ECF n°1
Evaluation en Cours de Formation n°1

Objectif fonctionnel :

- Créer un mini-réseau social

Ce mini-projet se présentera sous la forme d'une application sur une seul document (SPA – Single Page Application) qui peut, bien sûr, faire appel à de multiples fichiers de style et de script. Les différents affichages produits sur ce document seront appelés **vues**.

Pour le stockage et la restitution d'informations, vous devrez créer et utiliser une API REST et une base de données.

Le choix des technologies est libre (langages et frameworks).

Sa réalisation est proposée en 3 itérations.

Objectif pédagogique :

- Evaluer la capacité de l'apprenant à :
  - Itération 1 : Dupliquer et dans une moindre mesure Adapter ;
  - Itération 2 : Adapter et dans une moindre mesure Transposer ;
  - Itération 3 : Transposer.

- **L'objectif à atteindre est, à minima, la fin de l'itération 1.**

Modalités :

- Vous avez jusqu'à **18:00** pour fournir un livrable.

- Le livrable à fournir se présentera sous la forme d'un dépôt GIT auquel les formateurs auront accès. Le dépôt devra contenir :
  - Le code source backend ;
  - Le code source frontend ;
  - Un export de la base de données au format SQL.

➔ **Pensez à faire en sorte que chaque itération soit réalisée sur une branche distincte de votre dépôt.**

- Chaque apprenant a droit à **4 « jokers »**. Un « joker » est une demande d'assistance auprès d'un de vos formateurs :
  - 2 « jokers » pour résoudre une problématique plutôt frontend ;
  - 2 « jokers » pour résoudre une problématique plutôt backend.

I. Itération 1/3 :

Objectif :

- Réaliser une **mise en page soignée** ;
- S'authentifier pour accéder son profil utilisateur.

Vues :

- Connexion (accueil)
- Inscription
- Profil utilisateur

Fonctionnalités :

- **Vue de Connexion** (*accueil*) :
  - Contient un formulaire de saisie avec e-mail et mot de passe.
  - Contient un lien qui permet d'afficher la **vue d'inscription**.

➢ Si l'utilisateur saisit un e-mail et mot de passe correct et valide, la **vue de profil utilisateur** s'affiche en pleine page.
➢ Si l'utilisateur saisit un e-mail et/ou mot de passe incorrect et valide, un **message d'erreur** du type "**e-mail ou mot de passe incorrect**" s'affiche.
➢ Si l'utilisateur clique sur le lien d'inscription, la **vue d'inscription** s'affiche à la place de la vue de connexion.

- **Vue d'Inscription** :
  - Contient un formulaire de saisie avec prénom, nom, date de naissance, e-mail, mot de passe.
  - Contient un lien qui permet d'afficher la **vue de connexion**.

➢ Si l'utilisateur saisit un prénom, un nom, une date de naissance, un e-mail et un mot de passe et valide, la **vue de connexion** s'affiche avec le message "**Inscription réussie ! Vous pouvez utiliser maintenant vos identifiants.**".
➢ Si l'utilisateur omet de saisir une information, un ou plusieurs **messages d'erreur** s'affichent indiquant à l'utilisateur les libellés de chaque champ et la nature du problème.
  - Pour des raisons de sécurité, vous devez **obligatoirement** implémenter ces vérifications **coté serveur**.
  - Pour améliorer l'ergonomie et se prémunir des appels inutiles au serveur, vous pouvez **optionnellement** implémenter ces vérifications coté client.

➢ Si l'utilisateur clique sur le lien de connexion, la vue de connexion s'affiche à la place de la **vue d'inscription**.

- **Vue de Profil utilisateur :**
  - Contient un encart de type fiche utilisateur contenant le prénom, le nom et l'âge (pas sa date de naissance) de la personne.
  - Contient un lien de déconnexion qui permet de réafficher la **vue de connexion**.

II. Itération 2/3 :

Objectif :

- Améliorer la sécurité des formulaires ;
- Pouvoir écrire des messages sur son propre profil.

Fonctionnalités :

- Vue d'Inscription :
  - Le formulaire contient un champ supplémentaire de **confirmation de mot de passe** (double saisie du mot de passe).

