

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Martin Sakač**

**APLIKACIJA ZA PREPORUKE FILMOVA -  
POLUSTRUKTURIRANE BAZE PODATAKA -  
MONGODB**

**PROJEKT**

**TEORIJA BAZA PODATAKA**

**Varaždin, 2022.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Martin Sakač**

**Matični broj: 45998/17–R**

**Studij: Informacijsko i programsko inženjerstvo**

**APLIKACIJA ZA PREPORUKE FILMOVA - POLUSTRUKTURIRANE  
BAZE PODATAKA - MONGODB**

**PROJEKT**

**Mentor:**

dr. sc. Bogdan Okreša Đurić

**Varaždin, veljača 2022.**

*Martin Sakač*

### **Izjava o izvornosti**

Izjavljujem da je moj 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 potvrdio prihvatanjem odredbi u sustavu FOI Radovi*

---

## Sažetak

U ovom radu koristit će se polustrukturiran baza MongoDB koja radi sa dokumentima umjesto sa tablicama kao što to rade relacijske baze. Pomoću Node.js-a, Expressa i MongoDB-a kreirat će se web aplikacija za preporuku filmova. U radu će biti prikazan opis aplikacijske domene i teorijski uvod. Zatim slijedi prikaz modela baze podataka. Nakon što se upoznamo sa modelom, prikazat ćemo implementaciju u navedenim tehnologijama te prikazi izgled korisničkog sučelja. Na kraju će biti dan zaključak

**Ključne riječi:** Polustrukturirane baze, MongoDB, NoSQL, Node.js, filmovi, preporuke.

# Sadržaj

1.	Opis aplikacije domene . . . . .	1
2.	Teorijski uvod . . . . .	2
3.	Model baze podataka . . . . .	3
4.	Implementacija . . . . .	5
5.	Primjeri korištenja . . . . .	9
6.	Zaključak . . . . .	17
	Popis literature . . . . .	18
	Popis slika . . . . .	19
	Popis isječaka koda . . . . .	20

# 1. Opis aplikacije domene

Tema aplikacije je izrada sustava za preporuku filmova. Cilj aplikacije je skupiti sve važne korisničke podatke u bazu podataka. Na temelju tih podataka prikazat će se preporuka filmova koji odgovaraju korisnikovom ukusu.

Unutar aplikacije, administrator će kreirati kategorije i filmove. Svakom filmu biti će moguće dodati više kategorija ovisno o vrsti filma. Korisnik će se moći registrirati i prijaviti. Nakon registracija prikazani će biti svi filmovi koji se nalaze u bazi podataka. Korisnik će biti u mogućnosti komentirati odnosno ocijeniti filmove koje je pogledao. Na temelju korisničkih ocjena izvući ćemo faktor preporuke za svaku od kategorije filmova. Ovisno o tome koje kategorije film sadrži i koje faktore korisnik ima za svaku pojedinu kategoriju, prikazat će se preporučeni filmovi.

Za realizaciju ove aplikacije koristit će se sljedeće tehnologije:

- MongoDB - polustrukturirana baza podataka
- Node.js - programski alat baziran na JavaScript okruženju za razvoj i povezivanje baze podataka
- Express - JavaScript mikrookvir za razvoj podataka u Node.js okolini
- Mongoose - paket za jednostavniju upotrebu MongoDB u Node.js okolini
- Embedded JavaScript - jezik predložaka za generiranje HTML oznaka pomoću JavaScripta

Motivacija za korištenje ovih tehnologija je veoma jednostavna. MongoDB je savršena baza podataka za korištenje Node.js-a. MongoDB sadrži podatke, ne kao tablice, već kao zbirku dokumenata u JSON formatu.

## 2. Teorijski uvod

NoSQL odnosno polustrukturirana baza podataka je relativno nova vrsta baza podataka. Unatoč tome danas je nezaobilazni dio za funkcioniranje mnogih projekata i proizvoda diljem svijeta. Najveća razlika između polustrukturiranih i relacijskih baza podataka je način spremanja podataka. Kod standardnih relacijskih baza podaci se spremaju u fiksne tablice dok se kod polustrukturiranih baza spremaju u takozvane dokumente. Primjerice MongoDB u svoje baze podataka pohranjuje kolekcije. Te kolekcije bi se mogle usporediti sa tablicama u klasičnim relacijskim bazama no postoji bitna razlika. Shema podataka unutar kolekcije je proizvoljna. Za razliku od relacijskih baza podataka, nije potrebno predefinirati shemu podataka, već se shema može prilagoditi podacima koji će se spremati u tu kolekciju. Unutar kolekcije nalaze se dokumenti sa svojim atributima. Kao što je rečeno, ti dokumenti odnosno njihove sheme su dinamične te su podaci unutar njih spremljeni u JSON (eng. JavaScript Object Notation) podatkovnom formatu.[1], [2]

