

Rapport du projet web

Application gestion des parrainages

2015-2016

Elaboré par : Anaïs GENTY - Sandra MICHAUT - Arielle MAROUANI - Hafsa
RIAHY (groupe J)

Encadré par : Monsieur Thomas COMES

Table des matières

1	Introduction générale	2
2	Structure de l'application	2
3	Répartition du travail	7
4	Problèmes rencontrés et solutions apportées	7
5	Conclusion générale	8

1 Introduction générale

Le projet web consiste en l'élaboration d'une application web. Plusieurs sujets sont proposés et tous obligent la présence d'un certain nombre de fonctionnalités dont l'authentification ou encore un profil utilisateur éditable.

Parmi les sujets proposés, nous avons retenu le sujet traitant la gestion des parrainages inter promo de l'ENSIIE. Ce projet a pour objectif de permettre aux élèves de l'école de gérer leurs relations de parrainages inter promo à l'école.

Pour cela, notre groupe a utilisé l'IDE PHPStrom pour créer un projet regroupant nos fichiers codés en PHP (fonctions), HTML (affichage des pages et formulaires), JS (validation des formulaires), CSS (apparence de l'application) et SQL (création et gestion de la base de données) ainsi que d'autres documents tels qu'un état de l'avancement du projet ou un schéma relationnel de la base de données. Cet IDE nous permet également de tester notre application sur un serveur local préalablement configuré et avec la base de données intégrée.

Après avoir déterminé un plan initial du site, les différentes fonctionnalités nécessaires à son bon fonctionnement et un schéma de la base de données, nous nous réparties ces différentes tâches et avons commencé à implémenter les solutions correspondantes. Une fois les premiers fichiers finis, des tests nous ont permis de déterminer s'ils fonctionnaient ou non puis de les corriger jusqu'à arriver à un résultat final correct et fonctionnel renvoyant ce que nous souhaitions. La structure de l'application sera explicitée plus en détails dans la section correspondante. Il en va de même pour la répartition du travail ainsi que les problèmes rencontrés et les solutions qui leur ont été trouvées.

2 Structure de l'application

L'application se décompose en un certain nombre de fonctionnalités, et les fichiers de code qui leurs sont associés. Les fonctionnalités disponibles diffèrent selon le type de l'utilisateur (visiteur, élève inscrit ou administrateur). Un utilisateur déconnecté sera toujours considéré comme un visiteur (accès aux fonctionnalités de recherche et visualisation de l'arbre) et un utilisateur connecté pourra être soit un simple élève ou ancien élève soit un administrateur (attribut *admins* dans la table *eleve* de la base de données soit à 'y' soit à 'n'). Cependant, un utilisateur s'inscrivant via l'application n'aura jamais la possibilité de devenir administrateur (attribut *admins* automatiquement mis à 'n').

- La première fonctionnalité de l'application est la création d'un profil. Les fichiers utilisés pour la création d'un profil sont les suivants :

- *creer_profil1.php* (1)
- *creation_donnees_perso.php* (2)
- *profil_vision.php* (3)
- *profil_modif.php* (4)

L'utilisateur, une fois sur la page d'accueil (fichier *accueil.php*), clique sur le bouton *Sign up* et se retrouve sur la page codée par (1). Il a alors la possibilité de s'inscrire en donnant un nombre minimal d'informations (nom, prénom, mail et mot de passe). Une fois les données rentrées, il les valide et elles sont transmises au fichier (2) via la méthode *get* du formulaire HTML (la méthode *post* ne fonctionnant pas avec PHPStrom). Ainsi, il est connecté et on lui donne son login de la forme : nom.prenom. Il peut alors choisir de finaliser son inscription en allant vers (4). Ou il peut simplement le visionner via (3).

