

SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Hrvoje Lesar

## APLIKACIJA ZA PREPORUKE

PROJEKT

TEORIJA BAZA PODATAKA

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

VARAŽDIN

Hrvoje Lesar

Matični broj: 0016133479

Studij: Organizacija poslovnih sustava

## APLIKACIJA ZA PREPORUKE

### PROJEKT

Mentor:

dr. sc. Bogdan Okreša Đurić

Varaždin, Siječanj 2024.

*Hrvoje Lesar*

**Izjava o izvornosti**

Izjavljujem da je ovaj projekt izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrđio prihvaćanjem odredbi u sustavu FOI Radovi*

---

# Sadržaj

|   |    |
|---|----|
| <b>1. Uvod</b>                              | 1  |
| <b>2. Polustrukturirane baze podataka</b>   | 2  |
| 2.1. Polustrukturirani podaci               | 2  |
| 2.2. Vrste polustrukturiranih baza podataka | 2  |
| 2.3. MongoDB                                | 3  |
| 2.4. Sustavi za preporuke                   | 3  |
| <b>3. Model baze podataka</b>               | 5  |
| 3.1. Korišteni skup podataka                | 5  |
| 3.2. Pretvorba podataka                     | 5  |
| 3.3. Implementacija                         | 7  |
| 3.3.1. Najbolje ocjenjene knjige            | 7  |
| 3.3.2. Kolaborativno filtriranje            | 8  |
| 3.3.3. Dodavanje, brisanje, promjena ocjena | 8  |
| <b>4. Primjeri korištenja</b>               | 9  |
| <b>5. Zaključak</b>                         | 11 |
| <b>Popis literature</b>                     | 12 |
| <b>Popis slika</b>                          | 13 |
| <b>Popis isječaka koda</b>                  | 14 |

# 1. Uvod

Kroz projekt će biti teoretski obrađene polustrukturirane baze podataka, polustrukturirani podaci, jedna od često korištenih polustrukturiranih baza podataka MongoDB i teoretske osnove sustava za preporuke.

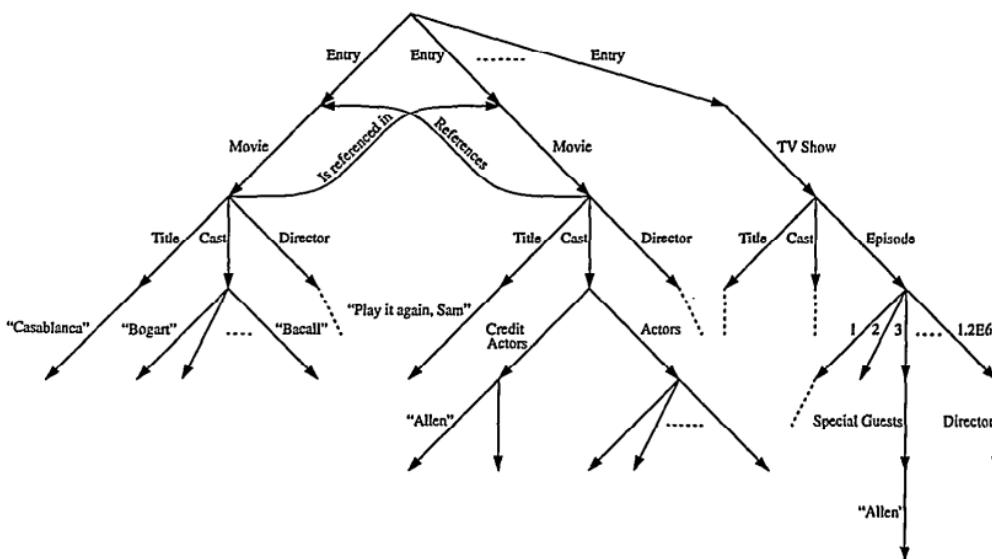
Cilj rada je implementirati aplikaciju za preporuke knjiga. Ovaj cilj će se postići korišteњem programskog jezika Go u kojem će se izrađivati web aplikacija. Aplikacija mora imati mogućnost komunikacije s bazom podataka MongoDB i način za postavljanje upita. Odabran je programski jezik Go zbog relativne jednostavnosti jezika i mogućnosti brze iteracije i izrade aplikacija, također ima vrlo dobru podršku za web.

