Chapitre II

Introduction

Ce projet vous propose de créer un site de rencontres.

Vous devrez donc concevoir une application permettant à deux potentielles âmes soeurs de se rencontrer, de l'inscription au contact final.

Un utilisateur devra donc pouvoir s'inscrire, se connecter, compléter son profil, parcourir et rechercher d'autres utilisateurs, les liker [1], et chatter avec ceux qui auront liké en retour.

1. "Liker" étant un choix d'un mauvais goût déplorable, vous êtes encouragés à trouver un verbe plus explicite pour cette action

Chapitre III

Consignes générales

Pour ce projet, vous êtes libres d'utiliser le langage de votre choix.

Vous pouvez utiliser un micro-framework, et toutes les librairies du monde pour ce projet.

On considerera qu'un "micro-framework" a un routeur, et éventuellement du templating, mais pas d'ORM, de validateurs ou de gestion de comptes utilisateurs [1]. Tant que vous respectez ces contraintes, vous êtes libre d'utiliser celui qui vous plaira.

Si vous avez besoin d'inspiration, on suggérera, pour les principaux languages :

- Sinatra pour Ruby.
- Express pour Node (oui, nous le considérons comme un micro-framework).
- Flask pour Python.
- Scalatra pour Scala.
- Slim pour PHP (Silex n'est pas autorisé en raison de l'intégration de Doctrine).
- Nickel pour Rust.
- Goji pour Golang.
- Spark pour Java.
- Crow pour C++.

Vous êtes libre d'utiliser le serveur web de votre choix, que ce soit `Apache`, `Nginx` ou même un `built-in web server`.

L'ensemble de votre application devra être au minimum compatible sur `Firefox` (>= 41) et `Chrome` (>= 46).

Votre site doit avoir une mise en page décente : c'est à dire au moins un header, une section principale et un footer.

Votre site devra être présentable sur mobile, et garder une mise en page acceptable sur de petites résolutions.

1. Cette définition fera foi en soutenance, même si un site internet en possède une autre.

Tous vos formulaires doivent avoir des validations correctes, et l'ensemble de votre site devra être sécurisé. Ce point est obligatoire et sera vérifié longuement en soutenance. Pour vous faire une petite idée, voici quelques éléments qui ne sont pas considérés comme sécurisés :

- Avoir des mots de passe "en clair" dans une base de données.
- Pouvoir injecter du code HTML ou JavaScript "utilisateur" dans des variables mal protégées.
- Pouvoir uploader du contenu indésirable.
- Pouvoir modifier une requête SQL.

Chapitre IV

Partie obligatoire

Vous devez donc réaliser une application web ayant les fonctionnalités suivantes :

IV.1 Inscription et connexion

L'application doit permettre à un utilisateur de s'inscrire, en demandant au minimum une adresse email, un nom d'utilisateur, un nom, un prénom et un mot de passe un tant soit peu sécurisé.

L'utilisateur doit ensuite être capable de se connecter avec son nom d'utilisateur et son mot de passe. Il doit également pouvoir recevoir un mail de réinitialisation de son mot de passe en cas d'oubli, et se déconnecter en un seul clic depuis n'importe quelle page du site.

IV.2 Profil de l'utilisateur

- Une fois connecté, un utilisateur doit pouvoir compléter son profil, en rajoutant des informations telles que :
  - Son sexe.
  - Son orientation sexuelle.
  - Une bio.
  - Une liste d'interêts, sous la forme de tags (ex : #bio, #geek, #piercing etc...). Ces tags doivent être réutilisables.
  - Des images, maximum cinq, dont une servant de photo de profil.

- A tout moment, l'utilisateur doit pouvoir modifier ces informations, ainsi que son nom, son prénom et son adresse email.

- L'utilisateur doit pouvoir consulter les personnes ayant consulté son profil, ainsi que les personnes qui l'ont "liké".

- L'utilisateur doit avoir un score de popularité public [1].

1. libre à vous de définir le terme "score de popularité", tant qu'il reste cohérent

- L'utilisateur doit être géolocalisé, à l'arrondissement près. Si l'utilisateur ne veut pas être géolocalisé, vous devez trouver un moyen de le géolocaliser malgré lui [2]. L'utilisateur doit pouvoir modifier sa localisation sur son profil.

IV.3 Parcours

L'utilisateur doit pouvoir aisément avoir une liste de suggestions qui lui correspondent.

- Vous ne devez afficher que les profils "intéressants", par exemple que des hommes pour une femme hétérosexuelle. Vous devez gérer la bisexualité. Si l'orientation n'est pas renseignée, l'utilisateur sera considéré bisexuel.

