Wirtualna Biblioteka

Adam Boros

sem. 2025Z

1 Wstęp

Aplikacja Wirtualna Biblioteka to narzędzie webowe umożliwiające użytkownikom zarządzanie zasobami bibliotecznymi w sposób zdalny. Celem aplikacji jest stworzenie platformy, która pozwala na łatwe przeglądanie oraz wypożyczanie książek - w założeniu ma realizować cechy fizycznej biblioteki w trybie zdalnym. Dzięki systemowi bazodanowemu, aplikacja zapewnia pełną kontrolę nad książkami, użytkownikami oraz historią wypożyczeń. Jest to idealne rozwiązanie zarówno dla indywidualnych użytkowników, jak i dla bibliotek, które chcą cyfryzować swoje zasoby i umożliwić dostęp do nich online.

2 Zakres funkcjonalny

1. Rejestracja i logowanie użytkowników

-Użytkownicy mogą rejestrować się w systemie, tworząc konto z unikalnym identyfikatorem (e-mail, hasło).

-Możliwość logowania się do systemu z wykorzystaniem hasła i loginu.

2. Zarządzanie kontem użytkownika

-Możliwość edytowania danych użytkownika, takich jak imię, nazwisko, adres e-mail, hasło.

-Przeglądanie historii wypożyczeń i terminów zwrotów.

3. Przeglądanie książek

-Wyświetlanie dostępnych książek w systemie, z możliwością filtrowania po kategorii, autorze, tytule czy dacie wydania.

-Każda książka ma opis (autor, wydawca, rok wydania, dostępność).

-Możliwość przeszukiwania książek według różnych kryteriów.

4. Wypożyczanie książek

-Użytkownicy mogą wypożyczać dostępne książki na określony czas.

5. Zarządzanie książkami przez administratorów

-Administratorzy mogą dodawać, edytować i usuwać książki z bazy.

-Możliwość zarządzania kategoriami książek (np. literatura piękna, naukowa, historia).

-Monitorowanie historii wypożyczeń, zwrotów i dostępności książek.

-Generowanie raportów na temat popularności książek, liczby wypożyczeń i opóźnionych zwrotów.

6. Wyszukiwanie zaawansowane

-Umożliwienie wyszukiwania książek według zaawansowanych kryteriów (autor, kategoria, data wydania, dostępność).

-Możliwość sortowania wyników wyszukiwania według popularności, daty dodania lub oceny.

3 Architektura aplikacji

Aplikacja *Wirtualna Biblioteka* została zaprojektowana z wykorzystaniem nowoczesnych technologii, w której skład wchodzą:

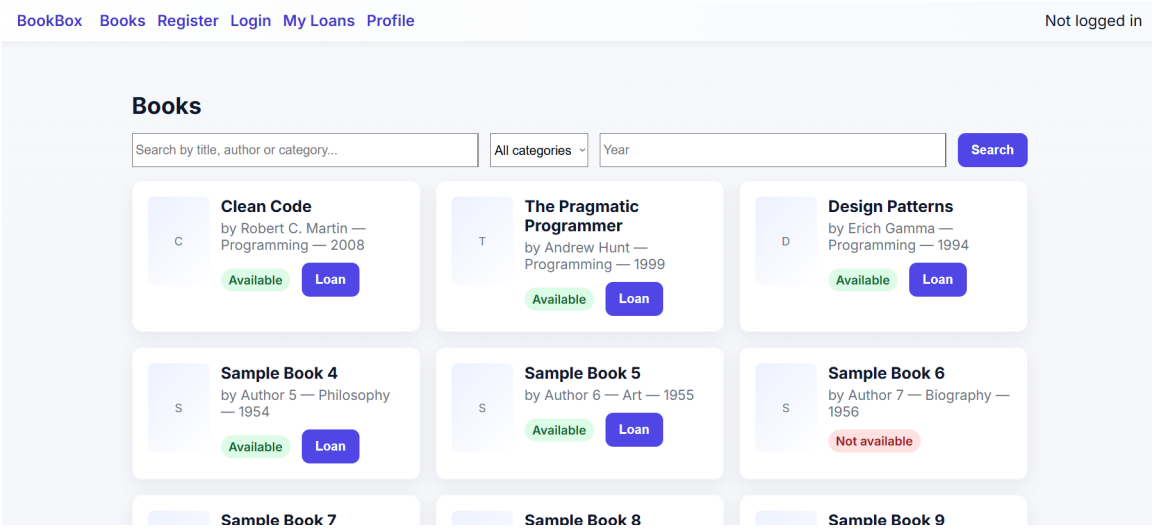
- **Backend:**
  - **Spring Boot** - framework do tworzenia aplikacji webowych w Javie.
  - **Spring Security** - biblioteka do zarządzania logowaniem użytkowników i uprawnieniami oraz bezpieczeństwem systemu.
  - **Spring Data JPA** - ułatwia komunikację z bazą danych.
- **Frontend:**
  - **React** - biblioteka JavaScript do budowania interaktywnych interfejsów użytkownika.
- **Baza danych:**
  - **MySQL** - relacyjna baza danych, w której przechowywane będą dane aplikacji (użytkownicy, książki, wypożyczenia).
- **Konteneryzacja:**
  - Aplikacja będzie uruchamiana w kontenerze Docker, co zapewni spójne środowisko pracy zarówno lokalnie, jak i na serwerze.

4 Realizacja aplikacji

Poniżej przedstawiono kluczowe elementy gotowej implementacji aplikacji *Wirtualna Biblioteka*, odnosząc je bezpośrednio do założeń z sekcji **Zakres funkcjonalny** oraz przyjętej architektury z sekcji **Architektura aplikacji**.

4.1 Strona główna i przeglądanie książek (funkcje z pkt 3 i 6)

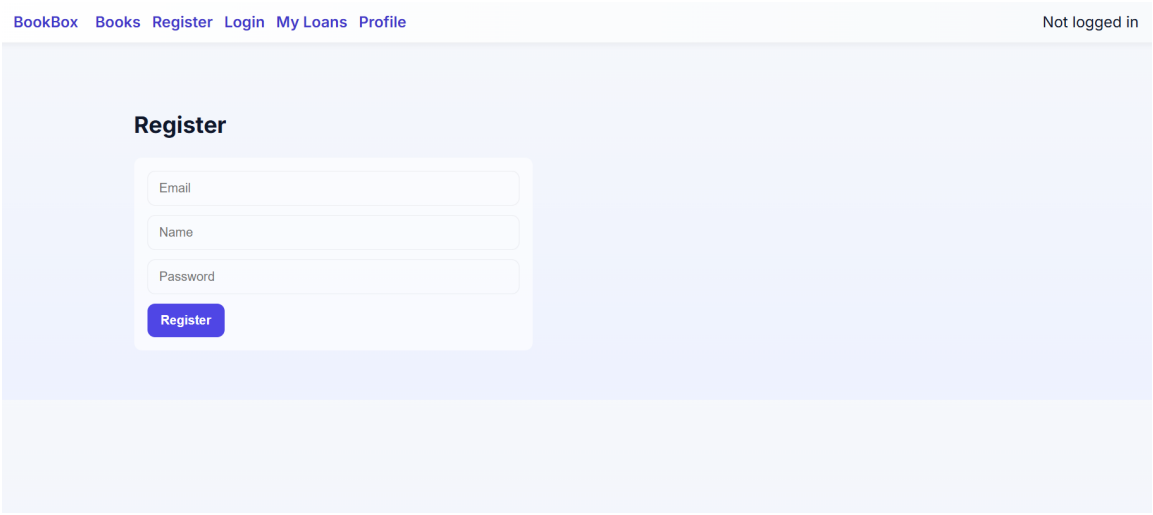
Po uruchomieniu aplikacji użytkownik widzi stronę główną z listą książek. Widok ten realizuje podstawową funkcję przeglądania zasobów bibliotecznych (pkt 3), a także wyszukiwanie i filtrowanie/sortowanie wyników (pkt 6). Przy każdej pozycji prezentowane są kluczowe informacje (np. tytuł, autor, rok wydania, dostępność), co odpowiada opisowi książki wskazanemu w pkt 3.



