

Module : Application Côté Client 2

Enseignants : UP Web

Classes : 4TWIN

Documents autorisés : OUI ☒ NON ☐

Nombre de pages : 6

Calculatrice autorisée : OUI ☐ NON ☒

Internet autorisée : OUI ☐ NON ☒

Date : 12/04/2023

Heure : 11h45

Durée : 01h30

NB : seul le code exécutable sera pris en considération lors de la correction

Toute carte wifi active est une tentative de fraude.

Étude de cas : (20 pts)

Une salle de cinéma envisage de créer un site web MovieDB qui permettra aux utilisateurs de consulter les informations sur les films. L'application permettra également aux utilisateurs de créer et de gérer une liste de souhaits de films. Pour réaliser ce projet, vous utiliserez la bibliothèque React JS 18. Afin de développer cette application, vous devrez respecter les consignes ci-dessous :

- L'utilisation de react-bootstrap et bootstrap est **facultative**.
- Respecter l'arborescence des composants suivante :

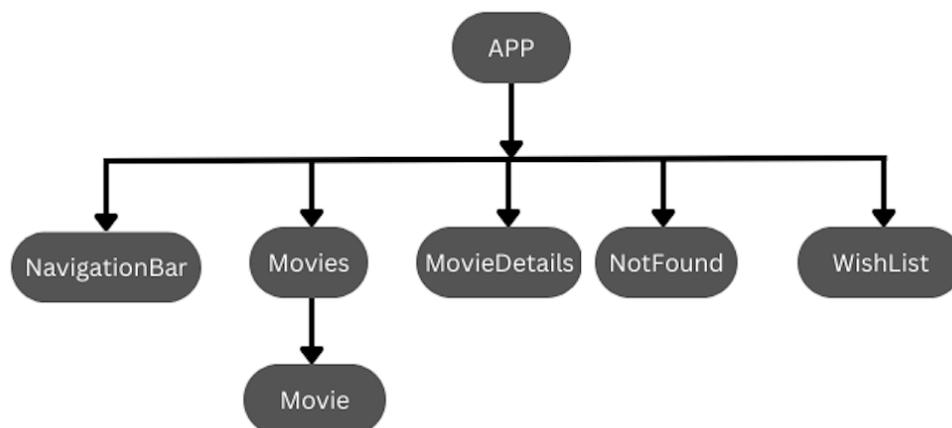


Figure 1 : Arborescence des composants

Sachant que :

- Tous les composants sont des composants fonctionnels sauf le composant **NotFound** est un composant de Classe.

Travail à faire :

Partie A (14,5):

- 1) Créer le fichier **movies.json** sous le dossier src contenant le code suivant :

```
[
  {
    "id": 1,
    "title": "Greyhound",
    "year": "2020",
    "img": "placeholder.jpg",
    "genre": "drama",
    "description": "Le lorem ipsum est, en imprimerie..... "
  },
  {
    "id": 2,
    "title": "Green Book",
    "year": "2018",
    "img": "placeholder.jpg",
    "genre": "comedy",
    "description": "Le lorem ipsum est, en imprimerie....."
  }
]
```

Figure 2 : Contenu du fichier **movies.json**

- 2) Créer les composants **Movies** et **Movie** afin d'afficher les informations des films en intégrant le contenu du fichier json. (2pts)