- Vous devez méler intelligemment [3] les profils :
  - Dans la même zone géographique que l'utilisateur.
  - Avec un maximum de tags d'interêt communs.
  - Avec un maximum de popularité.

- Vous devez afficher prioritairement les profils dans la même zone géographique que l'utilisateur.

- Cette liste devra être triable par âge, localisation, popularité et par tags en communs.

- Cette liste devra être filtrable par intervalle d'âge, localisation, intervalle de popularité et par tags.

IV.4 Recherche

L'utilisateur doit pouvoir effectuer une recherche avancée en sélectionnant un ou plusieurs critères tels que :

- Un intervalle d'âge.
- Un intervalle de score de popularité.
- La localisation.
- Un ou plusieurs tags d'interêt.

Tout comme la liste de suggestion, la liste de résultats devra être triable et filtrable par âge, localisation, popularité et par tags.

2. Et oui, c'est ce que font les sites de rencontre...
3. Faites au moins une petite pondération sur les différents critères.

IV.5 Profil des autres utilisateurs

Un utilisateur doit pouvoir consulter le profil des autres utilisateurs, qui doit contenir toutes les informations disponibles sur ce dernier, excepté l'adresse email et le mot de passe.

Quand un utilisateur regarde un profil, il doit apparaître dans l'historique des visites de ce dernier.

L'utilisateur doit également pouvoir :

- Si il possède au moins une photo, "Liker" un autre utilisateur. Quand deux personnes se "likent" mutuellement, on dira qu'ils sont "connectés", et doivent pouvoir engager la conversation. Si l'utilisateur courant ne possède pas encore de photo, il ne doit pas pouvoir faire cette action.
- Consulter le score de popularité.
- Voir si l'utilisateur est en ligne, et si ce n'est pas le cas, afficher la date de sa dernière visite.
- Reporter l'utilisateur comme étant un "faux compte".
- Bloquer l'utilisateur. Un utilisateur bloqué ne doit plus apparaître dans les résultats de recherche, et ne doit plus génerer de notifications.

L'utilisateur doit clairement voir si le profil consulté est connecté ou "like" le sien, et doit pouvoir "unlike".

IV.6 Chat

Lorsque deux utilisteurs sont connectés [4], ils doivent pouvoir "chatter" en temps réel [5]. L'implémentation du chat est totalement libre. L'utilisateur doit pouvoir voir, de n'importe quelle page, qu'il a recu un nouveau message.

IV.7 Notifications

Un utilisateur doit être notifié, en temps réel [6], des évènements suivants :

- L'utilisateur a reçu un "like".
- L'utilisateur a reçu une visite.
- L'utilisateur a reçu un message.

4. C'est à dire qu'ils se "likent" mutuellement.
5. On tolèrera une marge de 10 secondes.
6. Comme pour les messages, on tolèrera une marge de 10 secondes.

- Un utilisateur “liké” a “liké” en retour.
- Un utilisateur connecté ne vous “like” plus.

L’utilisateur doit pouvoir voir, de n’importe quelle page, qu’une notification n’a pas été lue.

Chapitre V

Partie bonus

Si la partie obligatoire a été réalisée entièrement et parfaitement, vous pouvez ajouter les bonus que vous souhaitez ; ils seront évalués à la discrétion de vos correcteurs. Vous devez néanmoins toujours respecter les contraintes de base.

Si l'inspiration vous manque, voici quelques pistes :

- Ajouter des stratégies Omniauth pour la connexion.
- Charger les images depuis Facebook et/ou Google+.
- Faire une carte intéractive des utilisateurs (ce qui implique une géolocalisation plus précise, via Javascript).

Chapitre VI

Rendu et peer-évaluation

Les consignes suivantes seront présentes dans le barème de soutenance. Soyez très attentifs lors de l'application de ces dernières car elles seront sanctionnées par un 0 sans appel.

VI.1 Consignes éliminatoires

- Votre code ne doit produire aucune erreur, warning ou notice, coté serveur et coté client, dans la console web.
- D'ailleurs, tout ce qui n'est pas explicitement autorisé est interdit.
- Enfin, la moindre faille de sécurité entrainera un 0. Vous devez au minimum gérer ce qui est indiqué dans les consignes générales, c'est à dire ne pas avoir de mot de passe en clair, être protégé contre les injections SQL, et avoir une validation de tous les formulaires de saisie et d'upload.

Vous pouvez poser vos questions sur le forum, Jabber, Slack, etc...

Bon courage à tous !