Rysunek 1: Strona główna aplikacji i przegląd katalogu książek (pkt 3 i 6).

4.2 Rejestracja użytkownika (funkcje z pkt 1)

Proces tworzenia konta realizuje założenia z pkt 1 sekcji **Zakres funkcjonalny**: rejestracja z użyciem unikalnego identyfikatora (adres e-mail) oraz hasła. Po poprawnym utworzeniu konta użytkownik może przejść do logowania i rozpocząć korzystanie z systemu.



Rysunek 2: Widok rejestracji użytkownika (pkt 1).

4.3 Logowanie i dostęp do funkcji użytkownika (funkcje z pkt 1)

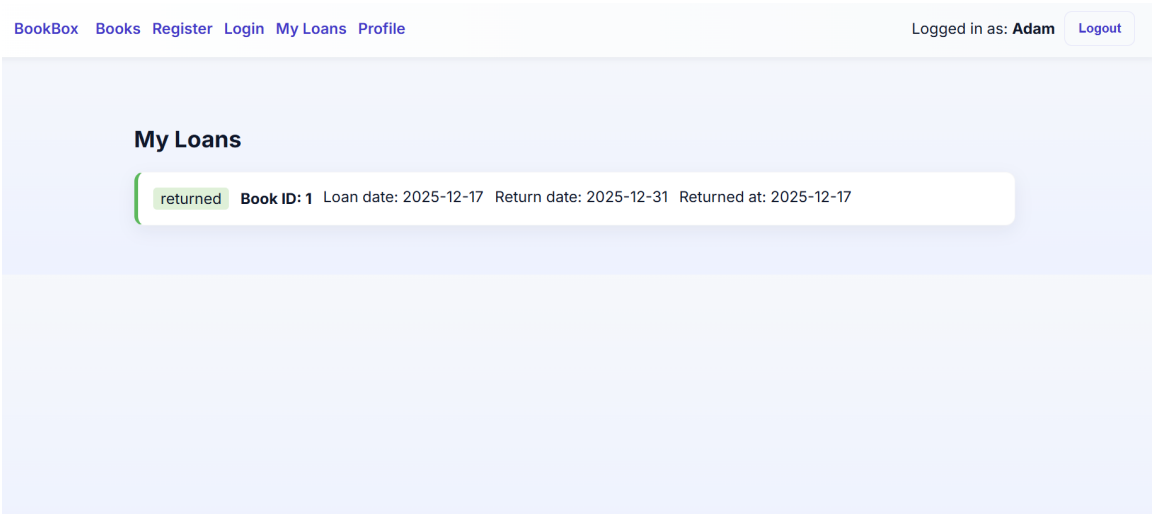
Logowanie realizuje wymagania z pkt 1: autoryzacja użytkownika przy użyciu loginu (e-mail) i hasła. Mechanizm ten jest spójny z architekturą opisaną wcześniej: backend oparty o Spring Boot i Spring Security kontroluje dostęp do zasobów, a interfejs React udostępnia dedykowane widoki zależnie od roli użytkownika.

Rysunek 3: Widok logowania do systemu (pkt 1).

