System rekomendacyjny polecający książki w bibliotece

Patryk Nikonowicz,
Ewa Pasterz,
Piotr Skibiński,
Paulina Szostek

Czewiec 2022

Spis treści

1 Opis tematu

Wybrany przez nas temat polega na stworzeniu systemu rekomendacyjnego polecającego książki w bibliotece. Nasz system jest zaimplementowany w formie strony internetowej połączonej z bazą danych. By otrzymywać rekomendacje trzeba założyć indywidualne konto czytelnika. Dane o użytkowniku i książkach są zapamiętywane między sesjami.

W naszym systemie jest dostępnych ok. 11000 książek, które użytkownik może oznaczyć jako przeczytane i je ocenić. Wszystkie książki posiadają informacje takie jak tytuł, autor, wydawnictwo, grupa docelowa czy język.

Do tworzenia rekomendacji przyjęliśmy podejście content-based. Książki będą polecane użytkownikowi na podstawie już wystawionych przez niego ocen i podobieństwa między ocenionymi książkami a tymi jeszcze nieprzeczytanymi.

2 Model danych

Nasze dane reprezentują rzeczywiste książki. Wyodrębniliśmy 13 atrybutów, które przetrzymujemy dla każdej pozycji i zgodnie z którymi tworzymy rekomendacje. Poniżej wymieniliśmy atrybuty, które zdefiniowaliśmy:

1. Tytuł
2. Autor
3. Wydawca
4. Gatunki
5. Grupa docelowa
6. Poruszone tematy
7. Liczba stron
8. Język
9. Kraj pochodzenia
10. Data wydania

11. Ocena
12. Liczba wystawionych ocen
13. Liczba wypożyczeń w miesiącu

Poruszane tematy są innym atrybutem niż gatunki, choć mogą wydawać się podobne. Gatunek to przykładowo romans czy horror, a poruszane tematy to wampiry czy wojna stuletnia. Ocena będzie wyliczana jako średnia ocen wystawionych przez użytkowników. Tytuł nie jest uwzględniany w algorytmie rekomendacyjnym. Dane o książkach zostały pobrane z Kaggle i losowo uzupełnione o brakujące informacje.

3 Użytkownik

Każda osoba chcąca skorzystać z naszego systemu musi posiadać swój indywidualny profil. Do założenia konta potrzebny jest unikalny adres e-mail.

Zalogowany użytkownik może oznaczać książki obecne w naszej bazie jako przeczytane. Może im wystawić ocenę oraz oznaczyć już przeczytaną książkę jako jedną z ulubionych. Nasz system będzie zapamiętywał oceny użytkownika i na tej podstawie będziemy określać preferencje owego czytelnika.

4 Użyte technologie

Algorytm został zaimplementowany w języku Python przy użyciu biblioteki Pandas. Dane do biblioteki zostały pobrane z strony Kaggle. Jest to strona udostępniająca najróżniejsze bazy danych przeznaczone do nauki uczenia maszynowego. My skorzystaliśmy z bazy "Goodreads-books", którą następnie przerobiliśmy tak, aby spełniała nasze wymagania.

Frontend naszej aplikacji został wykonany jako strona internetowa przy użyciu TypeScript. Biblioteka, której użyliśmy to React. Backend naszej aplikacji został wykonany w C# przy użyciu ASP.NET Core. Aby umożliwić logowanie się użytkowników przeprowadziliśmy połączenie z dostawcą tożsamości. Dostawca tożsamości działa na podstawie biblioteki IdentityServer4.

5 Algorytm

5.1 Opis

Nasz algorytm działa jedynie dla użytkowników, którzy już oznaczyli co najmniej jedną książkę jako przeczytaną. Zatem nowy użytkownik powinien rozpocząć używanie systemu od ocenienia paru książek. Ewentualnie może filtrować bazę naszych książek by wyświetlić wysoko oceniane lektury o interesujących go cechach.

