

CyberBook – dokumentacja projektu wykonywanego w ramach zajęć:

Przetwarzanie danych w chmurach obliczeniowych

Mikołaj Marchewa

Wydział Fizyki i informatyki Stosowanej

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

15 grudnia 2021

1. CyberBook, czyli książkowa platforma społecznościowa

Głównym celem projektu było zrozumienie zasady działania, mechanizmów, przypadków użycia grafowej bazy danych *neo4j*. W efekcie powstała platforma społecznościowa zrzeszająca miłośników książek i pozwalająca na dzielenie się opiniami na ich temat, a także na obserwację aktywności innych użytkowników portalu.

Interfejs graficzny został zahostowany pod adresem:

<https://happybutter.github.io/Cyber-book-frontend/#/>

Rest API dostępne jest pod adresem:

<https://young-gorge-53186.herokuapp.com/>

Przykładowe konto użytkownika:

Login: **admin@admin.com**

Hasło: **admin**

2. Funkcjonalności

Po stronie użytkownika portalu:

- rejestracja w serwisie – tworzenie nowych kont,
- logowanie do już istniejącego konta,
- przeglądanie najlepiej ocenianych książek,
- przeglądanie recenzji książek,
- wyszukiwanie książki po tytule (lub jego fragmencie),
- obserwacja innych użytkowników portalu,
- wyszukiwanie innych użytkowników portalu (po imieniu/nazwisku lub ich fragmencie)

Po stronie administracji:

- dla administratora powstał specjalny endpoint dostępny pod adresem:

<https://young-gorge-53186.herokuapp.com/book/>

dzięki któremu z wykorzystaniem zapytania typu POST, administrator może dodać do forum nową książkę. Automatycznie utworzą się (lub nie – jeśli istnieją) węzły będące gatunkami literackimi oraz węzeł autora książki, a także relacje wiążące nowe węzły ze sobą. Struktura ciała zapytania na podstawie przykładowej pozycji:

```
1  {
2    "title": "Good to Great: Why Some Companies Make the Leap...And Others Don't",
3    "author": "Jim Collins",
4    "description": "lorem",
5    "publisher": "Harper Collins",
6    "page_count": 320,
7    "genres": [
8      "Business & Economics",
9      "Economics",
10     "Management"
11   ],
12   "ISBN": "9780062119209",
13   "language": "English",
14   "published_date": "Jul 19, 2011"
15 }
```

3. Projekt logiczny

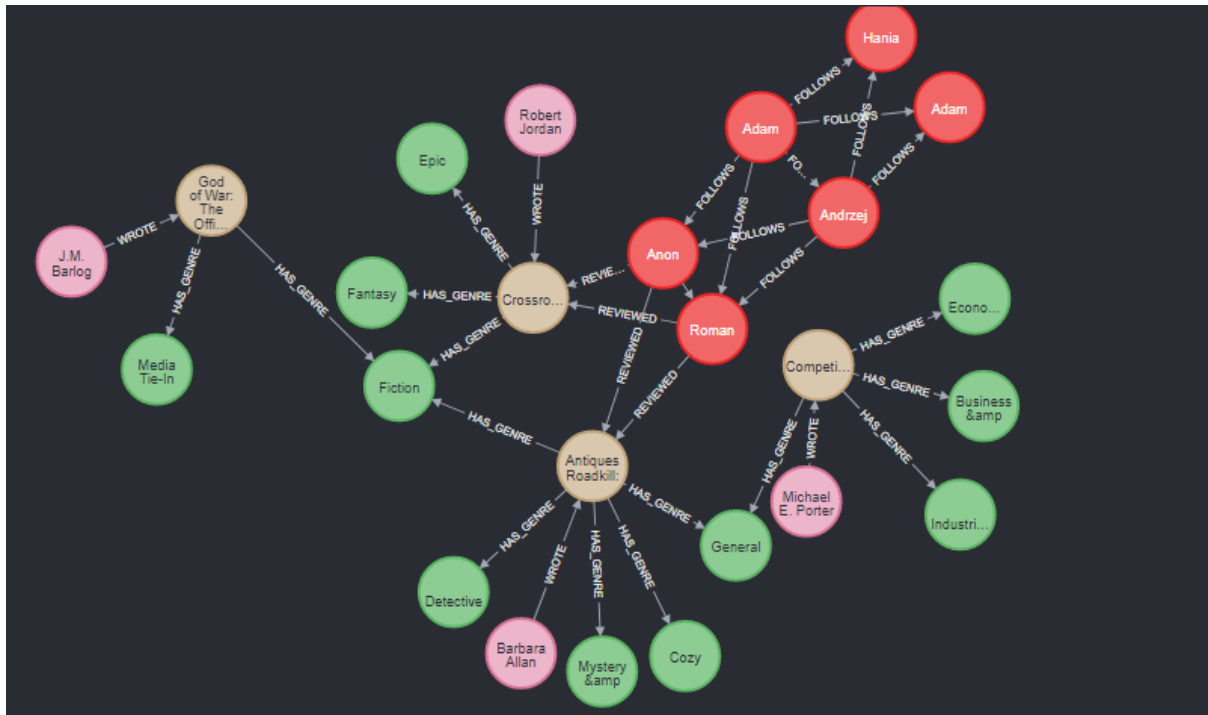
Węzły:

- User - użytkownik serwisu,
 - id,
 - firstName
 - lastName
 - email (pole unikalne)
 - password
- Book – książka,
 - ISBN (pole unikalne),
 - tilte,
 - publisher,
 - description,
 - pageCount,
 - published_date,
 - language,
 - rate,
 - voteCount
- Author – autor książki,
 - name
- Genre – gatunek literacki,
 - Name

Relacje międzywęzłowe:

- Book-[:HAS_GENRE]->Genre
- Author-[:WROTE]->Book
- User-[:FOLLOWS]->User
- User-[:REVIEWED]->Book

Baza została wypełniona wybranymi pozycjami ze zbioru książek dostępnego pod adresem:
<https://www.kaggle.com/bilalyussef/google-books-dataset>



4. Projekt funkcjonalny

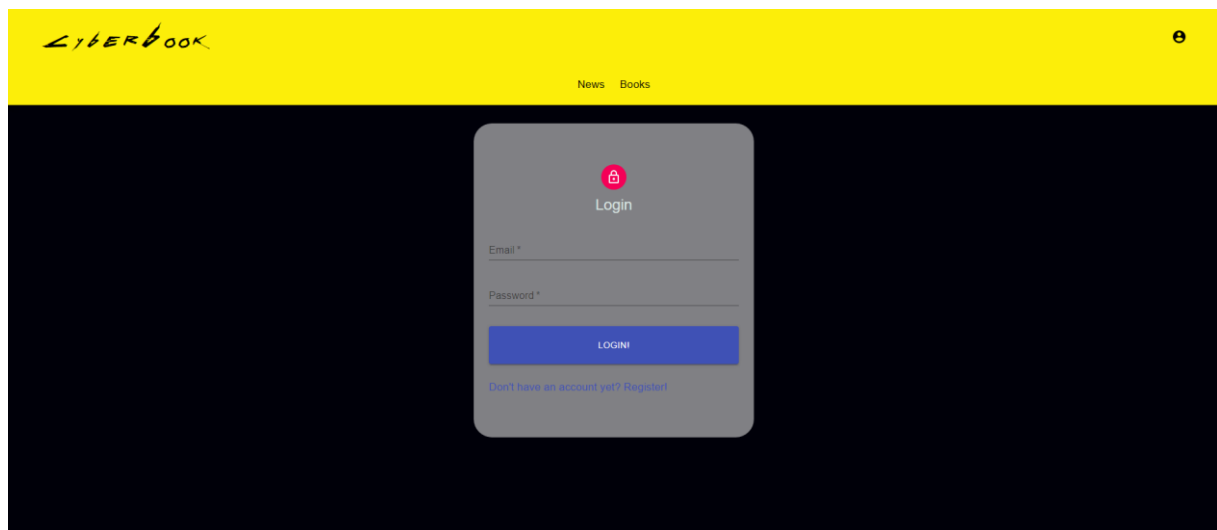
- Po wejściu na portal znajdujący się pod adresem:
<https://happybutter.github.io/Cyber-book-frontend/#/>
użytkownik serwisu otrzymuje możliwość utworzenia nowego konta:

The screenshot shows the 'Cyberbook' website header with a yellow background and a dark blue navigation bar containing 'News' and 'Books' links. The main content area is dark blue and features a light gray registration form. The form has a red user icon and the title 'Register'. It includes input fields for 'First name *', 'Last name *', 'Email *', and 'Password *'. A blue 'REGISTER' button is at the bottom of the form, followed by a link that says 'Already have an account? Login!'.

Co realizuje kwerenda:

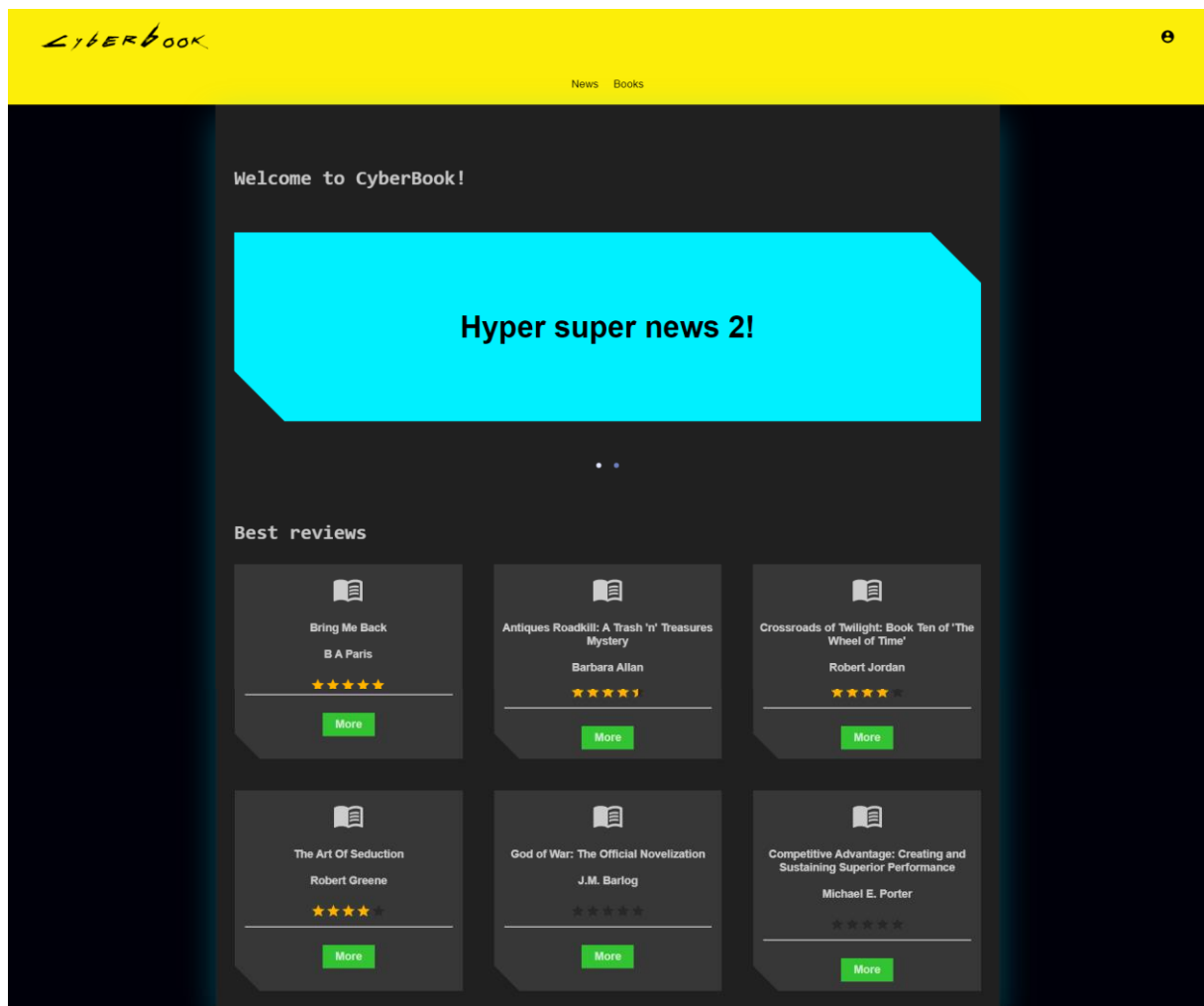
```
CREATE (u:User {  
    id: apoc.create.uuid(),  
    firstName: $firstName,  
    lastName: $lastName,  
    email: $email,  
    password: $password  
})  
RETURN u.id, u.firstName, u.lastName, u.email
```

