



UNIVERZITET U BIHAĆU  
TEHNIČKI FAKULTET  
ODSJEK: ELEKTROTEHNIKA  
SMJER: RAČUNARSTVO I INFORMATIKA

---

## OBJEKTNO ORJENTISANE BAZE PODATAKA

---

PROJEKTNI ZADATAK  
TEMA: ONLINE FILMOTEKA

**Profesor:** Prof.dr Ramo Šendelj  
**Asistent:** MA Zinaid Kapić

**Student:** Vojić Elma  
**Broj indeksa:** 1178

Bihać, februar 2024. godine

## Sažetak

Online filmoteka predstavlja revolucionarni koncept u digitalnom svijetu koji povezuje ljubitelje filma s neiscrpnom riznicom filmske umjetnosti putem suvremenog internet-skog sučelja. Kroz sistemsko planiranje i implementaciju objektno orijentisane baze podataka, osigurava se organizirano i strukturirano skladištenje filmskog sadržaja, obogaćeno informacijama o redateljima, glumcima i žanrovima. Ovakav pristup omogućuje korisnicima istraživanje svijeta filma stvarajući tako sveobuhvatno iskustvo koje nadmašuje klasične obrasce filmskih naslova.

Uz pomoć Laravel aplikacije, stvara se dinamična, stabilna i prilagodljiva web platforma koja korisnicima omogućuje brz i intuitivan pristup filmovima, kao i interakciju s njihovim sadržajem. Važno je istaknuti da ovaj projekt ne samo da nudi praktično rješenje za organizaciju i pristup filmskoj kolekciji, već predstavlja temelj za daljnji razvoj naprednih funkcionalnosti. Personalizirane preporuke kroz statističke podatke predstavljaju samo početak potencijala koji se može ostvariti kroz kontinuirani razvoj i inovacije.

**Ključne riječi:** Laravel, CRUD, PHP, API, baza podataka, statistika, upiti

## Abstract

The online film library represents a revolutionary concept in the digital world, connecting movie enthusiasts with an inexhaustible treasury of cinematic art through a modern internet interface. Through systematic planning and implementation of object-oriented database systems, it ensures organized and structured storage of film content, enriched with information about directors, actors, and genres. Such an approach enables users to explore the world of cinema, creating a comprehensive experience that surpasses traditional patterns of film consumption.

With the help of the Laravel application, a dynamic, stable, and adaptable web platform is created, allowing users fast and intuitive access to films, as well as interaction with their content. It is important to emphasize that this project not only provides a practical solution for organizing and accessing film collections but also lays the groundwork for further development of advanced functionalities. Personalized recommendations through statistical data, based on user preferences and behavior, represent just the beginning of the potential that can be realized through continuous development and innovation.

**Keywords:** Laravel, CRUD, PHP, API, database, statistics, queries

# Sadržaj

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Uvod</b>                               | <b>1</b>  |
| <b>2</b> | <b>Modeliranje aplikacije</b>             | <b>2</b>  |
| 2.1      | Opis aplikacije . . . . .                 | 3         |
| 2.2      | Dijagrami . . . . .                       | 5         |
| 2.2.1    | ER model . . . . .                        | 5         |
| 2.2.2    | Use Case dijagram . . . . .               | 8         |
| 2.2.3    | Klasni dijagram . . . . .                 | 9         |
| 2.2.4    | Sekvencijalni dijagram . . . . .          | 10        |
| <b>3</b> | <b>Implementacija aplikacije</b>          | <b>12</b> |
| 3.1      | Laravel Framework . . . . .               | 12        |
| 3.1.1    | MVC . . . . .                             | 13        |
| 3.1.2    | ORM . . . . .                             | 14        |
| 3.2      | phpMyAdmin . . . . .                      | 15        |
| 3.3      | XAMPP . . . . .                           | 15        |
| 3.4      | Insomnia . . . . .                        | 16        |
| 3.4.1    | REST API . . . . .                        | 16        |
| <b>4</b> | <b>Analiza rada aplikacije</b>            | <b>17</b> |
| 4.1      | Testiranje slučajeva use case-a . . . . . | 17        |
| 4.2      | Testiranje REST API-a . . . . .           | 31        |
| 4.3      | Testiranje rada baze podataka . . . . .   | 34        |
| <b>5</b> | <b>Zaključak</b>                          | <b>37</b> |

## Popis slika

|          |  |    |
|----------|--|----|
| Slika 1  | Početni izgled aplikacije "Online fimoteka" . . . . .    | 3  |
| Slika 2  | Izgled baze podataka <i>filmoteka</i> . . . . .          | 4  |
| Slika 3  | Izgled tabela sa nazivima kolona . . . . .               | 4  |
| Slika 4  | ER model . . . . .                                       | 6  |
| Slika 5  | Use case dijagram . . . . .                              | 8  |
| Slika 6  | Klasni dijagram . . . . .                                | 9  |
| Slika 7  | Sekvencijalni dijagram . . . . .                         | 11 |
| Slika 8  | Laravel Framework . . . . .                              | 12 |
| Slika 9  | MVC dijagram . . . . .                                   | 13 |
| Slika 10 | Primjeri upotrebe ORM-a . . . . .                        | 14 |
| Slika 11 | Odgovarajuće postavke za pokretanje aplikacije . . . . . | 15 |
| Slika 12 | Kreiran projekat i popis metoda . . . . .                | 16 |
| Slika 13 | Registracija . . . . .                                   | 17 |
| Slika 14 | Prijava . . . . .  | 18 |
| Slika 15 | Početna stranica . . . . .                               | 19 |
| Slika 16 | Filmovi stranica . . . . .                               | 19 |
| Slika 17 | Dodaj novi film . . . . .                                | 20 |
| Slika 18 | Uredi film . . . . .                                     | 21 |
| Slika 19 | Dodaj datoteku . . . . .                                 | 21 |
| Slika 20 | Popis i lokacija dodanih fajlova . . . . .               | 22 |
| Slika 21 | Prikaz datoteke1 . . . . .                               | 22 |
| Slika 22 | Prikaz datoteke2 . . . . .                               | 23 |
| Slika 23 | Prikaz datoteke3 . . . . .                               | 23 |
| Slika 24 | Redatelji stranica . . . . .                             | 24 |
| Slika 25 | Dodaj novog redatelja . . . . .                          | 25 |
| Slika 26 | Uredi redatelja . . . . .                                | 25 |
| Slika 27 | Glumci stranica . . . . .                                | 26 |
| Slika 28 | Dodaj novog glumca . . . . .                             | 27 |
| Slika 29 | Uredi glumca . . . . .                                   | 27 |
| Slika 30 | Žanrovi stranica . . . . .                               | 28 |

|          |   |    |
|----------|---|----|
| Slika 31 | Dodaj novi žanr . . . . .                           | 29 |
| Slika 32 | Uredi žanr . . . . .                                | 29 |
| Slika 33 | Nagrade stranica . . . . .                          | 30 |
| Slika 34 | POST metoda: Dodaj nagradu . . . . .                | 31 |
| Slika 35 | GET metoda: Popis svih nagrada . . . . .            | 32 |
| Slika 36 | PUT metoda: Uredi detalje o nagradi . . . . .       | 32 |
| Slika 37 | GET metoda: Ispis određene nagrade . . . . .        | 33 |
| Slika 38 | DEL metoda: Obriši nagradu . . . . .                | 33 |
| Slika 39 | Prikaz koda upita . . . . .                         | 34 |
| Slika 40 | Prikaz koda upita - nastavak . . . . .              | 35 |
| Slika 41 | Stranica statistika, Rezultati upita: 1 . . . . .   | 35 |
| Slika 42 | Stranica statistika, Rezultati upita: 2-4 . . . . . | 36 |
| Slika 43 | Stranica statistika, Rezultati upita: 5-6 . . . . . | 36 |

## Popis skraćenica

|                 |   |
|-----------------|---|
| <b>CRUD</b>     | Create, Read, Update, Delete                                      |
| <b>PHP</b>      | Hypertext Preprocessor  |
| <b>API</b>      | Application Programming Interface                                 |
| <b>REST API</b> | Representational State Transfer Application Programming Interface |
| <b>GUI</b>      | Graphical User Interface  |
| <b>ER</b>       | Entitetsko-Relacijski model                                       |
| <b>UML</b>      | Unified Modeling Language   |
| <b>XAMPP</b>    | Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P)  |
| <b>MVC</b>      | Model-View-Controller   |
| <b>ORM</b>      | Object-Relational Mapping   |
| <b>DBMS</b>     | Database Management System  |

# 1 Uvod

U svijetu digitalne revolucije, internet je postao neizostavan dio svakodnevnog života, transformirajući način na koji komuniciramo, radimo i zabavljamo se. U ovom kontekstu, online filmoteke predstavljaju jedan od najuzbudljivijih koncepta koji omogućuju pristup ogromnom svijetu filmske umjetnosti putem internetskog sučelja. Koristeći PHP programiranje i popularni Laravel framework, ovaj projekt nastoji izraditi online filmoteku koja će korisnicima pružiti ne samo pristup bogatom sadržaju filmova, već i interaktivno iskustvo istraživanja i uživanja u filmskom svijetu.

PHP je programski jezik koji se često koristi za razvoj web aplikacija zbog svoje jednostavnosti, fleksibilnosti i moćnih alata koje nudi. U kombinaciji s Laravelom, jednim od najomiljenijih PHP razvojnih okvira, stvara se solidna osnova za izgradnju robustnih i skalabilnih web aplikacija. Laravel se ističe po svojoj elegantnosti, jednostavnosti korištenja i bogatstvu funkcionalnosti koje olakšavaju razvoj kompleksnih aplikacija poput online filmoteke.

Osim tehnologije, važan aspekt izgradnje online filmoteke je upravljanje bazom podataka. Objektno-relacijsko mapiranje (ORM) predstavlja ključnu tehniku koja omogućava integraciju objektno orijentiranog programiranja s relacijskim bazama podataka poput MySQL. Kroz pravilno modeliranje i implementaciju baze podataka, osigurava se strukturirano skladištenje informacija o filmovima, glumcima, redateljima i drugim relevantnim entitetima koji čine online filmoteku.

Ovaj rad istražuje proces planiranja, dizajna i implementacije online filmoteke koristeći PHP, Laravel i MySQL. Kroz detaljno razmatranje tehnoloških i konceptualnih aspekata, cilj je stvoriti sveobuhvatan sistem koji će pružiti korisnicima jedinstveno iskustvo istraživanja, otkrivanja i uživanja u svijetu filma putem interneta.

## 2 Modeliranje aplikacije

Modeliranje aplikacije je proces stvaranja apstraktne reprezentacije aplikacije kako bi se razumjelo njeno ponašanje, struktura i interakcija s korisnicima i drugim dijelovima sistema. Ovaj složeni proces uključuje identificiranje entiteta (objekata), njihovih atributa i veza među njima, kao i definisanje operacija ili funkcionalnosti koje aplikacija pruža. Ovo modeliranje omogućuje jasniju sliku o tome kako će aplikacija funkcionisati prije nego što se započne s implementacijom, što olakšava razvojni proces i smanjuje mogućnost pogrešaka. U okviru modeliranja aplikacije primjer "Online filmoteka" pokazuje niz koraka koji su ključni za razvoj i implementaciju sistema.

Prvi korak u modeliranju aplikacije je identifikacija glavnih entiteta koji će činiti strukturu aplikacije. U ovom slučaju entiteti podrazumijevaju: Filmove, Redatelje, Glumce, Žanrove i Nagrade. Nakon identifikacije entiteta, potrebno je definisati attribute za svaki od entiteta. Atributi su ključne karakteristike entiteta koje omogućuju detaljno opisivanje i klasifikaciju podataka. Na primjer, entitet Film ima attribute kao što su naslov, godina izlaska, trajanje i ocjena, dok Redatelj sadrži attribute imena, prezimena, datuma rođenja i države. Također, bitno je naglasiti da se entiteti povezuju putem stranih ključeva koji se dodaju u obliku atributa kod određenog entiteta.

Nakon tačno definisanih atributa, važno je postaviti veze između entiteta. U tekstu "filmoteke", film je povezan s redateljima i žanrovima, glumac sa nagradama, nagrada sa filmovima, a žanrovi sa glumcima. Ove veze omogućuju povezivanje informacija i omogućuju složene upite i operacije nad podacima što je kasnije i prikazano.

Sljedeći korak je definisanje funkcionalnosti aplikacije. To uključuje operacije poput dodavanja novih ili brisanje postojećih filmova, ali i uređivanje filmova po različitim kriterijima. Navedene mogućnosti su prisutne za sve entitete u projektnom zadatku. Također, kod entiteta filmovi omogućeno je i dodavanje i pamćenje datoteka koje se mogu dodati za određeni film. Ove funkcionalnosti omogućuju korisnicima da interaktivno koriste aplikaciju i manipulišu podacima prema svojim potrebama.

Prethodno navedeni koraci odnose se na detaljno planiranje te kreiranje dijagrama i modela. Tek nakon što su jasno definisani svi dijagrami i pogledi pristupa se početku kodiranja odnosno realizaciji projekta putem nekog od alata kao što je *Laravel*.

Pored navedenog, modeliranje korisničkih interakcija igra ključnu ulogu u razvoju aplikacije. To uključuje procese kao što su registracija i prijava korisnika, pregled dostupnih filmova, brisanje te dodavanje novih podataka u bazu. Modeliranje korisničkih interakcija pomaže u razumijevanju načina na koje korisnici koriste aplikaciju i pruža temelj za razvoj GUI-a, odnosno korisničkog sučelja.

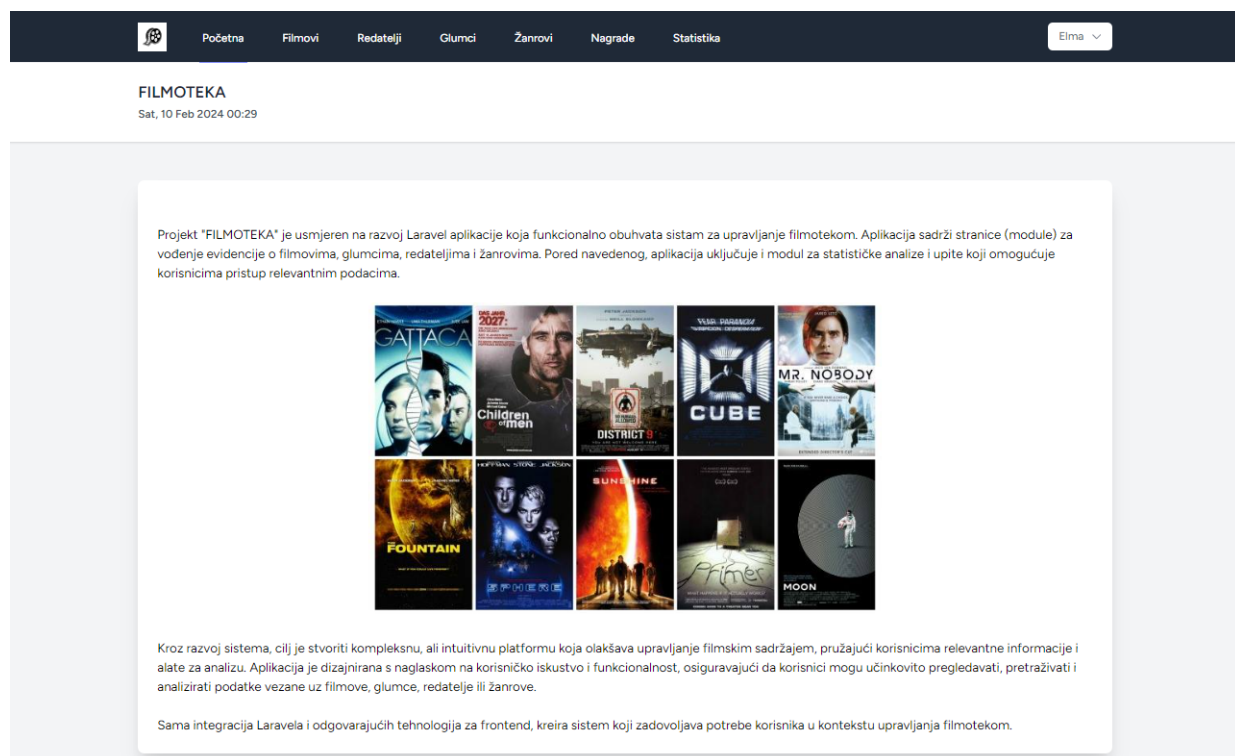
Kroz navedene korake, modeliranje aplikacije osigurava temelj za razvoj i implementaciju sistema "Online filmoteka", omogućujući jasno razumijevanje strukture, funkcionalnosti i interakcije entiteta unutar aplikacije.



## 2.1 Opis aplikacije

Laravel aplikacija "**Online filmoteka**" jeste primjer višestranične (engl. multipage) aplikacije koja se sastoji od nekoliko stranica: Početna, Filmovi, Glumci, Redatelji, Nagrade, Žanrovi i Statistika. Također, kreirane su i stranice za registraciju i prijavu u aplikaciju kao i za uređivanje podataka o profilu. Detaljan prikaz navedenih stranica je dat u poglavlju 4 ovoga rada.

