

# **Rapport devoir maison en Programmation WEB**

**Tender**

**Site d'appariement social**

Réalisé par : BENMEBAREK Rafik

## **1) Fonctionnalités disponibles :**

### **Création d'un compte :**

Un utilisateur qui ne dispose pas d'un compte Tender peut s'en créer un, en accédant à la page dédiée à cela se trouvant sur le bouton "Sign up" en haut à droite de l'écran.

L'utilisateur aura alors à fournir des informations sur lui (Nom, prénom, email, date de naissance, photo de profil etc..) et un profil lui sera alors créé et des informations supplémentaires sur ces critères de recherches seront demandés (sexe et tranche d'âge)

### **Système de Login**

Une fois le compte créé, l'utilisateur peut se connecter en cliquant sur le bouton "Login" en haut à droite dans la barre de navigation. En fournissant son mail et son mot de passe, un algorithme procède à la vérification de ces identifiants, le système procédera au login, dans le cas échéant il enverra un message d'erreur signalant qu'il y a un problème avec les identifiants fournis.

### **Modification de profil et critères**

Un utilisateur peut modifier à tout moments des informations sur lui, comme son nom, prénom ou plus important encore sa photo de profil. Le serveur s'occupe alors du traitement de la requête et change les champs à modifier dans la base de données et il en va de même pour les critères.

### **Système de likes/dislikes d'autres utilisateurs**

Une fois les critères établis, l'utilisateur peut maintenant voir les profils de tous les utilisateurs qui satisfont ses critères, il a le choix d'aimer (Like) ou pas (Dislike) une personne, ce qui va ajouter ce lien dans la base de données. Dans le cas où il y'a un like mutuel, ces deux utilisateurs disposent d'une page "Vos Matches" qui recense toutes les personnes avec qui ils ont matchés, ils peuvent alors consulter leur profils en cliquant sur leurs noms.

## 2) Principaux outils utilisés:

### **Bcrypt :**

Bcrypt est un package qui est responsable du hashage des mots de passes, et c'est ce hash qui est stocké dans la base de données. Dans le cas d'un login, on récupère ce Hash et c'est bcrypt qui vérifie si les deux paramètres sont égaux

### **express-fileupload :**

Middleware utilisé pour l'upload des photos de profils. Une profile soumise par un utilisateur est téléchargée et stockée sous le répertoire *public/img/upload/* sous un nom spécifique à l'utilisateur. Ce chemin est stockée dans la base de donnée, car c'est préférable à stocker l'image directement dans la base de donnée.

Si un utilisateur change de photo de profil, on ne supprime pas l'ancienne de nos serveurs, pour des raisons de sécurité.

### **Jquery form validator**

Lorsqu'on demande des informations à un utilisateur, et c'est grâce à ce plugin qu'on valide ces entrées, en écrivant les règles spécifiques à chaque champs, en cas de saisie erronée un message d'erreur s'affichera spécifiant la règle qui n'a pas été respectée

### **express-session :**

Ce package permet l'ajout de variables de sessions à notre serveur, ceci est très utile pour y stocker des informations sur l'utilisateur actuellement connecté, tel que son nom, prénom etc..

### **mysql :**

Malgré son nom, cet ORM fût utilisé pour gérer la liaison avec notre base de données MariaDB.

## **Bootstrap :**

Bootstrap a été utilisé pour la réalisation de ce site web du côté front-end, ce fut un framework très utile pour donner une belle forme au site web.

## **JSON :**

Pour le stockage des relation de like/dislike entre les utilisateurs, j'ai stocké un tableau JSON dans la base de données, contenant une liste d'objets ayant comme attributs "id" qui représente l'id d'un autre utilisateur et "likes" qui peut être soit 1 (Like) soit 0 (Dislike).

## **3) Fonctionnalités manquantes :**

### **Envoie de requêtes like/dislike**

Les likes et dislikes ont été fait manuellement, c'est à dire qu'il n'y a pas de système qui permettent d'accomplir cela de manière automatique. Vous remarquerez qu'à part un déplacement à gauche(dislike) ou à droite(like) il ne se passe pas grand chose.

Il ne manque alors qu'à ajouter ce système d'envoi de requete de like et dislike pour mettre à jour l'objet JSON et du coup, mettre à jour la liste de like/dislike.

### **Messagerie**

Par faute de temps, la messagerie n'a pas pu être mise en place, et comme c'est une fonctionnalité facultative, mon attention a été focalisée sur les autres aspects du site.