 <p>esprit Se former autrement HONORIS UNITED UNIVERSITIES</p>	EXAMEN : Application Côte Client 1	
	Semestre : 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> Session : Principale <input checked="" type="checkbox"/> Rattrapage <input type="checkbox"/>	
Documents autorisés : OUI <input checked="" type="checkbox"/> NON <input type="checkbox"/> Nombre de pages : 4 Calculatrice autorisée : OUI <input type="checkbox"/> NON <input checked="" type="checkbox"/> Internet autorisée : OUI <input type="checkbox"/> NON <input checked="" type="checkbox"/> Date : 13/01/2024 Heure: 9h Durée :1h30		

A noter avant de démarrer :

- 1- Seulement l'exécution du code est notée.
- 2- Les questions sont notées séparément. Si jamais vous voulez dépasser une question, vous pouvez faire ce qui vous convient pour pouvoir réaliser les questions d'après.
- 3- Vous devez utiliser json-server pour simuler la partie backend.

Enoncé :

Une entreprise souhaite développer un système de gestion de tâches pour ses employés. Chaque employé peut créer, mettre à jour et supprimer ses propres tâches. Les tâches sont organisées par projets.

Travail demandé :

- 1- Créer la classe décrivant les modèle de données suivantes : **(0.5 pt)**

Task
id ! : number title !: string dateD ! : string dateF !: string status ! : string project !: Project

Project
id! : number title !: string description! : string tasks !: Task []

- 2- Alimenter la liste des projets dans le fichier db.json par des données de votre choix tout en respectant le modèle de données donné ci-après. **(0.5pt)**
- 3- Au niveau du composant Navbar, ajoutez 2 liens, Projects et Tasks qui pointent respectivement sur les deux composants **projectsComponent** et **tasksComponent**. **(1pt)**
- 4- En cliquant dans l'url un path différent de Projects ou Tasks, une redirection se fait vers un nouveau composant **NotFoundComponent** que vous devez créer. **(1 pt)**
- 5- Par défaut le composant **loginComponent** sera chargé **(1pt)**
- 6- Lors du chargement du composant **LoginComponent**, une interface sera affichée comme illustre la Figure 1 : **(1pt)**

Projects Tasks

Nom d'utilisateur:

Nom d'utilisateur requis.

Mot de passe:

*

Le mot de passe doit avoir au moins 6 caractères.
Mot de passe doit commencer avec un caractère majuscule .

Se connecter

Figure1

- 7- Affichez un message adéquat pour chaque contrainte non respectée au niveau du formulaire sachant que : **(2.5pt)**
 - Tous les champs sont obligatoires
 - nom d'utilisateur contient au minimum 3 caractères
 - Mot de passe contient au moins 6 caractères et doit avoir le premier caractère majuscule.
 - Le bouton est désactivé tant que le formulaire est invalide.

NB : pour avoir le premier caractère en majuscule il faut respecter l'expression régulière suivante : **([^][A-Z].*)**

- 8- Une fois les contraintes sont respectées et en cliquant sur le bouton « Se connecter », une redirection vers le composant **ProjectComponent** **(0.5pt)**
- 9- Au niveau du composant **ProjectComponent**, récupérer la liste des projets à partir du fichier db.json **(1pt)**
- 10- Ajouter un bouton addTask devant chaque projet **(0.5)**

Projects	Tasks
Id: 1 Title: HealthConnect Description: HealthConnect is a digital healthcare platform that connects patients	Add Task
Id: 2 Title: EduTech Description: EduTech Innovate is an educational technology project	Add Task
Id: 3 Title: TechFarm Description: TechFarm is an agriculture technology project	Add Task

Figure 2

- 11- En cliquant sur le bouton « Add Task », un formulaire d'ajout d'un task dans le composant **addTaskComponent** sera affiché, comme illustré dans la Figure 3 : **(1.5pt)**
- La valeur de l'id du projet correspondant est envoyée dans l'url.

