Examen - M1 S1 2024

Objectif

Vous allez partir d'une application quasiment vierge, et créer le back-end et le front-end d'un système de gestion des livres pour une bibliothèque.

Le back-end est écrit avec le framework NestJS, et le front-end avec React. Le front-end utilise la librairie next. (La stack exacte vue en cours). Pour la base de données, vous utiliserez uniquement SQLite.

Dans un groupe (~ 4 personnes), vous vous séparerez le travail comme vous le souhaitez (certains uniquement sur le back, les autres uniquement sur le front, une personne par fonctionnalité, ...): le groupe entier sera noté sur l'intégralité de la codebase, qu'il ait participé à telle ou telle partie ou pas.

Votre objectif sera de rajouter le maximum de features demandées, tout en gardant un code le plus propre possible. La propreté, l'organisation de votre code et le respect des règles vues en cours seront ici plus importants que le nombre de fonctionnalités implémentées.

Vous pourrez fournir un rapport de la manière dont vous vous êtes organisés, des difficultés rencontrés etc.. si vous le souhaitez.

Rendu

Vous rendrez un repository Github, dans lequel on pourra voir l'évolution de votre codebase à travers l'historique de commit:

- Est-ce que les fonctionnalités sont arrivées petit à petit

- Est-ce que chaque commit n'a pas, pendant un moment, cassé la codebase sur `main` ou apporté des régressions

Le Github sera à rendre pour le **15 Novembre 2024 à 23h59**.

Tout changement qui sera appliqué sur le repository GitHub après cette date sera comptabilisé dans la correction, mais sera considéré comme du retard (pareil si vous ne m'envoyez pas de lien de repository GitHub).

Soit `x` le nombre de retard, le malus est de 2^(x-1) points.

Consignes

Pour le style de l'application, je n'ai pas fait de maquettes. Quand ce n'est pas spécifié, libre à vous de faire les composants qui vous paraissent le plus simple.

Quand c'est spécifié, merci de respecter le composant demandé, et de ne pas utiliser d'autres design system que Tailwind CSS, qui est déjà intégré dans l'application. Si la fonctionnalité fonctionne, mais qu'elle n'utilise pas le composant demandé, une pénalité sera appliquée.

Les composants front devront être correctement séparées, de manière à être ré-utilisable.

Les composants de base (bouton, modale, liste, input, ...), partagées à travers l'application, ne doivent contenir que la logique d'implémentation.

Vous séparerez en front :

- providers
- composants
- modèles
- pages

et en back :

- controller
- service
- repository
- DTO
- presenter
- modèles

Tous les composants de formulaire auront un label permettant de savoir à quoi correspond le champ. (On peut utiliser un composant spécifique par type d'input)

Vous ne retirerez pas de règle ESLint. Si il y a des erreurs sur le front-end ou sur le back-end, une pénalité sera appliqué (une erreur ignorée sans justification (un commentaire l'expliquant suffira) sera comptée comme une erreur).

Quand vous le pouvez, vous utiliserez la programmation fonctionnelle, en évitant donc au maximum les `if` et les `for` (lorsque faire se peut évidemment).

Vous préciserez explicitement les types :

- de vos variables
- de vos composants
- de vos fonctions

Point de départ

Modèle de données

Aucun type n'existe pour le moment dans l'API ou dans le front, libre à vous de les créer.

Routes existantes

FRONT

- `/` → page d'accueil

Fonctionnalités à implémenter

L'ordre n'est pas représentatif ici, les fonctionnalités sont simplement groupées par notions fonctionnelles.

Menu et layout

1. Implémenter un menu de navigation dans l'application

Votre application ne possède actuellement aucun moyen de naviguer entre les pages.

La forme de ce menu est libre, l'important est que ou que je sois dans l'application, je peux aller sur ces pages :

- Page d'accueil
- Liste des livres
- Liste des auteurs

2. Chaque page a un titre permettant de l'identifier facilement (tous ces titres doivent se présenter de la même manière)

3. Chaque page dont la route n'est pas une racine a un breadcrumb (exemple: https://mui.com/material-ui/react-breadcrumbs/)

Bibliothèque

Page bibliothèque (`/books`)

1. Je vois la liste de mes livres
    a. Je peux voir le nom, la date d'écriture et l'auteur de chaque livre.
    b. Je vois la note moyenne d'un livre (une fois les avis implémentés)
2. J'ai une barre de recherche me permettant de filtrer les livres par titre.
3. Je peux choisir le tri de mes livres
4. Je peux ouvrir une **modale** me permettant de créer un nouveau livre.
    a. Je choisis son titre, sa date de publication, son auteur

Détails d'un livre (`/books/:id`)

1. J'accède à la page de détail d'un livre quand je clique sur l'un dans la liste.
2. La page de détail contient :
    a. Le titre du livre
    b. Le prix du livre
    c. L'année de publication du livre
    d. Le nom de l'auteur (un lien vers la page de l'auteur est présent)
3. Je peux supprimer un livre. J'ai une **modale** de confirmation avant que l'opération ne se fasse réellement.
4. Chaque livre, sur sa page de détails, a un fil d'avis.
    a. Un avis va entre 1 et 5 étoiles.
    b. Ils sont affichés dans un **drawer** (https://mui.com/material-ui/react-drawer/)
    c. Ils peuvent s'accompagner d'un commentaire écrit qui est optionnel.
    d. La liste des avis est triable par date de création ascendante ou descendante

Auteurs

Page auteurs (`/authors`)

1. J'ai une page avec la liste des auteurs, pour chaque auteur je vois :
    a. Son nom.
    b. Sa photo.
    c. Le nombre de livres qu'il a écrit.
    d. La moyenne pondérée des avis de tous ses livres (une fois les avis implémentés)
2. Je peux filtrer mes auteurs via une barre de recherche.
3. Je peux ajouter un nouvel auteur via une **modale**.

Page détails d'un auteur (`/authors/:id`)

1. Je peux cliquer sur un auteur pour aller sur sa page de détails
    a. Son nom
    b. Sa photo
    c. Sa biographie
    d. La liste de ses livres avec un lien vers la page de détails d'un livre
2. Je peux modifier les informations d'un auteur
3. Je peux créer / supprimer des livres à un auteur
4. Je peux supprimer un auteur. J'ai une modale me permettant de confirmer ou d'annuler mon choix.

Rendu

Pour rendre le projet, vous devrez passer par un repository Github.

Dès qu'il est fait, vous pouvez m'envoyer le lien par mail.

Help

Si vous avez une question, vous pouvez me contacter sur la boîte mail `gerald.gallet@ext.junia.com`.

Github

Les commandes

`git add .` → préparer tous mes changements pour un commit

`git commit -m "<type>(<scope): <message>"` → créer votre commit.

`git push` → envoyer votre commit

`git pull` → récupérer le travail distant

`git switch <branch name>` → changer de branche

`git switch -c <branch name>` → créer une branche

`git commit --amend --no-edit` → envoyer mes changements dans le commit d'avant, sans changer son nom

`git commit --amend` → envoyer mes changements dans le commit d'avant, en changeant son nom