

## Zadaća 1 - Razvoj web aplikacije Filmoteka

**Poslužitelj:** spider.foi.hr

**Korijenska mapa zadatka:** /var/www/RWA/2022/zadaca\_01/{LDAP\_korisnicko\_ime}/

**Pristup preko web preglednika do zadatka:**

`http://spider.foi.hr/RWA/2022/zadaca_01/{LDAP_korisnicko_ime}`

**Aplikacija za rad s bazom podataka:** `https://phpmyadmin.spider.foi.hr/`

**Mapa postavki baze:** /var/www/RWA/2022/bazaPodaci/{LDAP\_korisnicko\_ime}.json

**Portovi APP:** /var/www/RWA/2022/portovi.js

**Portovi REST:** /var/www/RWA/2022/portovi\_rest.js

**Vanjski Servis "The movie data base" (TMDb):** <https://developers.themoviedb.org/>

**Način predaje:** Rješenje postaviti na poslužitelj do dogovorenog roka, u mapu za zadatak, a ne u mapu za vježbe. Nakon prijenosa postaviti prava (-rwxrwx---) za taj direktorij (chmod 770). **Zadaca postavljena na krivo mjesto ili nakon dogovorenog roka se ne ocjenjuje i rezultira s 0 bodova, a nepravilno definirana prava pristupa rezultira s oduzimanjem do 50% ostvarenih bodova.** Prava datoteka i poddirektorija unutar tog direktorija nisu bitna.

**Struktura zadatka:** U direktoriju smiju biti **samo datoteke i poddirektoriji vezane uz zadatak**, sve ostalo treba ukloniti (npr. skripte s vježbi). **Svugdje u zadaci treba koristiti relativne putanje.** Ako u tekstu zadatka piše naziv/struktura neke datoteke ili direktorija tada se treba držati tog naziva/strukture u ostalim slučajevima naziv/struktura je proizvoljna. **Multimedijalne datoteke postavljene na server moraju poštivati maksimalne veličine 500KB slike i 1MB video!** Ako **se ne pridržava zadane strukture direktorija, nazivlja datoteka i veličina datoteka** student se kažnjava s **oduzimanjem do 50% od ostvarenih bodova.** Poslužitelj **spider.foi.hr** prepoznaje velika i mala slova (eng. case sensitive) kod naziva direktorija, naziva datoteka i ekstenzija.

**Anketa:** Potrebno je ispuniti anketu vezanu uz zadatak na Moodle sustavu. Ako **se ne ispuni anketa ili se ne ispuni do kraja ili sadrži pogrešne informacije** oduzima se **do 50% od ostvarenih bodova.**

**Napomena:** Svi negativni bodovi se akumuliraju na ostale negativne bodove. Na primjer ne predana anketa, pogrešno definirana prava i pogrešna struktura mogu rezultirati s više od 50% negativnih bodova.

**Obrane zadatka:** Na kratkoj obrani na vježbama otvara se kod zadatka i objašnjava se što radi neki dio programskog koda koji nastavnih odredi. **Treba poznavati svaku liniju napisanog koda u zadaci u kontekstu programskog jezika u kojem je napisan, ali i logički koji je smisao koda u kontekstu zadatka.** **Ako student/ica ne da odgovor na većinu postavljenih pitanja smatra se da nije autor zadatka što znači 0 bodova iz zadatka, a time i potpis prema modelu praćenja.** Ako postoji sumnja o autorstvu zadatka, nastavnik zadržava pravo pozvati studenta/icu na konzultacije na dulju obranu zadatka u posebnom terminu konzultacija.

**Pomoć i pitanja:** Sva pitanja u vezi zadatka postavljaju se u **poseban forum za zadatak na sustavu Moodle.** Nema privatnih konzultacija u vezi zadatka osim ako nastavnik na vaše pitanje u forumu ne kaže da dodete na konzultacije zbog specifičnosti problem. Preporučujemo da ne šaljete kod drugima, ne

pokazujete kod drugima i ne radite zadaću zajedno iako ste cimeri ili se poznajete već 10 godina. **Zabranjeno je prepisivanje** zadaća; ako se utvrdi da je zadaća prepisana od drugog/ih studenta **svako dobiva 0 bodova**.