Nasza strategia dla użytkowników z większym stażem przyjmuje, że książki będą polecane na podstawie tytułów oznaczonych przez użytkownika jako ulubione. W naszej implementacji ograniczamy się do rekomendacji na podstawie 10 książek. Proces porównawczy polega na znalezieniu książek jak najbardziej podobnych do tych lubianych przez użytkownika, wykorzystując hybrydowo podobieństwo cosinusowe i odległość Hamminga. Odległość Hamminga jest wykorzystana do określenia podobieństwa poruszonych tematów i gatunków, gdyż te atrybuty mogą zawierać w sobie wiele wartości. Podobieństwo cosinusowe jest wykorzystane do określenia podobieństwa pozostałych wartości, z wyłączeniem oceny i częstości wypożyczeń. Wartości tych kolumn są wykorzystywane do posortowania znalezionych rekomendacji, tak by najpierw pokazać książki najpopularniejsze.

Podobieństwo cosinusowe polega na obliczeniu cosinus kąta między dwoma wektorami. Funkcja cosinus może przyjmować wartości z zakresu [-1,1]. Im większa wartość tym większe podobieństwo między wektorami. Podobieństwo wylicza się z następującego wzoru:

$$D(x, y) = cos(\theta) = \frac{x \cdot y}{||x|| \cdot ||y||} \tag{1}$$

Wektory x i y powinny przetrzymywać informacje o atrybutach książek. Atrybuty należy sprowadzić do postaci liczbowych, tak że dana wartość książki ocenionej przez użytkownika będzie równa 1, a każda inna będzie równa 0.

Odległość Hamminga natomiast określa liczbę różnych wartości pomiędzy dwoma wektorami. Gatunki i poruszane tematy zamieniamy na cyfry 0 bądź 1, w zależności od tego czy występują w danej książce. Te cechy będą też miały zawsze stałą kolejność, na przykład: romans, horror, kryminał. Kryminał, bez dodatkowych gatunków będzie miał wartość 001, a kryminał z elementami horroru: 011. Na takim ciągu liczbowym będzie liczona odległość

Hamminga.

By policzyć finalne podobieństwo między książkami, bierzemy pod uwagę podobieństwo cosinusowe oraz odległość Hamminga. Problemem jest to, że żeby książki były jak najbardziej podobne to potrzebujemy jak największego podobieństwa cosinusowego i jak najmniejszej odległości Hamminga, zatem nie możemy po prostu zsumować tych wartości. Dlatego zamiast zwykłej wartości Hamminga będziemy brali jej odwrotność, czyli liczbę elementów, których nie trzeba zmieniać - w takim wypadku im większa liczba tym lepiej. Patrząc na podany wcześniej przykład, z kryminałem i horrorystycznym kryminałem: klasyczna odległość Hamminga byłaby równa 1, a nasza „odwrócona"będzie równa 2.

Zaimplementowaliśmy też lekko odmienną strategię. Wciąż wykorzystywane są odległość Hamminga oraz podobieństwo cosinusowe, lecz zamiast liczyć je dla paru ulubionych książek rozważamy je dla „średniej" z książek najwyżej ocenionych przez użytkownika. „Średnia z książek" to książka o cechach najczęściej występujących wśród ulubionych książek użytkownika.

Użytkownik ma też możliwość wyświetlić rekomendacje na podstawie jednej książki. Algorytm takiej rekomendacji działa identycznie do sposobu pierwszego, jedynie podobieństwa nie są sumowane gdyż są liczone na podstawie jednej pozycji.