- La deuxième fonctionnalité de l'application est l'édition du profil. Les fichiers utilisés pour l'édition du profil sont les suivants :

- *validation_modif.php* (1)
- *visualisation_profil.php* (2)
- *profil_modif.php* (3)
- *profil_vision.php* (4)

Lorsque l'utilisateur est un membre et qu'il est connecté, il peut voir son profil à l'aide du lien *profil* en haut de chaque page et éditer à partir de la page d'accueil ou de la page de visualisation. En cliquant sur le lien *profil* (en haut de chaque page), il arrive sur la page (2) qui permet de visualiser toutes les informations qu'il a rentrées dans la base de données à l'aide de requêtes SQL données dans le fichier (4). S'il souhaite modifier ces données, il devra cliquer sur le bouton *modifier* pour arriver sur la page (3). Les données remplies dans la base de données sont déjà préremplies dans les cases du formulaire. Ainsi, il peut visualiser les champs déjà remplis et compléter / modifier / supprimer ce qu'il veut, sauf son nom et prénom s'il n'est pas administrateur (la qualité d'administrateur, le login et le mot de passe ne

comptent pas dans les informations modifiables et visionnalisables). Si le membre est un administrateur, il a la possibilité sur la page d'accueil de modifier son propre profil mais aussi le profil d'un membre dont il renseignera le login. Une vérification est faite pour la validité du login donné. Il est alors dirigé sur la page (3) pour faire les modifications. Un administrateur ne peut visualiser les informations d'un membre uniquement par ce moyen. Enfin, une fois que les informations ont été saisies, l'utilisateur envoie le formulaire sur la page (1) qui permettra de réaliser les changements, et d'afficher les types d'informations modifiés.

- La troisième fonctionnalité de l'application est la recherche de personne dans l'arbre ou la recherche de lien de parenté entre deux personnes. Les fichiers utilisés pour ces fonctionnalités de recherche sont les suivants :

- *form_recherche_lien_de_parente.php* (1)
- *form_recherche_personne.php* (2)
- *recherche_lien_de_parente.php* (3)
- *recherche_personne.php* (4)
- *resultat_lien_parente.php* (5)
- *resultat_personne.php* (6)

L'utilisateur, une fois sur la page d'accueil (fichier *accueil.php*), clique sur le lien menant vers la recherche de personne ou de lien de parenté et se retrouve sur la page codée par (1) ou (2). Il s'agit de formulaires HTML que l'utilisateur doit remplir (les données remplies par l'utilisateur sont validées par les fonctions contenues dans le fichier *checkForm.js*) et qui renvoient vers les fichiers (3) et (4). Le résultat est alors affiché sur les pages (5) et (6).

- La quatrième fonctionnalité de l'application est l'ajout de lien de parenté. Les fichiers utilisés pour l'ajout de lien sont les suivants :

- *add_lien.php* (1)
- *ajout_lien.php* (2)
- *profil_vision.php* (3)

L'utilisateur, une fois sur la page d'accueil (fichier *accueil.php*), clique sur

le lien menant vers l'ajout de lien de parenté et se retrouve sur la page codée par (1). Il est invité à remplir un formulaire contenant les informations sur la personne avec qui il souhaite créer un lien. Il est ensuite renvoyé sur la page (2) où il est invité à écrire un mail pour prévenir cette personne de l'ajout du lien si elle était déjà inscrite ou l'inviter à s'inscrire (si elle n'est pas inscrite, l'ajout du lien n'est pas possible). L'ajout du lien est permis grâce à la fonction `ajoutLien` du fichier (3).

- La cinquième fonctionnalité de l'application est la visualisation de l'arbre généalogique regroupant toutes les personnes inscrites sur l'application et leurs liens de parenté. Les fichiers utilisés pour l'affichage de l'arbre sont les suivants :

- *affichage_arbre.php* (1)

L'utilisateur peut visualiser tous les liens validés dans la base de données grâce à la page (1) qui appelle une fonction de recherche de tous les liens validés qui se trouve dans le fichier *profil_vision.php*.