**Bodovanje:** Točna skala bodova nije unaprijed poznata i određuje se tek nakon što završi predaja zadaće. Boduju se dijelovi koji su rađeni samostalno nakon vježbi. U bodovanje, uz činjenicu radi li aplikacija prema specifikaciji ulazi: kvaliteta ispunjenosti ankete, kvaliteta programskog koda, model baze podataka, dizajn aplikacije, HTML/CSS validacija, količina grešaka u implementaciji, kvaliteta provjere unosa podataka od strane korisnika (sigurnost) i sl. Dijelovi označeni ovakvom plavom bojom su za dodatne bodove što znači ako se **ne naprave**, može se ostvariti maksimalni broj bodova iz zadaće.

**Validacija:** Sve stranice kada se učitaju moraju ispravno prikazivati hrvatske znakove. Sve učitane stranice moraju:

- prolaziti HTML5 validaciju - potrebno je koristiti službeni W3 validator na stranici <http://validator.w3.org/>. Nakon validacije postaviti HTML5 ikonu s poveznicom u podnožje stranice koja radi validaciju stranice (<http://validator.w3.org/check?uri={uri}>).
- prolaziti CSS validaciju za vlastite CSS datoteke - potrebno je koristiti službeni W3 validator na stranici <http://jigsaw.w3.org/css-validator/>. Nakon validacije postaviti CSS ikonu s poveznicom koja radi validaciju stranice (<https://jigsaw.w3.org/css-validator/validator?uri={uri}>). Ikone CSS validacije se postavljaju u podnožje početne stranice za svaku vlastitu CSS datoteku.

**Dizajn:** Dizajn stranice je za sada proizvoljan, važno je da se stranica može normalno koristiti u preglednicima Firefox i Chrome/Chromium. U ovoj zadaći nije potrebno raditi prilagodbe za mobilne uređaje i tablete, već isključivo za pregled na računalu.

**Okviri/moduli/biblioteke:** U zadaći na strani klijenta i strani poslužitelja se smiju koristiti samo okviri/moduli/biblioteke obrađeni na nastavi. Konkretno za NodeJs mogu se koristiti oni moduli koji su instalirani na poslužitelju globalno i **ne smiju se instalirati dodatni moduli na razini klijenta**. Popis dostupnih modula dobiva se sa: **npm -g list**

**JavaScript klijentska strana:** Prilikom izrade zadaće potrebno se voditi principima **nenametljivog** (engl. unobtrusive) JavaScript-a (Hint: addEventListener). Cijeli JavaScript kod se nalazi u vanjskim datotekama za pripadajuće stranice u kojoj su definirane vlastite funkcije i događaji na HTML elemente (NE atributi upravljača događaja). Svi osluškivači se učitavaju nakon učitavanja cijelog HTML dokumenta.

**Baza podataka:** Svaki student/ica ima svoju vlastitu bazu podataka na serveru spider na koju se spaja s podacima dobivenim na 2. vježbama. Podsjećamo da je kod pristupa aplikaciji za rad s bazom podataka **potrebno proći dvije razine autentifikacije**. Ponekad je potrebno više puta proći prvu razinu (situacija kada se javlja iskočni prozor) koja je jednaka za sve studente i tek onda se otvara aplikacija phpMyAdmin gdje se prijavljuje sa svojim podacima koje ste dobili.

**Čisti kod:** U zadaći se treba voditi principima čistog koda što ukratko znači sljedeće. Kod organizirati u funkcije, klase i module. Moduli sadrže klase, a klase sadrže funkcije koje čine smislenu cjelinu. Funkcije ne bi trebale imati više od 35 linija programskog koda programskog koda, u što se ne broji definiranje

metode, njenih argumenata i lokalnih varijabli, prazna linija, linija samo s { ili }. U jednoj liniji nalazi se jedna instrukcija. Linija nema više od 120 znaka. Radi se uvlačenje koda s 4 znaka space ili 1 tab. Nazivi klasa/funkcija/varijabli su smisleni i oslikavaju svrhu klase/funkcije/varijable. **Nazivi datoteka, direktorija, klasa, funkcija, varijabli i sl. pišu se na hrvatskom jeziku.** Naravno principi čistog koda koji su spomenuti na OWT za HTML i CSS također treba koristiti.

**Komentari:** U zadaći ne smiju biti komentari koji objašnjavaju kod. Kod koji je zakomentiran jer se ne koristi preporuka je da se obriše, ali nije problem ako negdje slučajno ostane. Zakomentiran kod također treba znati objasniti.