5.2 Pseudokod

Poniżej znajduje się nasz algorytm zapisany w pseudokodzie. Nie uwzględniliśmy funkcji szukania najpopularniejszych książek oraz ulubionych książek użytkownika, gdyż sprowadzają się one jedynie do sortowania po ocenie i wypożyczeniach (w przypadku pierwszym) lub dacie przeczytania (w drugim przypadku).

```
Recommend(książka, n)
     x ← ToVector(książka, książka)
     xG, xT ← ToHemming(książka)
     similarity ← słownik przypisujący każdej książce
          stopień podobieństwa
     dla każdej nieprzeczytanej książki notRead
          y ← ToVector(notRead)
          yG, yT ← ToHamming(notRead)
          podobieństwo ← CosSimilarity(x,y) +
```

```
            HammingSimilarity(xG,yG) + HammingSimilarity(xT,yT)
        wstaw do similarity parę (notRead, podobieństwo)
    posortuj similarity malejąco
    zwróć n pierwszych książek z similarity

ToVector(x, comparison)
    wynik ← pusty wektor
    dla każdego a - atrybut książki (poza gatunkami i tematami)
        jeśli x.a == comparison.a to
            wstaw 1 do wynik
        w.p.p. wstaw 0 do wynik
    zwróć wynik
```

W powyższej funkcji należy zwrócić uwagę na porównanie atrybutów. Należy zaimplementować własne porównanie dat, by brać pod uwagę nie konkretne daty, tylko dłuższe okresy czasu, gdyż jeśli ktoś przeczytał książkę wydaną 03/12/2003, to nie interesują go inne książki wydane trzeciego grudnia tego roku, a raczej książki wydane w pierwszej dekadzie XXI wieku. My będziemy porównywali właśnie dekady.

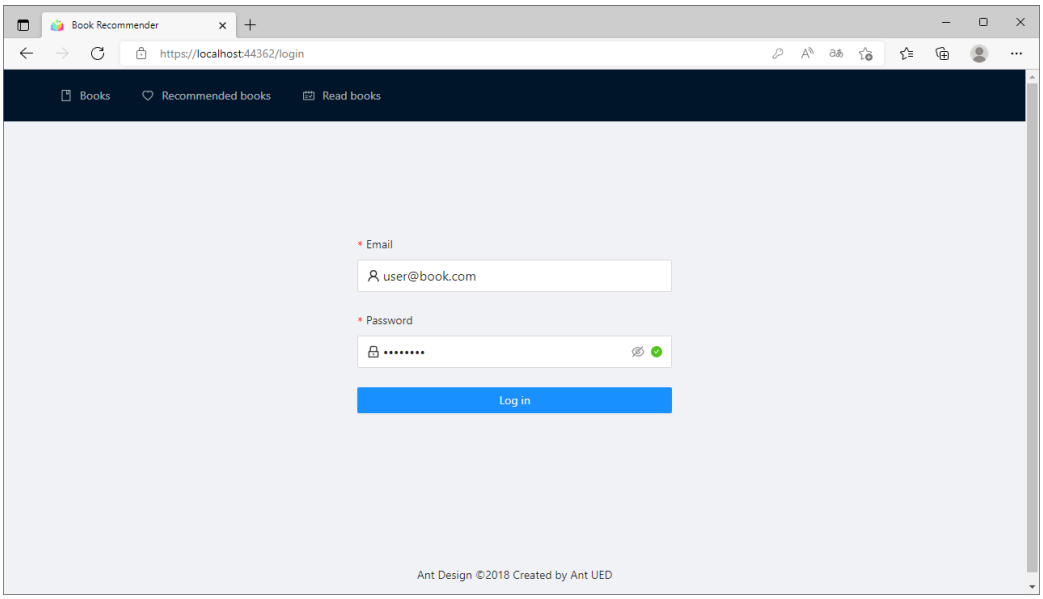
```
ToHamming(x)
    wynikG ← pusty wektor
    wynikT ← pusty wektor
    dla każdego gatunku g
        jeśli x.gatunki zawiera g to
            wstaw 1 do wynikG
        w.p.p. wstaw 0 do wynikG
    dla każdego poruszanego tematu t
        jeśli x.tematy zawiera t to
            wstaw 1 do wynikT
        w.p.p. wstaw 0 do wynikT
    zwróć (wynikG, wynikT)

CosSimilarity(x,y)
    zwróć x·y / ||x||·||y||
```