Web aplikacija mora imati mogućnost prikaza preporuka određenom korisniku. Preporuka se mora generirati na temelju drugih korisnika te se preporuke moraju mijenjati u slučaju promjene "ukusa" korisnika tj. korisnik i drugi korisnici imaju značaj i utjecaj na preporuke.

## 2. Polustrukturirane baze podataka

### 2.1. Polustrukturirani podaci

Glavna ideja polustrukturiranih podataka je predstavljanje podataka kao vrstu strukture koja ima nalik na graf ili stablo. Iako su dopušteni ciklusi između vrhova grafa, općenito takvu vrstu grafa možemo nazivati stablima [1]. Na slici 1 je prikazan primjer modela podataka formaliziran u strukturu grafa. Bridovi grafa su označeni tipom podatka ili više apstraktnim tipom koji se dalje grana. Prolaskom kroz graf moguće je primijetiti da postoje dva različita načina prema kojima je opisan film. U prvom su glumci, filmska ekipa direktno nabrojeni, dok u drugom postoji grananje na glumce i kreditirane glumce. Zadnja grana grafa opisuje TV seriju koja opet ima drugačiju strukturu u usporedbi s strukturom filmova.



Slika 1: Primjer polustrukturiranih podataka u obliku grafa [1]

Polustrukturirani podaci su organizirani u semantičke entitete, ali nisu striktno u skladu s formalno strukturiranim strogim tipovima podataka. Konceptualni model polustrukturiranih podataka mora sadržavati nekoliko svojstava kao što su reprezentacija nepravilnih i heterogenih struktura kao prikazanih na slici 1, hijerarhijske odnose uz nehijerarhijske vrste odnosa, kardinalnost, relacije n-niza, poredak i reprezentaciju mješovitog sadržaja [2].

### 2.2. Vrste polustrukturiranih baza podataka

Polustrukturirane baze podataka možemo podijeliti na četiri glavne vrste [3]:

1. Key-Value store (Pohrana ključeva i vrijednosti); Podaci se pohranjuju kao skup ključeva i vrijednosti. Ključevi su jedinstveni te se pristupa podacima povezivanjem ključa s vrijednosti. Vrijednosti ne moraju striktno biti informacije, mogu biti drugi ključevi.

2. Baze podataka temeljene na dokumentima; Mogu se definirati kao setovi ključeva i vrijednosti. Svaki dokument je identificiran unikatnim ključem. Tip dokumenta je definiran prema znam standardima koji su većini slučajeva XML ili JSON. Pristup podacima moguć je korištenjem ključa ili određenih vrijednosti.
3. Column-family (Obitelj stupaca); Podaci su postavljeni u stupce tj. sama struktura podataka i organizacija podatak se sastoji od stupaca super-stupaca, i obitelji stupaca. Struktura baze je definirana kroz super-stupce i obitelj stupca. Novi stupci se mogu dodavati po potrebi. Pristup podacima je moguć naznačujući obitelj stupca, ključ, stupac što će rezultirati dohvaćanjem vrijednosti.
4. Graf baze podataka; Ova vrsta se koristi kad se podaci mogu prikazati kao graf, jedan primjer su društvene mreže i veze između različitih korisnika.

## 2.3. MongoDB

MongoDB je polustrukturirana baza temeljena na dokumentima. Dokumenti su grupirani u kolekcije prema njihovoj strukturi. Dopušteno je spremanje dokumenta različitih struktura, no zbog boljih performansi preporuka je grupirati dokumente s istom ili sličnom strukturom [3]. MongoDB koristi BSON kao format za spremanje dokumenata. BSON je kratica za Binary JSON.

Svaki novi dokument je moguće identificirati poljem `_id`, te je za svaku kolekciju automatski kreiran indeks preko `_id` polja. Neke od najbitnijih karakteristika MongoDB-a su trajnost i moguće paralelno čitanje i pisanje podataka. Trajnost je omogućeno kroz kreiranje replika baze, MongoDB koristi master-slave (gospodar-rob) mehanizam za replikaciju. Omogućava definiranje jednog gospodara i jednog ili više robova. Gospodar je jedina replika kojoj je dozvoljeno pisanje i čitanje dok robovi služe samo kao sigurnosna kopija. U slučaju da se gospodar replika sruši, replika rob s najnovijim podacima je promovirana u novog gospodara. Sve replike su asinkrone što znači da izvršena ažuriranja nisu odmah vidljiva na svim replikama. MongoDB postiže paralelno čitanje i pisanje podataka zaključavanjem podataka. Podaci koji se trenutno ažuriraju su zaključani kako nebi bili pročitani zastarjeli podaci ili kako se nebi dogodilo više upisa u isto vrijeme te više nebi bilo moguće odrediti valjanost upisanih podataka.