**Datum i vrijeme:** Svi datumski podaci moraju se na ekranu prikazivati i unositi u obliku dan.mjesec.godina sati:minute:sekunde npr. 04.10.2022 23:55:55.

**Provjere unosa korisnika:** Preporuka je da se sav korisnički unos kontrolira kako ne bi bilo moguće srušiti aplikaciju krivim unosima ili napraviti napad. Preporuka je korištenje regularnih izraza jer nosi više bodova nego varijanta s if-ovima.

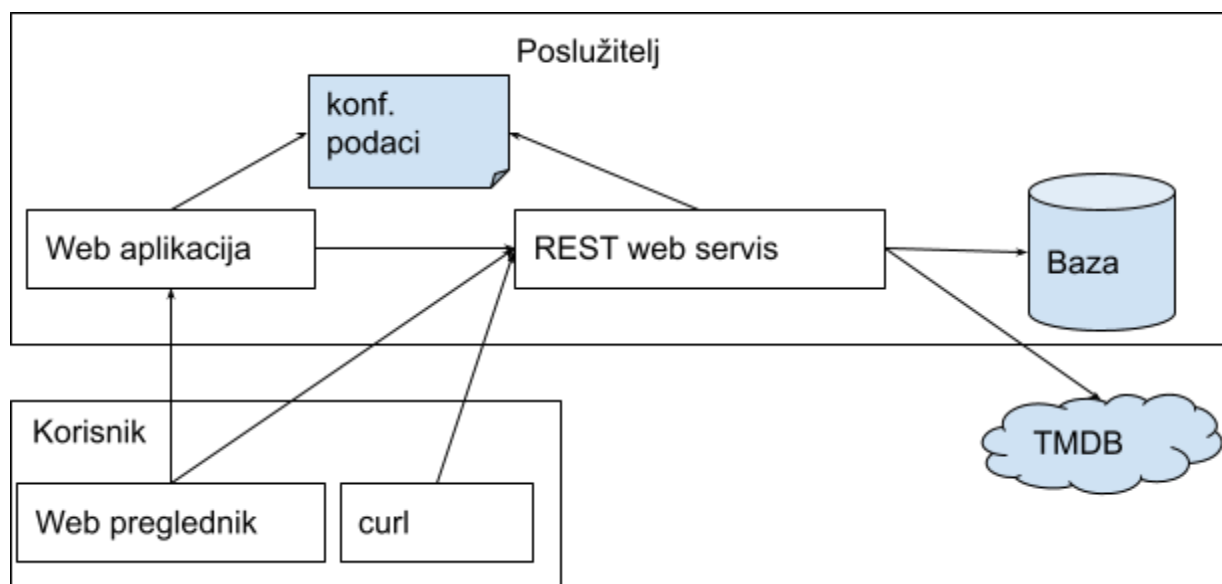
**Testiranje:** aplikaciju/servis treba testirati i s krivim podacima, provjeriti rubne slučajeve, a ne samo idealni scenarij korištenja. Kod testiranja servisa predlažemo curl naredbu jer možete lako više puta izvršiti. Također radi lakšeg testiranja bolje je napraviti prvo servisni dio, a tek onda aplikativni dio koji koristi REST.

#### Korisne TMDb adrese:

- Registracija: <https://www.themoviedb.org/signup>
- API ključ: <https://www.themoviedb.org/settings/api>
- Dokumentacija: <https://www.themoviedb.org/settings/api>
- API bazični URL: <https://api.themoviedb.org/3/>
- Lista žanrova: <https://developers.themoviedb.org/3/genres/get-movie-list>  
[https://api.themoviedb.org/3/genre/movie/list?api\\_key=xxx](https://api.themoviedb.org/3/genre/movie/list?api_key=xxx)
- Traženje filmova: <https://developers.themoviedb.org/3/discover/movie-discover>  
[https://api.themoviedb.org/3/discover/movie?api\\_key=xxx&language=en-US&sort\\_by=popularity.desc&include\\_adult=false&include\\_video=false&page=1&with\\_keywords=love](https://api.themoviedb.org/3/discover/movie?api_key=xxx&language=en-US&sort_by=popularity.desc&include_adult=false&include_video=false&page=1&with_keywords=love)
- Svi podaci o filmu: <https://developers.themoviedb.org/3/movies>  
[https://api.themoviedb.org/3/movie/500?api\\_key=xxx&language=en-US](https://api.themoviedb.org/3/movie/500?api_key=xxx&language=en-US)
- Putanje do slika: [https://image.tmdb.org/t/p/w600\\_and\\_h900\\_bestv2/{poster\\_path}](https://image.tmdb.org/t/p/w600_and_h900_bestv2/{poster_path})  
[https://www.themoviedb.org/t/p/w600\\_and\\_h900\\_bestv2/1yppMeTNQwDrzaUH4dRCx4mr8We.jpg](https://www.themoviedb.org/t/p/w600_and_h900_bestv2/1yppMeTNQwDrzaUH4dRCx4mr8We.jpg)

