

Cahier des Charges : Développement d'une Application Frontend en React pour une Gestion de Bibliothèque

Objectif

Développer une application frontend en React pour interagir avec l'API RESTful de gestion de bibliothèque. Cette application permettra aux utilisateurs de visualiser et de gérer les livres, auteurs, emprunteurs et emprunts.

Contexte

L'application doit offrir une interface utilisateur intuitive pour permettre aux administrateurs de la bibliothèque de gérer efficacement les collections de livres, les auteurs, les emprunteurs et les emprunts.

Fonctionnalités de l'Application

Gestion des Livres

- Afficher la liste de tous les livres.

- Afficher les détails d'un livre spécifique.

- Ajouter un nouveau livre.

- Modifier un livre existant.

- Supprimer un livre.

Gestion des Auteurs

- Afficher la liste de tous les auteurs.

- Afficher les détails d'un auteur spécifique.

Ajouter un nouvel auteur.

Modifier un auteur existant.

Supprimer un auteur.

Gestion des Emprunteurs

Afficher la liste de tous les emprunteurs.

Afficher les détails d'un emprunteur spécifique.

Ajouter un nouvel emprunteur.

Modifier un emprunteur existant.

Supprimer un emprunteur.

Gestion des Emprunts

Afficher la liste de tous les emprunts.

Afficher les détails d'un emprunt spécifique.

Enregistrer un nouvel emprunt.

Modifier un emprunt existant (par exemple, la date de retour).

Supprimer un emprunt.

Exigences Techniques

Technologies

React pour la construction de l'interface utilisateur.

Axios (ou Fetch API) pour les appels API.

React Router pour la navigation entre les différentes pages.

Pages et Composants

Livres

Liste des livres (BooksList)

Détail d'un livre (BookDetail)

Formulaire d'ajout/modification de livre (BookForm)

Auteurs

Liste des auteurs (AuthorsList)

Détail d'un auteur (AuthorDetail)

Formulaire d'ajout/modification d'auteur (AuthorForm)

Emprunteurs

Liste des emprunteurs (BorrowersList)

Détail d'un emprunteur (BorrowerDetail)

Formulaire d'ajout/modification d'emprunteur (BorrowerForm)

Emprunts

Liste des emprunts (LoansList)

Détail d'un emprunt (LoanDetail)

Formulaire d'ajout/modification d'emprunt (LoanForm)

Navigation

Utiliser React Router pour gérer la navigation entre les différentes pages de l'application.

Créer un menu de navigation pour accéder facilement aux sections Livres, Auteurs, Emprunteurs et Emprunts.

Appels API

Utiliser Axios (ou Fetch) pour effectuer les appels API vers l'API RESTful de la bibliothèque.

Créer un service API pour encapsuler toutes les opérations de requête.

Validation des Formulaires

Utiliser Formik pour la gestion des formulaires.

Utiliser Yup pour la validation des données des formulaires.

Gestion des Erreurs et Notifications

Afficher des messages d'erreur et de succès après les opérations CRUD.

Utiliser des composants de notification comme Snackbar de Material-UI pour informer les utilisateurs.

Étapes de Développement

Initialisation du Projet

Initialiser un nouveau projet React avec Create React App.

Installer les dépendances nécessaires (React Router, Axios, Formik, Yup, Material-UI/Bootstrap, Redux).

Création des Composants et Pages

Développer les composants pour chaque fonctionnalité (liste, détail, formulaire).

Configurer React Router pour la navigation entre les pages.

Appels API

Créer des services API pour encapsuler les appels à l'API REST.

Intégrer les appels API dans les composants de l'application.

Validation des Formulaires

Implémenter les formulaires avec Formik.

Ajouter la validation des données avec Yup.

Design et Interface Utilisateur

Utiliser Material-UI ou Bootstrap pour le design de l'interface utilisateur.

Créer un menu de navigation et des composants de mise en page.

Gestion des Erreurs et Notifications

Ajouter la gestion des erreurs pour les appels API.

Utiliser des composants de notification pour informer les utilisateurs des résultats des opérations.

Livrables

Code source de l'application React.

Documentation utilisateur pour expliquer comment utiliser l'application.

Documentation du développeur pour décrire l'architecture et les composants du code.