W powyższej funkcji licznik jest iloczynem skalarnym wektorów, a mianownik iloczynem ich norm.

```
HammingSimilarity(x,y)
     i ← długość wektorów x i y
     wynik ← 0
     dopóki i>0
          jeśli x[i]==y[i] to
               wynik++
          i--
     zwróć wynik
```

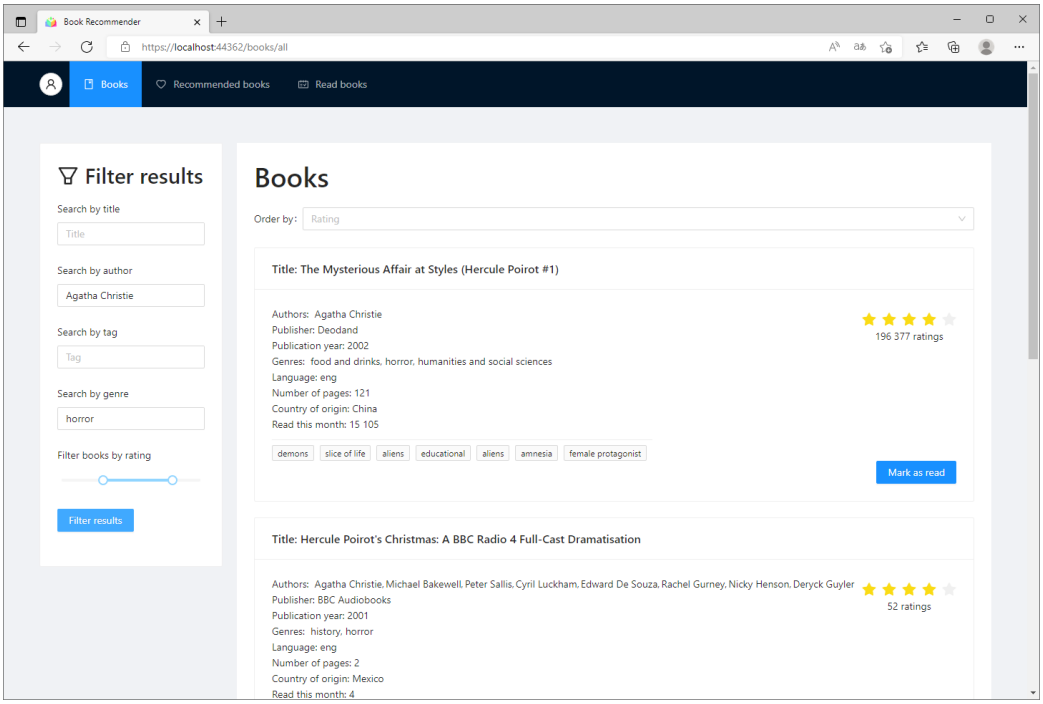
6 Aplikacja użytkownika

Po uruchomieniu aplikacji użytkownika pojawi się ekran logowania.



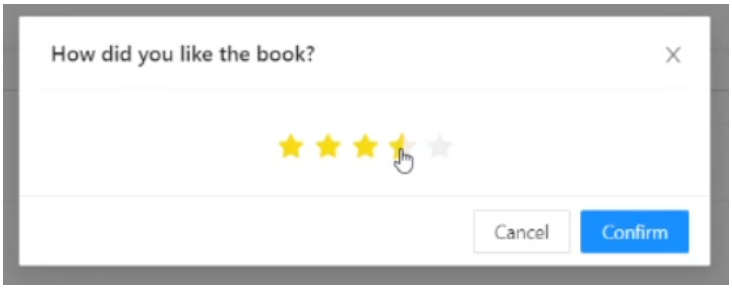
Rysunek 1: Ekran logowania

Po zalogowaniu wyświetla się oferta książek. Użytkownik może wyszukiwać książki po tytule, autorze, gatunku czy tagu oraz filtrować pozycje o ocenach w wybranym przez siebie zakresie. Wyniki domyślnie uporządkowane są malejąco po średniej ocenie, ale możliwa jest też kolejność alfabetyczna.