Ces fonctionnalités communiquent les unes avec les autres grâce à certaines fonctions que l'on peut retrouver dans *mise_en_page.php*. En effet, toutes les pages de l'application possèdent un lien vers le formulaire de contact avec l'administrateur ainsi qu'un lien de retour à l'accueil. De plus, la fonction *enTete* permet d'afficher en haut de page des liens de connexion et d'inscription lorsque l'utilisateur est déconnecté et des liens de déconnexion et menant vers le profil lorsqu'il est connecté. De plus, la page d'accueil a un menu permettant d'accéder à différentes fonctionnalités en fonction de l'état connecté ou déconnecté de l'utilisateur.

Ces fonctionnalités permettent ainsi de créer un ensemble cohérent grâce aux fichiers suivants :

- Les fichiers permettant de créer et gérer la base de données :

- *create_eleve.sql* (1)
- *fill_eleve.sql/csv* (2)
- *fill_assoc.sql/csv* (3)
- *fill_assoc_eleve.sql/csv* (4)
- *fill_liens.sql/csv* (5)

Le fichier (1) regroupe les commandes SQL permettant de créer la base de données et ses tables et les fichiers (2) à (5) permettent de remplir ces tables.

– Les fichiers permettant de gérer l'apparence de l'application :

- *style.css* (1)
- *style_email.css* (2)
- *style_form.css* (3)

Le fichier (1) gère l'apparence de l'ensemble de l'application alors que le fichier (2) gère celle de la page de contact administrateur et le (3) celle des pages de recherche (formulaires).

Avec un temps supplémentaire, l'idée aurait été d'utiliser le thème gratuit de Bootstrap, *Cerulean*, que l'on peut trouver sur Bootswatch pour faire le design de l'application.

– Les fichiers permettant la liaison entre les différentes fonctionnalités de l'application :

- *mise_en_page.php* (1)
- *accueil.php* (2)

Le fichier (1) regroupe différentes fonctions de base réutilisées dans la plupart des pages et le fichier (2) correspond à la page d'accueil de l'application qui contient des liens vers l'ensemble des fonctionnalités de l'application.

– Et d'autres fichiers :

- *checkForm.js* pour vérifier la validité des formulaires
- *email.php* pour contacter l'administrateur
- *config(_s).php* contenant les variables de session nécessaires au bon fonctionnement de l'application

3 Répartition du travail

Fonctionnalités de l'application	Description	Contributeurs
Inscription	Création d'un profil utilisateur sur l'application	Anaïs GENTY
Gestion du profil	Edition du profil utilisateur	Sandra MICHAUT
Recherche	Recherche d'une personne dans l'arbre ou d'un lien de parenté entre deux personnes	Sandra MICHAUT, Anaïs GENTY
Affichage	Visualisation de l'arbre	Sandra MICHAUT
Ajout de liens	Création de liens de parenté dans l'arbre par l'utilisateur	Anaïs GENTY
Création de la base de données	Création et gestion de la base de données ; requêtes SQL	Arielle MAROUANI, Anaïs GENTY, Sandra MICHAUT
Apparence	Réalisation de fichiers <i>.css</i> pour l'apparence de l'application	Sandra MICHAUT, Hafsa RIAHI
Connexion	Connecter l'utilisateur et ajouter les fonctionnalités en fonction du type d'utilisateur	Sandra MICHAUT
Autres : - page d'accueil - formulaires - validation de formulaires JS	Fichier <i>accueil.php</i> Fichiers <i>email.php</i> et <i>form_recherche_X.php</i> Fichier <i>checkForm.js</i>	Sandra MICHAUT, Anaïs GENTY Hafsa RIAHI, Anaïs GENTY Anaïs GENTY

TABLE 1 – Répartition des fonctionnalités de l'application entre les membres des groupes

Le travail a été réparti en fonction des différentes fonctionnalités et autres tâches nécessaires à la conception de l'application et à son bon fonctionnement. Certaines fonctionnalités, étant assez lourdes ou difficiles à coder, ont été réparties entre deux personnes permettant ainsi également de vérifier le travail de l'autre et de réduire le risque d'erreurs.