## 2.4. Sustavi za preporuke

Sustavi za preporuke se koriste za izradu kolekcije stavaka koje bi mogle interesirati određene korisnike. Dizajn sustava ovisi o domeni problema, proizvoda, stavaka koje se žele preporučiti te dostupnosti podataka i posebnih karakteristika prema kojima bi bilo moguće kreirati preporuku [4]. Sustavi za preporuke se razlikuju po načinu na koji analiziraju izvor podataka kako bi odredili srodnost između korisnika i stavka koje se mogu koristiti za identifikaciju podudarnih parova.

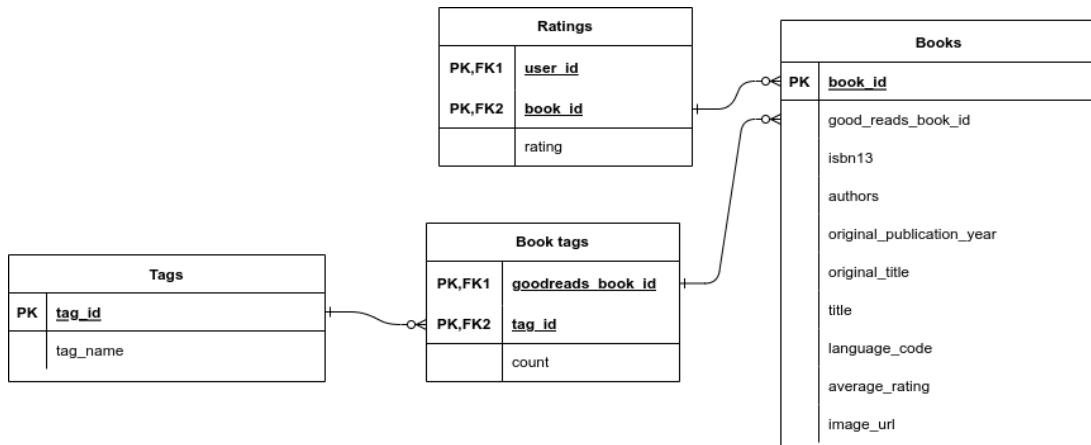
Načini pristupa rješavanju problema preporuka moguće je kategorizirati u nekoliko glavnih kategorija [5]:

1. Kolaborativno filtriranje; Kolaborativno filtriranje radi na principu prikupljanja povratnih informacija od korisnika. Povratne informacije su uglavnom u obliku ocjena za stavke u određenoj domeni. Iskorištavanjem sličnosti u ponašanju ocjenjivača se određuje manji broj korisnika kojima je moguće preporučiti neku stavku.
2. Preporuke temeljene na sadržaju; Daju preporuke uspoređujući prikaze opisa sadržaja koji opisuju neku stavku s prikazima sadržaja koji zanimaju korisnika.
3. Hibridni pristupi; Pokušava kombinirati kolaborativno filtriranje i preporuke temeljene na sadržaju kako bi preporuke bile što bolje i točnije.

## 3. Model baze podataka

### 3.1. Korišteni skup podataka

Skup podataka koji je korišten kao temeljni je goodreads-10k skup. Skup sadrži oko deset tisuća knjiga i oko šest milijuna ocjena knjiga, tagove i žanrove dodijeljene knjigama od strane korisnika. Inicialni skup je raspoređen u nekoliko .csv datoteka i može se prikazati sljedećim dijagramom:



Slika 2: Dijagram strukture podataka korištenog skupa podataka

Kroz dijagram na slici 2 je vidljivo da su podaci strukturirani na način pogodan za relacijske baze podataka. Korišteni skup će biti direktno importiran u MongoDB no potrebno će ga biti transformirati u skup više pogodan za rad s polustrukturiranim bazama. Kolekcije kreirane importiranjem podataka imaju nazive tags, ratings, book\_tags, books.

### 3.2. Pretvorba podataka

Sve navedene naredbe moguće je pokrenuti preko mongo shella u bazi podataka koja ima importiran prije definiran skup podataka. Pretvorba podataka započinje smanjivanjem dostupnih broja tagova, tj. korištenjem samo onih najpopularnijih. Pošto su tagovi iz izvornog skupa podataka definirani od strane korisnika mogu imati vrlo različite nazine i imati različita značenja za korisnike, time bi najpopularniji tagovi morali filtrirati one manje korisne.