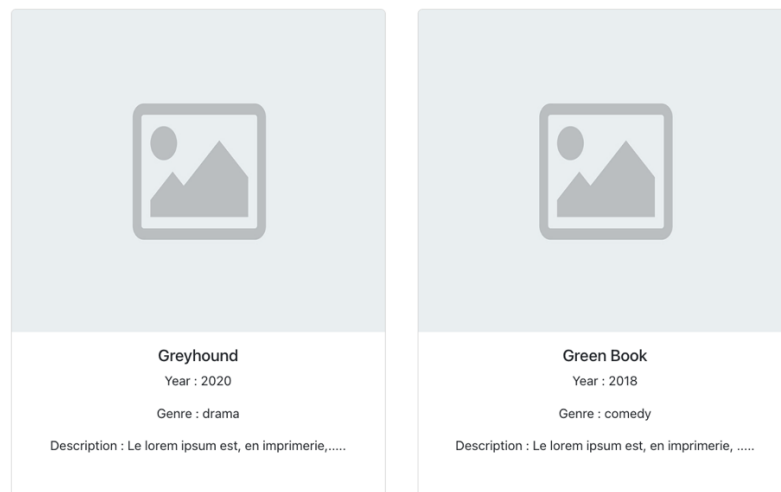
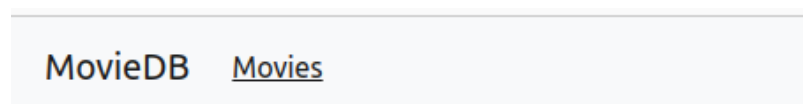


Figure 3 : Affichage des films

- 3) Créer le composant **NavigationBar** comme le montre la figure 4 en configurant les routes nécessaires (2pts)
 - La route active dans la **NavigationBar** doit être soulignée.



- 4) Créer le composant de classe **NotFound** qui sera affiché si l'utilisateur accède à une route qui n'existe pas. (2pts)
 - Un message « **Redirect to Movies page** » ne s'affiche qu'après le montage du composant, après 3 secondes il disparaît.
- 5) Créer une barre de recherche comme le montre la figure suivante qui a pour objectif de chercher un film bien déterminé à partir de son nom (**title**). (3,5 pts)
 - Si l'utilisateur clique sur le bouton « **Submit** » avec un paramètre de recherche vide, la liste de tous les films disponibles sera affichée.

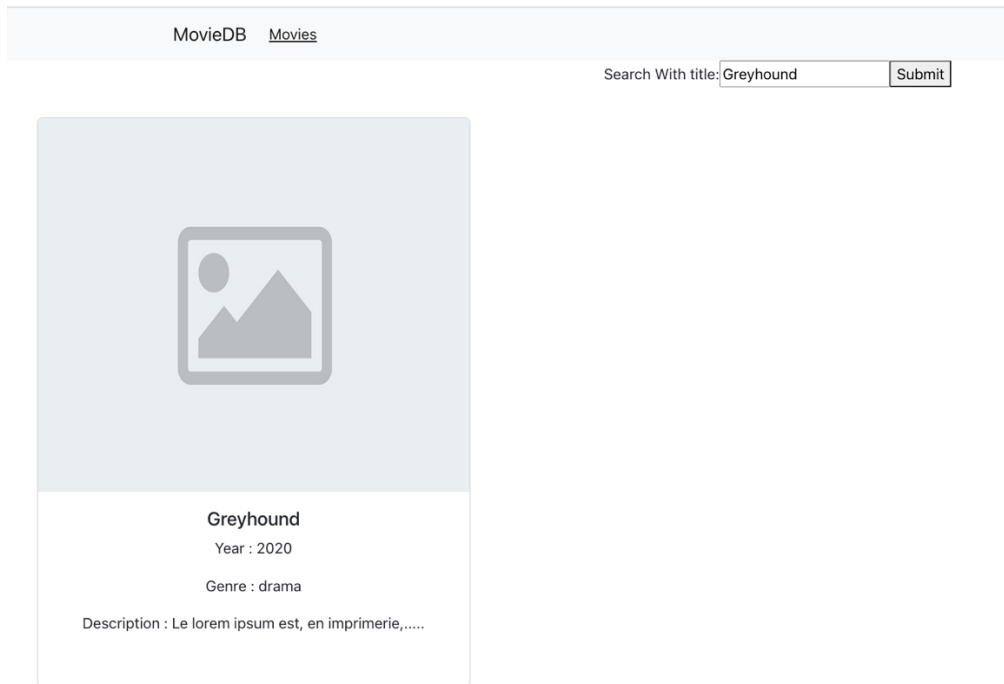


Figure 5 : Résultat de Recherche à partir de la liste des films

- Si le film n'existe pas un message « **No result found** » sera affiché dans l'interface.