Add Task

Title:

Date Debut:

jj/mm/aaaa

Data fin:

jj/mm/aaaa

Add

Figure 3

- 12- Une fois les contraintes sont respectées et en cliquant sur le bouton « Add» : **(2.5pt)**
- La valeur de l'id du projet correspondant est récupérer depuis l'url.
 - la tâche est ajoutée dans la liste des tasks
 - avec un statut par défaut est « To Do»
 - une redirection se fait vers le composant appelé **tasksComponent** après l'enregistrement dans le fichier db.json via json-server comme illustré dans la Figure 4.

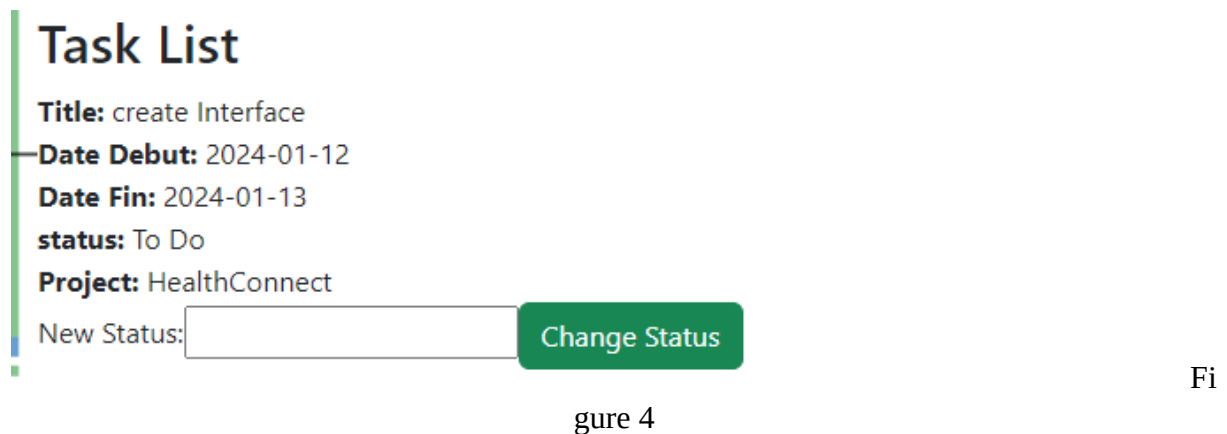
Task List

Title: create Interface Date Debut: 2024-01-12 Date Fin: 2024-01-13 status: To Do Project: HealthConnect Details
Title: Add Notification Date Debut: 2024-01-19 Date Fin: 2024-01-26 status: To Do Project: EduTech Details

Figure 4

13- En cliquant sur le lien « Details », le composant **DetailsTaskComponent** est affiché comme indiqué dans la Figure 4 : **(2,5 pts)**

- La valeur de l'id du task correspondant est envoyée dans l'url.
- Les informations affichées dans le composant **DetailTaskComponent** sont récupérées à partir du fichier db.json et correspondent aux données du task dont l'id est envoyé dans l'url
- Ajout d'un input et un bouton « change status »



The screenshot shows a web interface titled "Task List". Below the title, there are several fields displaying task information: "Title: create Interface", "Date Debut: 2024-01-12", "Date Fin: 2024-01-13", "status: To Do", and "Project: HealthConnect". At the bottom of this section, there is a form with the label "New Status:" followed by a text input field. To the right of the input field is a green button labeled "Change Status".

Figure 4

14- Dans la zone de saisie affichée dans la **Figure 4**, vous pouvez changer le status existant et en cliquant sur le bouton « change status », la valeur de l'attribut « status » est mis à jour **(2.5 pt)**

15- Au-dessous de la liste des tasks un bouton qui permet de calculer le nombre totale des tasks qui ont comme status « DONE » **(1,5pts)**

Bonne chance ☺