---

```

1 db.book_tags.aggregate([
2     { $group: { _id: "$tag_id", totalCount: { $sum: "$count", }, }, },
3     { $sort: { totalCount: -1, }, },
4     { $limit: 300, },
5     { $lookup: { from: "tags", localField: "_id", foreignField: "tag_id", as: "tag",
6         },
7         { $group: { _id: "$tag", }, },
8         { $unwind: "$_id", },
9         { $replaceRoot: { newRoot: "$_id", }, },
10        { $out: "most_popular_tags", },
11    })

```

---

### Isječak koda 1: Kreiranje najpopularnijih tagova

Sljedeće se dodaju žanrovi knjigama (žanrovi su izvedeni iz tagova). Definirani upit prolazi kroz svaki tag, provjerava koje knjige imaju dodijeljen trenutno selektirani tag, te ažurira sve dokumente u kolekciji books sa tagom tj. žanrom koji pripada knjizi.

---

```

1 db.most_popular_tags.find({}).map((tag) => {
2     db.book_tags.aggregate([
3         { $match: { tag_id: tag.tag_id } },
4         { $project: { goodreads_book_id: 1 } }
5     ]).map((tagIds) => {
6         db.books.updateMany({goodreads_book_id: { $in: tagIds }}, { $addToSet: {
7             genres: tag.name
8         }};
9     })

```

---

### Isječak koda 2: Dodavanje žanrova knjigama

Zadnji korak agregiranje korisnika i njihovih ocjena knjiga u jednu kolekciju. U ovom koraku se svakom korisniku dodjeljuje skup knjiga koje su ocijenili.

---

```

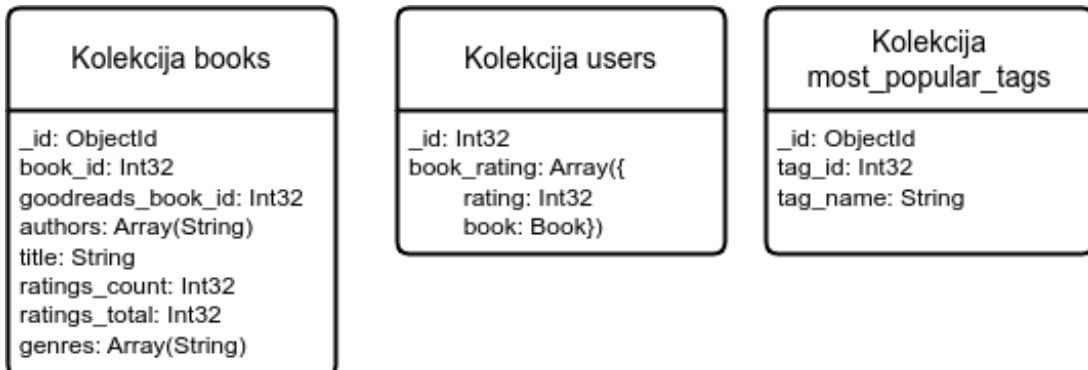
1 db.ratings.aggregate([
2     { $lookup: { from: "books", localField: "book_id", foreignField: "book_id", as:
3         "book", },
4         { $project: { _id: 1, user_id: "$user_id", rating: "$rating", book: {
5             $arrayElemAt: [ "$book", 0 ], }, }, },
6         { $group: { _id: "$user_id", book_ratings: { $push: { rating: "$rating", book:
7             "$book", }, }, }, },
8         { $out: "users", },
9     })

```

---

### Isječak koda 3: Dodjela knjiga korisnicima

Završne kolekcije u bazi se mogu reprezentirati kao što je prikazano na sljedećoj slici 3. Vidljivo je da su napravljene tri kolekcije books, users i most\_popular\_tags. Kolekcija books sadržava knjige i ima na novo dodano polje genres što je lista žanrova knjige. Kolekcija users su korisnici, identificirani samo svojim id-om, te svaki korisnik ima listu knjiga koje je ocijenio, svaki element liste je ocjena i cijeli zapis ocjenjene knjige.



