Romain THEODET CMI INFO
Christophe CARMAGNAC

Rapport INFO 405 BDD

Préambule

Au fil de ce TP, nous avons essayé de construire un site fonctionnel, sécurisé, et le plus robuste possible face aux attaques XSS, aux injections SQL et à diverses attaques (notamment l'exécution de code PHP via les images, comme donner *http://os-vps418.infomaniak.ch/etu_info/info_1_gr_1/?page=deconnection* comme URL d'image de sujet par exemple).
Nous effectuons ainsi quelques vérifications supplémentaires côté serveur, comme la taille minimale et maximale du login et du titre des sujets.
Nous avons de plus essayé de penser à l'avenir, et de faciliter l'ajout de nouvelles fonctionnalités sans avoir à convertir l'intégralité des données de la base de donnée. Nous allons donc vous détailler certains choix d'implémentation de notre base de donnée.

Tables BDD

– Utilisateur

La pièce maîtresse de ce projet est l'utilisateur. Chaque utilisateur est unique par son login et par son id, qui lui est attribué de manière automatique. Nous souhaitions en effet dissocier le nom d'utilisateur et l'identifiant interne, afin de laisser la possibilité théorique d'un changement de nom d'utilisateur, ou de plusieurs utilisateurs ayant le même nom sans trop de conflits. Autre avantage à avoir un id : l'uniformisation des tables, toutes désignables par un id de même type (entier).
À des fins de sécurité, nous stockons dans la base de donnée un sel et le hash de l'union de ce sel et du mot de passe, évitant ainsi de garder en clair le mot de passe de l'utilisateur.

– Specialite

Afin d'éviter les doublons, il n'existera qu'un nombre limité de spécialités, avec un nom associé et possiblement d'autres attributs à l'avenir. Cette table est composée d'un id et du nom de la spécialité.

– Competence

Chaque utilisateur est relié à un certain nombre de spécialités via une table Competence, qui associe seulement les id.

– Action

Chaque action a un nom et un nombre de points associé afin de ne pas hard-coder les valeurs de chaque action. Les actions sont fonctionnelles ainsi que le système de points, contrairement à la dernière démonstration.

– UtilisateurAction

Nous avons ajouté une table associant une action à un utilisateur. Cette table n'est pas vitale (nous aurions juste pu ajouter les points à l'utilisateur), mais elle permet plus de traçabilité et surtout permettrait en théorie de pouvoir modifier la valeur d'une action et de pouvoir recalculer les points de chaque utilisateur, impossible sans cette table. Elle contient aussi la date de l'action.

– Sujet

Chaque sujet a un identifiant unique, un titre limité à 256 caractères (en pratique nous imposons entre 4 et 255 caractères), une description / contenu de taille illimitée, un chemin vers une image, une date de création ainsi qu'une référence vers l'identifiant de son créateur.

– Favoris

Une table associant les id des utilisateurs et les id des sujets.

– Tag

Table listant tous les tags existants.

– SujetTag

Table associant les tags aux sujets.

– Message

Chaque message a l'identifiant de sa source et de sa destination, laissant donc l'éventualité de pouvoir envoyer des messages à d'autres utilisateurs sans devoir changer trop d'informations dans la base de donnée, ainsi qu'un timestamp précisant la date et l'heure de l'émission d'un message. Dans notre implémentation, aucun message n'est vraiment supprimé, seule une variable booléenne est changée. Le site n'est plus vraiment aux normes GDPR mais au prix d'une meilleure traçabilité permettant par exemple de résoudre des problèmes de harcèlement ou d'injures.

– Groupe

Les utilisateurs peuvent créer des groupes dont ils sont le propriétaire. Chaque utilisateur peut voir tous les groupes et demander à les rejoindre. Le créateur choisit de les accepter dans son groupe ou non. Notre groupe est donc composé d'un id, d'un nom (entre 3 et 127 caractères) et de l'id du créateur.

– UtilisateurGroupe

Table associant les utilisateurs et les groupes, avec un booléen validation pour savoir si l'utilisateur est en attente de validation ou s'il a bien rejoint le groupe.

Modèle relationnel

Utilisateur(id, login, password, salt, dateNaissance, niveauSql, description, points) ;
Specialite(id, name) ;
Competence($userId, $specialiteId) ;
Action(name, reward) ;
UtilisateurAction(userId, actionName, creationDate) ;

Sujet(id, title, description, picturePath, $creatorId) ;
Favoris($userId, $sujetId) ;

Tag(nameTag) ;
SujetTag($sujetId, $tagName) ;

Message(id, content, $sujetIdDestination, $senderId, creationDate, deleted) ;

Groupe(id, name, $creatorId) ;
UtilisateurGroupe($userId, $groupId, validation) ;

Conclusion

Ce TP nous a permis de travailler en collaboration sur un projet d'assez bonne envergure, d'approfondir nos bases en SQL et de comment profiter d'une base de données SQL pour enrichir un site web via PHP.