Prilikom otvaranja aplikacije korisniku se nudi mogućnost prijave u Online filmoteku na način da unese svoju email adresu i šifru. Ukoliko on nema kreiran profil za aplikaciju tada je potrebno da se registruje putem forme za registraciju koja zahtijeva unos ispravne email adrese, korisničkog imena te šifre koju je potrebno potvrditi.

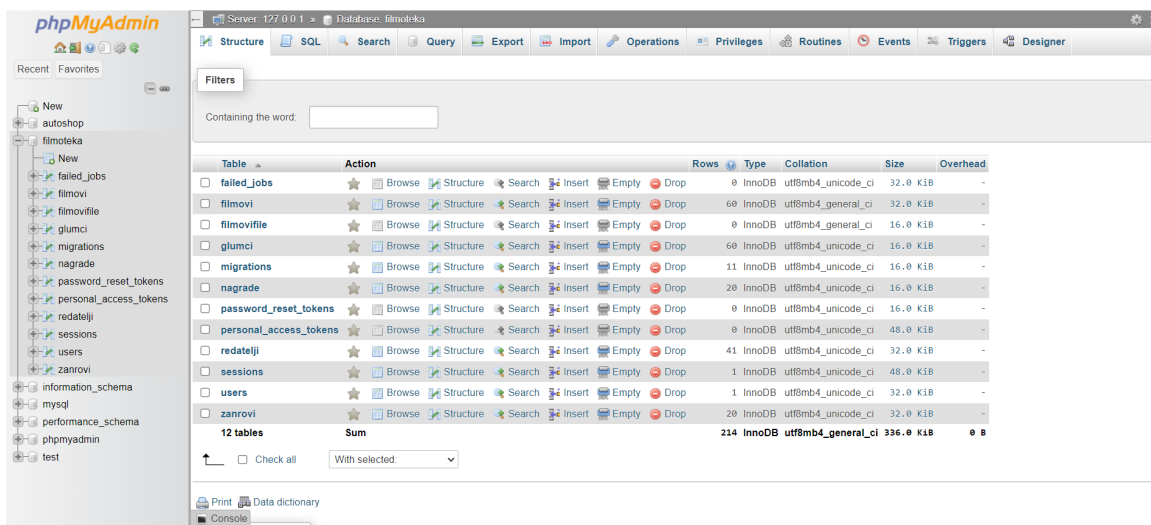


Slika 1: Početni izgled aplikacije "Online fimoteka"

Prijavom na aplikaciju korisnik je u mogućnosti proći kroz prethodno navedene stranice i njene funkcionalnosti. Bitno je naglasiti da početna stranica sadrži neke glavne informacije o ovoj online aplikaciji kao i da korisniku ispisuje trenutni datum i vrijeme. S druge strane stranica Statistika daje popis upita koji su kreirani na osnovu podataka iz baze podataka.

Putem localhost servera `http://localhost/phpmyadmin`, moguće je pristupiti bazi podataka koja je kreirana pod nazivom "filmoteka". Ovom bazom moguće je manipulirati ručno, ali i putem određenih naredbi (npr. `php artisan migrate` koja vrši migraciju svih

promjena napisanih u kodu u bazu. Struktura baze je prikazana na slici ispod.

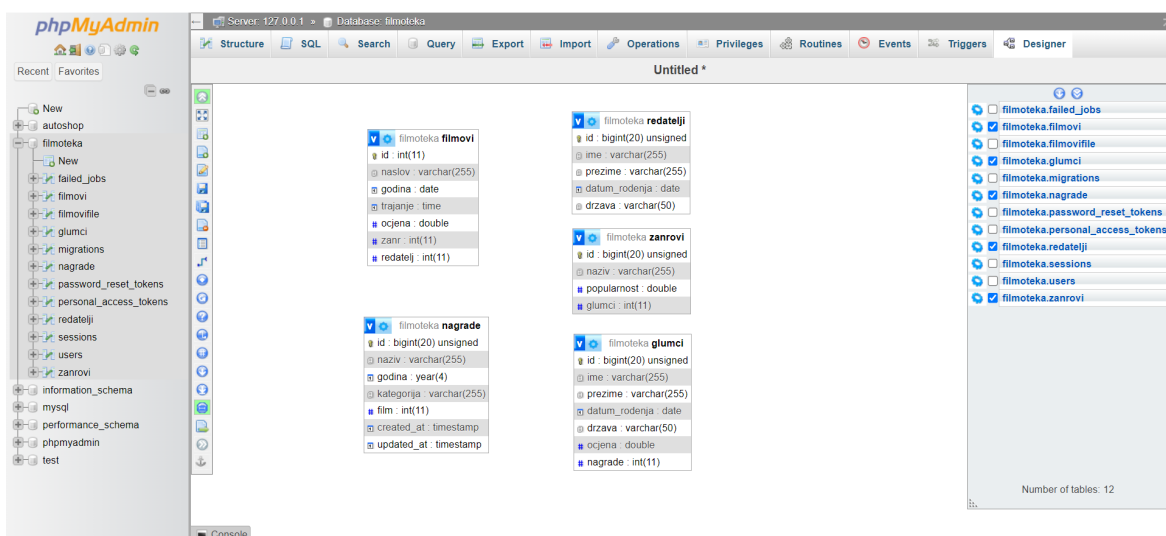


The screenshot shows the phpMyAdmin interface for a database named 'filmoteka'. The 'Structure' tab is selected, displaying a list of 12 tables. Each table entry includes a checkbox, the table name, a star icon, and a set of icons for actions like Browse, Structure, Search, Insert, Empty, and Drop. The tables are: failed\_jobs, filmovi, filmovifile, glumci, migrations, nagrade, password\_reset\_tokens, personal\_access\_tokens, redatelji, sessions, users, and zanrovi. A summary row at the bottom shows 12 tables with a total of 214 rows, using the InnoDB engine and utf8mb4\_general\_ci collation, with a total size of 336.0 KiB and 0 B overhead.

| Table   | Action                                      | Rows | Type   | Collation          | Size      | Overhead |
|---|---|------|--------|--------------------|-----------|----------|
| <input type="checkbox"/> failed_jobs            | ★ Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_ci | 32.0 KiB  | -        |
| <input type="checkbox"/> filmovi                | ★ Browse Structure Search Insert Empty Drop | 60   | InnoDB | utf8mb4_general_ci | 32.0 KiB  | -        |
| <input type="checkbox"/> filmovifile            | ★ Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_general_ci | 16.0 KiB  | -        |
| <input type="checkbox"/> glumci                 | ★ Browse Structure Search Insert Empty Drop | 60   | InnoDB | utf8mb4_unicode_ci | 16.0 KiB  | -        |
| <input type="checkbox"/> migrations             | ★ Browse Structure Search Insert Empty Drop | 11   | InnoDB | utf8mb4_unicode_ci | 16.0 KiB  | -        |
| <input type="checkbox"/> nagrade                | ★ Browse Structure Search Insert Empty Drop | 20   | InnoDB | utf8mb4_unicode_ci | 16.0 KiB  | -        |
| <input type="checkbox"/> password_reset_tokens  | ★ Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_ci | 16.0 KiB  | -        |
| <input type="checkbox"/> personal_access_tokens | ★ Browse Structure Search Insert Empty Drop | 0    | InnoDB | utf8mb4_unicode_ci | 48.0 KiB  | -        |
| <input type="checkbox"/> redatelji              | ★ Browse Structure Search Insert Empty Drop | 41   | InnoDB | utf8mb4_unicode_ci | 32.0 KiB  | -        |
| <input type="checkbox"/> sessions               | ★ Browse Structure Search Insert Empty Drop | 1    | InnoDB | utf8mb4_unicode_ci | 48.0 KiB  | -        |
| <input type="checkbox"/> users                  | ★ Browse Structure Search Insert Empty Drop | 1    | InnoDB | utf8mb4_unicode_ci | 32.0 KiB  | -        |
| <input type="checkbox"/> zanrovi                | ★ Browse Structure Search Insert Empty Drop | 20   | InnoDB | utf8mb4_unicode_ci | 32.0 KiB  | -        |
| 12 tables                                       | Sum   | 214  | InnoDB | utf8mb4_general_ci | 336.0 KiB | 0 B      |

Slika 2: Izgled baze podataka *filmoteka*

Entiteti (tabele) sa atributima (nazivi kolona) su prikazani na slici 3 ovog rada.



The screenshot shows the 'Structure' tab in phpMyAdmin for the 'filmoteka' database, displaying the column definitions for five tables: filmovi, nagrade, redatelji, zanrovi, and glumci. Each table's structure is shown in a separate panel. The columns and their data types are as follows:

- filmoteka.filmovi:** id (int(11)), naslov (varchar(255)), godina (date), trajanje (time), ocjena (double), zanr (int(11)), redatelj (int(11)).
- filmoteka.nagrade:** id (bigint(20) unsigned), naziv (varchar(255)), godina (year(4)), kategorija (varchar(255)), film (int(11)), created\_at (timestamp), updated\_at (timestamp).
- filmoteka.redatelji:** id (bigint(20) unsigned), ime (varchar(255)), prezime (varchar(255)), datum\_rođenja (date), drzava (varchar(50)).
- filmoteka.zanrovi:** id (bigint(20) unsigned), naziv (varchar(255)), popularnost (double), glumci (int(11)).
- filmoteka.glumci:** id (bigint(20) unsigned), ime (varchar(255)), prezime (varchar(255)), datum\_rođenja (date), drzava (varchar(50)), ocjena (double), nagrade (int(11)).

Slika 3: Izgled tabela sa nazivima kolona

Ovi entiteti su povezani putem stranih ključeva što je detaljno opisano kroz kreirane dijagrame za ovu online aplikaciju.

## 2.2 Dijagrami

Razvijanje dijagrama za aplikaciju od velike je važnosti jer dijagrami vizualno predstavljaju arhitekturu, funkcionalnosti i interakcije unutar aplikacije. Određeni dijagrami koji predstavljaju aplikaciju koriste se prije njene stvarne implementacije u određenom programskom jeziku kako bi se taj razvoj pojednostavio. Ukoliko postoje unaprijed jasno definisani dijagrami prilikom kreiranja tabela odnosno editovanju same migracije unutar projekta ne bi se trebali javiti problemi ili nedoumice kako se određeni dio aplikacije zove ili šta on sadrži. Kroz implementaciju dijagrama, dakle, unaprijed se znaju nazivi tabele sa kolonama.

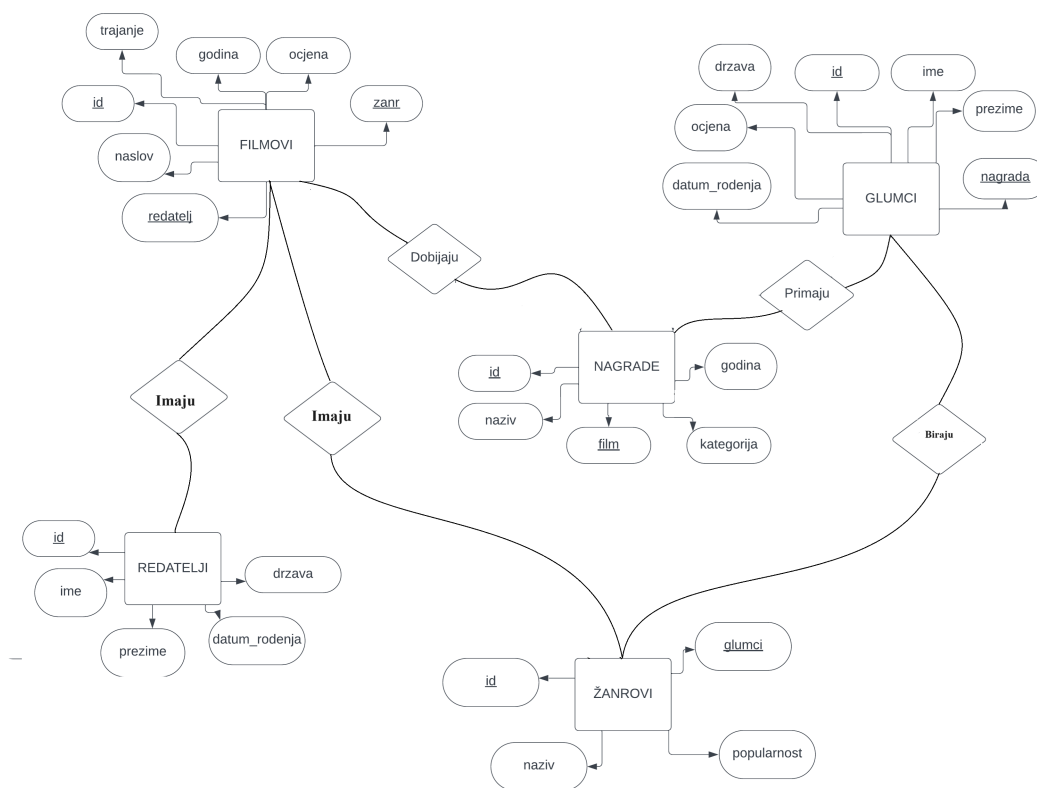
U svrhu analize aplikacije "Online filmoteka" kreirani su i detaljno opisani sljedeći dijagrami:

- 1 ER model
- 2 Use Case dijagram
- 3 Klasni dijagram
- 4 Sekvencijalni dijagram

### 2.2.1 ER model

ER model prikazuje ilustraciju strukture baze podataka kroz entitete, attribute i njihove međusobne veze. Koristi se za vizualizaciju logičke organizacije podataka u bazi podataka. Entiteti predstavljaju klase objekata u bazi, a atributi su karakteristike tih entiteta. Veze između entiteta označavaju kako su entiteti povezani u bazi podataka. Osim što omogućava jasno definisanje strukture podataka, ER model olakšava dizajn baze podataka i pruža temelj za implementaciju relacijskih baza podataka.

Pravougaonicima je označen entitet, a elipsama atribut entiteta na ER dijagramu. Atribut koji je podvučen označava primarni odnosno strani ključ. U nastavku je dat tekstualni opis svakog od navedenih entiteta sa pripadajućim atributima.



Slika 4: ER model

### Atributi entiteta Filmovi:

- 1 id - primarni ključ
- 2 naslov - naslov filma, tip:string
- 3 godina - datum izdavanja filma, tip:date u formatu dd/mm/yyyy
- 4 ocjena - ocjena filma, tip:double
- 5 trajanje - trajanje filma, tip:time u formatu HH:MM:SS
- 6 zanr - ocjena filma, tip:double
- 7 redatelj - redatelj filma, tip:int, strani ključ iz tabele Redatelji

### Atributi entiteta Glumci:

- 1 id - primarni ključ
- 2 ime - ime glumca, tip:string
- 3 prezime - prezime glumca, tip:string
- 4 datum-rođenja - datum rođenja glumca, tip:date u formatu dd/mm/yyyy

- 5 drzava - država porijekla glumca, tip:string
- 6 ocjena - ocjena glumca, tip:double
- 7 nagrada - nagrada koju je dobio glumac, tip:int, strani ključ iz tabele Nagrade

**Atributi entiteta Redatelji:**

- 1 id - primarni ključ
- 2 ime - ime redatelja, tip:string
- 3 prezime - prezime redatelja, tip:string
- 4 datum-rodenja - datum rođenja redatelja, tip:date u formatu dd/mm/yyyy
- 5 drzava - država porijekla redatelja, tip:string

**Atributi entiteta Žanrovi:**

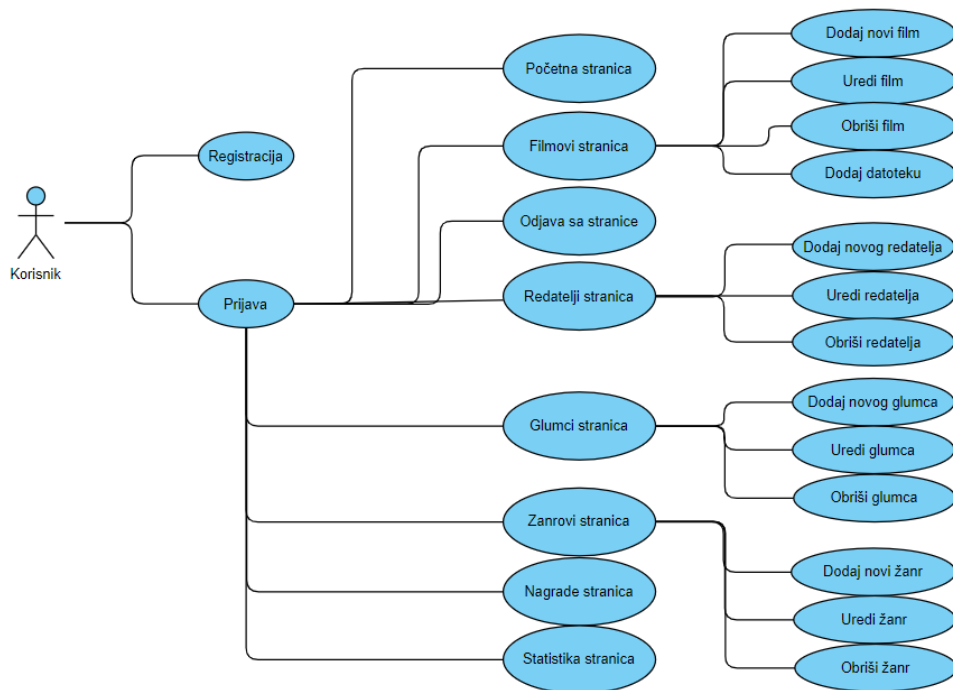
- 1 id - primarni ključ
- 2 naziv - naziv žanra, tip:string
- 3 popularnost - ocjena žanra, tip:double
- 4 glumci - glumci koji uglavnom glume u ovom tipu filma, tip:int, strani ključ iz tabele Glumci

**Atributi entiteta Nagrade:**

- 1 id - primarni ključ
- 2 naziv - naziv nagrade, tip:string
- 3 godina - godina dodjele, tip:year u formatu yyyy
- 4 kategorija - naziv kategorije za koju se nagrada dodjeljuje, tip:string
- 5 film - bilježi se za koje filmove je dodjeljena nagrada, tip:int, strani ključ iz tabele filmovi

### 2.2.2 Use Case diagram

Use Case diagram ili dijagram slučajeva korištenja/upotrebe prikazuje interakcije između korisnika/učesnika/aktera i sistema, a koriste se za definisanje funkcionalnosti sistema iz perspektive korisnika. Učesnici, prikazani kao crtež osobe koja ima linije za ruke, noge i tijelo te krug za glavu, su vanjski entiteti koji interagiraju sa sistemom, dok su use case-ovi, elipse sa nazivom, funkcionalnosti koje sistem nudi. Veze između učesnika i use case-ova pokazuju koje funkcionalnosti su dostupne svakom korisniku. Ovaj dijagram pruža jasnu sliku o tome kako korisnici koriste sistem i koje su im mogućnosti interakcije s njim.