- Lub możliwość zalogowania się przy użyciu wcześniej utworzonego konta:



```
MATCH (u:User)  
WHERE u.email=$email AND u.password=$password  
RETURN u.id, u.firstName, u.lastName, u.email
```

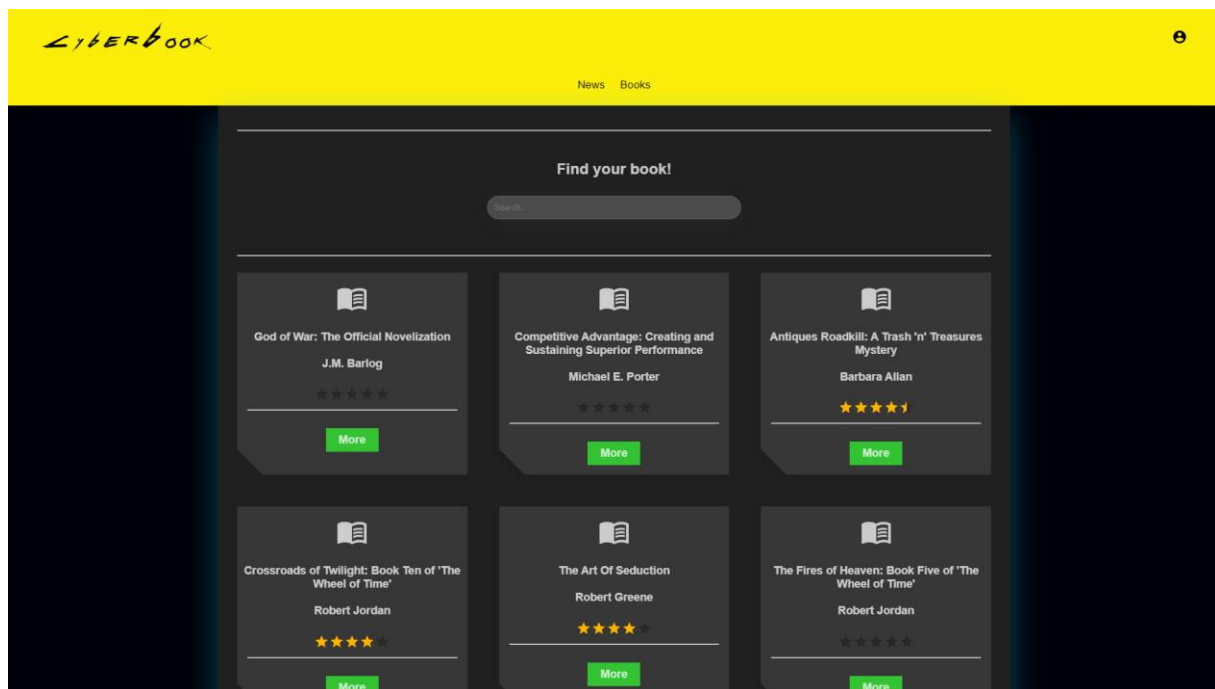
- Strona główna aplikacji – zakładka „News”:



Sekcja „Best reviews” wyświetla 6 najlepiej ocenianych książek spośród wszystkich dostępnych w bazie danych. Ich pobieranie realizuje kwerenda:

```
MATCH (a:Author)-[:WROTE]->(b:Book)
WITH [(b)-[:HAS_GENRE]->(g) WHERE g:Genre | g.name] AS genres, b, a
ORDER BY b.rate DESC
LIMIT 6
RETURN collect({author: a.name, book: b, genres: genres})
```

- Zakładka „Books”:



Zakładka “Books” pozwala na przeszukiwanie zasobów bazy i odnalezienie interesującej użytkownika książki:

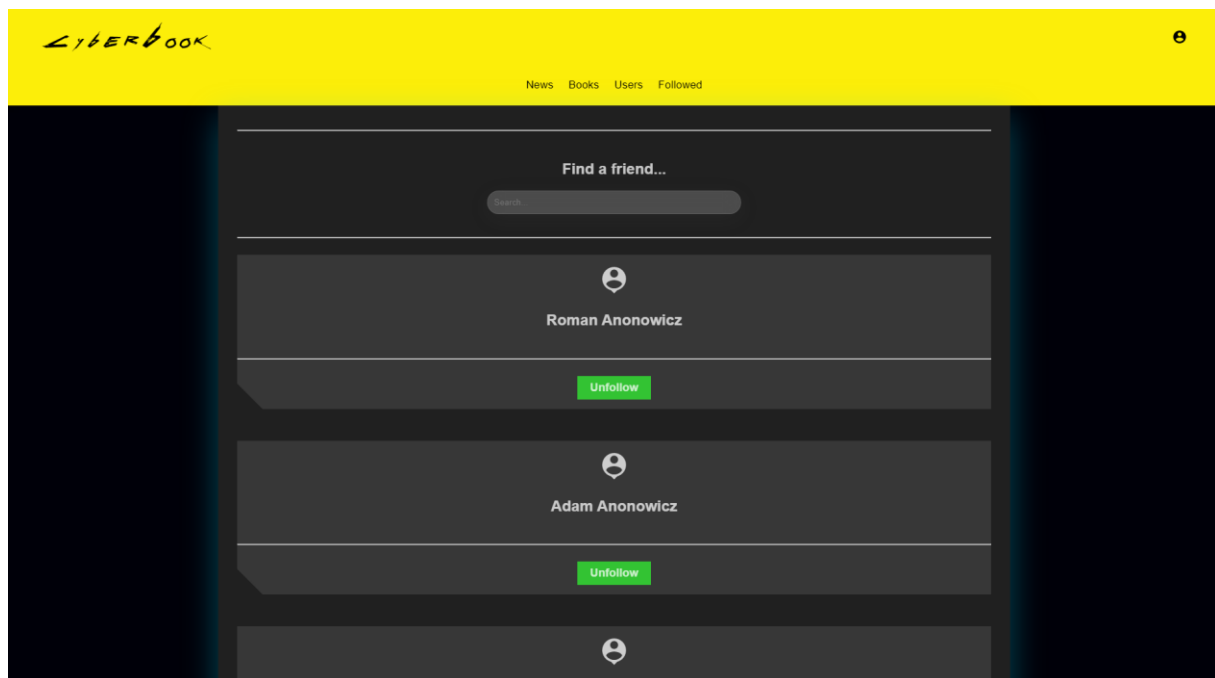
```
MATCH (a:Author)-[:WROTE]->(b:Book)
```

```
WHERE toLower(b.title) CONTAINS toLower($title)
```

```
WITH [(b)-[:HAS_GENRE]->(g) WHERE g:Genre | g.name] AS genres, b, a
```

```
RETURN collect({author: a.name, book: b, genres: genres})
```

