README Front

Installation :

Toutes les commandes sont à réaliser dans un terminal dans le dossier du projet react créé.

Ce projet utilise my-react-app. Je vous renvoie vers ce lien pour télécharger : <https://create-react-app.dev/docs/getting-started/> .

Ce projet utilise les packages axios, react-router-dom et react icons:

npm install axios

npm install -–save react-router-dom

<https://create-react-app.dev/docs/adding-a-router/>

<https://www.npmjs.com/package/react-icons>

Lancement :

Pour lancer l’application, il faut d’abord lancer le programme java s’occupant du back end. Ensuite, dans le terminal, il faut se placer dans le dossier du projet et lancer la commande :

npm start

Ainsi, le site devrait se lancer sur la page d’accueil correspondante.

**Important :**

Certaines fonctionnalités du projet ne sont plus implémentées correctement suite à l’évolution du backend. Par exemple, pour créer une question il faut une matière maintenant ce qui n’était pas le cas avant. Ainsi, il faut revoir certaines fonctionnalités.

Rooting :

Si c’est la première fois que vous lancer l’application, vous devriez arriver sur la page d’accueil pour un utilisateur non connecté. D’ici, vous avez 2 possibilités :

* Créer un compte en passant par la page dédiée, ce qui vous connectera automatiquement au compte.
* Se connecter à un compte déjà créer en passant par la page de connexion.

Une fois connecté, tous les utilisateurs ont accès à ces possibilités :

* Créer une question
* Réviser une matière
* Relancer une question (si une question proposée à était refusé)
* Consulter/Modifier son profil (accessible dans la barre de navigation)
* Se déconnecter

Le lien avec le back end n’étant pas fais sur tous ce qui concerne l’admin et l’admin root, tous les comptes ont aussi accès aux autres fonctionnalités :

* Approuver une question
* Supprimer une question
* Certifier ou supprimer un utilisateur (supprimer étant la suppression de la certification, réservé à l’admin root)

En ce qui concernes les fonctionnalités manquantes :

* La filière n’est pas prise en compte lors de la création d’utilisateur
* Tout ce qui concerne les matières et ues n’est pas implémenter.
* Les utilisateurs ne peuvent pas être certifier (et leur certification ne peut pas être retirer)
* Les listes de questions ne prennent pas en compte la certification de l’utilisateur connecté, ainsi, il peut supprimer ou valider n’importe quelle question.
* Les révisions ne sont pas liées au backend.

Cependant, il y a des lignes mise en commentaires qui servent d’ébauches à l’implémentation des différentes fonctionnalités manquantes.

Composantes :

Voici une liste avec explications des différentes composantes :

App.js, la page indiquant les différents liens accessibles.

Dans le dossier components :

-AuthContext.js est un contexte, il permet de garder certaines fonctions et variables et les réutiliser sur les différentes pages. On y retrouve les fonctions liées à l’authentification.

-Navbar.js, ce composant est la barre de navigation, elle s’adapte en fonction de si l’utilisateur est connecté ou non

-PrivateRoute.js, ce composant permet d’encadrer un lien afin que ce lien soit accessible uniquement par quelqu’un de connecté.

-PrivateRouteAdmin.js, ce composant permet d’encadrer un lien afin que ce lien soit accessible uniquement par un admin connecté.

Dans le dossier pages :

-Accueil.js, la page par défaut pour quelqu’un de non connecté, permet de renvoyer vers connexion ou création

-Création.js, la page de création de compte, la filière n’est pas prise en compte.

-Connexion, la page de connexion,

-Menu.js, la page par défaut une fois connecté, les 3 boutons du haut sont accessibles à tous et les 3 du bas uniquement aux admins

-CreationQuestion.js, la page de création d’une question

-Profil.js, la page du profil, on peut supprimer ou modifier son profil

-QuestionApprouveesList.js, la liste des question approuvées, à partir d’ici on peut en sélectionner une pour accéder aux détails.

-QuestionApprouveesDetails.js, le détail d’une question approuvé, on peut la remettre en attente ou la corriger directement.

-QuestionList.js, la liste des questions en attente d’une réponse d’un admin, on peut accéder au détail d’une question

-QuestionDetails.js, le détail d’une question en attente d’une réponse d’un admin, on peut refuser ou approuver la question.

-QuestionNopeList.js, la liste des questions de l’utilisateur qui ont été refusé par un admin, il peut accéder au détail d’une des questions

-QuestionNopeDetails.js, le détail d’une question refuser par un admin, on peut relancer ou supprimer la question.

-UtilisateurList.js, la liste des utilisateurs, permet d’accéder au détail

-UtilisateurDetails.js, le détail d’un utilisateur, permet de certifier ou supprimer une certification

-Revision.js, la page pour choisir sa matière et son mode de révision

-Question.js, la page sur laquelle se passe les révisions (sauf mode examen)

-QuestionExamen.js, la page de révision en mode examen