



Web Matcha

Résumé: Parce que l'amour aussi, ça s'industrialise.

Version: 6.0

Table des matières

I	Préambule	2
II	Introduction	3
III	Instructions générales	4
IV	Partie obligatoire	6
IV.1	Inscription et connexion	6
IV.2	Profil utilisateur	6
IV.3	Navigation	7
IV.4	Recherche	7
IV.5	Consultation de profil	8
IV.6	Chat	8
IV.7	Notifications	8
V	Partie bonus	10
VI	Rendu et évaluation par les pairs	11
VI.1	Évaluation par les pairs	11

Chapitre I

Préambule

Ce deuxième millénaire a changé et renforcé à jamais les habitudes et coutumes d'Internet. Les choix sont désormais guidés par la technologie, laissant de moins en moins de place au hasard. Les relations humaines, fondement de toute société moderne, se forment de plus en plus artificiellement grâce aux algorithmes des sites de rencontre et des réseaux sociaux, connectant les gens sur la base de critères très spécifiques.

Oui, le romantisme est mort, et Victor Hugo se retourne probablement dans sa tombe.

Chapitre II

Introduction

Ce projet vise à créer un site de rencontre.



Vous devez développer une application qui facilite les connexions entre deux partenaires potentiels, couvrant l'ensemble du processus, de l'inscription à la rencontre finale.

Les utilisateurs doivent pouvoir s'inscrire, se connecter, compléter leur profil, rechercher et consulter les profils d'autres utilisateurs, et exprimer leur intérêt pour eux avec un « like »¹. Ils doivent également pouvoir discuter avec ceux qui ont réciproquement leur intérêt.

1. Puisque « like » n'est pas un terme idéal, vous êtes encouragé à trouver une alternative plus explicite.

Chapitre III

Instructions générales

- Votre application ne doit produire aucune erreur, warning ou notice, côté serveur et côté client.
- Pour ce projet, vous êtes libre d'utiliser le langage de programmation de votre choix.
- Vous pouvez utiliser des micro-frameworks et toutes les bibliothèques nécessaires pour ce projet.
- Vous êtes libre d'utiliser des bibliothèques d'interface utilisateur telles que React, Angular, Vue, Bootstrap, Semantic, ou toute combinaison de celles-ci.
- Aucune vulnérabilité de sécurité n'est autorisée. Vous devez au minimum respecter les exigences de sécurité obligatoires, mais nous vous encourageons vivement à aller au-delà—tout en dépend.
- Nous définissons un « micro-framework » comme un framework qui inclut un routeur et éventuellement du templating, mais qui n'inclut pas d'ORM, de validateurs ou de gestionnaire de comptes utilisateurs.¹ Tant que vous respectez ces contraintes, vous êtes libre d'utiliser les outils de votre choix.
- Si vous avez besoin d'inspiration, nous suggérons d'utiliser les langages suivants comme choix principal :
 - Sinatra pour Ruby.
 - Express pour Node (oui, nous considérons cela comme un micro-framework).
 - Flask pour Python.
 - Scalatra pour Scala.
 - Slim pour PHP (Silex n'est pas autorisé en raison de son intégration avec Doctrine).
 - Nickel pour Rust.
 - Goji pour Golang.
 - Spark pour Java.
 - Crow pour C++.
- Vous devez utiliser une base de données relationnelle ou orientée graphe. La base

1. Cette définition fera autorité lors de la soutenance, quelles que soient les définitions alternatives trouvées en ligne.

de données doit être gratuite, comme MySQL, MariaDB, PostgreSQL, Cassandra, InfluxDB, Neo4j, etc. Vous devez créer vos requêtes manuellement, comme le font les développeurs expérimentés. Cependant, si vous êtes malin, vous pouvez créer votre propre bibliothèque pour simplifier la gestion des requêtes.

- Pour l'évaluation de ce projet, votre base de données doit contenir un minimum de 500 profils distincts.
- Vous êtes libre de choisir le serveur web qui convient le mieux à vos besoins, qu'il s'agisse d'**Apache**, **Nginx** ou d'un **serveur web intégré**.
- Votre application entière doit être compatible avec au moins les dernières versions de **Firefox** et **Chrome**.
- Votre site web doit avoir une mise en page bien structurée, comprenant au moins un en-tête, une section principale et un pied de page.
- Votre site web doit être adapté aux mobiles et maintenir une mise en page acceptable sur les petits écrans.
- Tous les formulaires doivent inclure une validation appropriée, et l'ensemble du site web doit être sécurisé. Il s'agit d'une exigence obligatoire qui sera largement évaluée lors de la soutenance. Pour vous donner une idée, voici quelques exemples de vulnérabilités de sécurité qui ne seront pas tolérées :
 - Stocker des mots de passe en clair dans votre base de données.
 - Permettre l'injection de HTML ou JavaScript dans des variables non protégées.
 - Permettre le téléchargement de contenu non autorisé.
 - Permettre les attaques par injection SQL.