Još jedna velika razlika između relacijskih i polustrukturiranih bazi podataka je u njihovoj sintaksi. Relacijske baze podataka koriste SQL jezik kako bi manipulirali podacima. U polustrukturiranim bazama koriste se funkcije za operacije nad podacima kao što su dodavanje, brisanje i ažuriranje. Polustrukturirane baze su također veoma fleksibilne. Zbog te fleksibilnosti, nije potrebno obraćati pažnju na ograničenja, primarne ključeve, vanjske ključene, kršenje pravila kod brisanja reda koji je vanjski ključ i slično. U takve baze možemo spremati kompleksne podatke kao što su objekti i nizovi. Ako je potrebna brza i fleksibilna baza podataka bez ograničenja između podataka, logički izbor su polustrukturirane baze podataka.[1], [2]

### 3. Model baze podataka

Prije bilo kakve izrade aplikacije potrebno je kreirati model baze podataka. Kao što je već rečeno, baza podataka koja će se koristiti je MongoDB. MongoDB se temelji na document modelu. MongoDB sprema podatke u dokumente po binarnoj reprezentaciji JSON-a. Taj format se još naziva i BSON (eng Binary JavaScript Object Notation). Svaki dokument u MongoDB bazi sadrži svoj specifični tip podataka.[1], [2]

Za ovu aplikaciju bila je potrebna baza u koju ćemo spremati filmove, korisnike i kategorije. Kako bi mogli preporučiti korisniku filmove, potrebna nam je neka kategorizacija prema kojoj možemo prepoznati uzorke po koji korisnik rangira filmove. Zbog toga također svaki film ima i svoje kategorije. Osim toga potrebno je bilo i spremište u koje će se spremati ocjene koje korisnik daje pojedinom filmu.

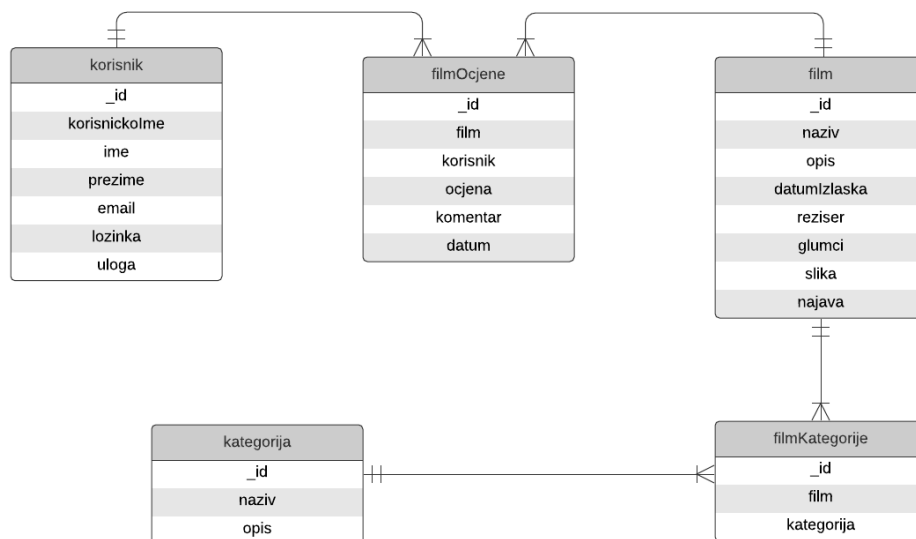
Na slici 1. prikazan je model baze podataka za Aplikaciju za preporuke filmova. Baza se sastoji od pet kolekcija:

- korisnik
- kategorija
- film
- filmKategorije
- filmOcjene

Kolekcija korisnik sadrži polja id, korisnickolme, ime, prezime, email, lozinku i ulogu. Kolekcija kategorije sadrži id, naziv i opis. Kolekcija filma sadrži id, naziv, opis, datum izlaska filma, režisera, glumce, sliku te najavu. Kolekcija filmOcjene nam služi kako bi spremali ocjene koje korisnik daje pojedinom filmu. Ta kolekcija sadrži id, film (id na neki film), korisnik (id na korisnika), ocjenu, komentar te datum. Kako bi spremali kategorije filmova potrebna nam je kolekcija. Za to služi kolekcija filmKategorije u koje spremamo id filma i id kategorije.

Naime, ovaj model i ne pokazuje najbolje snagu polustrukturiranih baza podataka iz jednostavnog razloga. Ovaj model se veoma jednostavno mogao svesti na samo tri kolekcije. U kolekciju film smjestili bi polje kategorije koje bi spremalo listu kategorija odnosno ključeve od kategorija. Osim toga također je bilo moguće dodati polje ocjene u koje bi spremali objekte ocjene sa svim potrebnim svojstvima. Zbog toga, ovaj model malo više vuče na standardni relacijski model jer smo koristili kolekcije "više-više" koji su se mogli spremati u jednu kolekciju..