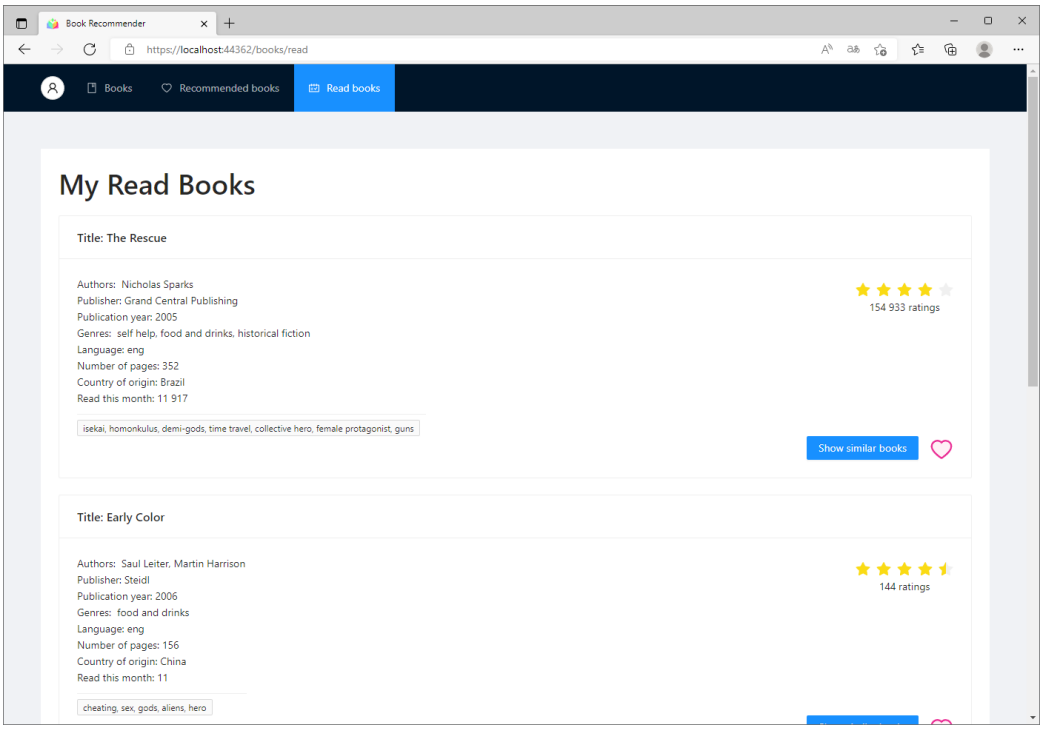
Rysunek 2: Oferta książek

Po odnalezieniu przeczytanej przez siebie książki użytkownik może ją oznaczyć jako przeczytaną za pomocą przycisku „**Mark as read**”. Następnie wyświetli się widoczne poniżej okno z prośbą o ocenę książki w skali 1-5.



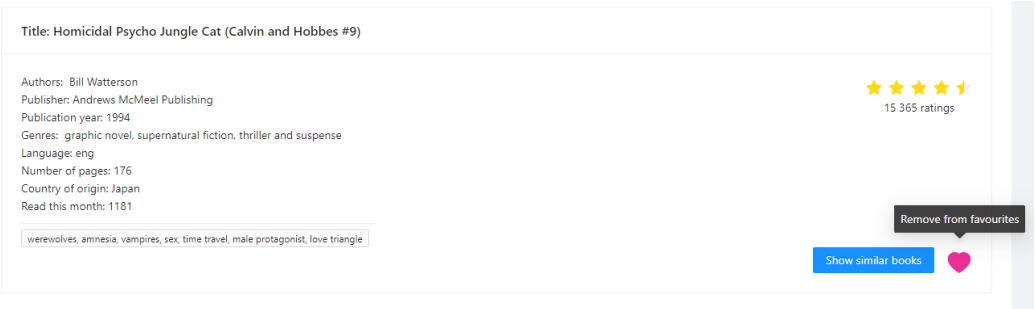
Rysunek 3: Ocena książki

Po oznaczeniu i ocenieniu książki, użytkownik może przejść do zakładki „**Read books**”, aby wyświelić wszystkie przeczytane przez siebie tytuły.



