucune checkbox n'a été coché, on le redirige vers le script index.php.

Ce fichier récupère dans un premier temps toutes les données du formulaire :

- **le code ppn** du module,
- **l'id du professeur,**

Projet Tuteuré 4

- **la date** sélectionné,
- **le ou les créneau(x)** sélectionné(s),

- On vérifie ensuite si des **checkbox de la colonne absences on été cochées**.

On alors une **double boucle imbriquées**.

Pour chaque étudiant absent et pour chaque horaires sélectionnées on insère un enregistrement dans la table absence, on stocke dans un tableau son nom et son prénom. On fait la même requête pour les retards avec aussi les 2 boucles imbriquées.

Une fois ces requêtes effectués on affiche dans un tableau html les données importantes à savoir:

- **le module** qui avait été sélectionné,
- **le professeur** qui avait été sélectionné,
- **la date** et les **créneaux**,
- **le nom** et prénom des étudiant **absent**,
- **le nom** et prénom des étudiant **en retard**,

IV) Conclusion

J'ai considéré ce projet comme un dernier test dans la programmation web, j'ai découvert encore une fois un tas de choses notamment sur le format de données JSON et ainsi que les requêtes AJAX. Enormément de connaissances vues en cours ont été utilisés et grâce à ce projet, elles ont été très bien assimilés.

Noga Lucas