

Rapport du projet de programmation web : Un site pour gérer les points associatifs.

Loïc DUBARD, Quentin Japhet, Aurélien Inacio, Oscar Potier

13 mai 2019

Table des matières

1	Approches importantes	3
1.1	Structure du site	3
1.2	Structure de la base de donnée	3
1.3	Fonctionnalités pour les élèves	3
1.4	Fonctionnalités pour les membres du BDE	4
1.5	Fonctionnalités pour les présidents d'associations	4
1.6	La récupération de mot de passe	4
2	Répartition des rôles	5
3	Problèmes rencontrés	5
3.1	OAuth et base de donnée distante	5
3.2	Envoi d'un mail	5
3.3	Design et graphisme du site	5

Introduction

Le but premier est de répondre au besoin du Bureau des élèves de l'ENSIIE qui consiste en un moyen de centraliser et de faciliter la constitution du classement des participations aux associations.

Ce classement doit être exportable dans un fichier csv pour des manipulations externes des coefficients des associations par le BDE.

1 Approches importantes

1.1 Structure du site

Nous avons donc pensé le site comme une plateforme épurée où la seule page accessible sans connexion est "index.php".

Il y a donc 3 type d'utilisateurs : les **élèves**, les **membres du bde** et les **présidents**.

Chaque type d'utilisateur a accès à des fonctionnalités différentes.

Un menu simple couronne la page web avec les onglets suivants :

- Accueil
- Connexion (ou Déconnexion de <Pseudo>)
- Mon Profil
- Mes Points
- Gestion BDE : uniquement accessible aux membres du BDE.
- Gestion Président : uniquement accessible si président d'une association.

1.2 Structure de la base de donnée

On a choisi d'utiliser une base de données découpée en 5 tables :

- users(id_user, firstname, lastname, pseudo, year, password, mail, bde, president) : contient tous les utilisateurs et leurs informations personnelles.
- associations(id_asso, name, president, coeff_asso) : contient la liste des associations ainsi que l'id du président.
- events(id_event, id_asso, name, date_ev, description, coeff_event) : contient la liste des événements et l'id de l'association à laquelle ils appartiennent.
- score(id_user, id_event, notation) : contient la liste des participations des utilisateurs à chaque événement.
- pointsassos(id_user, id_asso, notation, proposition) : contient la liste des points associatifs pour chaque élève pour chaque association.

Nous avons aussi créer les vues suivantes qui simplifient les requêtes utilisées dans le code :

- pointsassos_prop(id_user, id_asso, moyenne) : calcul la proposition de note d'un élève dans une association en faisant la moyenne de ses notes dans chaque association.
- leaderboard(id_user, moyenne) : calcul les points totaux de tous les élèves pour le semestre.

1.3 Fonctionnalités pour les élèves

Tous les élèves enregistrés ont accès à "**eleves.php**" qui permet aux élèves de consulter leurs participations aux événement des différentes associations et leurs points dans chaque association. Sur cette page ils peuvent aussi proclamer leurs participations aux événements pour chaque association (ils choisissent d'abord l'association par un simple formulaire constitué d'un select puis un des événements correspondants par un deuxième formulaire similaire)

Ils ont aussi accès à "**profil.php**" sur laquelle ils peuvent consulter les informations sur leur compte, modifier leur mot de passe par un formulaire classique à 3 champs (le mot de passe est hashé avec l'algorithme bcrypt avant d'être stocké dans la base de donnée). Le même type de

formulaire est utilisé pour modifier (ou en mettre une si il n'y en avait pas) l'adresse mail de récupération du mot de passe.

1.4 Fonctionnalités pour les membres du BDE

Les membres du BDE, sur la page "**bde.php**" peuvent choisir parmi deux menus de sélections : les promos et les associations des élèves à afficher. Ils peuvent ensuite afficher un tableau récapitulant ces données ou exporter ces données au format csv.

Un select déroulant permet de choisir l'ordre d'affichage du tableau : par nom ou pseudo ou points associatifs (si une seule association est sélectionnée).

Cette page est majoritairement faite en JS.

