

## Support à rendre

### Répartition du travail dans le binôme

- Réalisation/ implémentation d'une bonne partie de l'arbre des composants React, leurs interactions etc. (Hana)
- Spécification des Web Services à implémenter pour le site web (Fatou)
- Implémentation d'une bonne partie du serveur d'API (Fatou)
- Réalisation d'une bonne partie de la base de données (les collections, champs des documents, les classes pour interagir avec la base de données, l'implémentation des méthodes CRUD) (Fatou)
- Gérer l'inscription et connexion d'un utilisateur connexion et la redirection vers le forum + validation de l'inscription par l'administrateur (Hana, Fatou)
- Réalisation de la page principale (Hana)
- Dans le forum, affichage de la liste des messages (Hana, Fatou)
- Création de messages (possibilité de répondre à un message précédemment posté et de démarrer une nouvelle discussion) (Fatou)
- Pouvoir rechercher/ supprimer un certain message (Fatou)
- Visualisation du profil de l'utilisateur (comprenant toutes ses coordonnées et la liste des messages qu'il a publiés et la capacité de les supprimer, pouvoir visualiser les profils d'autres membres) (Hana, Fatou)
- Réalisation des options de déconnexion/ consulter la page de profil (Hana)
- Pouvoir afficher le panneau d'administrateur, dans le cas où l'utilisateur connecté est administrateur (Hana)
- Personnaliser le forum si l'on a le statut d'administrateur (Boutons pour donner/retirer le statut admin à un utilisateur) (Hana)
- Réalisation du design CSS (Hana)

PS : Cette répartition n'est qu'indicative, bien évidemment, nous nous sommes beaucoup entretenues lors de l'implémentation des fonctionnalités

### Problèmes rencontrés :

- Installation et configuration de MongoDB, MongoDB Compass...

### Ce qu'il reste à faire / nos choix de modélisation :

- Pour la création d'un nouvel utilisateur en fonction de l'approbation d'un administrateur, nous aurions aimé pouvoir intégrer un champ « isAuthorized » dans la collection « users ». Initialement, lorsqu'un utilisateur crée un compte, ce champ peut être défini par défaut sur false, signalant ainsi que l'utilisateur n'est pas encore membre, et non autorisé à se connecter sur le forum. Ce statut est modifié ultérieurement par un administrateur lorsqu'il approuve le statut de membre de

l'utilisateur. Lorsque l'utilisateur tente de se connecter, le système vérifie d'abord si son compte est autorisé en consultant la valeur du champ `isAuthorized`. Si celui-ci est `true`, l'utilisateur est autorisé à se connecter avec succès, sinon l'accès lui est refusé. Cette manière de faire nous permettrait de créer l'utilisateur dans la base de données sans regarder l'approbation ou non de l'administrateur, puis pouvoir bloquer un utilisateur non autorisé, au moment de la connexion.

- **Changement de la page courante** : nous avons choisi de ne pas garder l'utilisation des States pour changer la page courante (`signin_page`, `message_page`) mais plutôt : nous avons choisi d'utiliser `<BrowserRouter>`, `<Link>` et `history.push()` en raison de leur simplicité et de leur efficacité.