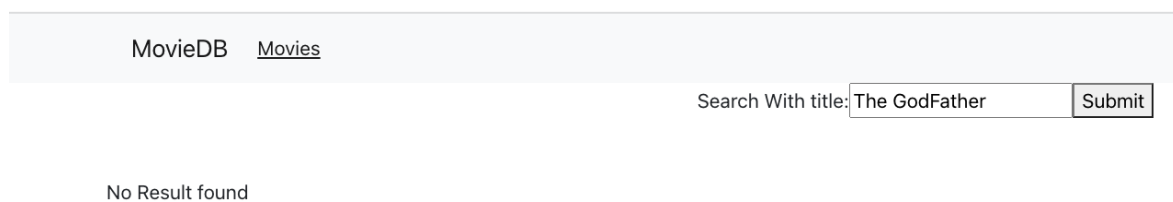


Figure 6 : Résultat de Recherche des films n'existe pas

- 6) Créer la nouvelle interface du composant **MovieDetails** qui affiche les détails d'un film. (1,5 pts)
 - Lorsqu'un utilisateur clique sur le titre d'un film, il sera redirigé vers la nouvelle interface.
 - Si aucun film n'est trouvé, un message « **Movie not found** » sera affiché à l'utilisateur pour l'informer que le film demandé n'est pas disponible.

