

# Dokumentacja projektu - System Biblioteczny

## Zależności

Projekt wykorzystuje następujące zależności:

### Backend:

- **Spring Boot:**
  - Spring Data JPA
  - Spring Web
  - Spring Security
  - Spring Validation
  - Lombok
  - MySQL Connector
  - JSON Web Token (JWT)

### Frontend:

- **React (z wykorzystaniem bibliotek):**
  - Material-UI (MUI)

## Opis

System biblioteczny jest aplikacją webową umożliwiającą zarządzanie książkami w bibliotece. Aplikacja wspiera funkcjonalności CRUD dla książek oraz umożliwia wypożyczanie książek przez użytkowników.

## Backend

Backend został zrealizowany w Spring Boot i obsługuje operacje:

- Tworzenie, odczyt, aktualizowanie i usuwanie danych książek.
- Obsługę wypożyczeń książek z weryfikacją dostępności.
- Autoryzację i uwierzytelnianie za pomocą JWT.

## Frontend

Frontend aplikacji został zrealizowany w React. Główne funkcjonalności to:

- Wyświetlanie listy książek wraz z ich szczegółami (tytuł, autor, dostępność).
- Wypożyczanie książek przez użytkowników.
- Edycja szczegółów książek z natychmiastową aktualizacją na backendzie.
- Przyjazny interfejs użytkownika dzięki Material-UI.

## Baza danych

Dane aplikacji są przechowywane w bazie MySQL.

## Endpointy

W projekcie dostępne są następujące endpointy:

### Endpointy publiczne:

- POST `/authenticate` - Logowanie użytkownika i generowanie tokenu JWT.
- POST `/register` - Rejestracja użytkownika

### Endpointy chronione (wymagają tokenu JWT):

#### Zarządzanie książkami:

- GET `/api/books` - Pobranie listy wszystkich książek.
- GET `/api/books/{id}` - Pobranie szczegółów konkretnej książki.
- POST `/api/books` - Dodanie nowej książki.
- PUT `/api/books/{id}` - Aktualizacja danych książki.
- DELETE `/api/books/{id}` - Usunięcie książki.

#### Zarządzanie wypożyczeniami:

- POST `/api/loans` - Wypożyczenie książki.
- GET `/api/loans` - Pobranie listy wypożyczeń użytkownika.
- DELETE `/api/loans/{id}` - Zakończenie wypożyczenia książki.

## Struktura projektu

- **Backend:** Projekt oparty na Spring Boot z modularnym podziałem na kontrolery, serwisy i repozytoria.
- **Frontend:** Komponenty React, zarządzające stanem aplikacji i komunikacją z API backendowym.