

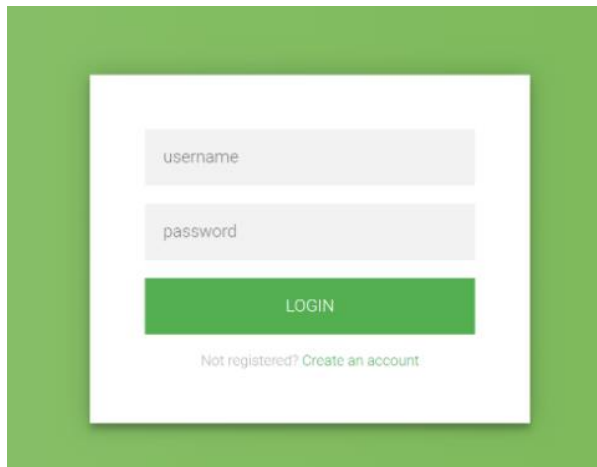
A- Configuration du store redux :

L'état initial est sous la forme suivante :

```
{
  "nom": "Funk",
  "age": "4",
  "admin": false,
  "MotDePasse": "e2EpziAy5RlpJgP",
  "pseudo": "Kaci_Reilly73",
  "prenom": "Rose",
  "couleur": "maroon",
  "Deviser": "kr",
  "Pays": "Spain",
  "avatar": "https://cloudflare-ipfs.com/ipfs/Qmd3W5DuhgHirLHGvixi6V76LhCkZUz6pnFt5AJBiyyHye/avatar/831.jpg",
  "email": "Wade34@yahoo.com",
  "photo": "https://loremflickr.com/640/480/people",
  "id": "8"
}
```

1. On ajoute l'objet au store redux lors de l'authentification
2. On supprime l'état lors de la déconnexion
3. Un utilisateur peut changer sa couleur préférée

B- Composante Login :

A login form with a green background. It features a white box containing two input fields: 'username' and 'password'. Below these fields is a green 'LOGIN' button. At the bottom of the white box, there is a link that says 'Not registered? Create an account'.

Cette Composante contient un formulaire d'authentification (Voir Image ci-dessus) :

1. Le nom d'utilisateur et le mot de passe sont obligatoires
2. Lors du click sur le bouton 'LOGIN' une requête axios est envoyée vers l'API <https://670ed5b73e7151861655eaa3.mockapi.io/Stagiaire> afin de vérifier l'existence du compte utilisateur
3. Au bout de trois tentatives, On désactive le bouton 'LOGIN'
4. Les erreurs sont affichées au-dessous du bouton 'LOGIN' sous forme d'une liste non ordonnée (ul) en couleur rouge
5. Un lien (Link) 'Create an account' qui mène vers la composante CreateAccount

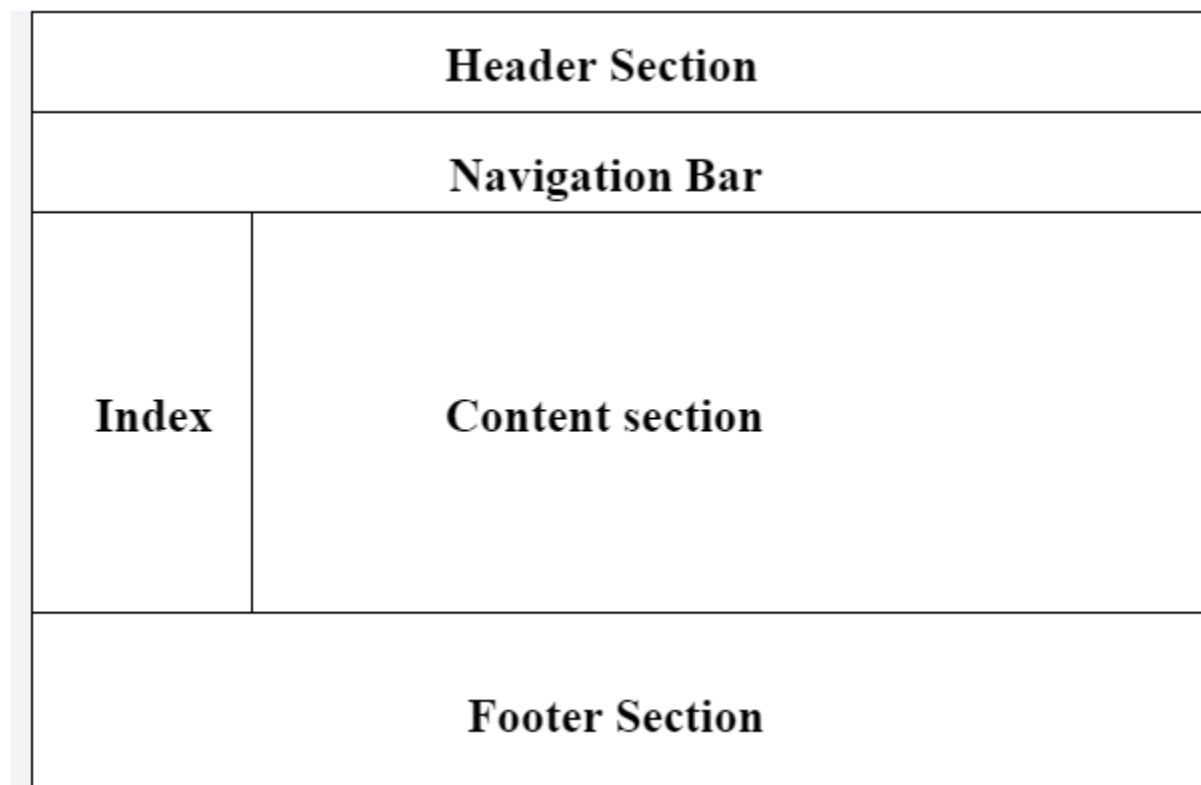
C- La Composante CreateAccount :