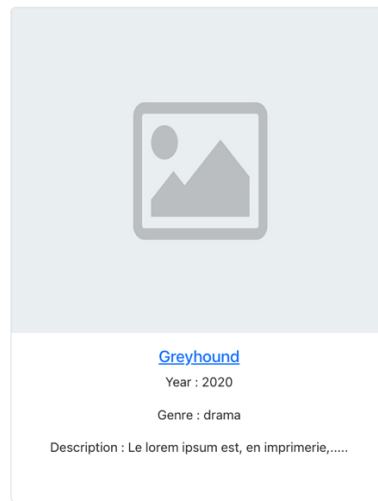


Figure 7 : Lien details d'un film

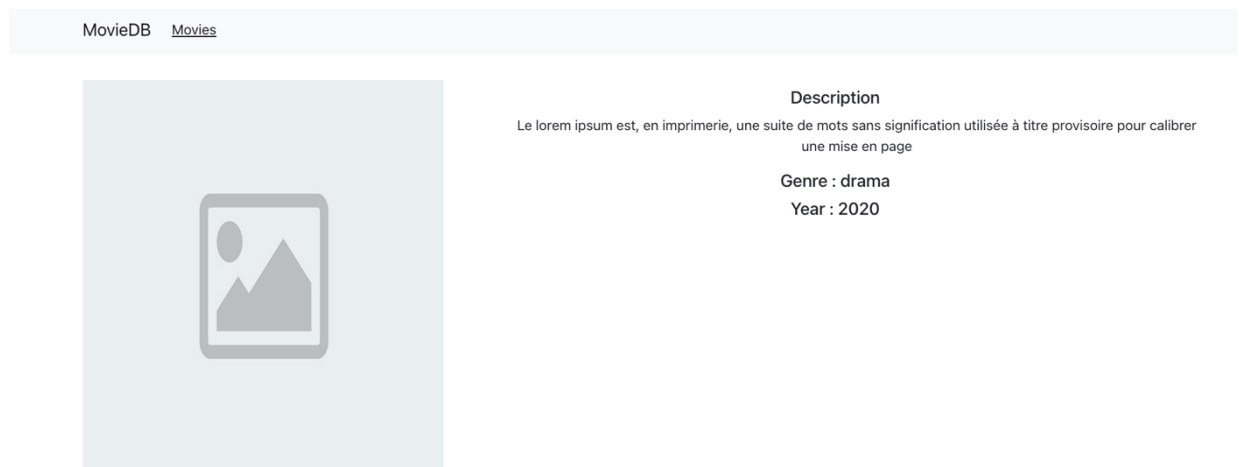


Figure 8 : Interface details d'un film

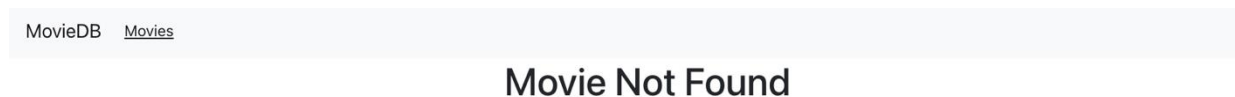


Figure 9 : Affichage du message "Movie not found"

7) Un utilisateur peut donner une note à un film comme présenté dans la figure 10 : (3,5pts)

- S'il n'y pas de note, un message « **No ratings yet** » sera affiché. Sinon, la moyenne des notes sera affichée.
- La note saisie par l'utilisateur doit être comprise entre 1 et 5. Sinon, une alerte JS sera affichée avec un message « **Please enter a rating between 1 and 5** ».

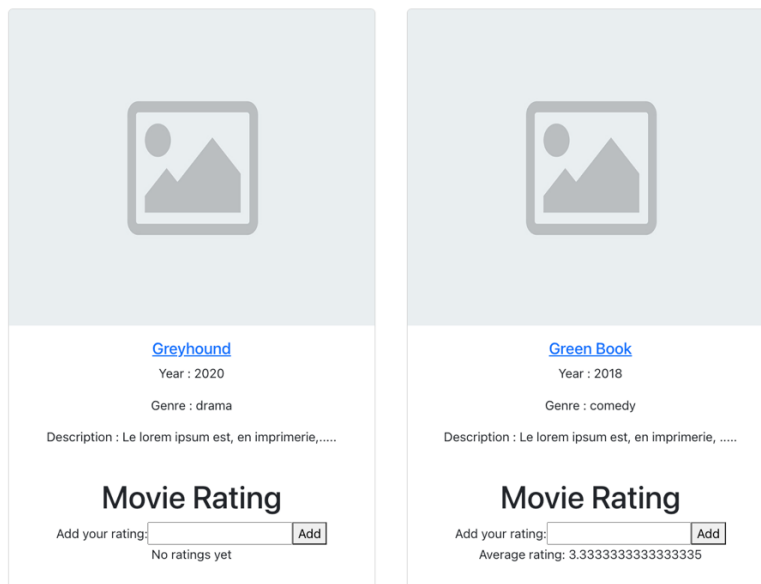


Figure 10 :Interface avec le rating

Partie B (5,5 pts):

Dans cette partie , nous souhaitons intégrer la fonctionnalité de liste de souhaits (Wishlist) à l'application MovieDB en utilisant **Redux Toolkit**. Les utilisateurs auront la possibilité d'afficher, d'ajouter et de supprimer des films de leur liste de souhaits.

Lorsqu'un utilisateur clique sur le bouton « **Add to wishlist** », le film sera ajouté à la liste des souhaits. (3pts)

- Un message sera affiché « **Added to wishlist** ».
- Le message va disparaître après 2 secondes.
- Le nombre d'éléments dans la wishlist sera incrémenté.

Si le film existe déjà dans la wishlist, il ne sera pas ajouté à nouveau et un message d'alerte « **Movie already exist** » sera affiché.