Il faudrait rajouter la possibilité pour un membre du bde de rajouter le rôle de BDE à un élève lambda ou de le supprimer.

1.5 Fonctionnalités pour les présidents d'associations

Les présidents d'associations peuvent sur la page "**president.php**" tout d'abord choisir dans un petit formulaire php constitué d'un select une des associations dont ils sont président.

Il vont ensuite pouvoir consulter la liste des événements correspondants à cette association dans un tableau où chaque ligne est un formulaire et chaque case un input de type text (ou textarea) sauf la dernière colonne qui contient trois boutons : modifier, détails et supprimer.

Ces boutons envoient des requêtes différentes qui sont interprétées par le php et permettent respectivement de supprimer un événement, envoyer les modifications sur l'événement et afficher les détails des gens ayant participé à un événement.

Il faut préciser que la suppression d'un événement dans la base de donnée ne fonctionne pas si il y a encore des gens qui y ont participé. Il vaut mieux donc supprimer ces gens avant la suppression de l'événement.

Le bouton "Détails" redirige vers la page "**event.php**" en définissant la variable de session "\$_SESSION['event']" qui contient l'id de l'événement concerné. On pourra, exactement de la même manière qu'avec les événements sur la page précédente, modifier la note de participation d'un élève à cet événement ou le supprimer de l'événement.

On pourra aussi, grâce à l'input de type texte avec autosuggestions en JS, rajouter un élève à un événement sur "**event.php**" ou rajouter un événement à une association sur "**president.php**".

Un dernier input avec autosuggestion permet de modifier une variable de session "\$_SESSION['userpassation']" et d'envoyer l'utilisateur vers la page "**passation.php**" pour qu'il confirme l'abandon de son rôle de président pour l'association choisie et qu'il le donne à un autre élève.

1.6 La récupération de mot de passe

Le système de récupération de mot de passe utilisé ici est constitué d'un premier formulaire sur **forgotten.php** qui prend le login (donc pseudo) et l'email de récupération, puis envoie à la

page **mail.php** un code de 10 chiffres aléatoires qui l'envoie par mail puis renvoie sur la page **recup.php** qui contient un simple formulaire demandant le code envoyé (le nombre d'essais maximum pour la session est de 3, ensuite il faut réenvoyer un nouveau code).

Si le code entré est le bon, la page change et demande maintenant un nouveau mot de passe et une confirmation de ce mot de passe. On est ensuite redirigé vers **authentification.php**.

2 Répartition des rôles

Le projet est versionné grâce à git et hébergé sur github. Nous avons décidé d'utiliser un Trello pour organiser les tâches à effectuer et par qui elles doivent l'être.

La répartition du travail a donc été originalement faite comme présentée ci-dessous :

- Loïc Dubard : accueil et menu (+css responsive du menu), système d'authentification et de récupération de mot de passe, fonctionnalités des présidents d'associations (président.php), écriture du rapport
- Quentin Japhet s'occupe des fonctionnalités du BDE (bde.php), les input avec suggestion automatique en JS et de l'envoi des codes de récupération de mot de passe par mail.
- Aurélien Inacio : eleve.php, correction du rapport
- Oscar Potier : CSS et profil.php

3 Problèmes rencontrés

3.1 OAuth et base de donnée distante

Le principe de base était d'utiliser le protocole OAuth pour se connecter directement à la base de donnée d'Arise, et en recréer une en local avec uniquement les nouvelles données. Nous n'avons pas réussi à l'implémenter avec docker, mais une fois en production sur les serveurs d'arise, ce sera facile de le faire.

3.2 Envoi d'un mail

La fonction mail() de php ne fonctionnant pas sur docker (pas de Mail Transfert Agent) nous avons choisi de recréer une fonction php qui fait une requête au serveur smtp de google avec un compte créé pour l'occasion et le mot de passe stocké en clair côté serveur pour envoyer un mail.

Si le site venait à être mis en production, nous utiliserons la fonction mail() de base du php.

3.3 Design et graphisme du site

Côté design, le site n'est pas un exemple. En effet, nous n'avons pas eu d'idées révolutionnaires pour rendre notre produit attrayant.