Slika 1: Model baze podataka

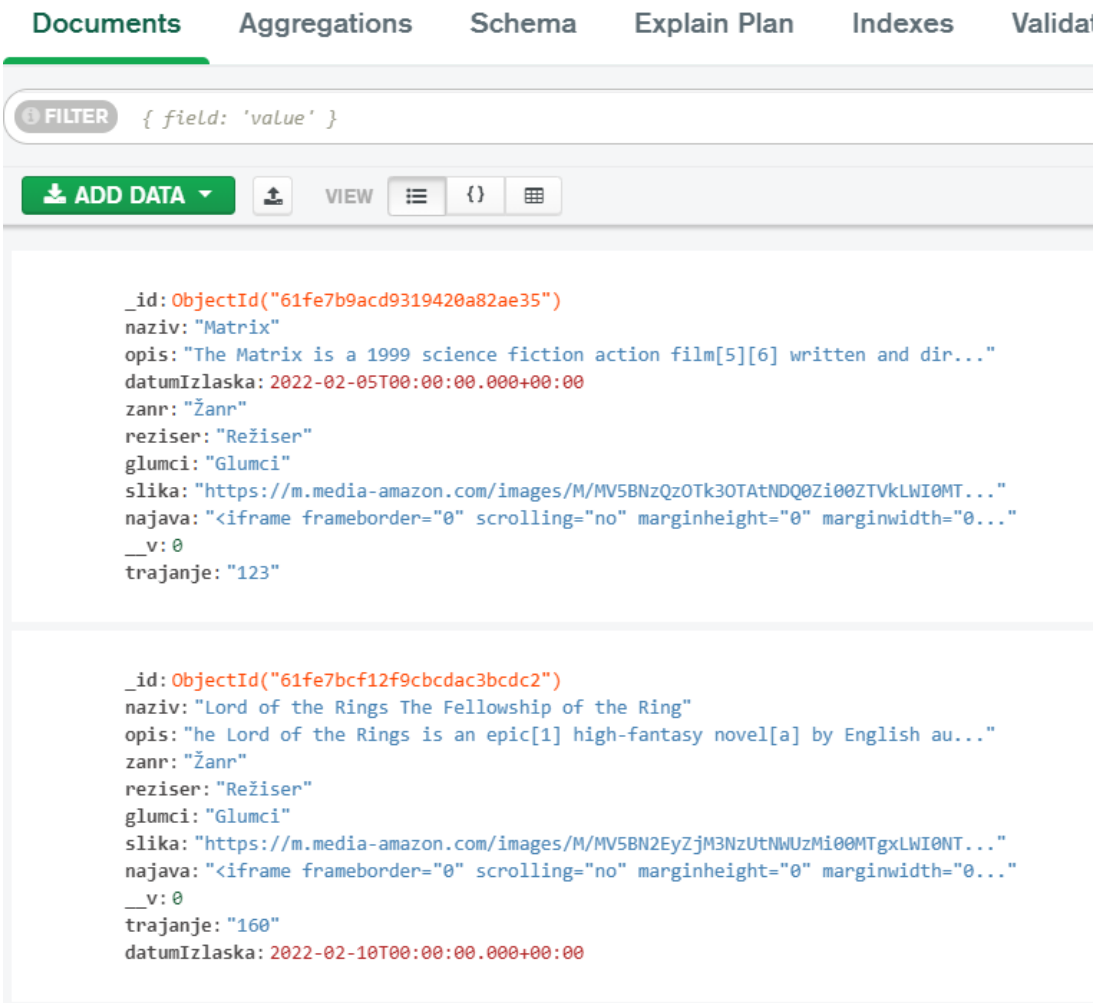
## 4. Implementacija

Za implementaciju baze korištene je JavaScript okruženje Node.js. Ova tehnologija danas je prilično popularna te je ova prilika za rad sa MongoDB iskorištena u Node.js. Prije same implementacije potrebno je bilo instalirati Node.js i MongoDB na računalo. Nakon što je Node.js instaliran, kreirana je nova projektna datoteka. U toj datoteci, pomoću terminala, prvo je inicijaliziran projekt sa komandom `npm init -y`. Nakon što je inicijaliziran projekt, dodani su svi potrebni paketi potrebni za kreiranje baze podataka i web aplikacije. Projektu smo dodali alate: `express` (pomaže u razvoju baze podataka u Node.js okolini), `mongoose` (paket za jednostavniju upotrebu MongoDB-a i kreiranje shema), `ejs` (jezik predložaka za generiranje HTML oznaka pomoću JavaScripta), `methodoverride` (mogućnost slanja i primanja PUT i DELETE zahtjeva), `express-session` (kreiranje korisničke sesije tokom uspješne prijave)

Za instalaciju alata koristi se komanda `npm i express mongoose ejs method-override express-session`. Zatim je još instaliran `nodemon` koji olakšava restartanje servera prilikom spremanja promjena sa komandom `npm i --save-dev nodemon`. Nakon što su projektu dodani svi potrebni alati kreirana je `server.js` datoteka u kojoj se dohvaćaju svi alati, usmjerivači i sheme podataka. Zatim se pomoću `mongoose` spajamo na lokalnu MongoDB bazu podataka te pokrećemo server na portu 9999