- Dla zalogowanych – zakładka „Users”:



Dzięki wyszukiwarce użytkownik może odnaleźć swoich znajomych i zaobserwować bądź przestać obserwować wystawiane przez nich recenzje. Wyszukiwanie realizuje następująca operacja:

```
MATCH (u:User)
WHERE toLower(u.firstName + u.lastName) CONTAINS trim(toLower($name))
OR toLower(u.lastName + u.firstName) CONTAINS trim(toLower($name))
WITH u
UNWIND u.id as ID
MATCH (u1:User {id: $userId}), (u2:User {id: ID})
WITH exists((u1)-[:FOLLOWS]->(u2)) as isFollowing, u2
WHERE NOT u2.id=$userId
RETURN collect({
    isFollowed: isFollowing,
    firstName: u2.firstName,
    lastName: u2.lastName,
    id: u2.id })
```

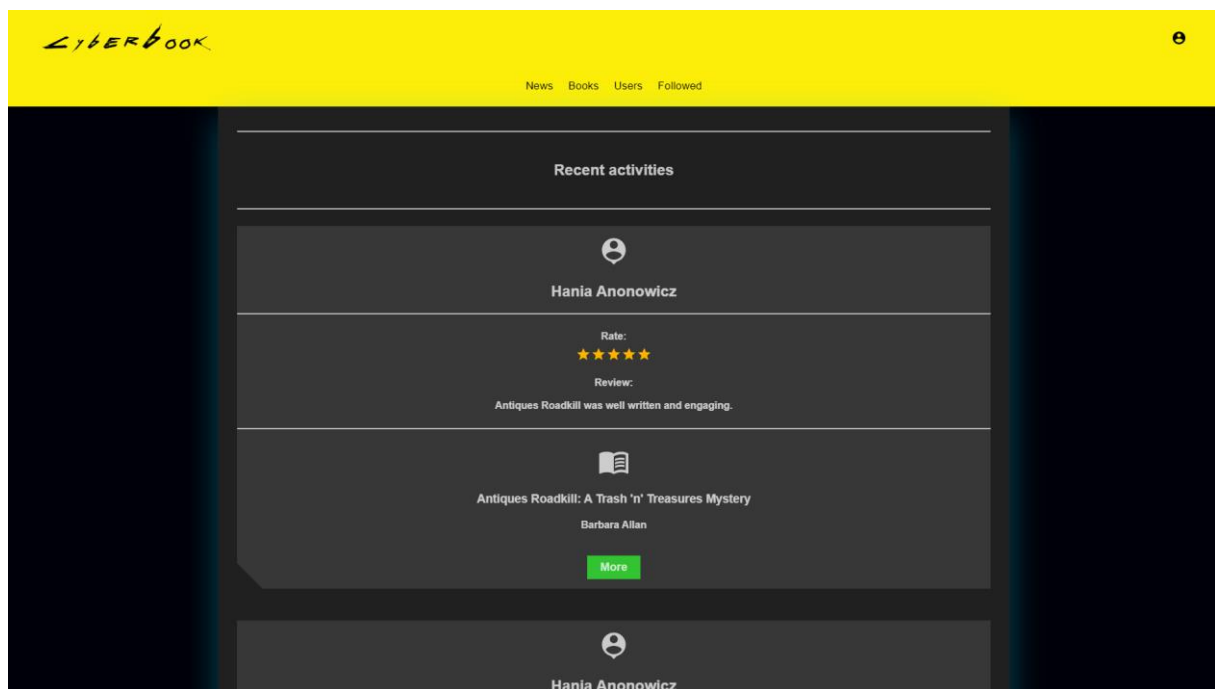
Obserwacja użytkownika polega na dodaniu nowej relacji *FOLLOWS*:

```
MATCH (a:User {id: $userId})
MATCH (b:User {id: $userIdToFollow})
MERGE (a)-[r:FOLLOWS]->(b)
RETURN r
```