Slika 3: Izvedene kolekcije

### 3.3. Implementacija

Sustav je implementiran u programskom jeziku Go. Za prikaz preporuka kreiran je web poslužitelj koji poslužuje web stranicu sa preporukama za određenog korisnika. Konkretno se preporučuju knjige. Aplikacija za komunikacijom s MongoDB koristi Mongo Driver biblioteku. Na web stranici postoji nekoliko vrsta preporuka:

- preporuke temeljene na kolaborativnom filtriranju,
- preporuke određene prema najvišim ocjenama knjiga
- preporuke određene prema najvišim ocjenama knjiga prema određenom žanru

Korisnici imaju mogućnost dodavanja, izmjene, brisanja ocjena knjigama, time potencijalno mijenjanju preporuke za "slične" korisnike i utječu na prosječnu ocjenu preporuka koje su određene prema najvišim ocjenama.

U nastavku svi implementirani MongoDB upiti će zbog lakšeg čitanja biti pretvoreni u upite koji se mogu izvršiti u mongosh.

#### 3.3.1. Najbolje ocjenjene knjige

Jedan od načina na koji se preporučuju knjige je pronalaženje knjiga s najboljim ocjenama korisnika. Za ovaj scenarij imamo dva slučajeva, preporuke knjiga koje imaju najbolje prosječne ocjene korisnika i preporuke knjiga koje imaju najbolje prosječne ocjene korisnika u određenom žanru.

Upit koji se postavlja bazi podataka za dohvatanje knjiga je sljedeći:

---

```

1 db.users.aggregate([
2   { $unwind: "$book_ratings" },
3   { $group: { _id: "$book_ratings.book", totalRatings: { $sum:
4     "$book_ratings.rating"}, count: { $sum: 1 } } },
5   { $project: { _id: "$_id._id", book: "$_id", totalRating: 1, count: 1,
6     averageRating: { $divide: ["$totalRating", "$count"] } } },
7   { $sort: { averageRating: -1 } },
8   { $project: { _id: "$_id._id", book: "$_id", totalRating: 1, count: 1,
9     averageRating: { $round: ["$averageRating", 2] } } },
10 ])

```

---

#### Isječak koda 4: Upit za najbolje ocjenjene knjige

Upit prolazi kroz kolekciju `users` te prvo za svakoga korisnika listu njihovih pretvara u jednu listu koja se nakon toga grupira prema knjigama, izračunava se ukupna ocjena i broj danih ocjena. Zatim treći korak `project` prikazuje polja objekta koja želimo prikazati i kreira prosječnu ocjenu za svaku knjigu. Knjige su sortirane silazno prema prosječnoj ocjeni te se prosječna ocjena zaokružuje na dvije decimale.

### 3.3.2. Kolaborativno filtriranje

Kolaborativno filtriranje je implementirano kroz algoritam K-najbližih susjeda (KNN - k-nearest neighbours). Prvo odabiremo korisnika za kojega želimo generirati preporuke potom radimo upit u bazu za sve korisnike koji nisu taj korisnik. Za svakog korisnika određujemo vrijednost koja označava kako je sličan odabranom korisniku. To se radi na način da ako korisnik ima ocjenjene knjige koje ima i odabrani korisnik dobiva brojčanu vrijednost ovisno o sličnosti ocjena. Tako će korisnik koji ima jednake ocjene knjiga kao odabrani imati vrijednost 1.0 što bi označavalo da su korisnici jednakim tj. prepostavljamo da vole jednake knjige, dok vrijednost od 0.0 označava da se korisnici uopće ne podudaraju. Poslije izračuna vrijednosti korisnici se sortiraju silazno prema izračunatim vrijednostima te se za preporuke uzimaju knjige koje nisu ocjenjene od odabranog korisnika, a jesu među ocjenjenim knjigama najsličnijih.

### 3.3.3. Dodavanje, brisanje, promjena ocjena

Što se tiče dodavanja, brisanja i promjena ocjena koriste se sljedeći upiti:

```
1 db.users.updateOne({ _id: userId }, { $push: { book_ratings: { rating: newRating,
  ↵   book: book } } })
```

#### Isječak koda 5: Dodavanje ocjene knjizi

U isječku koda 5 `userId`, `newRating` i `book` su varijable koje se proslijeđuju u upit. Sam upit za dodavanje pronalazi korisnika s vrijednošću `_id` jednakom `userId` te u listu `book_ratings` dodaje knjigu i ocjenu.

```
1 db.users.updateOne({ _id: userId }, { $set: { "book_ratings.$[x].rating": newRating
  ↵   } }, { arrayFilters: [ { "x.book.book_id": bookId } ] })
```

#### Isječak koda 6: Promjena ocjene

U isječku koda 6 se koriste `arrayFilters` kako bi se odabrao i ažurirao točan zapis u listi ocjena. Opcija `arrayFilters` postavlja `x` kao identifikator indeksa liste na kojem se nalazi knjiga s `bookId` te se u vrijeme pokretanja upita ažurira samo to polje liste.