4 Problèmes rencontrés et solutions apportées

Les principaux problèmes rencontrés étaient souvent d'ordre d'implémentation dans un nouveau langage (JavaScript) ou de façon à produire un résultat bien spécifique (affichage de l'arbre). Pour ce qui est de l'affichage de l'arbre, un problème rencontré a été le fait qu'il pouvait exister des liens en losange (un parrain a 2 filleuls qui ont le même

filieu) ce qui empêchait l'utilisation des listes en CSS. De plus l'utilisation du package *Image_GraphViz* a posé problème. Ainsi l'affichage est rudimentaire.

Nous avons également rencontré des problèmes au niveau de l'utilisation de l'IDE PHPStorm qui ne nous était pas familier, par exemple avec l'impossibilité d'utiliser la méthode `post` pour les formulaires HTML. Nous avons donc dû utiliser la méthode `get` malgré le fait qu'elle soit moins sûre puisque les données envoyées du formulaire se retrouvent dans la barre adresse ce qui pose surtout problème pour les mots de passe.

L'envoi d'emails depuis l'application n'apparaissait pas être évident non plus mais nous avons résolu ce problème grâce à l'action *mailto* du formulaire HTML. Cependant, le fonctionnement n'est pas optimal car cette méthode ne semble pas marcher dans tous les cas (dépend du navigateur, de sa version, etc).

Il n'était également pas évident de voir comment créer la base de données de façon à gérer les liens de parenté pour obtenir une structure pouvant être transposée en arbre pour table liens mais une fois le schéma de la base de données réalisé, le codage SQL est venu assez naturellement sans poser autant de problèmes qu'on n'aurait pu en attendre.

Un autre problème concernait la vérification des données rentrées par l'utilisateur dans les formulaires. La solution consistait à créer une fonction de validation en JavaScript pour le formulaire mais également à utiliser l'attribut *pattern* des formulaires HTML permettant de limiter le format des données rentrées par l'utilisateur.

De plus, dans l'esprit d'essayer de garantir un code propre, nous avons créé un fichier *mise_en_page.php* contenant des fonctions de mise en page utiles pour éviter l'oubli de balises HTML ou des erreurs de même type mais aussi pratiques car elles évitent d'avoir à récrire les mêmes lignes de code dans plusieurs fichiers différents. Dans ce même esprit, nous avons également fait usage des fonctionnalités de l'IDE PHPStorm qui nous permettait de repérer nos erreurs (grâce à *Inspect Code* notamment).

5 Conclusion générale

Le but de ce projet était de créer une application permettant aux élèves de l'école de gérer un arbre de parenté des liens de parrainages inter promo de l'ENSIIE.

Ainsi, le site s'organise de la façon suivante, avec une page d'accueil qui permet d'accéder aux différentes fonctionnalités de l'application. Initialement déconnecté, l'utilisateur arrivant sur la page d'accueil peut s'inscrire pour la première fois ou se connecter s'il est déjà inscrit. Tant qu'il est déconnecté, l'utilisateur n'a accès qu'aux pages de recherche et de visualisation de l'arbre. Une fois connecté, il a la possibilité d'accéder à son profil et de l'éditer mais aussi d'ajouter des liens de parrainages le concernant. Cependant,

si son parrain (sa marraine) ou son filleul (sa filleule) ne sont pas encore inscrits sur l'application, le lien ne pourra être créé. L'utilisateur a alors la possibilité d'envoyer un email à la personne en question pour l'inviter à s'inscrire sur l'application.

Les élèves s'inscrivent sur l'application puis différentes fonctionnalités leur sont offertes : la recherche d'une personne dans l'arbre, la recherche d'un lien de parenté entre deux personnes, la visualisation de l'arbre, l'ajout de lien si les deux personnes sont inscrites et la mise à jour de leur profil.

L'application reste cependant accessible aux personnes non inscrites, seulement avec des fonctionnalités réduites. Les visiteurs n'ont accès qu'à la visualisation de l'arbre et aux fonctionnalités de recherche.

Les membres de notre groupe possèdent tous un rôle de superuser et disposent, en plus des fonctionnalités accessibles par les autres utilisateurs, mettre à jour ou supprimer des profils d'autres utilisateurs ou modifier certaines tables dans la base de données non accessibles par les autres utilisateurs.