Après la prise en main de la plateforme, nous allons mettre en pratique les tests fonctionnels et non fonctionnels décrits dans le cahier des charges.

Dans une première partie, nous allons tester l’aspect exigences non fonctionnelles de la plateforme.

***Partie hacking :***

Nous avons estimé que la conservation des données au sein de la base de données doit être fiable, c’est à dire que les mots de passe doivent être hachées ou cryptées et non visibles par autrui.

En utilisant la base de données, lors de la création d’un professeur, nous pouvons remarquer que les mots de passe sont bien hachées, sauf pour le superviseur qui a été créé à la main.

Nous allons donc nous intéresser à la partie qui permet de hacher les mots de passe au sein du Model. Le hachage du mot de passe est réalisé grâce à la fonction Password\_Hash qui crée un nouveau hachage avec un algorithme de hachage afin de sécuriser les données.

***Partie accessibilité:***

La possibilité de traduire la page afin de permettre aux étrangers de l’utiliser n’a pas été implémentée, le site reste en Français.

Nous avons également estimé que pour une question de facilité d’usage, chaque fonctionnalité devrait être réalisable en maximum 3-4 lignes.

Pour la fonctionnalité,

=> création d’un compte, nous avons juste à cliquer sur le bouton “pas de compte, inscription” afin de renseigner ces informations nécessaires à la création du compte.

Sur le plan création d’un compte professeur, les tests sont validés, notamment, par le fait que l’on ne peut créer deux comptes avec la même adresse email affichant ainsi une erreur mais également le fait que la case CGU doit bien être cochée afin que le compte soit créé.

Premièrement, la vérification d’adresse mail est réalisée par le fait que dans le fichier bdd.sql, l’attribut adresse email, de la table professeur, soit précisé comme unique not null, indiquant qu’il est unique empêchant la création et l’insertion d’un professeur au sein de la base de données via la fonction addProfesseur du Model.

Deuxièmement, la vérification du CGU pour créer le compte se fait par le required au sein du html et également la fonction check\_inscription\_p qui vérifie que la session[‘inscrit’] a bien été activée afin de créer le compte sinon il affiche un message d’erreur.

=> création d’un compte superviseur, nous devons directement créer le compte sur la base de données ainsi le premier mot de passe n’est pas haché. Toutefois, si l’on modifie le mot de passe du superviseur crée à la main, ce dernier est hachée dans la base de données mais le mot de passe demandé est également le mot de passe en format haché et non celui de base

=>