### Isječak kôda 1: Isječak koda iz server.js datoteke

```
1  const express = require('express')
2  const mongoose = require('mongoose')
3
4  const Korisnik = require('./models/korisnik')
5  const Filmovi = require('./models/filmovi')
6  const Kategorije = require('./models/kategorije')
7  const FilmKategorije = require('./models/filmKategorije')
8  const FilmOcjene = require('./models/filmOcjene')
9
10 const korisnikRouter = require('./routes/korisnik')
11 const filmoviRouter = require('./routes/filmovi')
12 const kategorijeRouter = require('./routes/kategorije')
13 const filmKategorijeRouter = require('./routes/filmKategorije')
14
15 mongoose.connect('mongodb://localhost/tbprojekt', { useNewUrlParser: true,
16   useUnifiedTopology: true })
17
18 const app = express()
19 app.set('view engine', 'ejs')
20
21 app.use('/korisnik', korisnikRouter)
22 app.use('/filmovi', filmoviRouter)
23 app.use('/kategorije', kategorijeRouter)
24 app.use('/film-kategorije', filmKategorijeRouter)
25
26 app.listen(9999, () => {
27   console.log('Server live at 9999')
28 })
```



Slika 2: Kolekcija filmova

Na slici 2. vidljiv je izgled kolekcije dokumenata za filmove. Kako bismo dobili ovakav dokument, pomoću mongoose smo kreirali shemu koja je vidljiva na Isječku koda 2. Ova shema se instancirala u svakom preusmjerivaču odnosno routeru koji radi sa tom kolekcijom.

#### Isječak kôda 2: Shema kolekcije filmova

```

1 const mongoose = require('mongoose')
2 const filmoviSchema = new mongoose.Schema({
3   naziv:{
4     type: String,
5     required: true
6   },
7   opis:{
8     type: String,
9     required: true,
10  },
11  datumIzlaska:{
12    type: Date,
13    default: Date.now

```

```

14     },
15     reziser: {
16         type: String,
17         required: true
18     },
19     trajanje: {
20         type: String,
21         required: true
22     },
23     glumci: {
24         type: String,
25         required: true,
26     },
27     slika: {
28         type: String,
29         required: true
30     },
31     najava: {
32         type: String,
33     }
34 })
35 module.exports = mongoose.model('Filmovi', filmoviSchema)

```

Kreiran je novi objekt mongoose scheme te smo tom objektu dodali svojstva odnosno properties. Svako svojstvo ima svoj tip, dali je potreban unos ili nije te default vrijednost. Na kraju smo eksportali tu shemu kao model filmovi.

Nadalje ćemo na primjeru filmova prikazati na koji način se pomoću navedene scheme odnosno model, pomoću usmjerivača mogu dodavati, brisati, uređivati i dohvatiti dokumenti.

### Isječak kôda 3: Dohvaćanje filmova

```

1 router.get('/crud', async (req, res) => {
2     const filmovi = await Film.find().sort({ datumIzlaska: 'desc' })
3
4     filmoviZaStranicu = await dohvatiKategorijeZaSvakiFilm(filmovi)
5     res.render('filmovi/filmoviManage', { filmovi: Array.from(filmoviZaStranicu)})
6 })

```

Na isječku koda 3. možemo vidjeti da se radi o asinkronoj funkciji koja čeka da se dohvate filmovi. Filmovi se dohvaćaju sa funkcijom `find()` te ih sortiramo silazno sa funkcijom `.sort()` prema polju `datumIzlaska`. Također osim ovih funkcija postoje još mnoge, primjerice `findById()`, `findOne()`, `findOneAndUpdate()` i još mnoge..

### Isječak kôda 4: Kreiranja novog dokumenta filma

```

1 router.post('/novi', async (req, res) => {
2     const film = new Film({
3         naziv: req.body.naziv,
4         opis: req.body.opis,
5         datumIzlaska: req.body.datum,
6         zanr: req.body.zanr,
7         reziser: req.body.reziser,

```

```

8      trajanje: req.body.trajanje,
9      glumci: req.body.glumci,
10     slika: req.body.slika,
11     najava: req.body.najava
12   })
13   await film.save()
14   res.redirect('/filmovi/crud')
15 })

```

Na isječku koda 4 vidimo post usmjerivač koji kreira novi objekt filma prema shemi koja je kreirana pomoću mongoosa. Pomoću parametra request dohvacamo sve podatke te ih dodjeljujemo pojedinim svojstvima. Na kraju čekamo da se ti dokument spremi u kolekciju filmova sa funkcijom `save()`

#### Isječak kôda 5: Brisanje dokumenta

```

1 router.delete('/crud/:id', async (req, res) => {
2   await Film.findByIdAndDelete(req.params.id)
3   await FilmKategorije.find({film:req.params.id}).deleteMany()
4   await FilmOcjene.find({film:req.params.id}).deleteMany()
5   res.redirect('/filmovi/crud')
6 })

```

Na isječku koda 5 vidimo delete usmjerivač koji modela Film pronalazi prema id-u dokument u bazi podataka te ga izbriše pomoću funkcije `findByIdAndDelete()`. Nakon toga smo također obrisali kategorije i ocjene toga filma.

