

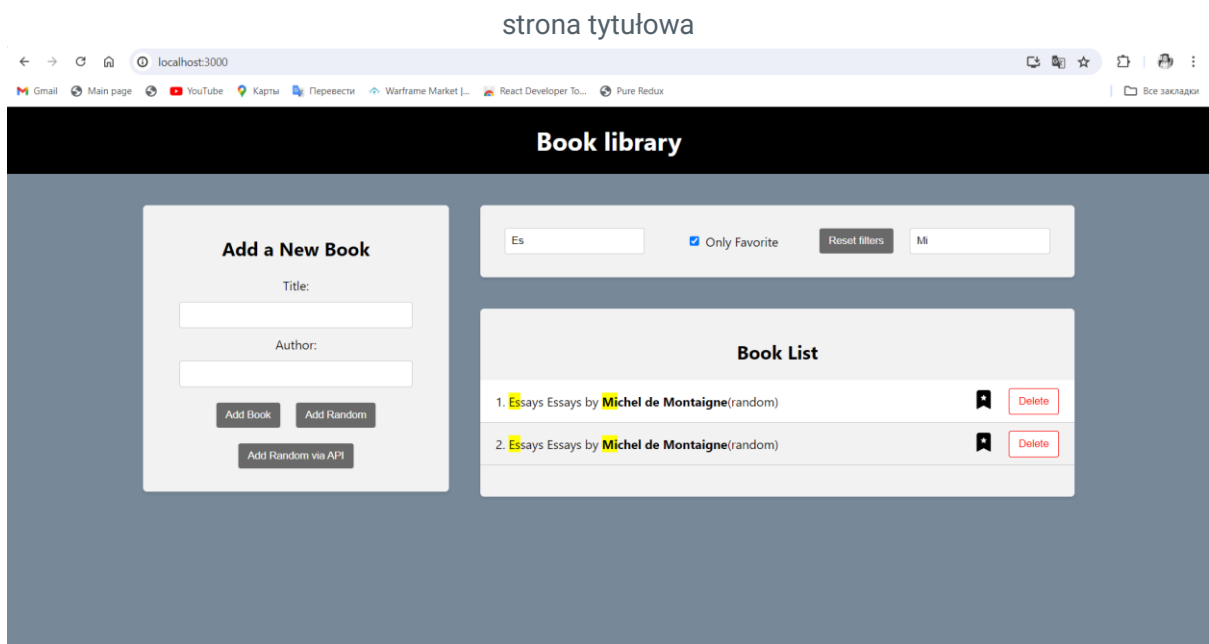
WAŻNE!!

Przed uruchomieniem należy zainstalować Node.js i npm. W folderach api i frontend trzeba zainstalować zależności za pomocą polecenia npm install. Następnie uruchom lokalny serwer w katalogu frontend poleceniem npm start.

Opis projektu

Stworzyłem aplikację "Library" na podstawie biblioteki React i Redux, używając stanów. Moja aplikacja zawiera trzy komponenty - listę książek, formularz do dodawania książek oraz listę do wyświetlania książek. W formularzu znajdują się trzy przyciski, każdy dodaje książki. Jeden przycisk - "Add Book" - dodaje książkę ręcznie, drugi dodaje losowo z pliku JSON, w którym przechowywane są obiekty książek, oraz jest przycisk, który zawiera książki, emulując dodawanie książek z zewnętrznego serwera. Aby wszystko działało, należy uruchomić serwer Node.js, instalując zależności za pomocą polecenia npm start.

Aplikacja zawiera również filtr książek, który śledzi filtry oraz ulubione książki, zmieniając wartość z true na false, gdy klikniesz ikonę. Jest też pole wyboru, które wyświetla tylko ulubione książki, a także formularz, który przechodzi niewielką walidację. Jeśli książki nie są wypełnione, w prawym górnym rogu pojawi się błąd informujący o pustym formularzu. Jeśli nie ma odpowiedzi z serwera, również pojawi się błąd. Dodatkowo jest spinner i ładowanie podczas oczekiwania na odpowiedź z serwera.