## Opis rada

Zadaća se sastoji od dva web poslužitelja (servera) kreiranih u programu Node gdje se svaki pokreće na svom portu i jednog sustava za upravljanje bazom podataka (SUBP). Prvi web poslužitelj pruža REST web servis i podiže se na vašem portu (vaše korisničko ime) koji se čita iz datoteke **portovi\_rest.js**. Drugi web poslužitelj predstavlja web aplikaciju i podiže se na vašem portu (vaše korisničko ime) koji se čita iz datoteke **portovi.js**. Oba web poslužitelja i SUBP nalaze se na istom fizičkom poslužitelju. U zadaći se koristi i vanjski REST servis TMDB koji sadrži informacije o filmovima, serijama i sl. Sljedeća slika prikazuje globalni pregled komponenata i veza u zadaći. Kao što se može vidjeti samo REST web servis ima pristup bazi podataka i TMDB servisu, dok web aplikacija nema pristup servisu te koristi isključivo REST web servis za dohvat svih podataka. **U realizaciji je potrebno primjenjivati asinkrono programiranje. Ni u jednom trenutku ne bi trebalo doći do blokiranja glavnog programa.**



## Baza podataka

Bazu podataka je potrebno samostalno modelirati i kreirati ERA model baze u alatu MySQL Workbench. Potrebno je kreirati poseban direktorij **dokumentacija** u kojem se nalazi izvorna datoteka modela baze iz alata MySQL Workbench, skripta .sql koja sadrži SQL kod za kreiranje baze generirana iz alata, slika ERA modela baze koja se izveze iz alata MySQL Workbench i **dokumentacija.html**. Slika ERA modela mora biti dovoljne veličine da se sve vidi čitati, mora imati jasno vidljive veze i sve attribute u tablici. Ova slika **SMIJE** biti veća od 1MB. Stranica dokumentacija.html sadrži informacije o vama kao autoru: slika, ime, prezime, e-mail. Također sadrži sliku ERA modela u manjoj veličini, a pritiskom na nju otvara se originalna slika koja je kreirana s alatom MySQL Workbench. U bazi podataka mora se nalaziti minimalno 30 podataka ukupno za sve glavne entitete. **Za potrebe testiranja u bazi moraju postojati dva korisnika. Prvi korisnik ima korisničko ime “obican”, ulogu “registrirani korisnik” i lozinku “rwa”. Drugi korisnik ima korisničko ime “administrator”, ulogu “admin” i lozinku “rwa”. Ako korisnici ne postoje u bazi, neće se bodovati dio koji ulazi u funkcionalnosti odgovarajuće uloge.**

## REST web servis

Ovaj servis predstavlja aplikativno programsko sučelje (API) prema bazi podataka i prema vanjskom servisu (TMDB), a koji koristi web aplikacija za pristup podacima.

Kod pokretanja poslužiteljskog programa web servisa koristi se:

```
node restServer.js nazivDatotekeKonfiguracije.csv &
```

Ako datoteka konfiguracije ne postoji pri podizanju servera, javlja se greška i poslužitelj prestaje s radom. Ako konfiguracijska datoteka postoji, radi se provjera postoje li svi potrebni konfiguracijski podaci unutar konfiguracijske datoteke. Konfiguracijski podaci koji se koriste su označeni s **REST** u tablici 3. Ako neki podatak nedostaje u datoteci ili ima pogrešnu vrijednost u odnosu na specifikaciju u tablici 3, javlja se poruka sa opisom koji podatak je problem i opisom razloga te poslužitelj prestaje s radom. Kada svi konfiguracijski podaci postoje u datoteci i kada su ispravni, poslužitelj čeka na zahtjev za obradu.