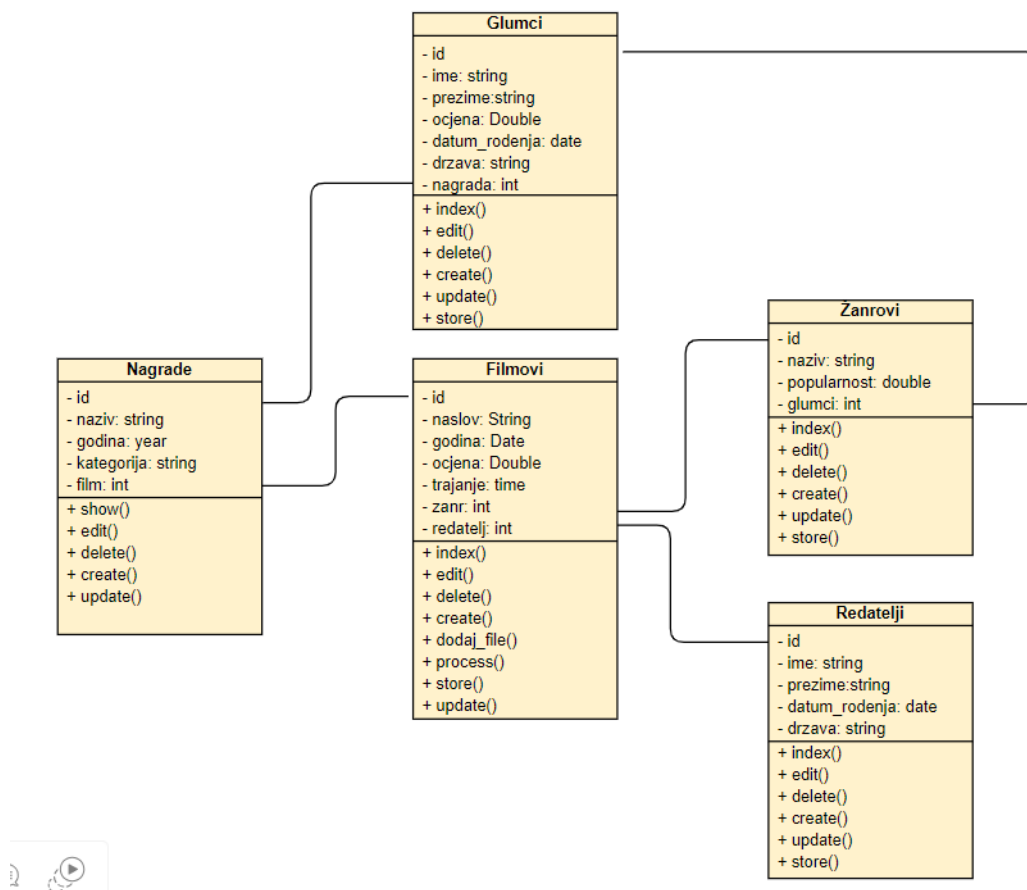
Slika 5: Use case dijagram

Sa ovog dijagrama jasno je uočljivo da sistem posjeduje samo jedan tip korisnika koji prvenstveno ima dvije opcije, da se registruje ili prijavi na aplikaciju. Nakon uspješne registracije te prijave njemu se nude višestruke opcije kao što su prikaz stranica: Početna, Filmovi, Glumci, Redatelji, Nagrade, Žanrovi te Statistika. Četiri (Filmovi, Glumci, Redatelji, Žanrovi) od navedenih sedam stranica imaju dodatne interakcije sa korisnikom kao što su dodavanje, brisanje ili uređivanje postojećeg entiteta u bazi podataka. Stranica Filmovi dodatno posjeduje i opciju za dodavanje fajla u bazu podataka.

Također, korisnik ima mogućnost izmjene svojih ličnih podataka kao i opciju za odjavu sa sistema.

### 2.2.3 Klasni dijagram

Klasni dijagram prikazuje strukturu sistema kroz klase, atribute i metode. Termin klasa se koristi u objektno orijentisanom programiranju za definisanje strukture podataka koja se sastoji od svojstava (atributa) i funkcija (metoda). Koristi se za modeliranje statičkih struktura u sistemu, uključujući objekte, njihove osobine i povezanost. Klase predstavljaju tipove objekata ili entiteta u sistemu, dok su veze između klasa pokazatelji odnosa među njima, poput nasljeđivanja ili asocijacije. Klasni dijagram pruža detaljan uvid u arhitekturu sistema i odnose između njegovih komponenti.



Slika 6: Klasni dijagram

Na dijagramu iznad uočava se 5 klasa koje su međusobno povezane. Naziv klase je dat u prvom redu svake od tablica, a nazivi atributa se nalaze u drugom redu zajedno sa oznakom tipa. Treći red tabele se odnosi na metode koje su ključne prilikom kreiranja kontrolera za određenu klasu.

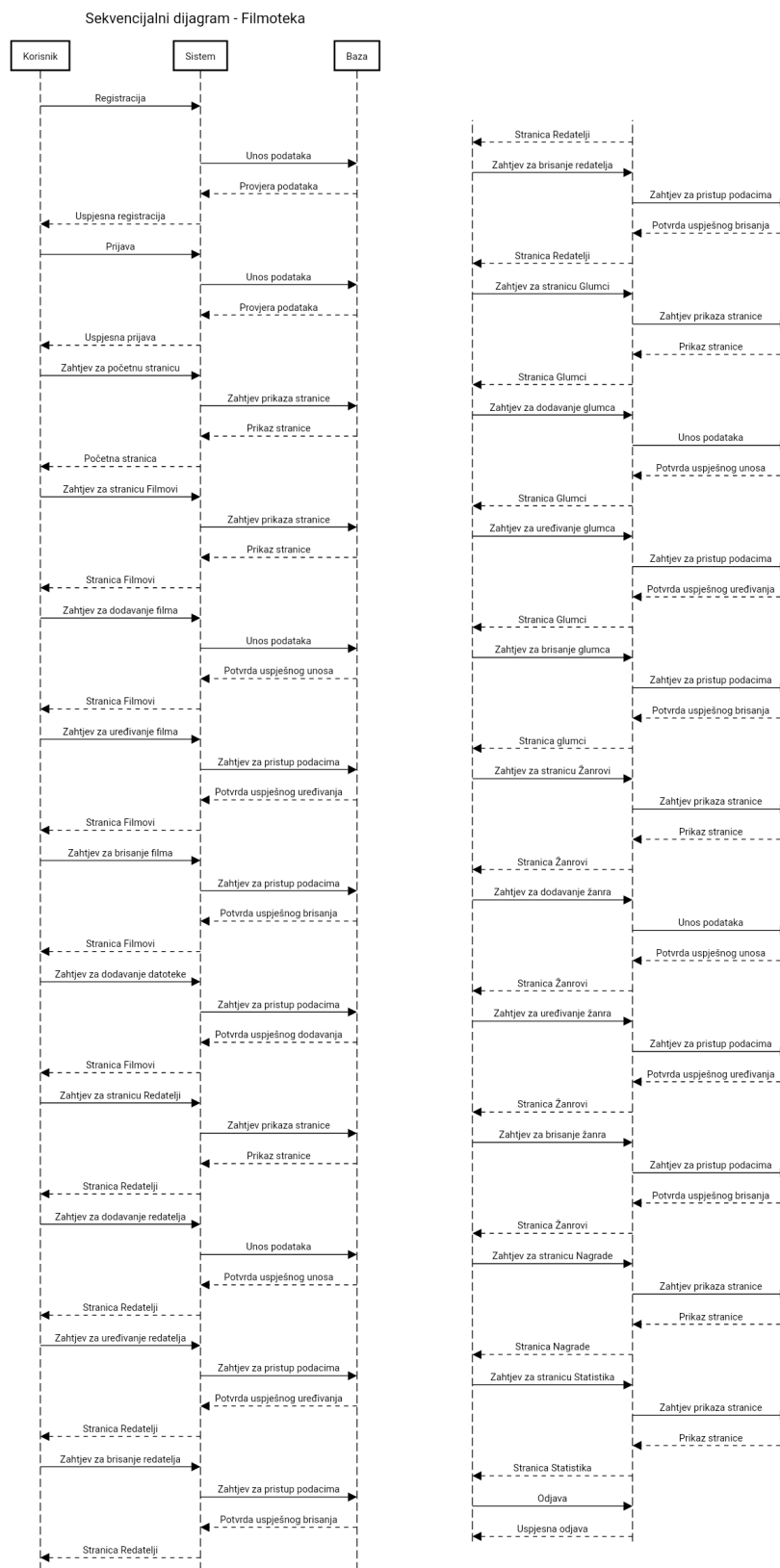
#### 2.2.4 Sekvencijalni dijagram

Sekvencijalni dijagram spada u podgrupu dinamičkih UML dijagrama, a to su interakcijski dijagrami. Sekvencijalni dijagram uvodi vremensku komponentu za opisivanje ponašanja sistema. Koriste se za prikaz rada sistema u realnom vremenu. Sekvencijalni dijagram pokazuje i način na koji objekti surađuju u nekom određenom scenariju, a ta interakcija između objekata prikazuje se putem poruka. Kod ovog dijagrama se definišu: objekti, veze i poruke. Također, sekvencijalni dijagram stavlja naglasak na vrijeme, odnosno na redoslijed gdje se odvija međudjelovanje učesnika u nekom sistemu. Vrijeme na ovom dijagramu uvijek ide od vrha prema dnu. Prema tome korisnici i sistem predstavljaju vremenske pravce. Pored toga, postoji i horizontalna dimenzija koja određuje objekte. Sekvencijalni dijagrami sadrže i poruke koje razmjenjuju sami učesnici unutar sistema. One se razmjenjuju u određenom vremenu između učesnika u sistemu.

Navedeni sekvencijalni dijagram prikazuje interakciju između različitih dijelova sistema ili objekata u vremenskom slijedu. Koristi se za modeliranje ponašanja sistema i prikazuje kako objekti komuniciraju jedni s drugima putem poruka ili poziva metoda tijekom izvođenja nekog procesa ili funkcionalnosti. Sekvencijalni dijagrami pružaju detaljan uvid u dinamiku izvođenja akcija u aplikaciji i koriste se za identifikaciju potencijalnih problema u komunikaciji između komponenti.

Dijagram prikazan na slici ispod je podjeljen na dva dijela radi preglednosti, desni dio se u pravilu nalazi ispod lijevog.





Slika 7: Sekvencijalni dijagram

### 3 Implementacija aplikacije

Implementacija aplikacije predstavlja ključni dio dokumentacije seminarskog rada. Ovaj dio dokumenta detaljno opisuje proces stvaranja i izgradnje online filmoteke, koristeći Laravel aplikaciju i lokalni server putem XAMPP-a.

U ovom dijelu dokumentacije, opisani su koraci poduzeti kako bi se aplikacija razvila i implementirala. To uključuje instalaciju i postavljanje Laravel projekta, konfiguraciju baze podataka putem phpMyAdmin-a, definisanje ruta, stvaranje kontrolera i pogleda, izradu modela, implementaciju sigurnosnih mehanizama, testiranje aplikacije te dokumentaciju koda i procesa.

Implementacija aplikacije uključuje i rješavanje različitih izazova i problema koji su se pojavili tijekom razvoja, kao i prilagodbu aplikacije prema specifičnim zahtjevima i potrebama korisnika. Cilj implementacijskog dijela dokumentacije je pružiti jasan uvid u proces razvoja aplikacije, njene ključne komponente, funkcionalnosti i arhitekturu.

Kombinacija tehnologija korištenih za izradu aplikacije "Online filmoteka" omogućava razvoj, testiranje i implementaciju aplikacije s visokim standardima sigurnosti, performansi i korisničkog iskustva. Svaka od u nastavku navedenih tehnologija ima svoju primjenu i prednosti zašto je korištena upravo ona.

Laravel pruža moćan i fleksibilan okvir za izgradnju web aplikacija, dok phpMyAdmin olakšava upravljanje bazom podataka. XAMPP omogućava lokalno hostiranje aplikacije radi bržeg razvoja i testiranja, dok Insomnia olakšava provjeru funkcionalnosti REST API-ja. Za pokretanje aplikacije i editovanje njene strukture korišten je Visual Studio Code.

#### 3.1 Laravel Framework

Laravel je popularan PHP web framework koji pruža elegantan i intuitivan način za izradu web aplikacija.



Slika 8: Laravel Framework

Ovaj framework omogućava brz i efikasan razvoj uz brojne ugrađene mogućnosti poput rutiranja, autentikacije, upravljanja sesijama, te integracije s bazom podataka. Laravel dolazi s bogatim skupom ugrađenih značajki koje olakšavaju razvoj sofisticiranih web aplikacija. Neki od ključnih značajki uključuju rutiranje, autentikaciju,

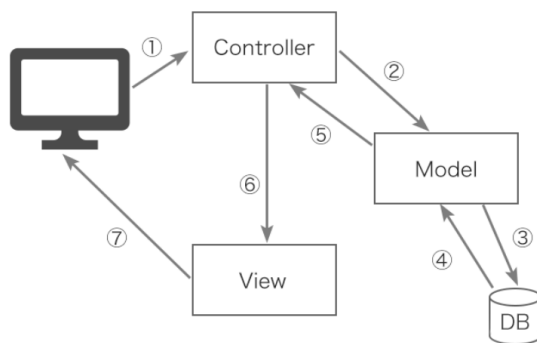
autorizaciju, upravljanje sesijama, ORM (Object-Relational Mapping) za rad s bazom podataka, te ugrađenu podršku za rad s RESTful API-ima. Osim toga, Laravel omogućava jednostavnu integraciju s različitim alatima i bibliotekama putem Composer-a, što pruža programerima fleksibilnost i mogućnost proširenja funkcionalnosti aplikacije prema njihovim potrebama.

Također, Laravel pruža programerima čist i čitljiv kod zahvaljujući svojoj elegantnoj sintaksi, a njegov intuitivni MVC (Model-View-Controller) arhitekturni obrazac pomaže u organizaciji koda. Sa svojom ekspresivnom i lako razumljivom sintaksom, Laravel omogućava programerima brz i efikasan razvoj aplikacija, bez obzira na njihovu razinu iskustva.

Još jedna prednost upotrebe ovog frameworka je što Laravel dolazi s detaljnom i sveobuhvatnom dokumentacijom koja pruža jasne upute i primjere za korištenje svih njegovih značajki, što olakšava razvoj aplikacija i rješavanje problema.

### 3.1.1 MVC

MVC ili Model-View-Controller je obrazac koji se često koristi u razvoju softvera, posebno u web aplikacijama. Prednosti MVC arhitekturnog obrasca uključuju jasno razgraničenje odgovornosti između komponenata, što olakšava razvoj, testiranje i održavanje aplikacija. Također, omogućava višestruku upotrebu koda, poboljšava čitljivost koda i omogućava paralelno razvijanje različitih dijelova aplikacije.