Zadnji upit je brisanje ocjene. Brisanje se radi korištenjem `$pull` naredbe koja će iz liste maknuti zapise koji zadovoljavaju zadane uvjete.

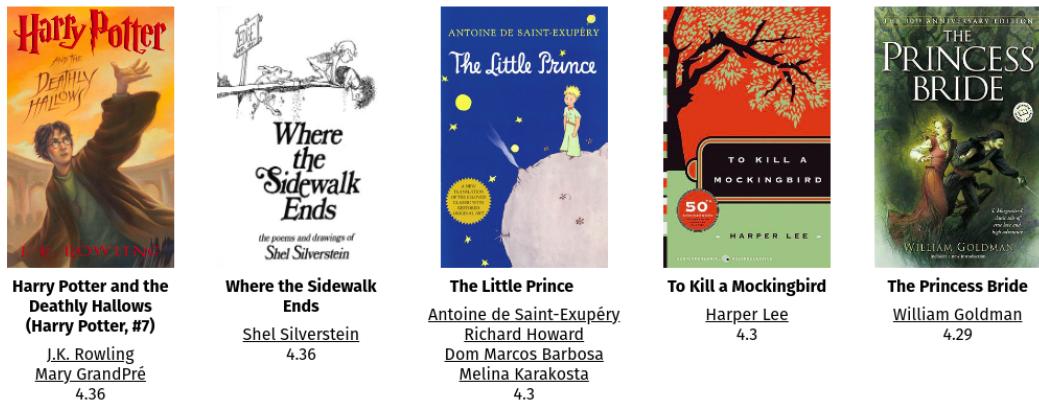
```
1 db.users.updateOne({ _id: userId }, { $pull: { book_ratings: { "book.book_id": 
  ↵   bookId } } })
```

#### Isječak koda 7: Brisanje ocjene

## 4. Primjeri korištenja

Slika 4 prikazuje neke od knjiga koje su dane kao preporuke prema najboljim ocjenama. Ocjene knjiga su vidljive ispod naslova knjige i autora.

### Najpopularnije knjige



Slika 4: Najpopularnije knjige

Slika 5 prikazuje neke od knjiga koje su dane kao preporuke prema najboljim ocjenama uz odabrani tag tj. žanr. Prikazane su samo knjige koje imaju dodijeljen odabrani žanr high-fantasy.

### Najpopularnije knjige prema odabranom tagu



Slika 5: Najpopularnije knjige prema odabranom tagu

Slika 6 prikazuje korisnika koji ima najsličnije ocjene knjiga trenutno. Možemo primijetiti da oba korisnika imaju dane ocjene za knjige The Da Vinci Code, Memoirs of a Geisha i druge. Neke od ocjena su također jednake što više pridonosi sličnosti korisnika. U zagradi pored Korisnik ID: 127 je zapisan broj koji odgovara sličnosti između ova dva korisnika.



Slika 6: Prikaz sličnog korisnika trenutnom odabranom



Slika 6: Prikaz sličnog korisnika trenutnom odabranom

Slika 7 prikazuje preporuke za odabranog korisnika na slici 6. Sve preporuke su uvijek knjige koje taj korisnik nije ocijenio i nalaze se u skupu knjiga koje su ocijenili slični korisnici.