REST servis nudi API za resurse prema specifikaciji u Tablici 1. **Sva komunikacija je u JSON obliku unutar tijela poruke zahtjeva ili odgovora, osim ako nije drugačije definirano parametrom.** Za dodatne bodove moguće je dodati XML varijantu i dodatni parametar **?vrsta=[XML | JSON]** u kojem se može definirati želi li se XML ili JSON varijanta komunikacije. Kod svih zahtjeva (osim na `api/konf`) moraju se poslati korisničko ime i lozinka za pristup servisu **?korime={rest.korime}&lozinka={rest.lozinka}**

Ako podaci o korisničkom imenu ili lozinki nisu poslani ili su u pogrešnom obliku (prema specifikaciji u tablici 3) vraća se “nevaljani zahtjev” (**400**), a ako podaci jesu poslani i jesu u ispravnom obliku, ali nisu točni (korisničko ime ili lozinka) vraća se odgovor “neautoriziran pristup” (**401**). Nakon što je autentifikacija uspješna obrađuje se zahtjev prema specifikaciji u Tablici 1. Na mjestima u tablici 1 gdje je prazno (bez opisa) vraća se “metoda nije implementirana” (**501**). Ako se pokuša dohvatiti resurs koji nije u specifikaciji tada servis vraća “nema resursa” (**404**). Ako se pokuša dohvatiti ispravni resurs, ali s krivim parametrima (npr. `api/filmovi?str=10`) tada se vraća “neočekivani podaci” (**417**). Svi zahtjevi koji u putanji imaju **api/tmdb** preuzimaju podatke direktno s vanjskog servisa i prosljeđuju se njegove poruke greške ako se dogode. Svi ostali zahtjevi rade s vašom bazom podataka. **Sve poruke grešaka se vraćaju unutar odgovora pod atributom “greska”.**

Servis ima specijalni resurs **api/konf** koji učitava nove konfiguracijske podatke i radi provjeru kao kod podizanja poslužitelja. Ako je sve ok učitano koristi dalje za rad nove podatke i vraća da je sve “ok” (**200**) kao odgovor. Ako dođe do bilo koje od već opisanih grešaka konfiguracijske datoteke **svi** zahtjevi osim resursa `api/konf` ne prolaze i vraća se poruka “servis nedostupan” (**503**) sve dok se konfiguracija opet uspješno ne učitava nekim novim pozivom na ovaj resurs.

Tablica 1. Pregled resursa, metoda i parametara REST servisa

| Resurs<br>Parametri  | GET<br>čitanje   | POST<br>kreiranje  | PUT<br>ažuriranje               | DELETE<br>brisanje                         |
|--|--|--|---------------------------------|--|
| api/korisnici  | vraća listu korisnika  | kreira novog korisnika   |                                 |  |
| api/korisnici/<br><i>{korime}</i>  | vraća određenog korisnika  | “metoda nije dopuštena” (405)  | Ažurira određenog korisnika     |  |
| api/korisnici/<br><i>{korime}</i> /aktivacija  |  | “metoda nije dopuštena” (405)  | Aktivira novog korisnika        |  |
| api/korisnici/<br><i>{korime}</i> /prijava   |  | Provjerava podatke prijave i vraća je li korisnik uspješno prijavljen              |                                 |  |
| api/filmovi/<br>?stranica= <i>broj</i><br>&brojFilmova= <i>broj</i><br>[&datum= <i>datum</i> ]<br>[&zanr= <i>idZanr</i> ]<br>[&naziv= <i>dioNazivFilma</i> ]<br>[&sortiraj=[ <i>d</i>   <i>z</i>   <i>n</i> ]] | vraća listu određenog broja filmova za neku stranicu ( <i>broj</i> ), opcionalno filtrirano po datumu unosa ( <i>datum</i> ) i/ili žanru ( <i>idZanr</i> ) i/ili nazivu filma ( <i>dioNazivFilma</i> ), te može sortirati uzlazno po datumu ( <i>d</i> ), nazivu ( <i>n</i> ) ili žanru ( <i>z</i> ) | Uz dobivene podatke čita sve podatke o filmu s vanjskog servisa i dodaje novi film |                                 |  |
| api/filmovi/ <i>{id}</i>   | Vraća podatke određenog filma ( <i>id</i> )  | “metoda nije dopuštena” (405)  | Ažurira podatke određenog filma | Briše određeni film                        |
| api/zanr/  | Vraća listu žanrova  | Dodaje novi žanr   |                                 | Briše sve žanrove za koje ne postoji film. |
| api/zanr/ <i>{id}</i>  | Vraća podatke određenog žanra ( <i>id</i> )  | “metoda nije dopuštena” (405)  | Ažurira postojeći žanr          | Briše određeni žanr                        |
| api/tmdb/zanr  | Prosljeđuje listu žanrova koji vrati servis TMDB   |  |                                 |  |
| api/tmdb/filmovi<br>?ključaRijec= <i>rijec1</i><br>[, <i>rijec2</i> ]<br>&stranica= <i>broj</i>  | Prosljeđuje listu filmova koju vrati servis TMDB, a čiji filmovi imaju ključnu riječ ili više njih   |  |                                 |  |