#### Isječak kôda 6: Ažuriranje dokumenta

```

1 router.put('/uredi/:id', async (req, res) => {
2   await Film.findById(req.params.id)
3   .then((model) => {
4     return Object.assign(model, {
5       naziv: req.body.naziv,
6       opis: req.body.opis,
7       datumIzlaska: req.body.datum,
8       reziser: req.body.reziser,
9       trajanje: req.body.trajanje,
10      glumci: req.body.glumci,
11      slika: req.body.slika,
12      najava: req.body.najava
13    });
14  })
15  .then((model) => {
16    return model.save()
17  });
18  res.redirect('/filmovi/crud')
19 })

```

Na isječku koda 6 pronalazimo film pomoću funkcije `findById()` te nakon što pronademo taj film dodamo mu nove vrijednosti iz zahtjeva. Nakon što su nove vrijednosti dodane, taj dokument ponovno spremamo u bazu podataka.

## 5. Primjeri korištenja

Otvaranjem web stranice prikazuje se prijava. Korisnik se najprije mora registrirati sa svojim podacima. Korisničko ime je jedinstveno za svakog korisnika. Nakon što se korisnik registrirao, server ga prebacuje na stranicu prijave gdje je potrebno unijeti korisničko ime i lozinku kako bi se uspješno prijavili. Nakon što se korisnik prijavi prikazuje se početna stranica sa svim filmovima.

U navigaciji imamo 5 gumba. Korisnik se pomoću navigacije može odjaviti pritiskom na Odjavi se. Zatim imamo početnu. To je stranica na koji je korisnik preusmjeren nakon prijave.

Sljedeća stranica je Kategorije CRUD. Ovdje su korisniku vidljive sve kategorije koje trenutno na stranici postoje zajedno sa opisom. Svaka kategorija se može urediti i obrisati pritiskom na tipke Uredi i Obriši. Također u gornjem desnom kutu vidljiva je tipka za dodavanje nove kategorije koja korisnika preusmjerava na stranicu za dodavanje kategorije. Na toj stranici korisnik mora unesti naziv i opis kategorije te potvrditi unos pritiskom na gumb Spremi.

Stranica Filmovi CRUD prikazuje korisniku sve filmove, zajedno sa kategorijom koju ti filmovi sadrže, opisom, te slikom. Ovdje korisnik može urediti film, obrisati film te urediti kategorije. Pritiskom na uredi kategoriju korisniku se prikazuju sve trenutne kategorije za taj film. Iz dropdown liste može odabrati novu kategoriju te ju dodati navedenom filmu. Također moguće je izbrisati kategorije filma ako je to potrebno. U gornjem desnom kutu vidljiva je tipka za dodavanje novoga filma koja otvara stranica sa formom. Na toj stranici prikazani su svi potrebni podaci koje korisnik mora unijeti kako bi uspješno kreirao novi film.

Na stranici Filmovi CRUD korisnik može pritisnuti na neki film te se onda prikazuju detalji tog filma. Na kraju te stranica vidljiva je tipka za dodavanje ocjene gdje korisnik unosi svoj komentar i ocjenu od 1 do 5. Ukoliko je korisnik već ocjenio taj film, gumb za dodavanje ocjene nije vidljiv. Nadalje su prikazani svi komentari koje su ostali korisnici napisali za taj film te je vidljiva njihova ocjena. Korisnik kraj svojeg komentara ima gumb za brisanje tog komentara. Nakon što obriše komentar ponovno može dodati komentar tom filmu.

Zadnja stranica preporučeno ujedno je i tema ovog projekta. Na temelju ocjena koje korisnik daje nekom filmu, te ocjene se također daju i toj kategoriji. Primjerice ukoliko korisnik ocjeni film X sa ocjenom 5 koji ima kategorije A i B, tada kategorije A i B također imaju ocjene 5. Ovaj proces je personaliziran za svakog korisnika. Sve ocjene za pojedinu kategoriju se sumiraju te podjele sa 5 pošto je to najveća moguća ocjena. Zatim sredinu ukupne ocjene kategorije i sumu ocjena kategorije pomnožimo i dobijemo faktor neke kategorije za korisnika. Kad neki film ima dvije ili više kategorije, njihovi faktori se zbroje te time dobijemo faktor preporuke za neki film. Taj faktor pridodamo svakom filmu koji korisnik nije ocjenio te ih sortiramo silazno. Na taj način prikazujemo filmove na stranici Preporučeno.

Nadalje slijede slike web aplikacije.