➢ Si l'utilisateur ne saisit pas deux fois le même mot de passe, un message d'erreur s'affiche indiquant à l'utilisateur la nature du problème.
➢ Si l'utilisateur ne saisit pas une adresse e-mail au format d'adresse e-mail, un message d'erreur s'affiche indiquant à l'utilisateur la nature du problème
➢ Si l'utilisateur saisi une adresse e-mail qui correspond à une adresse déjà enregistrée en base de données, un message d'erreur s'affiche indiquant à l'utilisateur la nature du problème.

Comme précédemment :

  - Pour des raisons de sécurité, vous devez **obligatoirement** implémenter ces vérifications **coté serveur**.
  - Pour améliorer l'ergonomie et se prémunir des appels inutiles au serveur, vous pouvez **optionnellement** implémenter ces vérifications coté client.

- Vue de Profil utilisateur :
  - Contient un encart pour la saisie de messages et un encart pour l'affichage des messages.

Encart pour la saisie de messages :

  - Contient un formulaire de saisie suivi d'un bouton de validation ; ce formulaire permet de démarrer un sujet de discussion

➢ Si l'utilisateur saisit un message dans le formulaire et valide, le nouveau sujet de discussion contenant le nouveau message s'affiche dans l'encart pour l'affichage des messages.
➢ Tant que l'utilisateur ne saisit pas de message dans le formulaire, le bouton de validation est désactivé.

Encart pour l'affichage des messages :

  - Contient une liste de sujets de discussions du plus récent au plus ancien. Chaque sujet de discussions comporte un message qui se compose du corps du message, sa date de publication, le prénom et le nom de son auteur. Chaque message est suivi d'un lien supprimer.

➢ Si l'utilisateur clique sur le lien supprimer, le message ainsi que le sujet de discussion associé sont :
  ➔ Supprimés en base de données ;
  ➔ Supprimés de l'affichage.

III. Itération 3/3 :

Objectif :

- Améliorer le profil utilisateur ;
- Pouvoir répondre à des messages sur un autre profil utilisateur.

Fonctionnalités :

- Vue de Profil utilisateur :
  - Contient un encart pour la recherche d'utilisateur. Cet encart contient un formulaire pour la saisie d'une adresse e-mail. Le champ de saisie fonctionne selon le principe des champs d'« autocomplétion ».

Encart pour la recherche d'utilisateur :

- Si l'utilisateur saisi tout ou une partie d'une adresse e-mail utilisateur, une liste d'utilisateurs s'affiche sous le formulaire.
  - ➔ La liste s'affiche automatiquement sous le formulaire sans avoir à effectuer de validation ;
  - ➔ La liste est actualisée au fur et à mesure de la saisie d'une adresse e-mail afin de restreindre la taille de la liste ;
  - ➔ La liste affiche le prénom, le nom et l'adresse e-mail de l'utilisateur sous la forme d'un lien.

- Si l'utilisateur clique sur le lien formé par le prénom, le nom et l'adresse e-mail d'un utilisateur, la vue de profil de cet utilisateur s'affiche.

Encart pour l'affichage des messages :

- Chaque sujet de discussions comporte 1 ou plusieurs messages présentés de façon arborescente (le 1er message est à gauche, le suivant indenté, etc.). Chaque message est suivi de 2 liens :
  - ➔ « Répondre » dans tous les cas ;
  - ➔ Et « Supprimer » uniquement si l'utilisateur est l'auteur du message ou que le message fait partie d'un sujet de discussion dont l'utilisateur est le propriétaire.
- Le prénom et le nom de l'auteur de chaque message constitue un lien qui, lorsqu'il est suivi, permet d'afficher la vue de profil de l'auteur du message.

- Si l'utilisateur clique sur le lien répondre, un formulaire s'affiche à la suite du message. Ce formulaire permet de saisir un message. Tant que l'utilisateur n'a pas saisi de message, le bouton valider est désactivé. Si utilisateur écrit un message et valide, le nouveau message du sujet de discussion s'affiche sous le précédent selon un affichage arborescent.