4.4 Wypożyczanie i zwroty (funkcje z pkt 4 oraz historia z pkt 2)

Użytkownik może wypożyczyć dostępną książkę na określony czas (pkt 4). Po wykonaniu akcji system aktualizuje stan dostępności książki oraz zapisuje zdarzenie w historii wypożyczeń

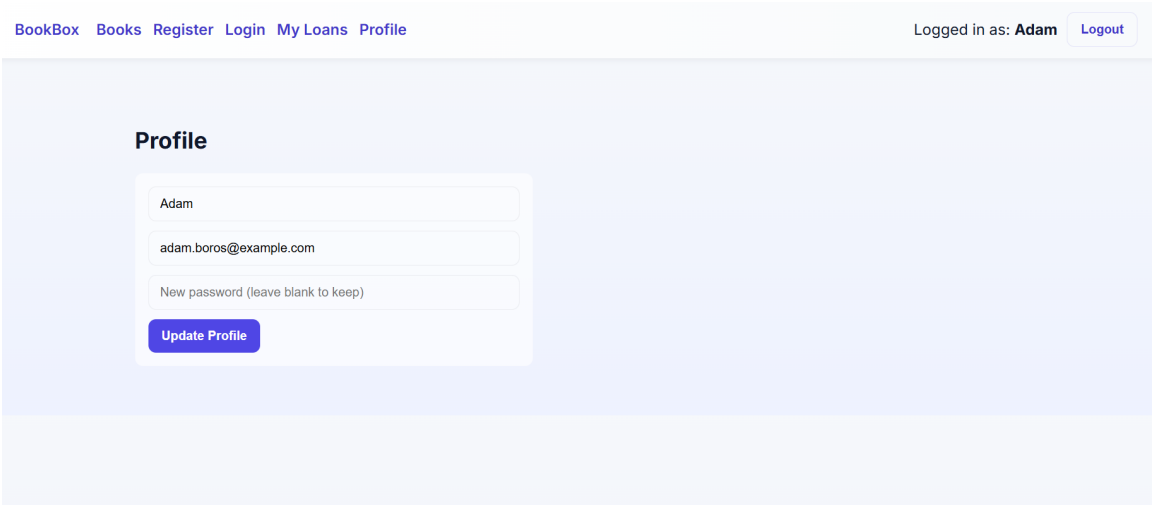
użytkownika (pkt 2). Widok na rys. 4 przedstawia perspektywę użytkownika po zwrocie książki (zaktualizowany status wypożyczenia), co stanowi bezpośredni dowód działania mechanizmu zwrotów i spójności danych w bazie.



Rysunek 4: Perspektywa użytkownika: wypożyczenie po zwrocie / zaktualizowany status (pkt 4 oraz pkt 2).

4.5 Zarządzanie kontem użytkownika (funkcje z pkt 2)

W aplikacji zaimplementowano możliwość edycji podstawowych danych konta (pkt 2), w szczególności zmianę adresu e-mail oraz hasła. Funkcja ta pozwala utrzymać aktualność danych i zwiększa bezpieczeństwo konta użytkownika, pozostając spójna z przyjętym podejściem do autoryzacji i kontroli dostępu.