Slika 9: MVC dijagram

U kontekstu web razvoja, MVC se često koristi jer omogućava odvajanje logike poslovnih podataka od prezentacijskog sloja, što rezultira boljom organizacijom koda i lakšim upravljanjem složenim aplikacijama. Njegove glavne tri komponente su:

M Model je komponenta koja predstavlja logiku poslovne logike i podatke aplikacije. Ovdje se obično nalazi logika za pristupanje i validaciju podataka i poslovne operacije. Model direktno komunicira s bazom podataka ili drugim izvorima podataka kako bi dohvaćao ili ažurirao informacije.

Npr. `app->Models->Filmovi.php`

V View(pogled) je komponenta koja je odgovorna za prikazivanje korisničkog sučelja korisniku. Ovdje se obično nalazi HTML, CSS i JavaScript kôd koji korisnik vidi i s kojim interagira. View prima informacije od kontrolera (controller) i prikazuje ih korisniku na odgovarajući način.

Npr. *resources->views->filmovi->index.blade.php*

C Controller je komponenta koja upravlja komunikacijom između modela i pogleda. Kontroleri obrađuju zahtjeve korisnika, izvršavaju odgovarajuće akcije, dohvaćaju ili ažuriraju podatke iz modela i zatim prenose te informacije na odgovarajući pogled. Kontroleri su "posrednici" između modela i pogleda.

Npr. *app->Http->Controllers->FilmoviController.php*

Kreiraje modela i kontrolera u Laravel aplikacijama (za Filmove na primjer) se vrši upisivanjem u terminal sljedeće naredbe: *php artisan make:model -mcr Filmovi*. Ovu naredbu potrebno je upotrijebiti za svaku od tabela u bazi podataka.

### 3.1.2 ORM

ORM ili Object-Relational Mapping je tehnika programiranja koja omogućava mapiranje podataka iz relacijskih baza podataka u objekte u programskom jeziku. Ova tehnika omogućava programerima da koriste objektno orijentisane koncepte i pristupe prilikom rada s podacima, što olakšava razvoj aplikacija i integraciju s bazama podataka. Umjesto korištenja složenih SQL upita za komunikaciju s bazom podataka, programeri mogu koristiti ORM biblioteke ili frameworke koji pružaju jednostavne API-je za izvršavanje operacija poput upita, dodavanja, ažuriranja i brisanja podataka.

```
app > Http > Controllers > NgradeController.php
9  class NgradeController extends Controller
10 {
11     public function index()
12     {
13         return Ngrade::all();
14     }
15
16     public function store(Request $request)
17     {
18         return Ngrade::create($request->all());
19     }
20
21     public function show(string $id)
22     {
23         return Ngrade::find($id);
24     }
25
26     public function update(Request $request, $id)
27     {
28         $ngrade = Ngrade::find($id);
29         $ngrade->update($request->all());
30
31         return $ngrade;
32     }
33
34     public function destroy($id)
35     {
36         return Ngrade::destroy($id);
37     }
38 }
39
```

Slika 10: Primjeri upotrebe ORM-a

Jedna od glavnih prednosti ORM-a je smanjenje potrebe za pisanjem SQL upita, što rezultira čistim i čitljivijim kodom. ORM omogućava programerima da se fokusiraju na logiku aplikacije umjesto na detalje komunikacije s bazom podataka. Također, ORM pruža veću razinu apstrakcije, što olakšava migraciju između različitih sistema upravljanja bazama podataka (DBMS) bez potrebe za promjenom koda aplikacije.

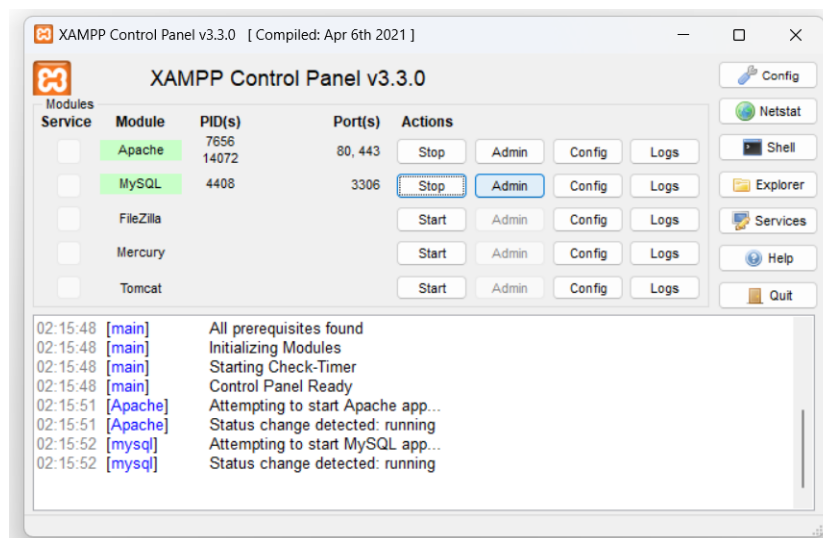
Uz to, ORM omogućava lakše testiranje aplikacija jer programeri mogu koristiti mock objekte ili lažne baze podataka tijekom testiranja, što povećava pouzdanost i kvalitetu softvera. Međutim, važno je napomenuti da ORM nije uvijek najbolje rješenje za sve situacije, posebno u slučajevima kada je potrebna velika brzina ili složeni SQL upiti.

### 3.2 phpMyAdmin

phpMyAdmin je alat za upravljanje MySQL bazama podataka putem web sučelja. Omogućava administriranje baza podataka, uključujući kreiranje, uređivanje i brisanje tablica, upravljanje korisnicima i privilegijama te izvoz i uvoz podataka. Primjer je prikazan na slici 2 ovog rada.

### 3.3 XAMPP

XAMPP je popularan paket koji sadrži Apache web server, MySQL bazu podataka, PHP interpreter i druge komponente potrebne za izradu web aplikacija. Omogućava lokalno hostiranje web aplikacija na računaru što olakšava razvoj, testiranje i implementaciju aplikacije.

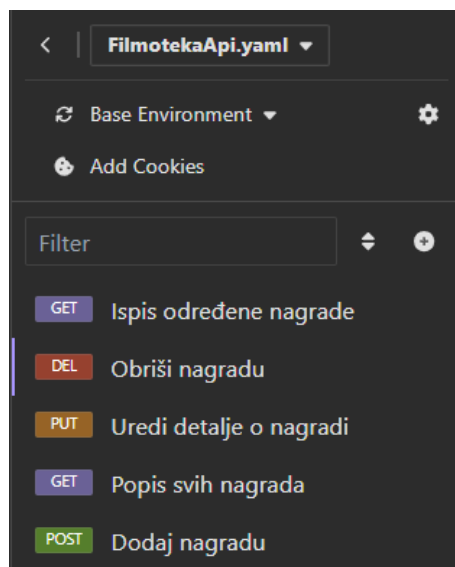


Slika 11: Odgovarajuće postavke za pokretanje aplikacije

Nakon pokretanja servera na iznad pokazat način, pokreće se aplikacija kroz terminal naredbom *php artisan serve*

### 3.4 Insomnia

Insomnia je alat za testiranje REST API-ja koji omogućava slanje HTTP zahtjeva na API te pregled i analizu odgovora. Koristan je za provjeru ispravnosti i funkcionalnosti API-ja te olakšava otklanjanje greški i testiranje aplikacije.



Slika 12: Kreiran projekat i popis metoda

#### 3.4.1 REST API

REST API ili Representational State Transfer Application Programming Interface predstavlja stil koji se često koristi za razvoj web usluga. Osnovna načela REST-a promoviraju skalabilnost, fleksibilnost i jednostavnost dizajna, čime omogućuju efikasnu komunikaciju između različitih softverskih sistema putem HTTP protokola. Ključni koncepti u REST arhitekturi uključuju resurse, HTTP metode i predstavljanje resursa. Resursi su ključni entiteti u REST arhitekturi, a svaki od njih ima jedinstveni URL koji ga identificira.

Standardne HTTP metode poput GET, POST, PUT i DELETE koriste se za izvođenje operacija nad resursima, omogućujući dohvat, stvaranje, ažuriranje i brisanje podataka. Predstavljanje resursa često se vrši u formatima poput JSON-a ili XML-a, pružajući jednostavnu i čitljivu strukturu podataka.

REST API se široko koristi u web razvoju zbog svoje jednostavnosti i fleksibilnosti. Omogućuje programerima da izgrade skalabilne web usluge koje lako integriraju različite aplikacije i platforme. REST API-ji su ključni u modernom web razvoju jer omogućuju razmjenu podataka na jednostavan i standardiziran način, čime se potiče razvoj kompleksnih aplikacija i poboljšava korisničko iskustvo.

## 4 Analiza rada aplikacije

Analiza rada stranice je zahtjevan proces koji od analizatora traži da provjeri i najsitniji detalj o aplikaciji koja se nalazi pred njim. Ovaj tip analize podrazumijeva testiranje aplikacije, a samo testiranje se odvija kroz tri dijela.

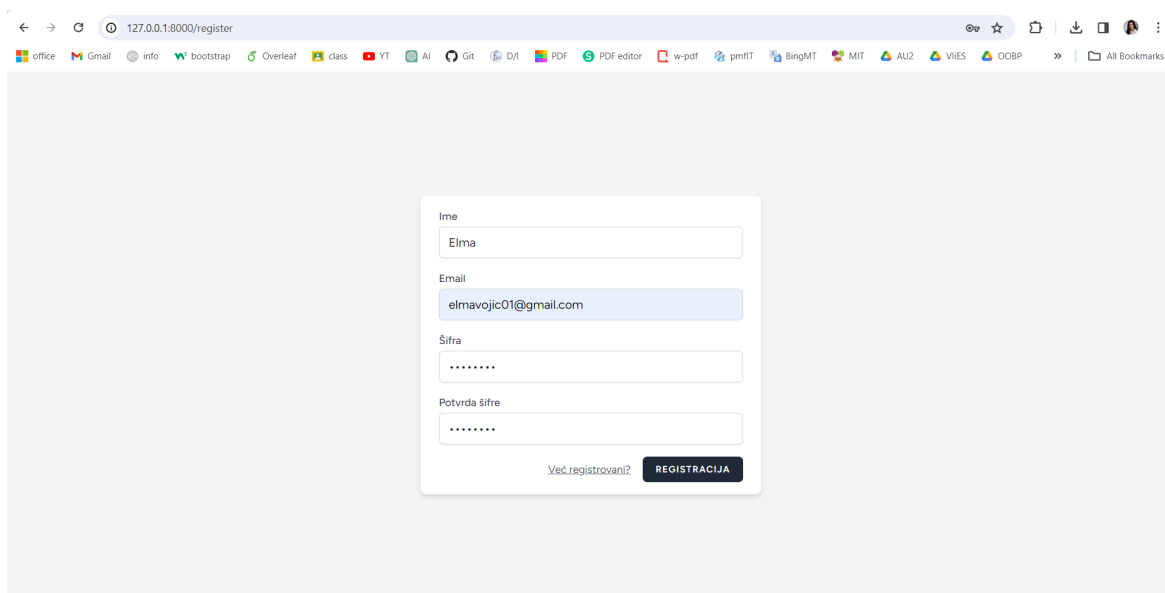
Prvi dio je analiza svakog od slučajeva use case dijagrama. Dakle, potrebno je testirati svaki od navedenih slučajeva. Jednostavnije objašnjeno, potrebno je provjeriti funkcionalnost svake od mogućnosti koju nudi aplikacija "Online filmoteka". Drugi dio jeste testiranje REST API-a putem Insomnie. Zadnji, ali nimalo manje važan dio analize jeste, testiranje rada baze podataka. Ovo se postiže kreiranjem složenijih upita.

### 4.1 Testiranje slučajeva use case-a

Testiranje slučajeva korištenja podrazumijeva testiranje rada svakog od dijelova use case dijagrama koji su prikazani na slici 5 ovoga rada uz naglasak na činjenici da sistem ima *samo jedan tip korisnika*.

#### Use case: Registracija

Slika ispod prikazuje izgled stranice za registraciju. Korisnik treba unijeti svoje lične podatke u formi koja je navedena te kliknuti na button "REGISTRACIJA". Nakon unesenih podataka, isti se provjeravaju kroz sistem (validacija unesenih podataka) te se ukoliko su svi ispravno navedeni korisniku ispunjava zahtjev, a ukoliko je došlo do unosa nekog pogrešnog podatka korisnik dobija informaciju o greški.



The image shows a web browser window with the address bar displaying "127.0.0.1:8000/register". The browser's bookmark bar is visible with various icons. The main content area displays a registration form with the following fields and values:

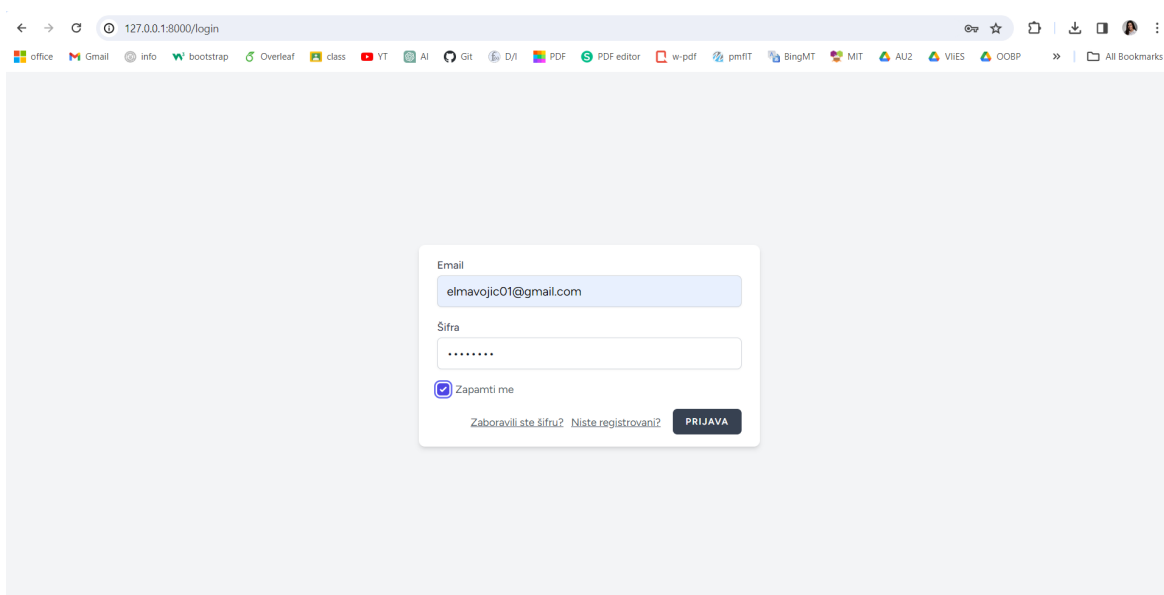
- Ime:** Elma
- Email:** elmavojic01@gmail.com
- Šifra:** [masked with dots]
- Potvrda šifre:** [masked with dots]

Below the form fields, there is a link "Već registrovani?" and a dark button labeled "REGISTRACIJA".

Slika 13: Registracija

Također, uočljivo je da se korisniku nudi opcija da klikne na link "Već registrovani?" koji ga vodi na stranicu "**Prijava**".

**Use case: Prijava** Ova stranica predstavlja novi slučaj korištenja aplikacije "online filmoteka", a sastoji se od forme za unos ličnih podataka, email adresa i šifre, s kojima se korisnik registrovao. Prilikom prijave, ako je unio pogrešne podatke korisniku će se javiti da postoji određena greška u njegovom zahtjevu. Međutim, ukoliko korisnik ima problem sa prijavom, zaboravio je svoju šifru, tada on može odabrati link "Zaboravili ste šifru?" koji će otvoriti stranicu koja će mu pomoći u rješavanju ovog problema. Također, korisniku se nudi opcija da se prebaci na stranicu za registraciju putem linka "Registracija" ili da odabere opciju da njegovi podaci za prijavu budu sačuvani (opcija "Zapamti me").



