

## Consignes générales

Vous devez créer une **application web** suivant les fonctionnalités indiquées dans ce document. Cette application permet à l'utilisateur de se **connecter** et de gérer sa **bibliothèque de livres**.

## Modalités

- L'examen dure **2 heures**.
- Vous avez accès aux **slides** des séances durant l'année.
- Vous avez accès au **code des démos** et **solutions des exercices**.
- Vous ne pouvez **pas** accéder à **internet**.
- Vous ne pouvez **pas** utiliser d'outils d'**intelligence artificielle générative**.

## Contraintes techniques

- Votre application web doit être composée d'un **frontend** et d'un **backend**
- Votre frontend doit être développé en **TypeScript** avec **React**
- Votre frontend doit être une **Single Page Application**
- Le routage des pages du frontend doit se faire avec la librairie **react-router(-dom)**
- Votre backend doit être développé en **TypeScript** avec **Express**
- Votre backend doit suivre les **principes REST**
- Votre backend ne doit proposer que des routes **pertinentes** à cette application
- Votre frontend doit communiquer avec votre backend au moyen de **requêtes HTTP**
- L'authentification des requêtes du backend doit se faire par **JWT**

## Mise en place

**Téléchargez** l'archive **examen\_js.zip** disponible sur EvalMoodle et **décompressez-la**. Vous trouverez dedans :

- **frontend** : un **boilerplate** pour commencer le développement de votre frontend. Il correspond principalement au résultat de la commande **npm create vite@latest**.
  - Le boilerplate contient une image **cover.jpg** dans le dossier **src/assets**. Cette image servira d'image par défaut pour les fonctionnalités demandées.
- **backend** : un **boilerplate** pour commencer le développement de votre backend. Il correspond principalement au clone du repository **jwt-ts-api-boilerplate** vu au cours.
  - Le boilerplate contient en plus un fichier **books.ts** dans le dossier **services** et une interface **Book** dans le fichier **types.ts**. Vous pourrez y retrouver une liste de livre par défaut à utiliser pour votre examen.
  - Nous attirons votre attention sur le fait que la liste des utilisateurs par défaut comportent un utilisateur « admin » avec comme mot de passe « admin ».
- **.git** : un dossier git caché. Il nous permet de retrouver facilement le code que vous avez développé durant l'examen. **Veuillez ne pas le supprimer ni faire de commit**.

Dans ces boilerplates, vous retrouverez également les dossiers **node\_modules** déjà installés pour fonctionner sur les machines de l'école sans internet. En cas de problème, vous pourrez retrouver ces éléments fournis en redécompressant l'archive.

# Fonctionnalités

## Frontend

Chaque page de votre application web doit afficher les liens de navigation suivants :

- **Accueil** : page d'accueil, route /
- **Bibliothèque** : page de bibliothèque, route /library

L'application doit également proposer une page de détails d'un livre à la route /book/:id même s'il n'y a pas de lien de navigation pour cette page.

Chaque page de votre application web doit également afficher un footer affichant votre prénom et votre nom.

### Page d'accueil

La page d'accueil de votre application web doit afficher un formulaire de login tant que l'utilisateur n'est pas connecté. Si la connexion échoue, la page d'accueil doit afficher un message d'erreur en dessous du formulaire.

Quand l'utilisateur est correctement connecté, la page d'accueil affiche à la place un message de bienvenue avec le nom de l'utilisateur, et un bouton de déconnexion.

### Page de bibliothèque

Si l'utilisateur n'est pas connecté, la page de bibliothèque n'affiche qu'un message d'erreur invitant l'utilisateur à se connecter.

Autrement, la page de bibliothèque affiche la liste des livres de la bibliothèque de l'utilisateur. Pour chaque livre, la page doit afficher son nom et un bouton amenant à la page de détails de ce livre.

### Page de détails d'un livre

Si l'utilisateur n'est pas connecté, la page de détails n'affiche qu'un message d'erreur invitant l'utilisateur à se connecter.

Autrement, la page de détails affiche le nom du livre, son auteur, son année d'écriture et son image de couverture. Si aucune image n'est disponible, la page de détails affiche l'image par défaut **cover.jpg** à la place.

## Backend

Le backend doit permettre à n'importe quel utilisateur de se connecter. En plus de ça, les utilisateurs connectés peuvent lire la liste de tous les livres et lire les données d'un livre.

Le backend ne doit pas proposer d'autres fonctionnalités. Les fichiers et fonctions du boilerplate qui ne sont plus utiles peuvent être supprimés pour plus de clarté.

## Soumission

Quand vous avez terminé l'examen, veuillez **effacer** les dossiers **node\_modules** de vos applications frontend et backend. **Compressez** ensuite votre dossier contenant les deux applications au format **.zip**. Renommez le fichier résultant en **NOM\_PRENOM.zip** en remplaçant votre nom et votre prénom. **Soumettez** finalement ce fichier sur **EvalMoodle**.

## Évaluation

Vous serez évalués principalement sur les fonctionnalités de l'application sur base d'un scénario de tests fonctionnels. L'esthétique de votre site web n'est pas évaluée, sauf si elle bloque certaines des fonctionnalités.

Nous vérifierons également la qualité du code fourni. Vous devez créer du code facilement lisible et maintenable, et qui respecte les contraintes demandées.