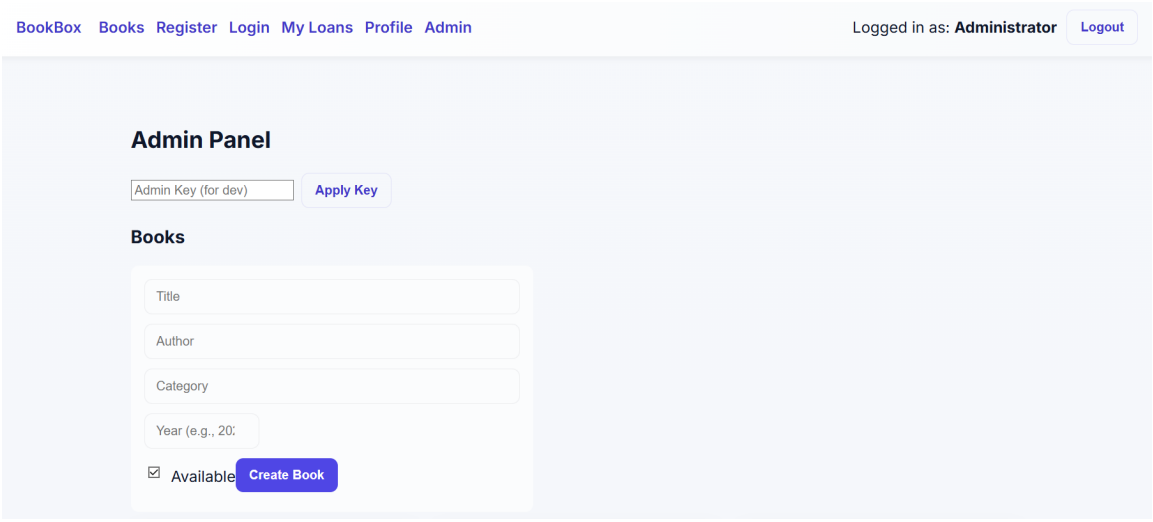


Rysunek 5: Edycja profilu: zmiana e-maila i hasła (pkt 2).

4.6 Panel administratora: zarządzanie książkami (funkcje z pkt 5)

Zgodnie z pkt 5 sekcji **Zakres funkcjonalny**, administrator posiada narzędzia do zarządzania katalogiem książek (dodawanie/edycja/usuwanie). Rysunek 6 prezentuje fragment panelu admi-

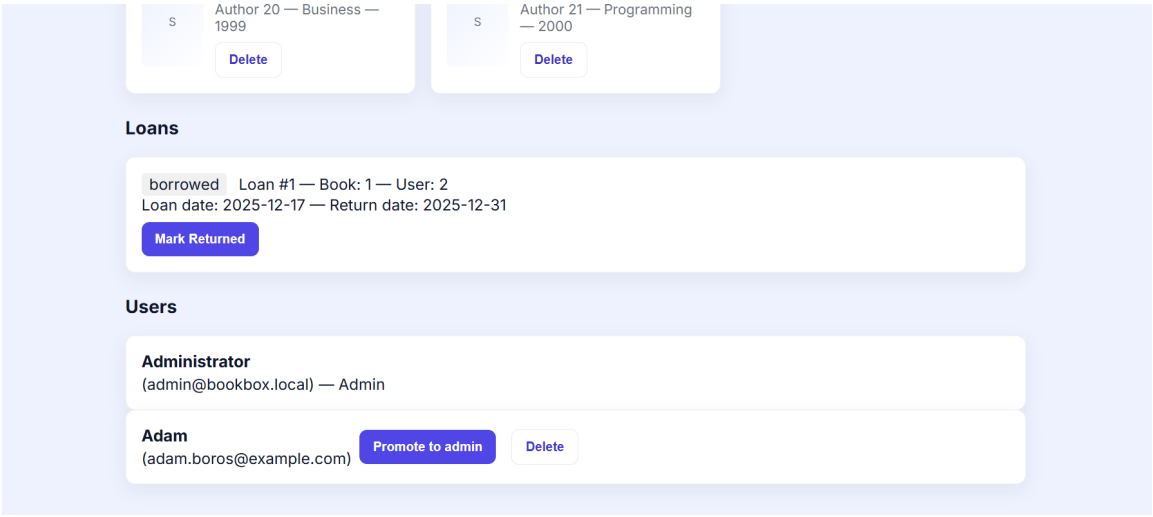
nistracyjnego związany z dodawaniem nowych pozycji do bazy, co przekłada się bezpośrednio na aktualizację zasobów widocznych później na stronie głównej (pkt 3).



Rysunek 6: Panel administratora: dodawanie książek do bazy (pkt 5).