Prijavi se

Prijava

Korisničko ime:

Lozinka:

Prijavi se Registracija

Slika 3: Prijava

**Registracija**

Korisničko ime:

Ime:

Prezime:

Email:

Lozinka:


[Registriraj se](#) [Prijava](#)

Slika 4: Registracija



FILMOVI

Novi Film



### Lord of the Rings The Fellowship of the Ring

10. 02. 2022.

he Lord of the Rings is an epic[1] high-fantasy novel[a] by English author and scholar J. R. R. Tolkien. Set in Middle-earth, intended to be Earth at some distant time in the past, the story began as a sequel to Tolkien's 1937 children's book The Hobbit, but eventually developed into a much larger work. Written in stages between 1937 and 1949, The Lord of the Rings is one of the best-selling books ever written, with over 150 million copies sold[2] The title refers to the story's main antagonist, the Dark Lord Sauron, who in an earlier age created the One Ring to rule the other Rings of Power given to Men, Dwarves, and Elves, in his campaign to conquer all of Middle-earth. From homely beginnings in the Shire, a hobbit land reminiscent of the English countryside, the story ranges across Middle-earth, following the quest to destroy the One Ring mainly through the eyes of the hobbits Frodo, Sam, Merry and Pippin.

Znanstvena Fantastika
Akcija
160 min
Više

Uredi Film
Uredi kategorije
Obriši


### Matrix

05. 02. 2022.

The Matrix is a 1999 science fiction action film[5][6] written and directed by the Wachowskis.[a] It is the first installment in The Matrix film series, starring Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, and Joe Pantoliano. It depicts a dystopian future in which humanity is unknowingly trapped inside a simulated reality, the Matrix, which intelligent machines have created to distract humans while using their bodies as an energy source.[7] When computer programmer Thomas Anderson, under the hacker alias "Neo", uncovers the truth, he "is drawn into a rebellion against the machines"[7] along with other people who have been freed from the Matrix.