Chapitre IV

Partie obligatoire

Vous devez développer une application web avec les fonctionnalités suivantes :

IV.1 Inscription et connexion

L'application doit permettre à un utilisateur de s'inscrire en fournissant au minimum son adresse e-mail, son nom d'utilisateur, son nom de famille, son prénom et un mot de passe sécurisé. Les mots anglais couramment utilisés ne doivent pas être acceptés comme mots de passe.

Après l'inscription, l'utilisateur doit recevoir un e-mail avec un lien unique pour vérifier son compte.

Les utilisateurs doivent pouvoir se connecter en utilisant leur nom d'utilisateur et leur mot de passe. Ils doivent également avoir la possibilité de demander un e-mail de réinitialisation de mot de passe s'ils l'oublient. De plus, les utilisateurs doivent pouvoir se déconnecter en un seul clic depuis n'importe quelle page du site.

IV.2 Profil utilisateur

- Une fois connectés, les utilisateurs doivent compléter leur profil en fournissant les informations suivantes :
 - Genre.
 - Préférences sexuelles.
 - Une biographie.
 - Une liste d'intérêts utilisant des tags (par exemple, #vegan, #geek, #piercing, etc.), qui doivent être réutilisables
 - Jusqu'à 5 photos, dont une désignée comme photo de profil.
- Les utilisateurs doivent pouvoir modifier ces informations à tout moment, ainsi que mettre à jour leur nom de famille, prénom et adresse e-mail.
- Les utilisateurs doivent pouvoir voir qui a consulté leur profil.
- Les utilisateurs doivent également pouvoir voir qui les a « likés ».

- Chaque utilisateur doit avoir une « note de popularité » publique¹.
- Les utilisateurs doivent être localisés via le positionnement GPS jusqu'à leur quartier, avec leur consentement explicite. Si un utilisateur refuse le suivi de localisation GPS, il doit fournir manuellement sa localisation approximative (ville ou quartier) pour utiliser les fonctionnalités de matching. Cette saisie manuelle de la localisation est requise pour que l'application fonctionne correctement.² Les utilisateurs doivent également avoir la possibilité de modifier leur localisation dans leur profil à tout moment.

IV.3 Navigation

Les utilisateurs doivent pouvoir accéder facilement à une liste de profils suggérés qui correspondent à leurs préférences.

- Vous devez suggérer des profils « intéressants ». Par exemple, une femme hétérosexuelle ne devrait voir que des profils masculins. Vous devez également gérer la bisexualité. Si un utilisateur n'a pas spécifié son orientation, il doit être considéré comme bisexuel par défaut.
- Les correspondances doivent être déterminées intelligemment³ en fonction de :
 - La proximité avec la localisation géographique de l'utilisateur.
 - Le plus grand nombre de tags partagés.
 - La « note de popularité » la plus élevée.
- La priorité doit être donnée aux utilisateurs de la même zone géographique.
- La liste des profils suggérés doit être triable par âge, localisation, « note de popularité » et tags communs.
- Les utilisateurs doivent pouvoir filtrer la liste en fonction de l'âge, de la localisation, de la « note de popularité » et des tags communs.

IV.4 Recherche

Les utilisateurs doivent pouvoir effectuer une recherche avancée en sélectionnant un ou plusieurs critères, tels que :

- Une tranche d'âge spécifique.
- Une plage de « note de popularité ».
- Une localisation.
- Un ou plusieurs tags d'intérêt.

Comme pour la liste suggérée, les résultats de recherche doivent être triables et filtrables par âge, localisation, « note de popularité » et tags d'intérêt.

1. Vous êtes responsable de définir ce que signifie « note de popularité », tant que vos critères sont cohérents.
2. Note : Cette approche respecte les exigences du RGPD concernant le consentement explicite pour le traitement des données. Bien que certains sites de rencontre puissent utiliser des méthodes de suivi alternatives, ce projet met l'accent sur les pratiques de développement respectueuses de la vie privée.
3. Prenez en compte plusieurs critères.