Slika 14: Prijava

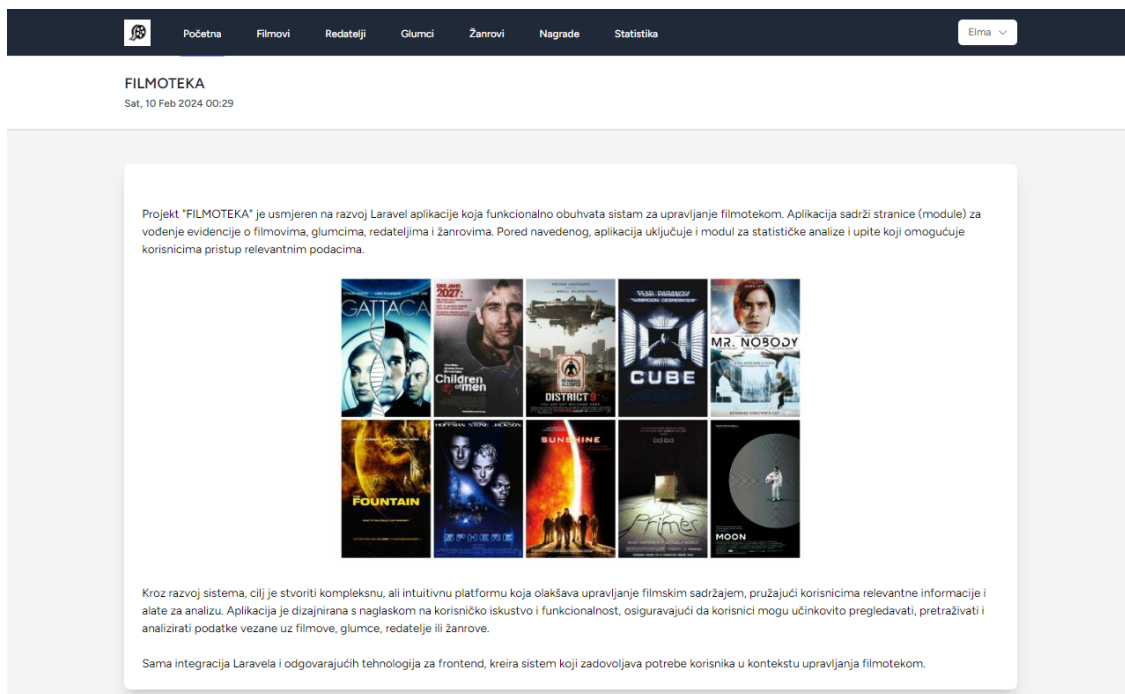
Preduslov da se ovaj use case ispravno izvrši je taj da korisnik ima kreiran profil. Sljedeći korak nakon popunjavanja forme za prijavu je kliknuti na button "PRIJAVA". Ispravno izvršen ovaj korak će rezultirati prikazom početne stranice projekta koja je spremljena pod nazivom "**Početna**".

#### **Use case: Stranica Početna**

Slika 15 koja je prikazana u nastavku ovog rada je prikaz početne stranice. Ona sadrži navigacijsku traku sa linkovima svih stranica koji su sastavni dio projekta, tijelo stranice sa opisom i slikom filmoteke. Još jedan njen značajan segment jeste taj što prikazuje trenutni datum i vrijeme.

Preduslov za ovaj use case je da je korisnik prijavljen na sistem. Također, ovaj uslov vrijedi za sve use case-ove koji slijede.

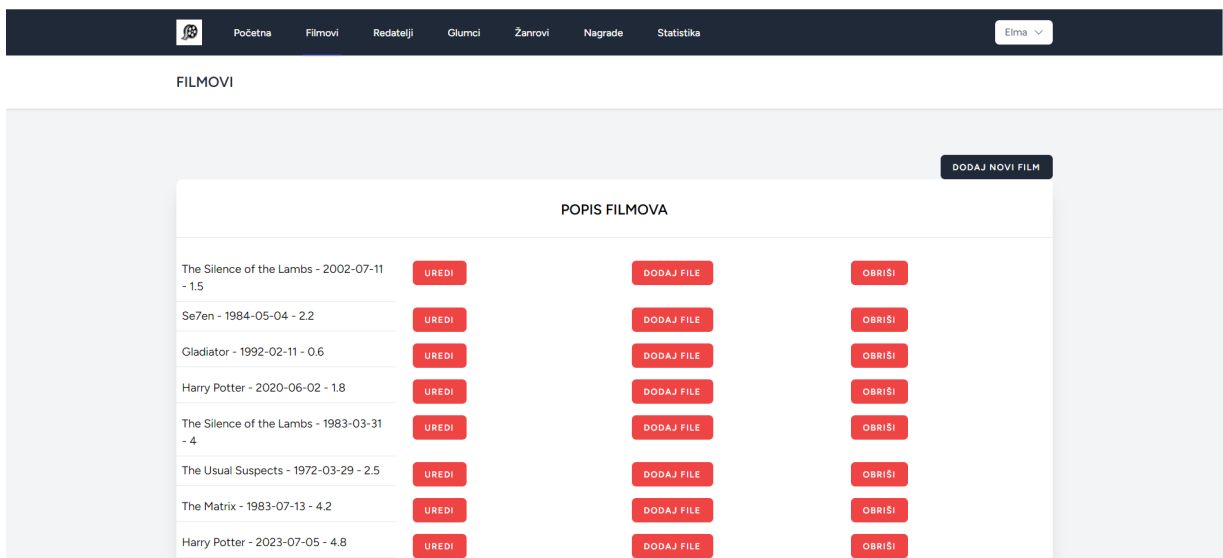




Slika 15: Početna stranica

### Use case: Stranica Filmovi

Stranica filmovi sadrži popis svih filmova koji su spremljeni u bazi podataka. Također nudi korisniku mogućnost dodavanja, uređivanja, brisanja, ali i dodavanja datoteke za određeni film. Svako od navedenih opcija se pristupa putem button-a sa nazivom. Slika ispod prikazuje ovaj slučaj korištenja.

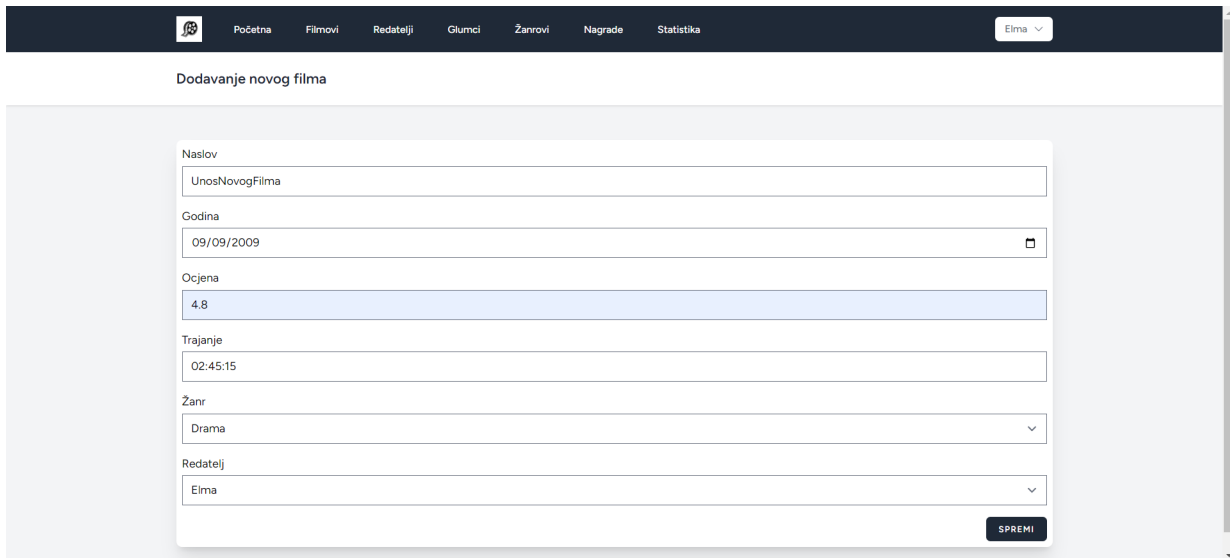


Slika 16: Filmovi stranica

Sljedeća 4 use case-a prikazuju dodatne opcije stranice filmovi.

### Use case: Dodaj novi film

Ovaj use case prikazuje stranicu sa formom za dodavanje novog filma. Forma sadrži input polja sa nazivima atributa koje ima svaki Film. Kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Polja za unos žanra i redatelja su select option tipa. Novi film se dodaje klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica filmovi.



The screenshot shows a web application interface for adding a new film. At the top, there is a dark navigation bar with a logo and links: Početna, Filmovi, Redatelji, Glumci, Žanrovi, Nagrade, and Statistika. A user profile dropdown labeled 'Elma' is on the right. Below the navigation bar, the page title is 'Dodavanje novog filma'. The main content area contains a form with the following fields: 'Naslov' (text input with placeholder 'UnosNovogFilma'), 'Godina' (date input with value '09/09/2009' and a calendar icon), 'Ocjena' (text input with value '4.8'), 'Trajanje' (text input with value '02:45:15'), 'Žanr' (select dropdown with 'Drama' selected), and 'Redatelj' (select dropdown with 'Elma' selected). A 'SPREMI' button is located at the bottom right of the form.

Slika 17: Dodaj novi film

### Use case: Obriši film

Ovaj use case se odnosi na mogućnost brisanja određenog filma iz baze podataka. Izvršenje ovog slučaja je moguće samo ako u bazi postoji određeni red. Brisanje filma se vrši klikom na button "OBRIŠI" koji se nalazi u istom redu kao i button za uređivanje.

### Use case: Uredi film

Ovaj use case prikazuje stranicu sa formom za uređivanje postojećeg filma, a otvara se preko buttona "UREDITI" koji se nalazi pored ispisanih podataka o filmu koji se želi urediti. Forma sadrži input polja sa nazivima atributa koje ima svaki Film. Kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Uređeni podaci o filmu se spremaju, a promjene se izvršavaju klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica filmovi.

Uređivanje podataka o filmu

Naslov  
UnosNovogFilma

Godina  
09/09/2009

Ocjena  
4.8

Trajanje  
02:45:15

Žanr  
Naučna fantastika

Redatelj  
Elma

SPREMI

Slika 18: Uredi film

### Use case: Dodaj datoteku

Ovaj use case se odnosi na mogućnost dodavanje datoteke za određeni film iz baze podataka. Izvršenje ovog slučaja započinje klikom na button "DODAJ DATOTEKU" koji se nalazi u istom redu kao i buttoni za brisanje i uređivanje podataka o filmu. Klikom na button otvara se sljedeća stranica na kojoj se vrši dodavanje datoteke odabirom opcije "Choose file".

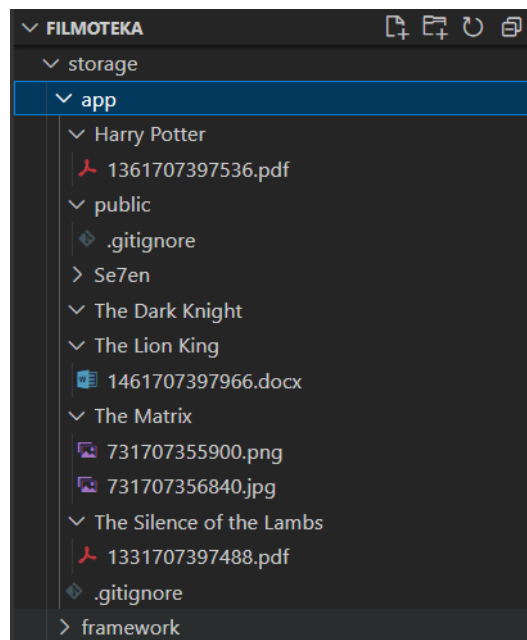
Dodavanje datoteke za film

Choose File Naslovna.jpg

DODAJ

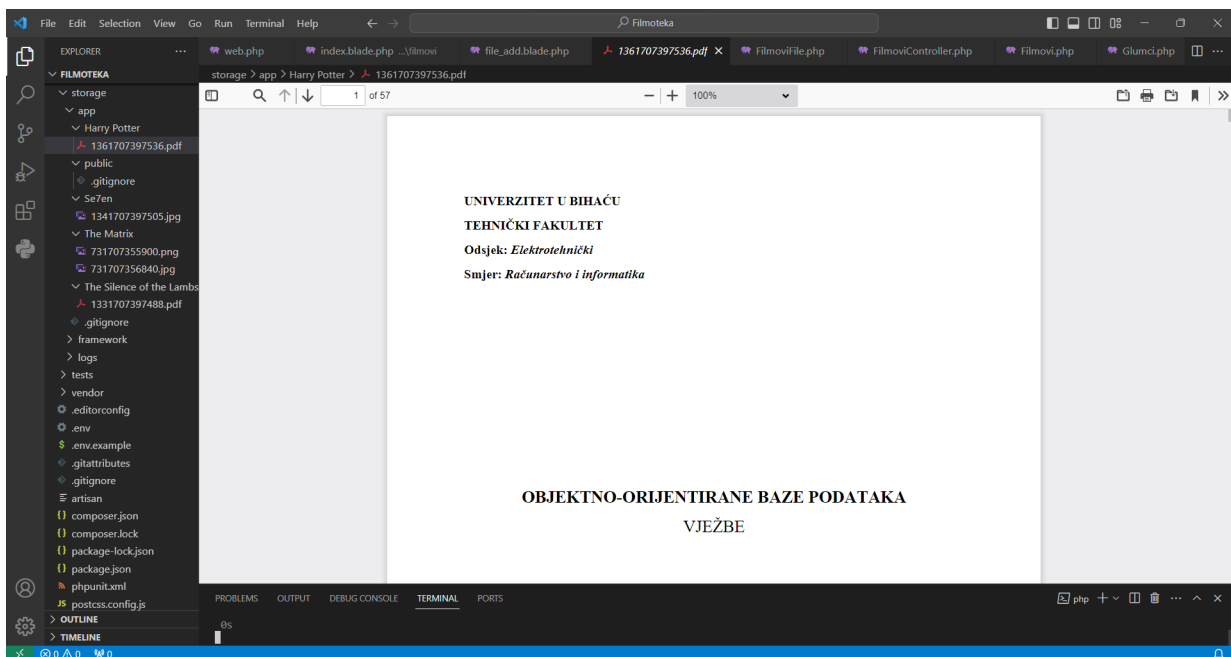
Slika 19: Dodaj datoteku

Nakon što je za nekoliko filmova iz baze dodana datoteka, struktura aplikacije "Online filmoteka" se izmijenila i sada izgleda kao na slici ispod. Uočljivo je da je za 5 filmova u bazi dodan jedan ili više fajlova različitog tipa (.pdf, .docx, .jpg, .png).

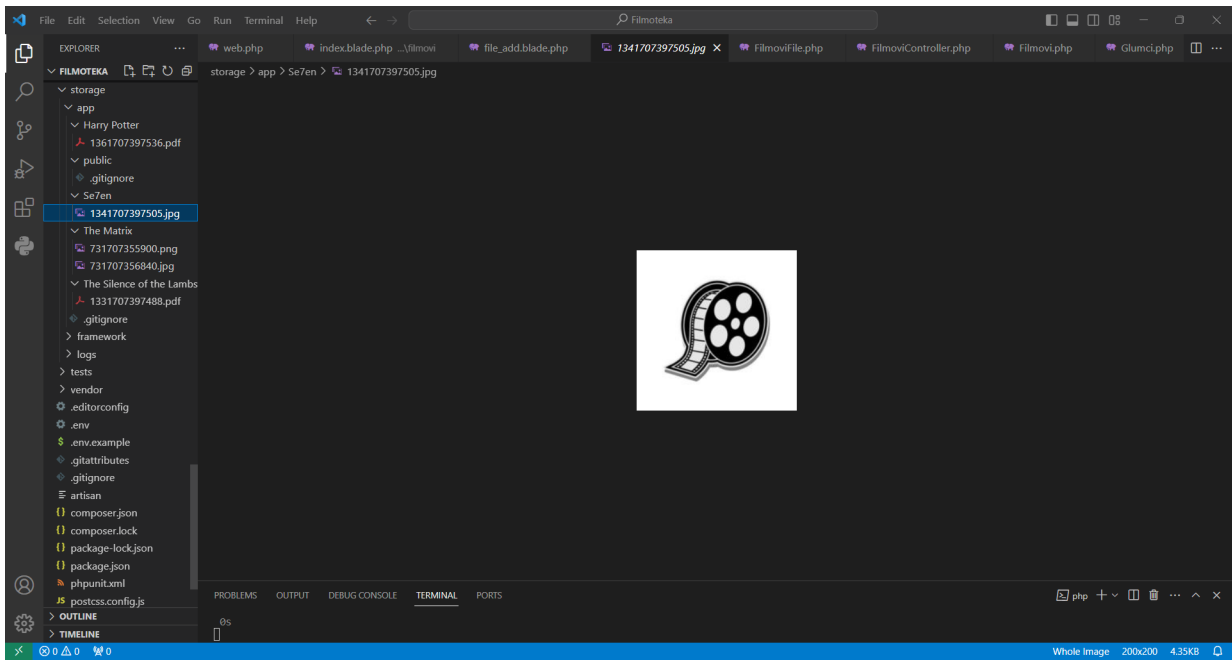


Slika 20: Popis i lokacija dodanih fajlova

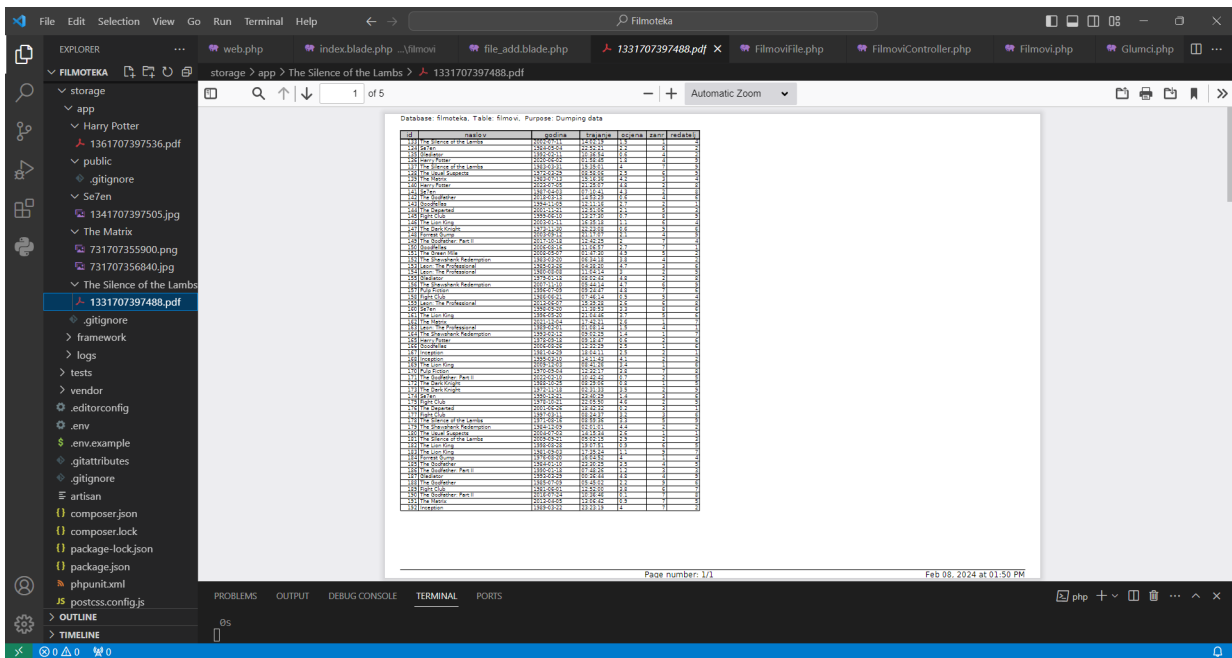
Primjeri kako izgledaju prethodno navedene datoteke su dati na slikama 21, 22 i 23 ovog rada.