4.7 Panel administratora: obsługa wypożyczeń i użytkowników (funkcje z pkt 5)

W ramach rozszerzeń administracyjnych (pkt 5) zaimplementowano widoki umożliwiające: monitorowanie wypożyczeń oraz oznaczanie zwrotów, a także przegląd listy użytkowników wraz z operacjami administracyjnymi (np. usunięcie konta lub nadanie uprawnień administratora). Funkcje te wspierają kontrolę nad systemem i zapewniają zgodność z wymaganiem „monitorowania historii wypożyczeń, zwrotów i dostępności".



Rysunek 7: Panel administratora: oznaczanie zwrotów, lista użytkowników oraz akcje administracyjne (pkt 5).

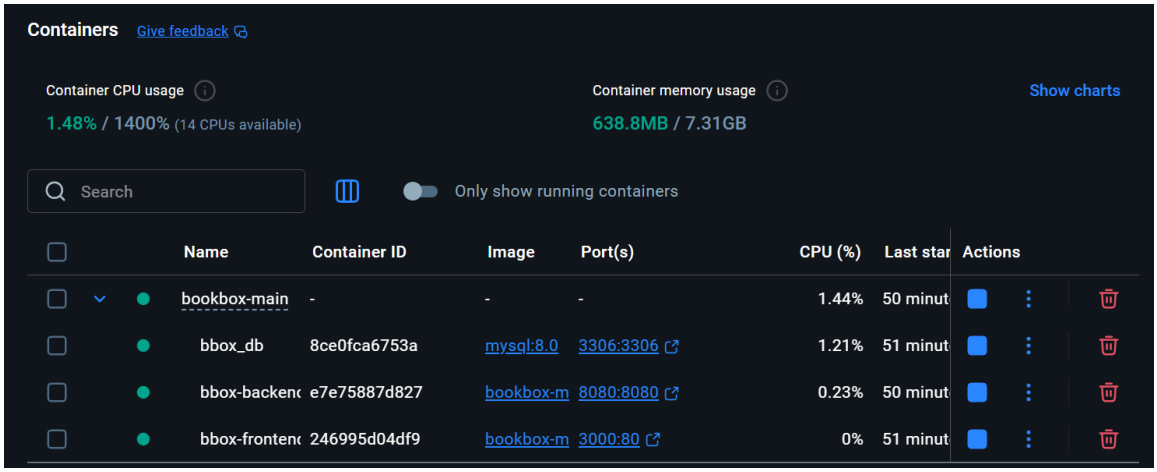
4.8 Spójność z architekturą (sekcja 3)

Całość implementacji jest zgodna z przyjętą architekturą: warstwa frontendowa (React) odpowiada za interakcję użytkownika i prezentację danych, natomiast backend (Spring Boot) realizu-

je logikę biznesową, autoryzację (Spring Security) oraz operacje na danych (Spring Data JPA) przechowywanych w relacyjnej bazie MySQL. Uruchamianie w środowisku kontenerowym Docker zapewnia powtarzalność konfiguracji i ułatwia wdrożenie aplikacji.

4.9 Konteneryzacja i uruchomienie środowiska (sekcja 3)

Zgodnie z założeniami opisanymi w sekcji **Architektura aplikacji** aplikacja została uruchomiona w środowisku kontenerowym Docker, co zapewnia powtarzalność konfiguracji i szybkie odtworzenie środowiska na innym komputerze. W ramach jednego zestawu kontenerów uruchomiono bazę danych MySQL (`bbox_db`, mapowanie portu `3306:3306`), backend Spring Boot (`bbox-backend`, `8080:8080`) oraz frontend React (`bbox-frontend`, `3000:80`). Taki podział odpowiada warstwowej architekturze (frontend/backend/baza) i ułatwia niezależne rozwijanie oraz wdrażanie poszczególnych komponentów.



Rysunek 8: Działające kontenery Docker aplikacji (frontend, backend, MySQL) wraz z mapowaniem portów.

5 Dostęp do aplikacji

Repozytorium projektu

github.com/adampb13/BookBox