1. Contient un formulaire contenant des informations similaires a celles du magasin Redux (nom, âge, admin, ...)
2. Tous les champs sont obligatoires

3. Le mot de passe doit avoir au moins :
 - a. Lettre en Majuscule
 - b. Lettre en Minuscule
 - c. Un Chiffre
 - d. Un caractère spécial
 - e. 8 caractères
4. Le Mot de passe doit être confirmé
5. Si le formulaire est valide une requête axios post est envoyé vers <https://670ed5b73e7151861655eaa3.mockapi.io/Stagiaire> afin d'ajouter le nouvel utilisateur
6. En cas de succès d'ajout d'un nouvel utilisateur, on es redirigé vers la page d'authentification.

D- Layout

La Composante 'Layout' contient les Composantes suivantes



La couleur de l'arrière-plan est lue depuis le magasin redux

HeaderSection :

1. Contient le logo
2. Le nom et le prénom de l'utilisateur connecté lus depuis le magasin redux
3. Un bouton 'Se Déconnecter', lorsqu'on clique sur le bouton on redirige l'utilisateur vers le formulaire 'Login' et on efface le contenu du magasin redux



NavigationBar et Index :

Ces composantes contiennent le menu de navigation, sauf que la composante NavigationBar l'affiche horizontalement et la composante Index l'affiche verticalement.

Il y a deux menus :

Menu Admin : Affiché pour les utilisateurs de type admin

Menu Visiteur : affiché pour les visiteurs (avec attribut admin=false)

Composante	Description	Admin	Visiteur
Accueil	Affiche un message d'accueil	oui	oui
VoirMonProfile	Affiche les détails de l'utilisateur connecté. Les données sont lues depuis le magasin redux. Les données sont affichées en lecture seule.	Oui	Oui
ModifierCouleur	Affiche la couleur de l'utilisateur connecté. Une select pour choisir la nouvelle couleur. Un bouton de validation. Lors de la validation : on change la couleur dans le magasin redux, on change la couleur de l'utilisateur avec un appel API. Pour les visiteurs (non admin) ayant moins de 15 ans , On affiche un message au lieu du formulaire.	Oui	Oui
ListeUtilisateurs	Affiche la liste des utilisateurs avec CRUD opérations sous forme d'une table. Chaque ligne du tableau est une Composante séparée Il faut prévoir la création des Composantes DetailsUtilisateur, ModifierUtilisateur	Oui	Non
AjouterUtilisateur	Ajoute un nouvel utilisateur via un appel API	Oui	Non

Lorsqu'on clique sur un lien du menu, on affiche le contenu au niveau de la zone 'Content Section'

Footer :

1. Contient une adresse
2. Des Liens avec icones pour les réseaux sociaux (Facebook, instagram,...)

E- Développement

Un utilisateur (non admin) peut ajouter une demande (Titre, description) cette demande est soit acceptée ou rejetée par l'admin (sinon elle reste en attente).

L'utilisateur peut voir la liste de ses demandes (En attente, Approuvées, Rejetées) Un utilisateur peut annuler une demande si elle est encore en attente.

Ainsi l'admin peut voir les demandes en attente, les demandes qu'il a approuvées, ainsi que celles qu'il a rejetées, l'admin peut changer l'état d'une demande à n'importe quel moment

F- Références :

CRUD Links using axios

Lire liste (GET)	<pre> axios.get(`https://670ed5b73e7151861655eaa3.mockapi.io/Stagiaire`) .then((res) => { const persons = res.data; console.log(persons); this.setState({ persons }); }); </pre>
------------------	--

Lire un seul (GET)	<pre> axios.get(`https://670ed5b73e7151861655eaa3.mockapi.io/Stagiaire /`+id).then((res) => { const persons = res.data; alert(JSON.stringify(persons)); }); </pre>
Supprimer (Delete)	<pre> axios.delete(`https://670ed5b73e7151861655eaa3.mockapi.io/Stagi ai re/`+id).then(res=>{ console.log(res); this.fetchData(); }).catch(err=>{ console.log(err); }) </pre>
Modifier (PUT)	<pre> axios.put('https://670ed5b73e7151861655eaa3.mockapi.io/Stagiaire /`+id,{nom:"Ali",age:20,groupe:101}) .then (res=> { console.log(res); this.fetchData(); }).catch(err=>{ console.log(err); }) </pre>
Ajouter (POST)	<pre> axios.post('https://670ed5b73e7151861655eaa3.mockapi.io/Stagi aire',{id:Math.random(),nom:"Ali",age:20,groupe:101}) .then (res=> { console.log(res); this.fetchData(); }).catch(err=>{ console.log(err); }) </pre>

G- Travail à faire :

Héberger le code sur github.

Déployer le code en ligne.

Ajouter plus de fonctionnalités à l'application.

Rédiger un rapport.

Faire une présentation PowerPoint.