Zakończenie obserwacji, to usunięcie istniejącej krawędzi grafu:

```
MATCH (a:User {id: $userId})-[r:FOLLOWS]->(b:User {id: $userIdToUnfollow})
DELETE r
```

- Dla zalogowanych – zakładka „Followed”:




W sekcji „Followed” użytkownik znajdzie posortowane chronologicznie (począwszy od najnowszych) recenzje obserwowanych użytkowników. Wyszukanie tych rekordów polega na wykonaniu poniższej kwerendy:

```
MATCH (u:User {id: $userId})-[:FOLLOWS]->(u2:User)-[r:REVIEWED]->(b:Book)-[:WROTE]-(a:Author)
WITH collect({firstName: u2.firstName, lastName: u2.lastName, authorId: u2.id}) as author,
      collect({ISBN: b.ISBN, rate: b.rate, title: b.title, author: a.name}) as book, r
RETURN collect({ author: author,
                book: book, rate: r.rate,
                review: r.review, creationDate: r.creationDate })
```


- Szczegóły wybranej książki:

CYBERBOOK
e

News Books Users Followed



Antiques Roadkill: A Trash 'n' Treasures Mystery

Barbara Allan

★★★★★

Publisher:

Kensington Publishing Corp.

Description:

Determined to make a new start in her quaint hometown on the banks of the Mississippi, Brandy Borne never dreams she'll become the prime suspect in a murder case... Moving back in with her eccentric, larger-than-life mother, Brandy Borne finds small-town Serenity anything but serene. It seems an unscrupulous antiques dealer has swindled Vivian out of the family's heirlooms. But when he is found run over in a country lane, Brandy becomes Murder Suspect Number One—with her mother coming in a very close second... The list of other suspects is impressive—the victim's business seems to have been based on bilking seniors out of their possessions. And when the Borne "girls" uncover a few very unsavory Serenity secrets, they become targets for a murderer whose favorite hobby seems to be collecting victims. Don't miss Brandy Borne's tips on antiques! "Cozy mystery fans will love this down-to-earth heroine with the wry sense of humor and a big heart!"—Nancy Pickard "With its small-town setting and quirky characters, Antiques Roadkill is fun from start to finish. Dive in and enjoy!"—Laurien Berenson

Your rate

★★★☆☆

Your review...

Add Review

Hania Anonowicz★★★★★

Antiques Roadkill was well written and engaging.

Anon Anonowicz★★★★★

Ok.

Wyświetlenie szczegółów danej książki pozwala na przeczytanie najnowszych opinii, a także napisanie własnej recenzji. W efekcie dodania nowego komentarza ocena średnia książki zostanie ponownie przeliczona. Operacja ta jest zamknięta w obrębie jednej transakcji:

1. Dodanie recenzji.

```
MATCH (a:User {id: $userId})  
MATCH (b:Book {ISBN: $isbn})  
MERGE (a)-[r:REVIEWED {rate: $rate, review: $review, creationDate: $creationDate}]->(b)  
RETURN r
```

2. Przeliczenie wartości średniej oceny

```
MATCH (b:Book {ISBN: $isbn})<-[r:REVIEWED]-()  
WITH avg(r.rate) AS averageRate, count(*) as countReviews, b  
SET b.rate = averageRate, b.voteCount = countReviews  
RETURN averageRate, countReviews
```

5. Bibliografia

- <https://neo4j.com/>
- <https://www.npmjs.com/package/neo4j-driver>
- <https://beta.reactjs.org/>
- <https://reactrouter.com/>
- <https://www.kaggle.com/bilalyussef/google-books-dataset>
- <https://pages.github.com/>