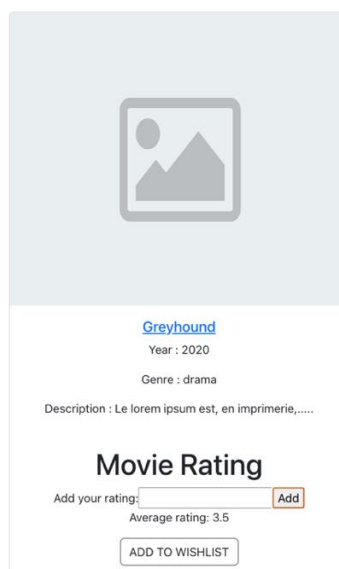


Figure 11 :Interface Movie avec bouton «Add to Wishlist »

- 2) Si l'utilisateur clique sur le lien de la wishlist dans la barre de navigation, il sera redirigé vers l'interface de la wishlist. **(0,5 pt)**

Faire le nécessaire afin d'avoir une barre de navigation comme le montre la figure suivante :

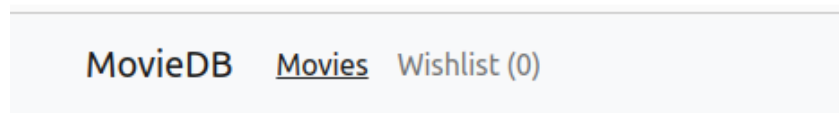


Figure 12 : Barre de navigation finale

- 3) Lorsqu'un utilisateur clique sur le bouton « **X** » situé à côté d'un film spécifique, ce film sera supprimé de la liste. **(1pt)**
- 4) Si l'utilisateur clique sur « **Clear Wishlist** », tous les films seront supprimés de la wishlist. **(1pt)**

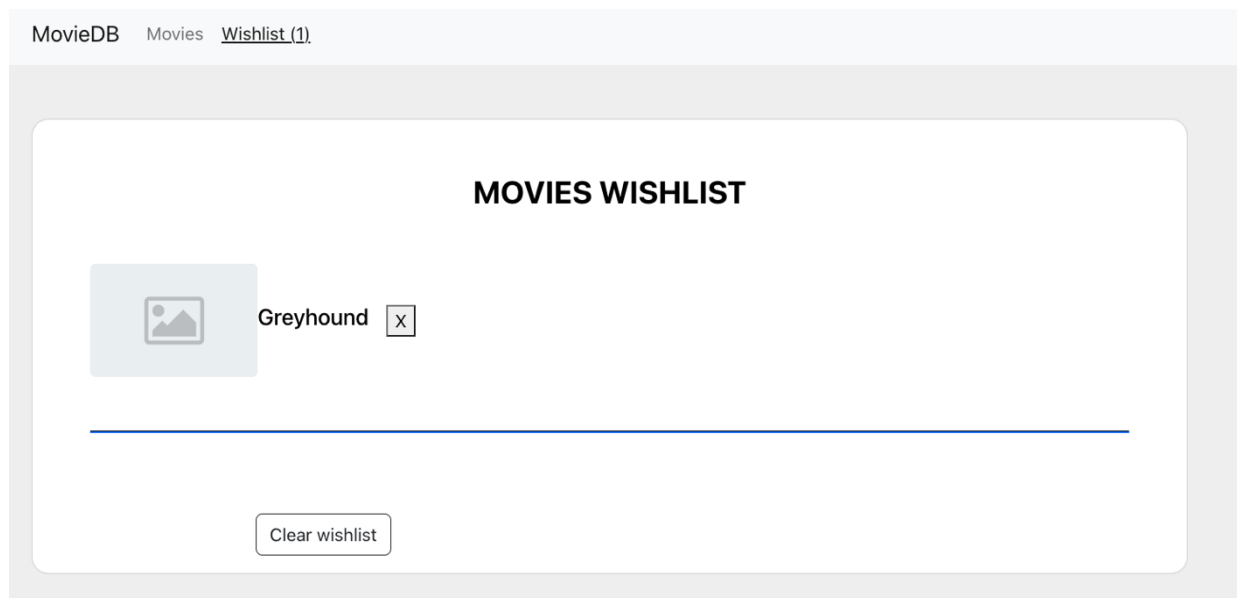


Figure 13 : Interface "Wishlist"

Bon Courage