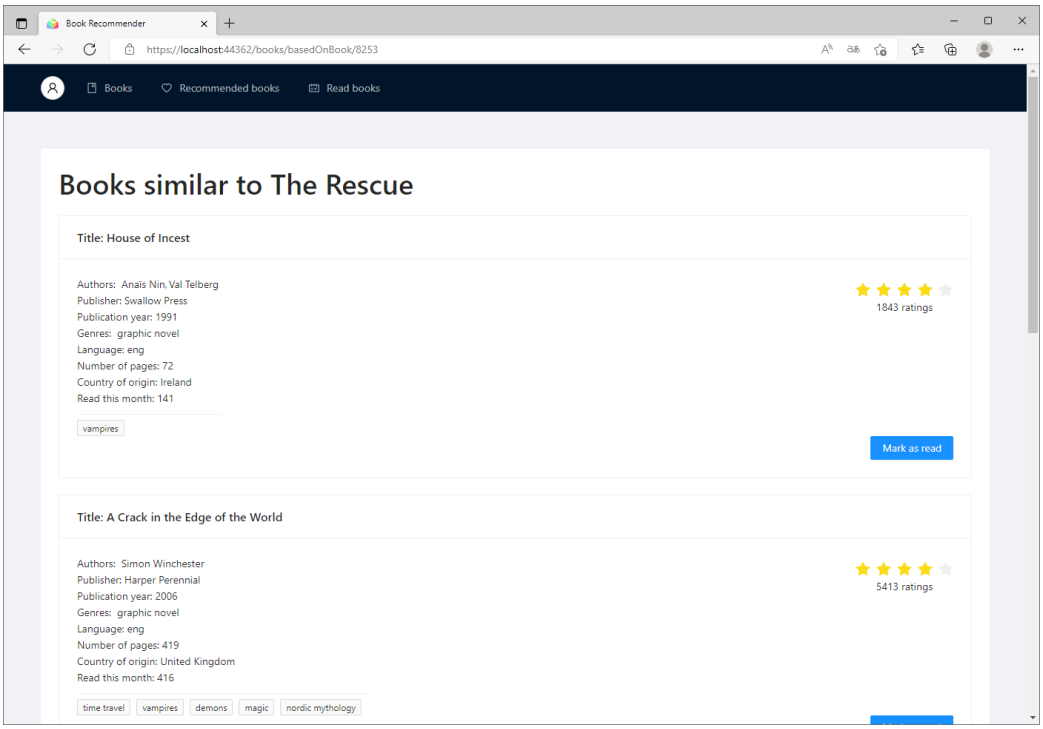
Rysunek 4: Lista przeczytanych książek

Użytkownik może oznaczyć książkę jako ulubioną poprzez kliknięcie na ikonę serca przy wybranym tytule. Zamalowana ikona serca oznacza, że książka jest na liście ulubionych książek, w przeciwnym przypadku ikona jest pusta w środku.