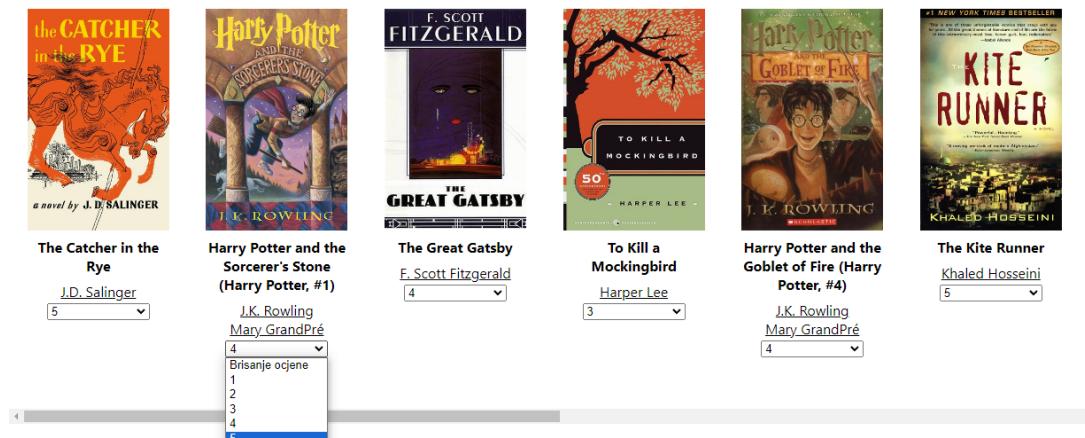
Preporučene knjige (Kolaborativno filtriranje) (20)



Slika 7: Preporuke

Slika 8 sadrži prikaz načina promjene ocjene.

### Moje ocjene knjiga



Slika 8: Promjena, brisanje ocjene

## 5. Zaključak

Kroz rad su prikazane osnovne ideje polustrukturiranih baza podataka, objašnjeni način na koji ovakve vrste baza podataka pohranjuju podatke i struktura tih podataka. Opisan je MongoDB korišten za pohranu podataka i pokretanje upita na podacima, te je navedena osnovna ideja sustava za preporuke, različite vrste sustava i načina implementacija.

U poglavljiju 3 prikazana je struktura podataka koji se koriste za preporučivanje knjiga. Opisan je proces pretvorbe podataka iz odabranog skupa podataka u polustrukturirane. Za implementaciju aplikacije korišten je programski jezik Go. Odabran je zbog svoje relativne jednostavnosti i brzine razvijanja aplikacija. Ima dobru podršku za komunikaciju s bazom MongoDB. Implementirana je web aplikacija za preporuke knjiga.

Web aplikacija omogućuje korisnicima ocjenjivanje knjiga. Ovisno o ocjenama korisnika predlaže knjige koje su ocijenili slični korisnici. Slične korisnike određuje koristeći algoritam K-najbližih susjeda. Svakom promjenom ocjena, ocjenjivanjem nove knjige ili brisanjem ocjena korisnik će potencijalno dobiti nove preporuke.

# Popis literatúre

- [1] P. Buneman, „Semistructured data,” *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 1997., str. 117–121.
- [2] R. Ganguly i A. Sarkar, „Evaluations of conceptual models for semi-structured database system,” *International Journal of Computer Applications*, sv. 50, br. 18, 2012.
- [3] V. Abramova i J. Bernardino, „NoSQL databases: MongoDB vs cassandra,” *Proceedings of the international C\* conference on computer science and software engineering*, 2013., str. 14–22.
- [4] P. Melville i V. Sindhiani, „Recommender systems.,” *Encyclopedia of machine learning*, sv. 1, str. 829–838, 2010.
- [5] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang i T. Zhou, „Recommender systems,” *Physics reports*, sv. 519, br. 1, str. 1–49, 2012.

# **Popis slika**

|    |  |    |
|----|--|----|
| 1. | Primjer polustrukturiranih podataka u obliku grafa [1] . . . . . | 2  |
| 2. | Dijagram strukture podataka korištenog skupa podataka . . . . .  | 5  |
| 3. | Izvedene kolekcije . . . . .                                     | 7  |
| 4. | Najpopularnije knjige . . . . .                                  | 9  |
| 5. | Najpopularnije knjige prema odabranom tagu . . . . .             | 9  |
| 6. | Prikaz slicnog korisnika trenutnom odabranom                     | 10 |
| 7. | Preporuke . . . . .  | 10 |
| 8. | Promjena, brisanje ocjene . . . . .                              | 10 |

# **Popis isječaka koda**

|    |   |   |
|----|---|---|
| 1. | Kreiranje najpopularnijih tagova . . . . .  | 6 |
| 2. | Dodavanje žanrova knjigama . . . . .        | 6 |
| 3. | Dodjela knjiga korisnicima . . . . .        | 6 |
| 4. | Upit za najbolje ocjenjene knjige . . . . . | 8 |
| 5. | Dodavanje ocjene knjizi . . . . .           | 8 |
| 6. | Promjena ocjene . . . . .                   | 8 |
| 7. | Brisanje ocjene . . . . .                   | 8 |