Slika 21: Prikaz datoteke1



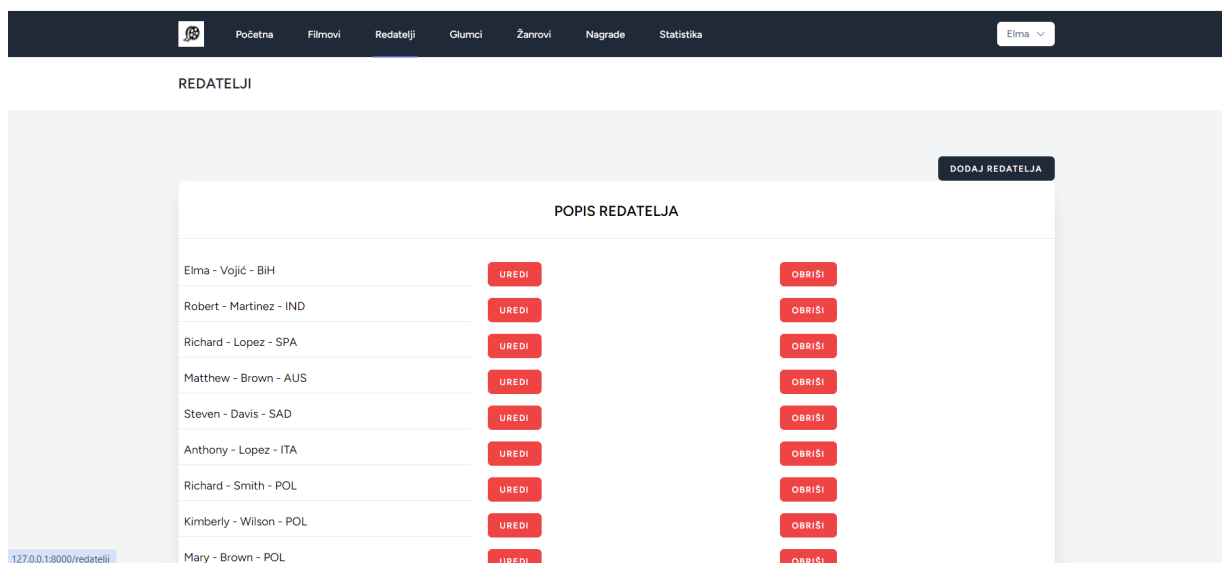
Slika 22: Prikaz datoteke2



Slika 23: Prikaz datoteke3

### Use case: Stranica Redatelji

Stranica redatelji sadrži popis svih redatelja koji su spremljeni u bazi podataka. Također nudi korisniku mogućnost dodavanja, uređivanja i brisanja određenog redatelja. Svako od navedenih opcija se pristupa putem button-a sa nazivom. Slika ispod prikazuje ovaj slučaj korištenja.



Slika 24: Redatelji stranica

Sljedeća 3 use case-a prikazuju dodatne opcije stranice redatelji.

### Use case: Obriši redatelja

Ovaj use case se odnosi na mogućnost brisanja određenog redatelja iz baze podataka. Izvršenje ovog slučaja je moguće samo ako u bazi postoji barem jedan popunjen red. Brisanje se vrši klikom na button "OBRIŠI" na stranici redatelji koji se nalazi u istom redu sa podacima o redatelju i button za uređivanje.

### Use case: Dodaj novog redatelja

Ovaj use case se otvara klikom na button "DODAJ REDATELJA" na stranici redatelji i prikazuje stranicu sa formom za dodavanje novog redatelja. Forma sadrži input polja sa nazivima atributa koje ima svaki redatelj. Kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Novi redatelj se dodaje klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica redatelji.

Ime redatelja

NoviRedatelj

Prezime redatelja

PrezimeRedatelj

Datum rođenja

05/05/1985

Država porijekla

RUS

SPREMI

Slika 25: Dodaj novog redatelja

### Use case: Uredi redatelja

Ovaj use case prikazuje stranicu sa formom za uređivanje postojećeg redatelja, a otvara se preko buttona "UREDJI" koji se nalazi pored ispisanih podataka o redatelju koji se želi urediti. Forma sadrži input polja sa nazivima atributa koje ima svaki redatelj, a kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Uređeni podaci o redatelju se spremaju, a promjene se izvršavaju klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica redatelji.

Ime redatelja

Jessica

Prezime redatelja

Brown-Smith

Datum rođenja

03/22/1992

Država porijekla

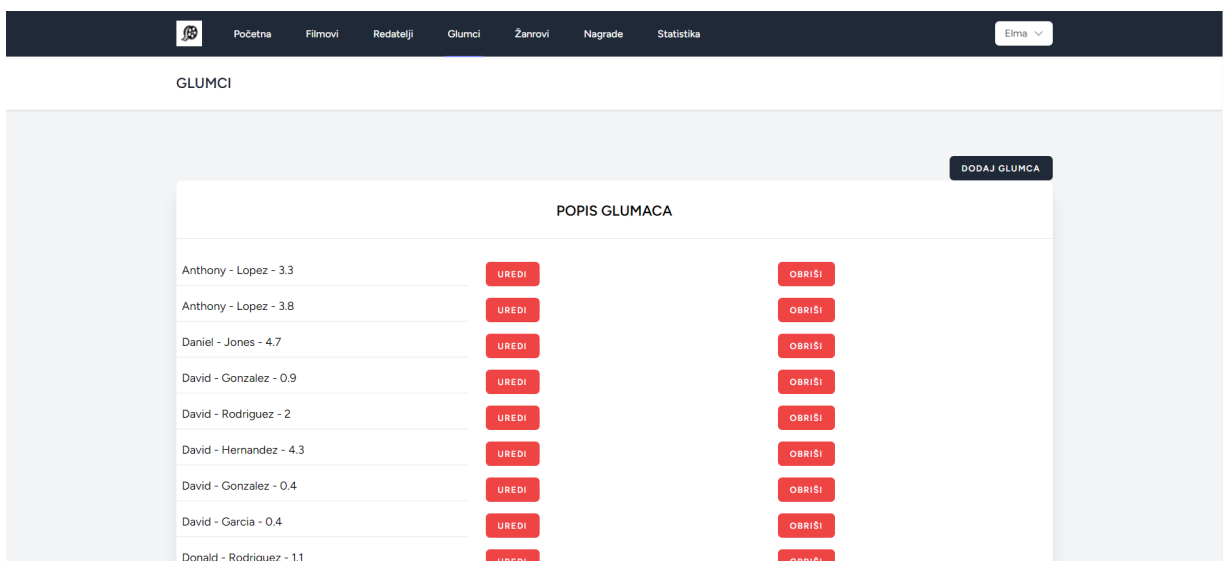
UKR

SPREMI

Slika 26: Uredi redatelja

### Use case: Stranica Glumci

Stranica glumci sadrži popis svih glumaca koji su spremljeni u bazi podataka. Također nudi korisniku mogućnost dodavanja, uređivanja i brisanja određenog glumca. Svakoj od navedenih opcija se pristupa putem button-a sa nazivom. Slika ispod prikazuje ovaj slučaj korištenja.



Slika 27: Glumci stranica

Sljedeća 3 use case-a prikazuju dodatne opcije stranice glumci.

### Use case: Obriši glumca

Ovaj use case se odnosi na mogućnost brisanja određenog glumca iz baze podataka. Izvršenje ovog slučaja je moguće samo ako u bazi postoji barem jedan popunjen red. Brisanje se vrši klikom na button "OBRIŠ" koji se nalazi u istom redu sa podacima o glumcu i button za uređivanje na stranici glumci.

### Use case: Dodaj novog glumca

Ovaj use case se otvara klikom na button "DODAJ GLUMCA" na stranici glumci i prikazuje stranicu sa formom za dodavanje novog glumca. Forma sadrži input polja sa nazivima atributa koje ima svaki glumac, a koji su navedeni u ER dijagramu na slici 3 ovog rada. Kako bi ovaj use case bio uspješno izvršen potrebno je da se sva polja popune sa odgovarajućim podacima. Novi glumac se u bazu podataka dodaje klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica glumci.



Početna Filmovi Redatelji Glumci Žanrovi Nagrade Statistika Elma

Dodavanje novog glumca

Ime glumca  
GlumacIme

Prezime glumca  
GlumacPrezime

Datum rođenja  
10/10/2010

Država porijekla  
HRV

Ocjena  
4.8

Nagrade  
Srebrni medvjed

SPREMI

Slika 28: Dodaj novog glumca

### Use case: Uredi glumca

Ovaj use case prikazuje stranicu sa formom za uređivanje postojećeg glumca u bazi podataka, a otvara se preko buttona "UREDJI" koji se nalazi pored ispisanih podataka o glumcu koji se želi urediti. Forma sadrži input polja sa nazivima atributa koje ima svaki glumac, a kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Uređeni podaci o glumcu se spremaju preko postojećih, a promjene se izvršavaju klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica glumci.

Početna Filmovi Redatelji Glumci Žanrovi Nagrade Statistika Elma

Uređivanje podataka o glumcima

Ime glumca  
David

Prezime glumca  
Smith-Gonzales

Datum rođenja  
08/24/2018

Država porijekla  
BRA

Ocjena  
2.4

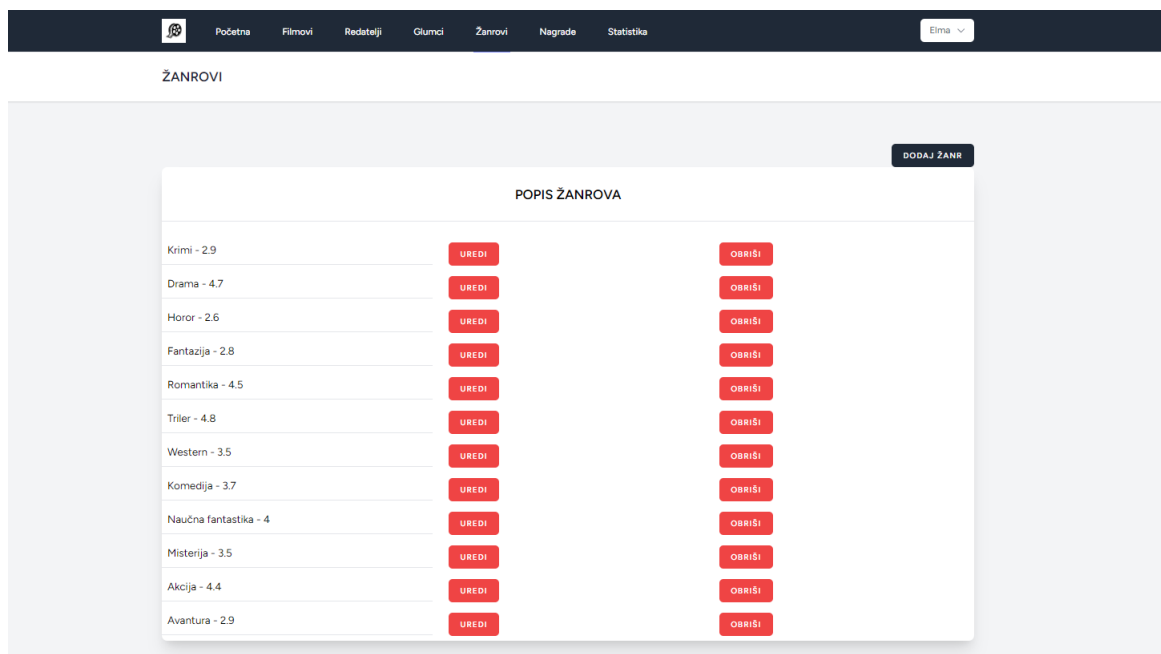
Nagrade  
Cannes Film Festival

SPREMI

Slika 29: Uredi glumca

### Use case: Stranica Žanrovi

Stranica žanrovi sadrži popis svih žanrova koji su spremljeni u bazi podataka. Također nudi korisniku mogućnost dodavanja, uređivanja i brisanja određenog žanra filma. Svako od navedenih opcija se pristupa putem button-a sa nazivom. Slika ispod prikazuje ovaj slučaj korištenja.



Slika 30: Žanrovi stranica

Sljedeća 3 use case-a prikazuju dodatne opcije stranice žanrovi.

### Use case: Obriši žanr

Ovaj use case se odnosi na mogućnost brisanja određenog žanra iz baze podataka, a pristupa mu se preko stranice žanrovi. Izvršenje ovog slučaja je moguće samo ako u bazi postoji barem jedan popunjen red. Brisanje se vrši klikom na button "OBRIŠI" koji se nalazi u istom redu sa podacima o žanru i button za uređivanje.

### Use case: Dodaj novi žanr

Ovaj use case se otvara klikom na button "DODAJ ŽANR" na stranici žanrovi i prikazuje stranicu sa formom za dodavanje novog žanra. Forma sadrži input polja sa nazivima atributa koje ima svaki redatelj. Kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Novi žanr se dodaje klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica žanrovi.

Početna Filmovi Redatelji Glumci Žanrovi Nagrade Statistika Elma ▾

Dodavanje novog žanra

Naziv žanra  
NoviZanr

Popularnost žanra  
5

Glumac  
Matthew ▾

SPREMI

Slika 31: Dodaj novi žanr

### Use case: Uredi žanr

Ovaj use case prikazuje stranicu sa formom za uređivanje postojećeg žanra u bazi podataka, a otvara se preko buttona "UREDITI" koji se nalazi pored ispisanih podataka o žanru koji se želi urediti. Forma sadrži input polja sa nazivima atributa koje ima svaki žanr, a kako bi ovaj use case bio uspješno izvršen potrebno je da se sva input polja popune sa odgovarajućim podacima. Uređeni podaci o žanru se prepisuju preko postojećih, a promjene se izvršavaju klikom na button "SPREMI", nakon čega se korisniku ponovo prikazuje stranica žanrovi.

