**Contrôle Continu 3**

**Groupe : DEV202**

**Durée : 2h30**

L'objectif de ce projet est de créer une application de gestion des taches des utilisateurs.

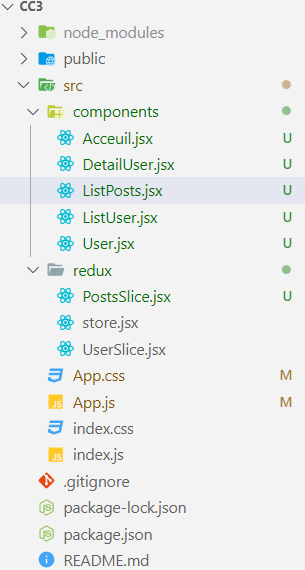
Dans ce projet, nous allons utiliser les endpoints d'API suivants :

|  |  |
| --- | --- |
| EndPoint | Description |
| https://dummyjson.com/users | Qui retourne une liste d'utilisateurs |
| https://dummyjson.com/posts | Qui retourne la liste des posts de chaque utilisateur |

1. Crier un projet React (0,25 point)

2. Ajouter les bibliothèques suivantes (0,25 point) :

1. React Icons
2. React Router
3. Axios
4. Redux Toolkit
5. React Redux

3. Structurer votre projet de la façon suivante (0,25 point):

* 1. Le dossier Component : Contient les composants de notre application
  2. Le dossier Redux : Contient les programmes de redux.

4. Dans le fichier UserSlice.jsx, créer un slice puis charger tous la liste de tous les utilisateurs retournés par l'endpoint (https://dummyjson.com/users) dans le store de redux (ces données seront rechargées au lancement du composant App) (1 point).

5. Dans le fichier PostsSlice.jsx, créer un slice puis charger toutes taches retournées par l'endpoint (https://dummyjson.com/posts) dans le store de redux (ces données seront rechargées au lancement du composant App) (1 point)

6. Dans le fichier Store.jsx, créer le store de redux puis associe ce dernier a votre projet (1 point).

7. Dans le fichier User.js, Créer le composant User qui rend l'interface suivant :



Nom & prenom de l'utilisateur

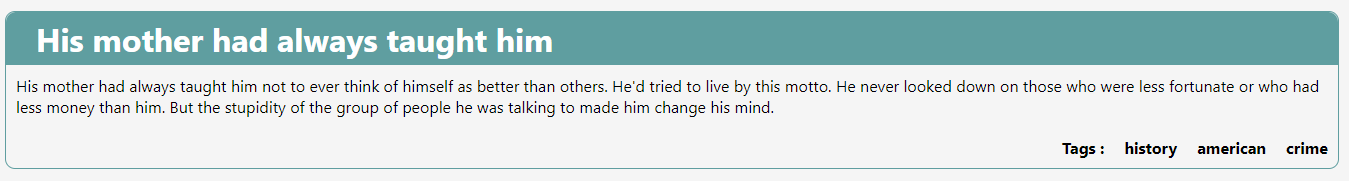
Image de l'utilisateur

L'image, le nom et le prénom de l'utilisateur sont passe à ce composant en tant que props (1 point).

7. Dans le fichier ListUser.js, Créer le composant ListUser qui va afficher une liste des utilisateurs dans le store de redux sous format d'une liste avec 4 User par ligne (2,25 points).

8. Créer le fichier DetailUser.jsx, Créer le composant DetailUser qui sera affiche après une clique sur le bouton Détail User. Ce composant va afficher le détail de l'utilisateur dont ID est passe dans l'URL. Les informations qui seront affiche : L'image, FirstName, LastName, Age, gender, Email, phone, poids, hauteur, date naissance, groupe sanguin et la couleur de l'œil (4 points).

9. Créer le fichier ListPosts.jsx, Créer le composant ListPosts qui sera affiche après une clique sur le bouton Liste des posts. Ce composant va afficher la liste des posts de l'utilisateur dont ID est passe dans l'URL. Chaque post est affiche de la facon suivante : (4 points)



tags

Body

Title

10. Au démarrage de l'application le composant Acceuil sera affiche, ce composant comprend deux parties :

- **La partie Header** : qui sera affiche dans toutes les pages.

- **La partie content** : qui sera modifié selon la page affiche par l'utilisateur. Par défaut, elle sera occupée par le composant ListUser (6 points).



Header

Content

par defaut ListUser