- Si l'utilisateur clique sur "supprimer" sur un message, le message est supprimé. Si tous les messages d'un sujet de discussion sont supprimés, le sujet de discussion est supprimé.

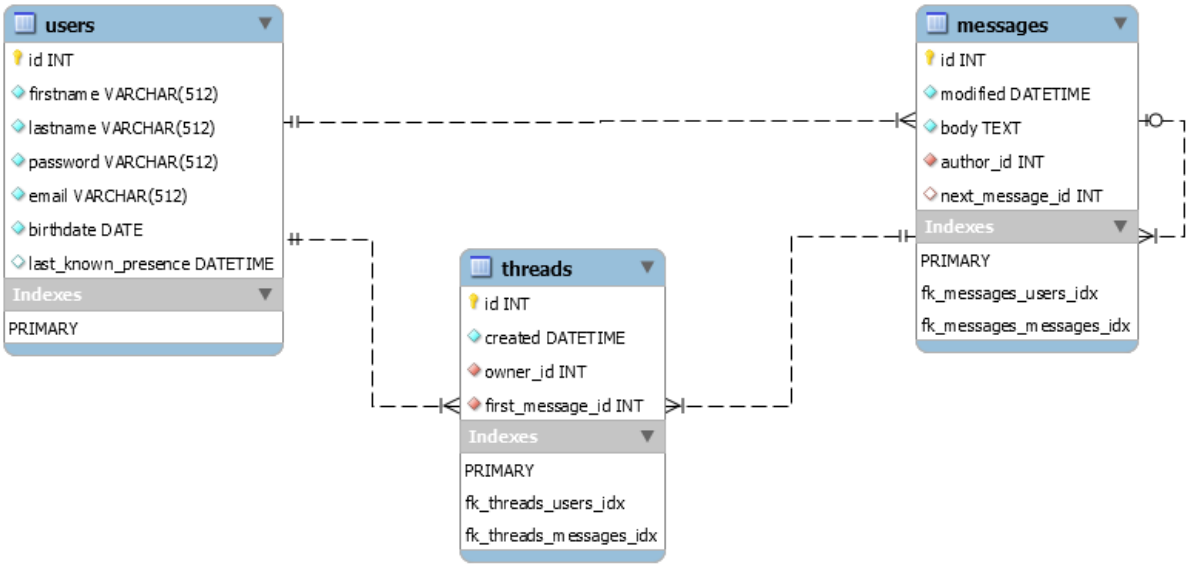
Encart de type fiche utilisateur

- L'encart de type fiche utilisateur contient un champ supplémentaire indiquant si l'utilisateur est en ligne (pastille verte) ou hors ligne (pastille rouge).

IV. Aides :

Dictionnaire de données :

Voici, ci-après, une proposition de modèle de données qui sera susceptible d'évoluer en fonction de vos besoins.



- `users` (table des utilisateurs)
  - `id` (identifiant, clé primaire)
  - `firstname`
  - `lastname`
  - `email`
  - `password`
  - `birthdate` (date de naissance)
  - `last_known_presence` (date de dernière connexion)

- `thread` (table des sujets de discussion)
  - `id` (identifiant, clé primaire)
  - `first_message_id` (obligatoire, clé étrangère vers le premier message)
  - `owner_id` (obligatoire, clé étrangère vers l'utilisateur)
  - `created` (date de création du sujet de discussion)

- `messages` (table des messages)
  - `id` (identifiant, clé primaire)
  - `body` (corps du message)
  - `author_id` (obligatoire, clé étrangère vers l'utilisateur auteur du message)
  - `next_message_id` (optionnel, clé étrangère vers le message suivant)
  - `created` (date de création du message)

Authentification :

Si, en cas de rechargement de la page, l'utilisateur est toujours connecté et est toujours sur la vue avant le rechargement c'est mieux. Sinon, ça n'est pas pénalisant.

Afficher l'état en ligne/hors ligne :

D'une part, envoyer à intervalle de temps régulier une requête au serveur pour enregistrer l'état de l'utilisateur (présent à telle heure) et d'autre part, récupérer à intervalle de temps régulier l'utilisateur ou la liste des utilisateurs dont l'état présent était confirmé il y'a moins d'un certain nombre de secondes ou minutes.