Početna Filmovi Redatelji Glumci Žanrovi Nagrade Statistika Elma ▾

Uređivanje podataka o žanrovima

Naziv žanra  
NoviZanr

Ocjena popularnosti  
3.9

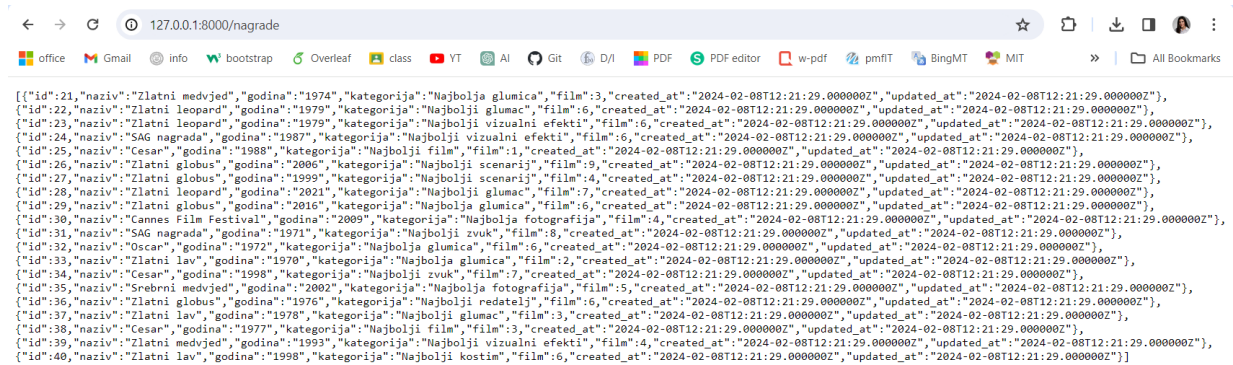
Glumac  
Matthew ▾

SPREMI

Slika 32: Uredi žanr

## Use case: Stranica Nagrade

Stranica nagrade sadrži popis svih nagrada koji su spremjeni u bazi podataka. Ova stranica se razlikuje od rpothodno navedneih stranica obzirom da je razvijena samo u kontekstu REST API-a. Ona dakle nema svoju HTML stranicu iste strukture kao na primjer stranica Filmovi.



Slika 33: Nagrade stranica

## Use case: Stranica Statistika

Stranica statistika je detaljno prikazana u dijelu koji opisuje testiranje rada baze podataka. Ona predstavlja stranicu iste strukture kao što je početna stranica i prikazuje rezultate određenih upita.

## Use case: Odjava sa stranice

Odjava sa stranice se vrši klikom na padajući meni u desnoj strani navigacijske trake te odabirom opcije "Log Out".

Use case testiranje je obuhvatilo sve prikaze stranica sa mogućnostima koje su kreirane za projekat objektno-orjentisane baze podataka pod nazivom "Online filmoteka".

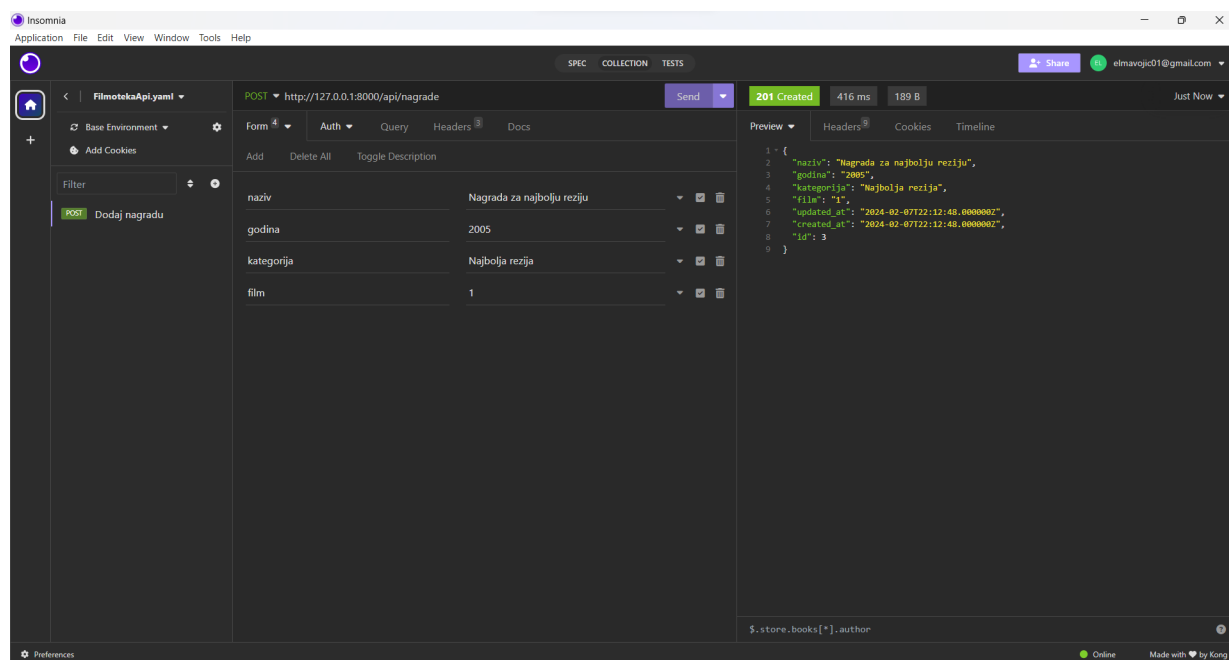
## 4.2 Testiranje REST API-a

Za testiranje API-a korištena je Insomnia. Kroz Insomniju, programeri mogu temeljito testirati funkcionalnosti, identificirati i rješavati probleme te osigurati da API pruža očekivano ponašanje. Ovaj alat olakšava razvoj, debugiranje i dokumentaciju API-a, što je ključno za održavanje visoke kvalitete i pouzdanosti softverskih aplikacija.

Također, kreiran je i ORM model u kontroleru *NagradaController.php* koji olakšava rad sa Insomniom, odnosno, koji olakšava testiranje REST API-a.

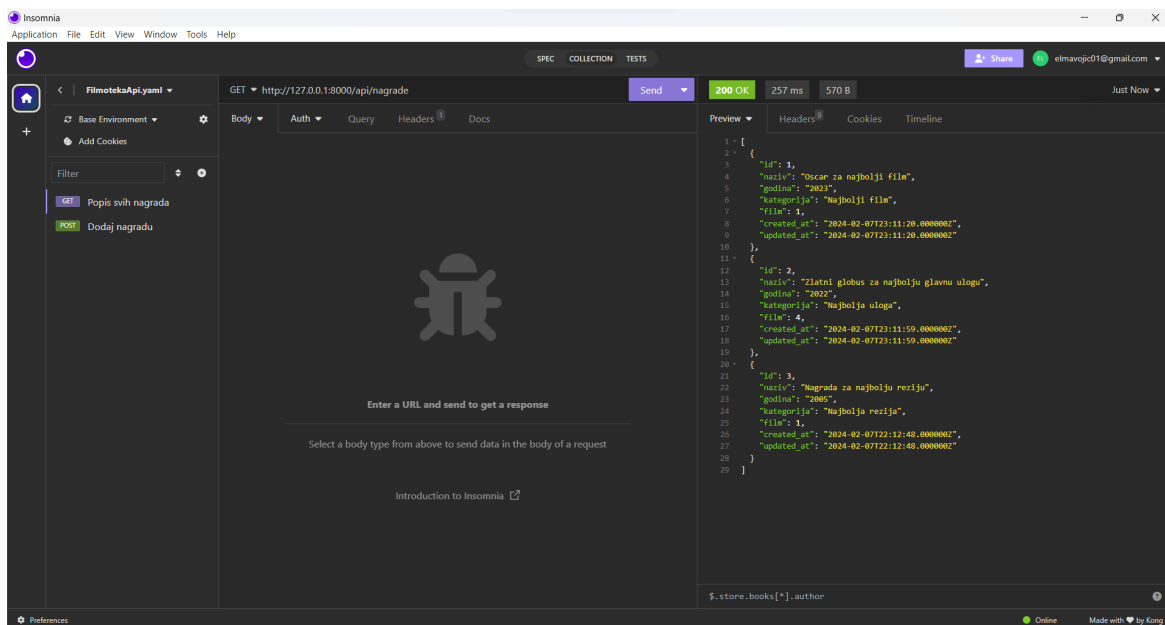
Na početku je kreiran novi projekat te **FilmotekaApi.yaml** datoteka u kojoj je izvršeno kreiranje metoda GET, POST, PUT i DEL koje su neophodne za testiranje. Slika 12 ovog rada prikazuje popis naziva svih metoda koje su kreirane za "Online filmoteku". Sve što je potrebno za izvršenje ovih metoda je dodavanje tačnog linka odnosno URL-a sa pokrenute aplikacije.

Prva kreirana metoda je **POST metoda za dodavanje nagrada**. Za kreirane atribute je potrebno popuniti podatke za svaku Nagradu. Primarni ključ (id) je automatski popunjen kao i vrijeme kreiranja i update određene nagrade, dok su ostali podaci ručno uneseni u bazu podataka putem ovog alata. Na slici ispod je s desne strane prikazan rezultat ove metode.



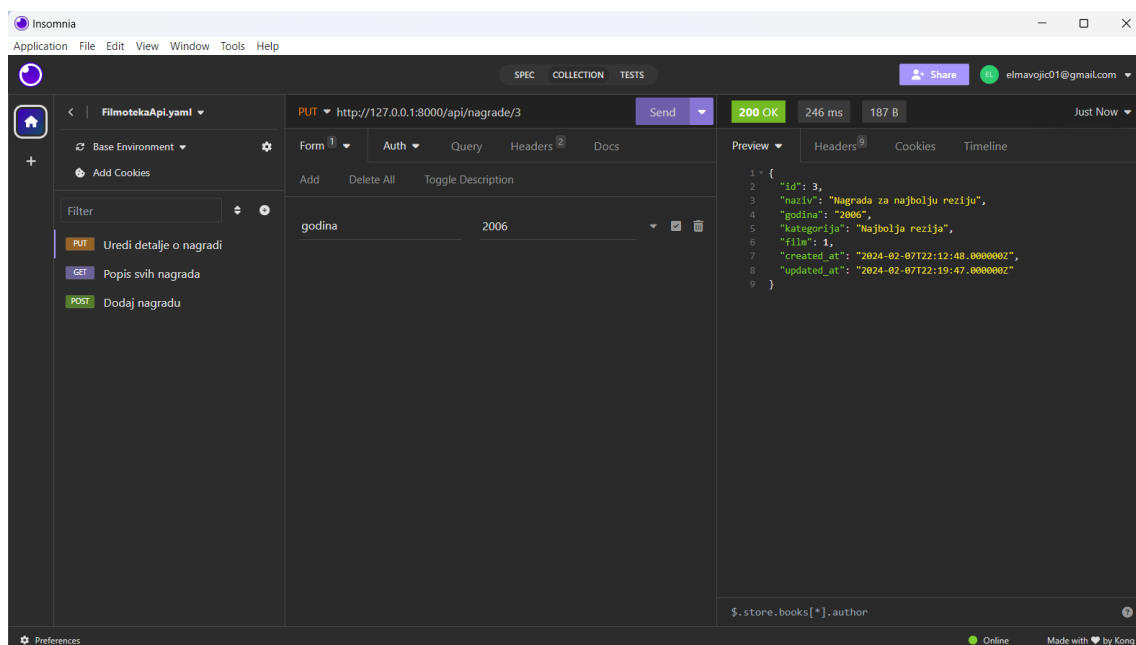
Slika 34: POST metoda: Dodaj nagradu

Sljedeća kreirana metoda je **GET metoda koja ispisuje sve nagrade** iz baze podataka zajedno sa nazivima atributa.



Slika 35: GET metoda: Popis svih nagrada

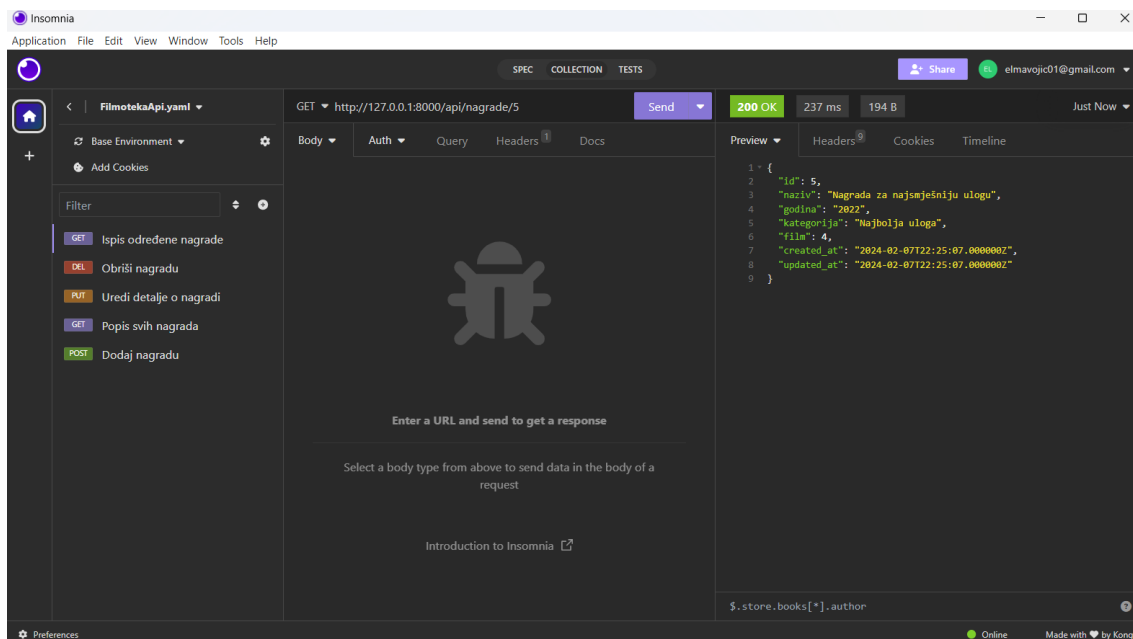
**PUT metoda za uređivanje nagrade** je također kreirana kako bi se mogli mijenjati pojedini podaci za određenu nagradu iz baze podataka.



Slika 36: PUT metoda: Uredi detalje o nagradi

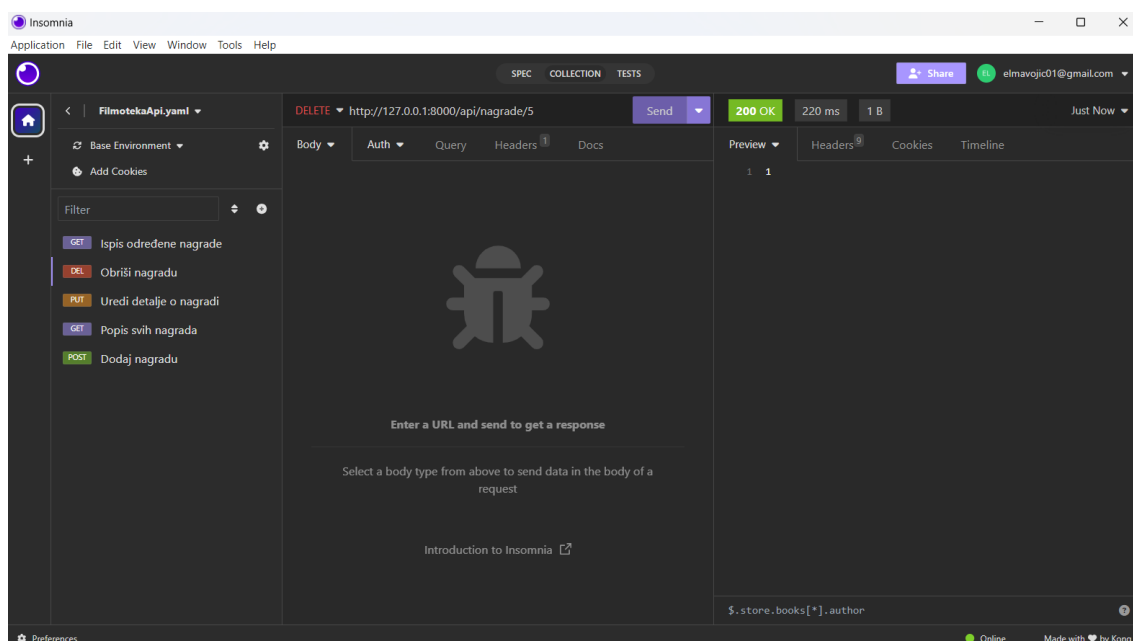
Također, kreirana je još jedna **GET metoda koja ispisuje samo određenu nagradu**. URL koji je potreban za ovaj ispis je isti kao URL za popis svih nagrada s tim

da se na kraju dodaje još i id nagrade za koju se ispisuju podaci, na primjer na kraj se doda samo /5.



Slika 37: GET metoda: Ispis određene nagrade

Slično kao kod ispisa određene nagrade vrši se i brisanje putem **DEL** metode za brisanje nagrade.



