

TP #1

(Travail d'équipe permis)

Remise : dimanche 10 avril 2022

Memo

Application Web monopage React/Firebase

Tâches à accomplir

Vous devez compléter quelques fonctionnalités simples de l'application « Memo » dont le code de départ est fourni dans le dépôt GitHub suivant :

<https://github.com/582-4PA-H22/tp1-memo>

Suggestion : faites un « fork » de ce dépôt à votre compte GitHub, puis « clonez-le » localement sur votre poste de travail.

Consultez les copies d'écrans distribuées avec l'énoncé.

A. (15 points) Configuration Firebase

- i. Créez une nouvelle appli Firebase dans votre projet existant ;
- ii. Créez une collection Firestore nommé « *memo* » ;
- iii. Copiez et utilisez votre *objet de configuration* Firebase dans votre application React ;
- iv. Une fois votre solution complétée, revenez à ce point et configurer les « *règles de sécurité* » Firestore pour restreindre l'accès à vos données uniquement aux utilisateurs connectés et avec les privilèges minimaux comme expliqué en classe (ce n'est pas encore fait).

B. (0 point) Installation des modules Node . js, test, et synchro GitHub

- i. Utilisez npm pour installer toutes les dépendances Node . js du projet ;
- ii. Testez que l'application fonctionne localement (npm start) ;
- iii. Synchronisez votre code avec votre dépôt GitHub de solution ;
- iv. N'oubliez pas de faire une dernière synchronisation de votre dépôt GitHub avant la remise du travail.

C. (25 points) Gestion des utilisateurs

- i. Utilisez `Firestore Authentication` pour permettre l'ouverture de connexion avec le *fournisseur d'authentification* Google ;
- ii. Lorsqu'un utilisateur est connecté adéquatement, l'information de son profil est affichée dans le composant `Utilisateur` ;
- iii. Pour chaque utilisateur de l'application, créez un document dans votre collection `Firestore` « *memo* » avec comme identifiant du document l'identifiant de l'utilisateur retourné par `Firestore Auth`, et comme propriétés l'information suivante : *nom complet* et *adresse courriel*.

Attention : vous devez créer ce document une seule fois, c'est-à-dire lorsque l'utilisateur se connecte à plusieurs reprises, pas besoin de recréer le document, il suffit d'utiliser l'option pour fusionner le document.

- iv. Gérez le bouton de déconnexion adéquatement pour fermer la connexion.

D. (20 points) Ajout d'une tâche

- i. Gérez l'événement « *submit* » du formulaire pour ajouter une tâche dans la sous-collection `Firestore` « *taches* » de l'utilisateur connecté ;
- ii. N'oubliez pas de désactiver (en JavaScript) l'action par défaut d'un formulaire HTML (sinon il y aura rechargement de la page !!) ;
- iii. Videz le formulaire après l'ajout d'une tâche et remettez le curseur dans la boîte de texte pour l'ajout de la prochaine tâche ;
- iv. L'ajout d'une nouvelle tâche doit déclencher un changement d'état pour permettre à `React` de rafraîchir le UI du composant.

E. (30 points) Affichage des tâches

- i. Produisez le code nécessaire pour chercher toutes les tâches appartenant à l'utilisateur connecté dans la collection `Firestore` adéquate au chargement du composant, et lorsque l'état des tâches change ;
- ii. Produisez l'affichage des tâches à partir du résultat obtenu en (i) ;
- iii. Écrivez et utilisez une fonction JavaScript adéquate pour afficher la date et heure dans le format visible dans les captures d'écran (notez que `Firestore` retourne un objet spécial pour la date et vous devez le traiter adéquatement avant de l'utiliser pour l'affichage) ;
- iv. Remplacez les mots « basculer » et « compléter » par les boutons d'icônes *Material-UI* comme dans la capture d'écran fournie. **Vous n'avez pas** à implémenter les fonctionnalités associées à ces boutons dans cet exercice !

Remise (10 points)

- (3 points) Votre **code source** doit être remis dans un **dépôt GitHub** nommé « tp1-memo » ; **Synchronisez une dernière fois votre code avant la remise !**
- (3 points) Votre solution doit être hébergée sur **GitHub Pages** et le lien vers le site Web doit être affiché sur la page d'accueil du dépôt GitHub ; N'oubliez pas de configurer correctement **Firebase Authentication** pour permettre l'accès à partir du nom de domaine de **GitHub Pages** (lisez le message d'erreur affiché lorsque cette configuration n'a pas encore été complétée 😊) ;
- (4 points) Remettez par Léa un seul **fichier** texte qui contient l'URL de votre **dépôt GitHub** de code source (pas le site Web de la solution !)