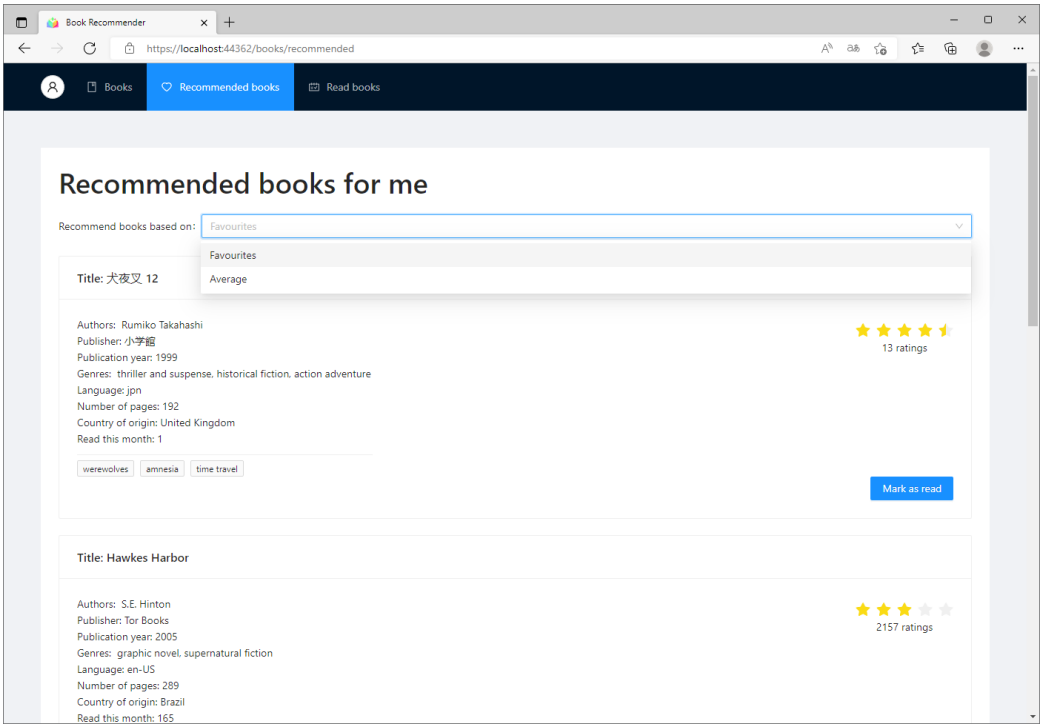
Rysunek 5: Dodawanie/usuwanie książki do/z ulubionych

Istnieje możliwość wyświetlenia książek podobnych do przeczytanego tytułu. Aby wyświetlić rekomendacje na podstawie jednej książki należy wcisnąć przycisk „**Show similar books**”.



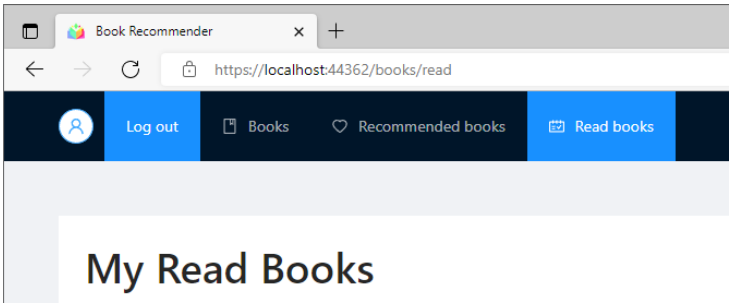
Rysunek 6: Spis książek podobnych do wybranej książki

Po wybraniu zakładki „**Recommended books**”w menu widocznym na górnym panelu ekranu użytkownik może wyświetlić indywidualnie dobraną listę książek polecanych przez nasz system. Możliwe są dwie opcje polecania: na podstawie najwyżej ocenionych książek oraz na podstawie ulubionych. Wybranie opcji „**Favorites**”generuje rekomendacje na podstawie ulubionych pozycji, natomiast opcja „**Average**”generuje rekomendacje na podstawie książki wytworzonej na podstawie najwyżej ocenionych lektur.



Rysunek 7: Spis polecanych książek

Aby się wylogować należy nacisnąć na ikonę użytkownika w lewym górnym roku, a następnie wcisnąć „**Log out**”.



Rysunek 8: Wylogowywanie