Slika 38: DEL metoda: Obriši nagradu

### 4.3 Testiranje rada baze podataka

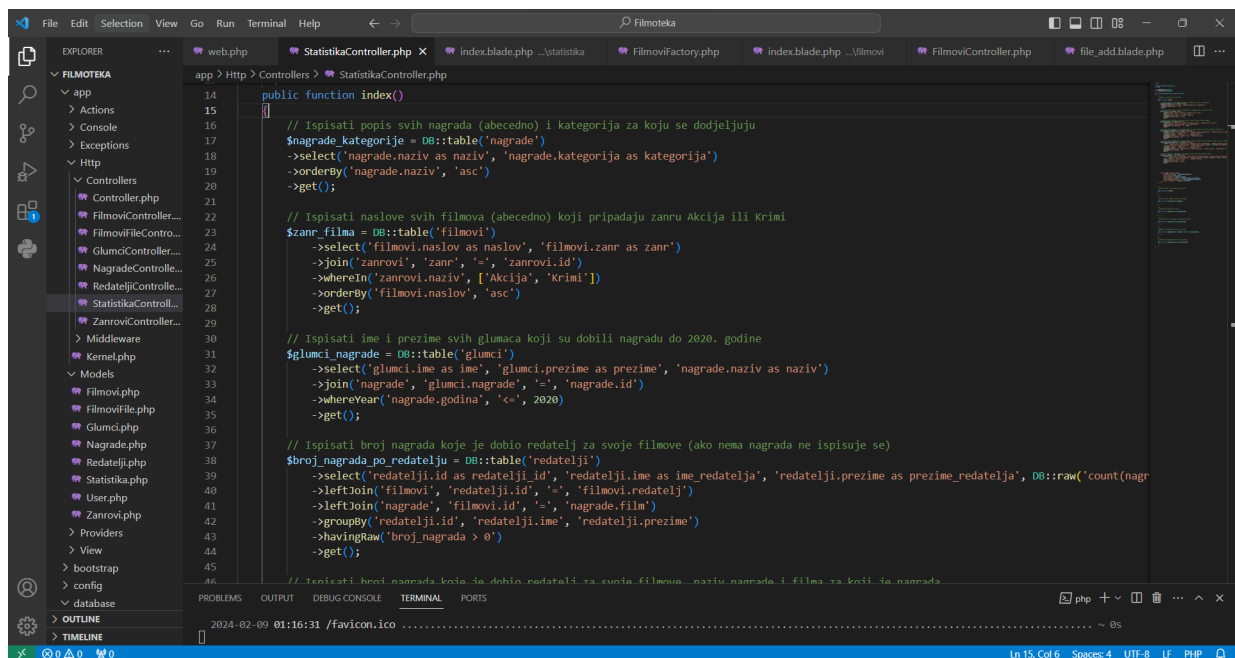
Treći dio analize aplikacije uključuje testiranje rada baze podataka. U svrhu kreiranja ovog projekta kreirana je baza podataka pod nazivom *filmoteka* koja ima strukturu prethodno prikazanu u radu na slici 2.

Testiranje baze podataka putem izvođenja upita ključno je za osiguranje ispravnosti i učinkovitosti njenog funkcioniranja. Ovaj pristup omogućava programerima i inženjerima baze podataka evaluaciju kako se sistem ponaša pod različitim scenarijima i uvjetima. Primarni fokus testiranja putem upita leži u verifikaciji tačnosti podataka, brzine izvršavanja upita te ispravnosti sistema.

Prilikom testiranja upita, važno je osigurati da su oni napisani tačno i da daju očekivane rezultate, a samo testiranje baze podataka na ovaj način pruža mogućnosti za otkrivanje i ispravljanje pogrešaka te optimizaciju performansi, čime se postiže stabilnost i pouzdanost sistema.

Kako bi se izbjeglo ručno dodavanje velikog broja podataka u bazu podataka korišteni su *dummy podaci* koji su dodani uz upotrebu *factorys* i *seeders*.

Upiti za testiranje su napisani u metodi `index()` kontrolera stranice na koju se oni i ispisuju. Kod za ove upite se nalazi u datoteci koja je na lokaciji: `app->Http->Controllers->StatistikaController.php` i prikazan je na slikama 39 i 40 koje su date u nastavku ovog rada.



```
public function index()
{
    // Ispisati popis svih nagrada (abecedno) i kategorija za koju se dodjeljuju
    $nagrada_kategorije = DB::table('nagrada')
    ->select('nagrada.naziv as naziv', 'nagrada.kategorija as kategorija')
    ->orderBy('nagrada.naziv', 'asc')
    ->get();

    // Ispisati naslove svih filmova (abecedno) koji pripadaju žanru Akcija ili Krimi
    $zanr_filma = DB::table('filmovi')
    ->select('filmovi.naslov as naslov', 'filmovi.zanr as zanr')
    ->join('zanrovi', 'zanr', '=', 'zanrovi.id')
    ->whereIn('zanrovi.naziv', ['Akcija', 'Krimi'])
    ->orderBy('filmovi.naslov', 'asc')
    ->get();

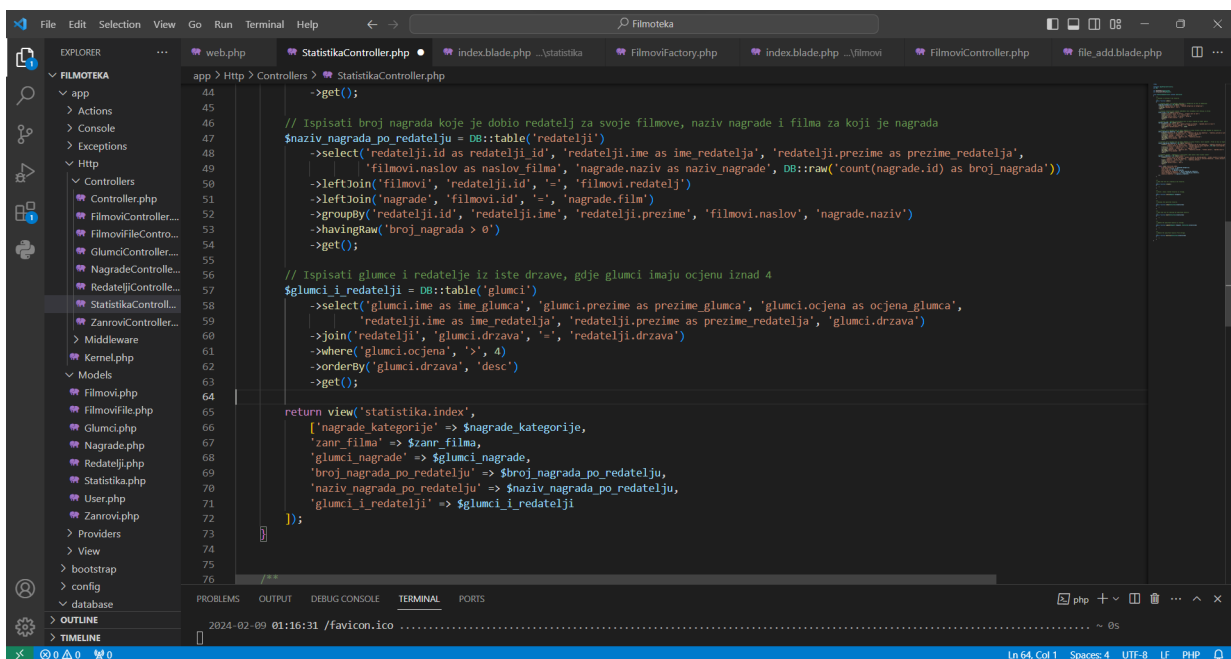
    // Ispisati ime i prezime svih glumaca koji su dobili nagradu do 2020. godine
    $glumci_nagrade = DB::table('glumci')
    ->select('glumci.ime as ime', 'glumci.prezime as prezime', 'nagrada.naziv as naziv')
    ->join('nagrada', 'glumci.nagrada', '=', 'nagrada.id')
    ->whereYear('nagrada.godina', '<=', 2020)
    ->get();

    // Ispisati broj nagrada koje je dobio redatelj za svoje filmove (ako nema nagrada ne ispisuje se)
    $broj_nagrada_po_redatelju = DB::table('redatelji')
    ->select('redatelji.id as redatelj_id', 'redatelji.ime as ime_redatelja', 'redatelji.prezime as prezime_redatelja', DB::raw('count(nagrade.id) as broj_nagrada'))
    ->leftJoin('filmovi', 'redatelj_id', '=', 'filmovi.redatelj')
    ->leftJoin('nagrada', 'filmovi.id', '=', 'nagrada.film')
    ->groupBy('redatelj_id', 'redatelj.ime', 'redatelj.prezime')
    ->havingRaw('broj_nagrada > 0')
    ->get();

    // Ispisati broj nagrada koje je dobio redatelj za svoje filmove i naziv nagrada i filma za koji je nagrada
}
```

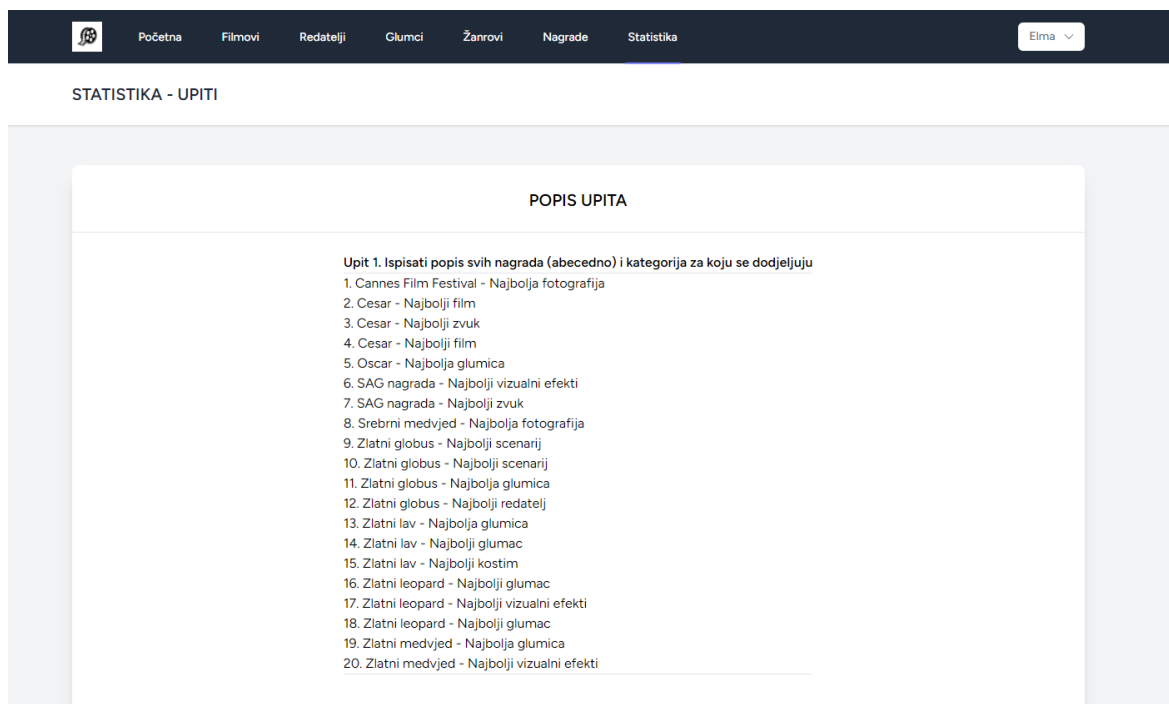
Slika 39: Prikaz koda upita





Slika 40: Prikaz koda upita - nastavak

Rezultati navedenih 7 upita su dati na slikama ispod, 41-43, a prikazani su na stranici Statistika projekta "Online filmoteka".



Slika 41: Stranica statistika, Rezultati upita: 1

**Upit 2. Ispisati naslove svih filmova (abecedno) koji pripadaju žanru Akcija ili Krimi**

1. Fight Club
2. Gladiator
3. Gladiator
4. Harry Potter
5. The Matrix

**Upit 3. Ispisati ime i prezime svih glumaca koji su dobili nagradu do 2020. godine**

1. David Smith-Gonzales - Cannes Film Festival
2. GlumacIme GlumacPrezime - Srebrni medvjed

**Upit 4. Ispisati broj nagrada koje je dobio redatelj za svoje filmove (ako nema nagrada ne ispisuje se)**

1. Robert Martinez - 4
2. Richard Lopez - 4
3. Matthew Brown - 3
4. Thomas Davis - 1
5. Richard Johnson - 1

Slika 42: Stranica statistika, Rezultati upita: 2-4

**Upit 5. Ispisati broj nagrada koje je dobio redatelj za svoje filmove, naziv nagrade i filma za koji je nagrada**

1. Redatelj: Robert Martinez - 1, Film: Forrest Gump, Naziv nagrade: Zlatni globus
2. Redatelj: Robert Martinez - 1, Film: Forrest Gump, Naziv nagrade: Zlatni leopard
3. Redatelj: Robert Martinez - 1, Film: Harry Potter, Naziv nagrade: SAG nagrada
4. Redatelj: Robert Martinez - 1, Film: The Lord of the Rings, Naziv nagrade: Zlatni globus
5. Redatelj: Richard Lopez - 1, Film: Gladiator, Naziv nagrade: Zlatni medvjed
6. Redatelj: Richard Lopez - 1, Film: The Lion King, Naziv nagrade: Cesar
7. Redatelj: Richard Lopez - 1, Film: The Lion King, Naziv nagrade: SAG nagrada
8. Redatelj: Richard Lopez - 1, Film: The Lion King, Naziv nagrade: Zlatni leopard
9. Redatelj: Matthew Brown - 1, Film: Inception, Naziv nagrade: Cesar
10. Redatelj: Matthew Brown - 1, Film: The Dark Knight, Naziv nagrade: Cannes Film Festival
11. Redatelj: Matthew Brown - 1, Film: The Dark Knight, Naziv nagrade: Zlatni globus
12. Redatelj: Thomas Davis - 1, Film: Forrest Gump, Naziv nagrade: Zlatni globus
13. Redatelj: Richard Johnson - 1, Film: The Lion King, Naziv nagrade: Srebrni medvjed

**Upit 6. Ispisati glumce i redatelje koji su iz iste države i gdje glumci imaju ocjenu iznad 4**

- UK - Daniel Jones - Jessica Davis  
UK - Daniel Jones - Robert Martinez  
UK - Richard Johnson - Jessica Davis  
UK - Richard Johnson - Robert Martinez  
SAU - Michael Garcia - John Hernandez  
SAU - Michael Garcia - Michael Davis  
RUS - Thomas Lopez - NoviRedatelj PrezimeRedatelja  
RUS - Thomas Lopez - Thomas Anderson  
ITA - Kimberly Johnson - Robert Smith  
ITA - Kimberly Johnson - Anthony Lopez  
ITA - Kimberly Johnson - Matthew Gonzalez  
ITA - Kimberly Johnson - Jennifer Lopez  
ITA - Kimberly Johnson - John Hernandez  
ITA - Kimberly Johnson - Jessica Jones  
IND - Jessica Anderson - James Johnson  
IND - Jessica Anderson - Robert Martinez  
IND - Jessica Anderson - Michael Hernandez  
IND - Jessica Anderson - Joseph Wilson

Slika 43: Stranica statistika, Rezultati upita: 5-6

## 5 Zaključak

Kroz proces razvoja online filmoteke koristeći objektno orijentisnu bazu podataka i Laravel aplikaciju, ostvaren je značajan napredak u stvaranju pristupačne i intuitivne platforme za ljubitelje filma. Ovaj projekt nije samo tehnički izazov, već i duboko iskustvo u razumijevanju potreba korisnika i razvijanja sistema koji odgovara njihovim očekivanjima.

Jedna od ključnih prednosti korištenja objektno orijentisane baze podataka je njezina sposobnost da reflektira stvarni svijet kroz jasno definirane entitete poput filmova, redatelja, glumaca i žanrova. Ova struktura omogućuje efikasno povezivanje različitih elemenata filmske industrije te olakšava korisnicima pristup relevantnim informacijama.

Proces modeliranja, implementacije i testiranja ključan je za stvaranje stabilne i funkcionalne web aplikacije. Kroz opis aplikacije i detaljne dijagrame, uključujući ER model, Use Case dijagram, Klasni dijagram i Sekvencijalni dijagram, definisane su arhitektura i funkcionalnosti aplikacije. Implementacija korištenjem Laravel Frameworka, podržanog alatima poput phpMyAdmin-a, XAMPP-a i Insomnia-e, pružila je korisno okruženje za razvoj i testiranje.

Analiza rada aplikacije kroz testiranje slučajeva use case-a, REST API-a i rada baze podataka omogućila je dublji uvid u funkcionalnosti i performanse "Online filmoteke". Utvrđeno je da je pravilno modeliranje ključno za funkcionalnost aplikacije, dok su alati poput Insomnie omogućili detaljno testiranje API-a, a sam phpMyAdmin je znatno olakšao administraciju baze podataka.

Unatoč postignutim uspjesima, ovaj projekt ostavlja prostor za daljnje unaprjeđenje i proširenje funkcionalnosti. Moguće nadogradnje uključuju dodavanje sistema preporuka, integraciju s vanjskim API-ima za prikupljanje dodatnih podataka o filmovima te implementaciju naprednih algoritama za pretraživanje i filtriranje sadržaja.

## Literatura

- [1] Laravel: <https://laravel.com/docs/10.x/readme>
- [2] Insomnia: <https://docs.insomnia.rest/insomnia/get-started>
- [3] XAMPP: <https://www.apachefriends.org/>
- [4] MVC: <https://www.geeksforgeeks.org/mvc-framework-introduction/>
- [5] PHP: <https://www.php.net/docs.php>
- [6] Visual Paradigma: <https://online.visual-paradigm.com/>