Akcija
Znanstvena Fantastika
123 min
Više

Uredi film
Uredi kategorije
Obriši



Slika 5: Filmovi CRUD



# LORD OF THE RINGS THE FELLOWSHIP OF THE RING

Dodaj kategoriju

Odaberi kategoriju ▼

Odustani Spremi

Kategorije filma

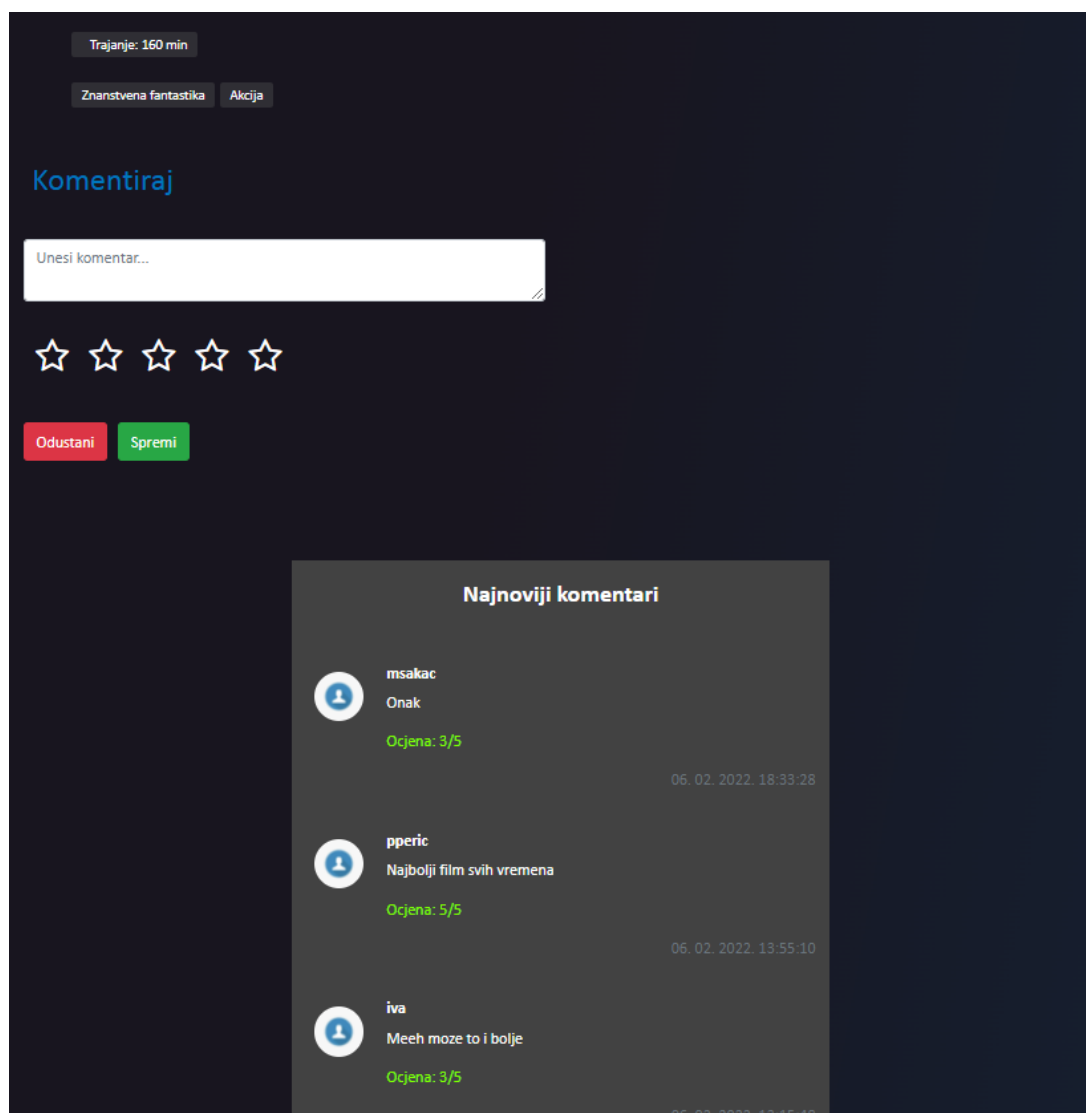
Znanstvena fantastika

Obriši

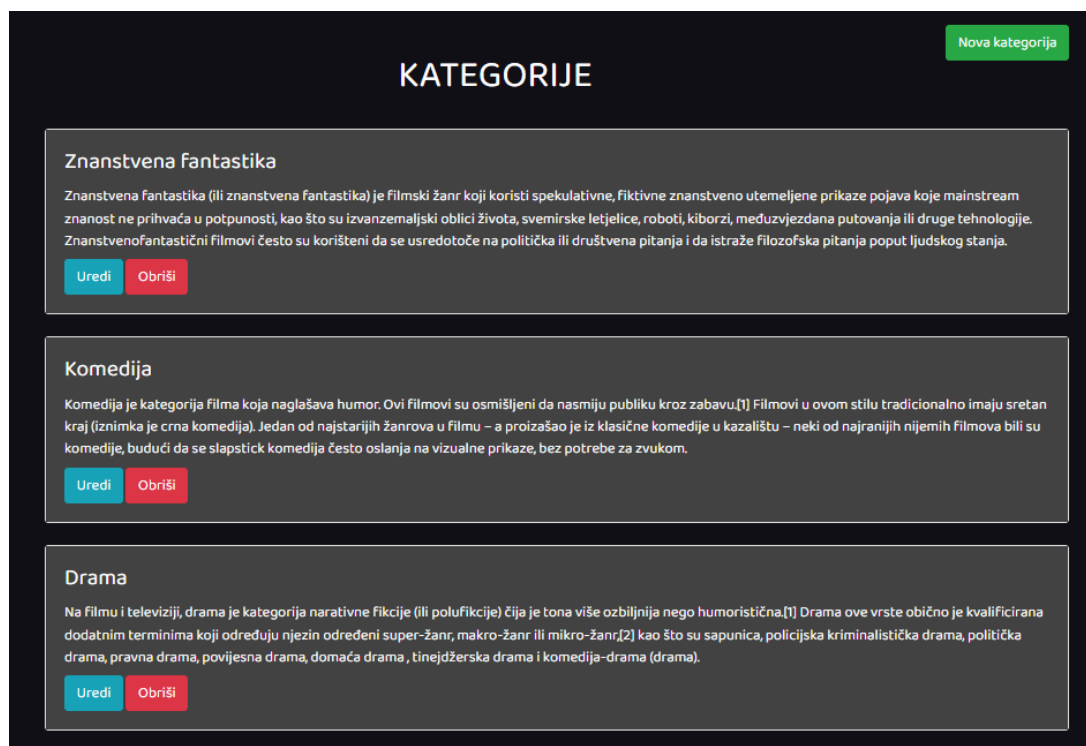
Akcija

Obriši

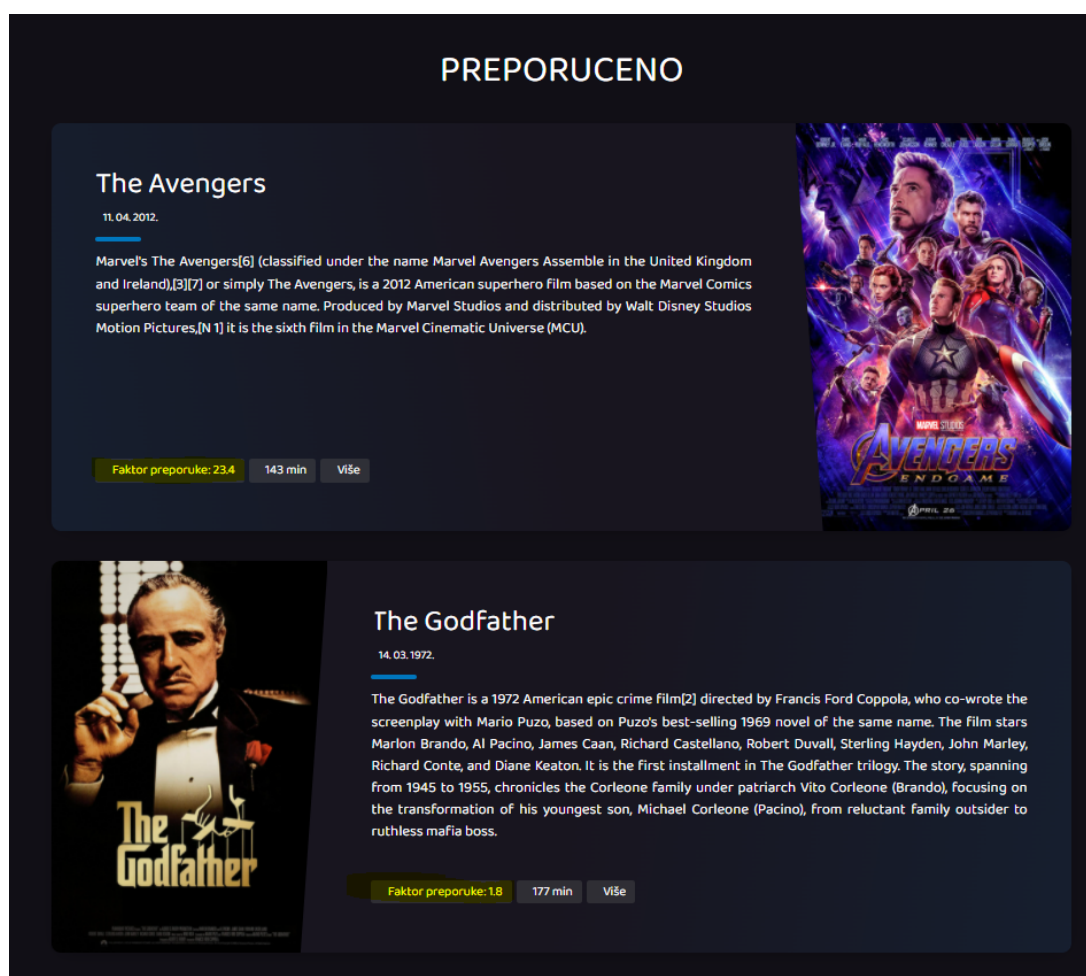
Slika 7: Azuriranje kategorije filma



Slika 8: Ocjenjivanje filma



Slika 9: Kategorije CRUD



Slika 10: Stranica preporuka

## 6. Zaključak

U sklopu ovog projekta izrađena je aplikacija za preporuku filmova. Ova tema bila je vrlo zanimljiva no isto tako i kompleksna. Postoje veoma kompleksne metode kojima se može procijeniti koji film, knjiga ili muzika će se korisniku svidati na temelju sakupljenih podataka o korisniku. Korištenje polustrukturiranih baza dolazi sa mnogim prednostima, no te prednosti bilo je pomalo teško iskoristiti pošto je u glavi i dalje relacijski model. Ovaj projekt izrađen je na način da se nauči veoma popularan stack sa Node.js. Također MongoDB sa paketom mongoose uvelike olakšava rad sa podaci u Node.js-u. Ako je potrebna brza i fleksibilna baza podataka za razvoj web aplikacija sa novim tehnologijama, logički izbor su polustrukturirane baze podataka odnosno MongoDB.