IV.5 Consultation de profil

Les utilisateurs doivent pouvoir consulter les profils des autres utilisateurs. Les profils doivent afficher toutes les informations disponibles, à l'exception de l'adresse e-mail et du mot de passe. Lorsqu'un utilisateur consulte un profil, cela doit être enregistré dans son historique de visites.

L'utilisateur doit également pouvoir :

- « Liker » la photo de profil d'un autre utilisateur. Lorsque deux utilisateurs se « likent » mutuellement, ils seront considérés comme « connectés » et pourront commencer à discuter. Si l'utilisateur actuel n'a pas de photo de profil, il ne peut pas effectuer cette action.
- Retirer un « like » précédemment donné. Cela empêchera les notifications ultérieures de cet utilisateur, et la fonction de chat entre eux sera désactivée.
- Consulter la « note de popularité » d'un autre utilisateur.
- Voir si un utilisateur est actuellement en ligne, et sinon, consulter la date et l'heure de sa dernière connexion.
- Signaler un utilisateur comme « faux compte ».
- Bloquer un utilisateur. Un utilisateur bloqué n'apparaîtra plus dans les résultats de recherche et ne générera plus de notifications. De plus, discuter avec lui ne sera plus possible.

Les utilisateurs doivent clairement voir si le profil qu'ils consultent les a « likés » ou s'ils sont déjà « connectés ». Ils doivent également avoir la possibilité de « unlike » ou de se déconnecter de ce profil.

IV.6 Chat

Lorsque deux utilisateurs sont connectés⁴, ils doivent pouvoir « chatter » en temps réel.⁵

L'implémentation de la fonctionnalité de chat vous appartient. Cependant, les utilisateurs doivent pouvoir voir, depuis n'importe quelle page, lorsqu'ils reçoivent un nouveau message.

IV.7 Notifications

Les utilisateurs doivent recevoir des notifications en temps réel⁶ pour les événements suivants :

- Lorsqu'ils reçoivent un « like ».
- Lorsque leur profil a été consulté.
- Lorsqu'ils reçoivent un message.
- Lorsqu'un utilisateur qu'ils ont « liké » les « like » également en retour.
- Lorsqu'un utilisateur connecté les « unlike ».

4. C'est-à-dire qu'ils se sont mutuellement « likés ».

5. avec un délai maximum de 10 secondes.

6. Avec un délai maximum de 10 secondes.

Les utilisateurs doivent pouvoir voir, depuis n'importe quelle page, lorsqu'ils ont des notifications non lues.



Pour des raisons de sécurité, tous les identifiants, clés API, variables d'environnement, etc., doivent être stockés localement dans un fichier .env et exclus de Git. Le stockage public des identifiants peut entraîner l'échec du projet.

Chapitre V

Partie bonus

Voici des fonctionnalités bonus possibles que vous pouvez implémenter pour gagner des points supplémentaires :

- Ajouter des stratégies OmniAuth pour l'authentification des utilisateurs.
- Permettre aux utilisateurs de créer une galerie photo personnelle avec téléchargement par glisser-déposer et édition d'image de base (par exemple, recadrer, pivoter, appliquer des filtres).
- Développer une carte interactive des utilisateurs, nécessitant une localisation GPS plus précise via JavaScript.
- Intégrer un chat vidéo ou audio pour les utilisateurs connectés.
- Implémenter une fonctionnalité pour planifier et organiser des rendez-vous ou événements réels pour les utilisateurs matchés.



La partie bonus ne sera évaluée que si la partie obligatoire est parfaite. « Parfait » signifie que les fonctionnalités obligatoires ont été entièrement implémentées et fonctionnent sans aucun dysfonctionnement. Si vous n'avez pas rempli TOUTES les exigences obligatoires, vos fonctionnalités bonus ne seront pas évaluées.

Chapitre VI

Rendu et évaluation par les pairs

Soumettez votre travail dans votre dépôt **Git** comme d'habitude. Seul le travail présent dans votre dépôt sera évalué lors de la soutenance. Assurez-vous de vérifier les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.

VI.1 Évaluation par les pairs

- Votre code ne doit produire aucune erreur, warning ou notice, côté serveur ou côté client (dans la console web).
- Tout ce qui n'est pas explicitement autorisé est strictement interdit.
- Toute faille de sécurité entraînera une note de 0. Au minimum, vous devez implémenter les mesures de sécurité décrites dans les instructions générales. Cela inclut :
 - S'assurer que les mots de passe ne sont pas stockés en clair dans la base de données.
 - Se protéger contre les attaques par injection SQL.
 - Valider toutes les entrées de formulaire et les téléchargements de fichiers.