Na zrzucie ekranu w prawym górnym rogu widać komunikat o błędzie, który pojawia się, gdy odpowiedź z serwera nie nadchodzi, czyli jest odrzucona

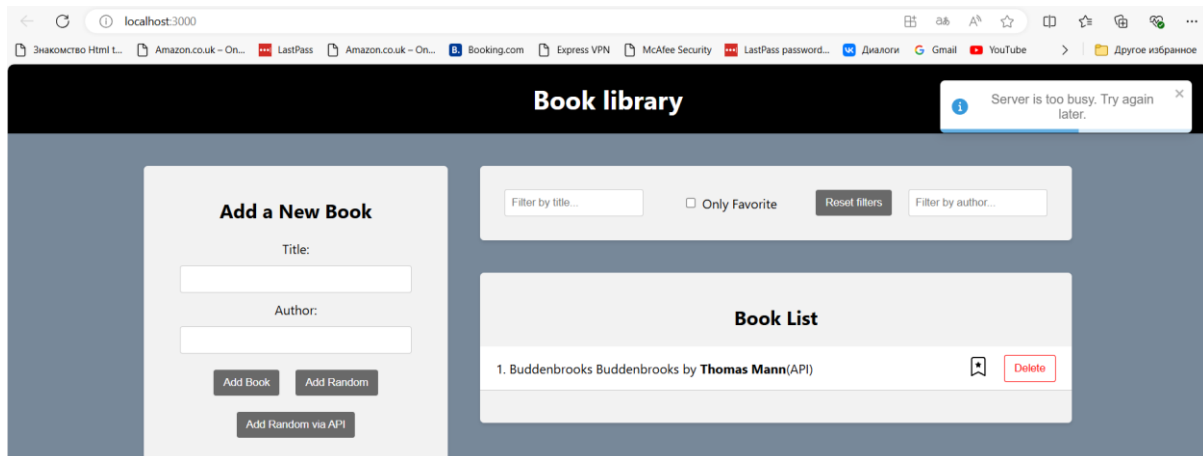
W projekcie wykorzystane zostały następujące moduły środowiska programistycznego:

1. React - Biblioteka JavaScript do budowania interfejsów użytkownika.
2. Redux - Kontener stanu aplikacji dla JavaScript, często używany z React.
3. JavaScript - Język programowania używany do tworzenia dynamicznych i interaktywnych elementów na stronach internetowych.
4. HTML - Język znaczników używany do tworzenia struktury stron internetowych.
5. CSS - Język arkuszy stylów używany do opisu wyglądu i formatowania dokumentów napisanych w HTML.
6. JSON - Format wymiany danych używany do reprezentowania obiektów i danych strukturalnych w sposób czytelny dla człowieka i maszyny.

Opis funkcjonalności

1. **Zapewnienie formularza i dodanie książki:** Użytkownik wypełnia formularz zawierający szczegóły książki, której chce dodać do systemu. Dane są walidowane, a następnie książka jest zapisywana w lokalnej bazie danych.
2. **Dodawanie losowej książki z bazy danych:** System wybiera losową książkę z istniejącej bazy danych i prezentuje ją użytkownikowi. Po potwierdzeniu, książka jest zapisywana w lokalnej bazie danych.

3. **Dodawanie książki z zewnętrznego serwera:** Użytkownik wysyła zapytanie do zewnętrznego serwera w celu uzyskania danych o książce. Po otrzymaniu danych, książka jest prezentowana użytkownikowi do potwierdzenia i zapisywana w lokalnej bazie danych.
4. **Filtrowanie książek:** Użytkownik może filtrować książki według tytułu i nazwiska autora. System wyświetla tylko te książki, które spełniają kryteria wyszukiwania.
5. **Dodawanie książki do ulubionych:** Użytkownik może dodawać książki do listy ulubionych. Po dodaniu książka jest zaznaczana jako ulubiona w systemie.



Jeśli limit zapytań na serwerze został osiągnięty, w prawym górnym rogu możesz zobaczyć odpowiedni błąd, lub jeśli serwer nie odpowiada w ogóle - błąd sieciowy.