# Popis literature

- [1] A. Boicea, F. Radulescu i L. I. Agapin, „MongoDB vs Oracle – Database Comparison,” *2012 Third International Conference on Emerging Intelligent Data and Web Technologies*, 2012., str. 330–335. DOI: 10.1109/EIDWT.2012.32.
- [2] S. Chickerur, A. Goudar i A. Kinnerkar, „Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications,” *2015 8th International Conference on Advanced Software Engineering Its Applications (ASEA)*, 2015., str. 41–47. DOI: 10.1109/ASEA.2015.19.

# Popis slika

1.	Model baze podataka . . . . .	4
2.	Kolekcija filmova . . . . .	6
3.	Prijava . . . . .	10
4.	Registracija . . . . .	11
5.	Filmovi CRUD . . . . .	12
6.	Azuriranje filma . . . . .	13
7.	Azuriranje kategorije filma . . . . .	14
8.	Ocjenjivanje filma . . . . .	15
9.	Kategorije CRUD . . . . .	16
10.	Stranica preporuka . . . . .	16



# Popis isječaka koda

1.	Isječak koda iz server.js datoteke . . . . .	5
2.	Shema kolekcije filmova . . . . .	6
3.	Dohvaanje filmova . . . . .	7
4.	Kreiranja novog dokumenta filma . . . . .	7
5.	Brisanje dokumenta . . . . .	8
6.	Ažuriranje dokumenta . . . . .	8