## Web aplikacija

Ova web aplikacija pruža korisničko sučelje za rad s filmovima, TV serijama i sl. Kod pokretanja poslužiteljskog programa web servisa koristi se:

**node appServer.js nazivDatotekeKonfiguracije.csv &**

Pri podizanju poslužitelja ako datoteka konfiguracije ne postoji javlja se greška i poslužitelj prestaje s radom. Ako konfiguracijska datoteka postoji radi se provjera postoje li svi potrebni konfiguracijski podaci. Konfiguracijski podaci koji su potrebni su označeni s **APP** u tablici 3. Ako neki podaci ne postoje ili imaju pogrešnu vrijednost u odnosu na specifikaciju u tablici 3, ispisuje se poruka u konzolu koji podatak je problem s razlogom te poslužitelj prestaje s radom. Kada svi konfiguracijski podaci postoje u datoteci i kada su ispravni, poslužitelj šalje probni zahtjev na REST web servis. Ako ne dobije odgovor ili dobije odgovor 400 ili 401 javlja se odgovarajuća poruka o grešci u konzoli i poslužitelj prestaje s radom. Ako dobije od REST web servisa odgovor 200 tada čeka na zahtjeve od korisnika.

U aplikaciji postoje 3 uloge korisnika (admin > korisnik > gost). Svaka uloga više razine ima pristup pogledima niže razine. U tablici 2 dan je opis funkcionalnosti po ulogama s aspekta korisnika. U svim pogledima ako se dohvaća više podataka treba implementirati straničenje. Maksimalan broj elemenata po stranici definiran je s konfiguracijskim parametrom **app.broj.stranica**. Na svim stranicama treba biti navigacija te ni jedna stranica ne smije biti slijepa ulica. Svaka stranica mora imati gumb za prijavu ili odjavu ovisno o statusu prijavljenosti. Ne smije se dozvoliti neovlašteni pristup stranicama.

Na stranicama “Filmovi pregled”, “Filmovi pretraživanje” i “Galerija slika” treba napraviti takvo straničenje, pretraživanje i/ili sortiranje da se stranica NE osvježava. Varijanta s osvježavanjem nosi manje bodova.

**Tablica 2. Pregled funkcionalnosti Web aplikacije**

| Uloga    | Stranica/e                    | Opis  |
|----------|-------------------------------|---|
| Gost     | Početna                       | Prikazuje popis žanrova u bazi podataka i nasumice prikazuje 2 filma za svaki žanr koji postoje u bazi podataka.  |
| Gost     | Dokumentacija                 | Prikazuje stranicu dokumentacija.html opisanu ranije.   |
| Gost     | Registracija, Prijava, Odjava | Gost se može se registrirati kao novi korisnik. Kod registracije šalje se e-mail s tokenom za aktivaciju računa. Lozinka se u bazu ne smije spremati u čitljivom obliku i treba koristiti sol kod spremanja. Dok se račun ne aktivira nije moguće napraviti prijavu. Kod prijave treba koristiti autentikacija dvije razine prvo lozinka s JWT i nakon toga TOTP. Jednom prijavljeni korisnik može se odjaviti u bilo kojem trenutku. |
| Korisnik | Profil                        | Vidi svoje podatke (ime, prezime, korisničko ime, e-mail, ...) i može ih ažurirati. Korisničko ime i e-mail se <u>ne mogu ažurirati</u> .   |
| Korisnik | Filmovi pregled               | Vidi popis svih filmova koji postoje u bazi, može filmove filtrirati po žanru i/ili datumu unosa. Datumsko filtriranje čita podatke od unesenog datuma do današnjeg datuma. Može uzlazno sortirati filmove po nazivu filma, datumu ili nazivu žanra. Može kliknuti na naziv filma i dobiti sve podatke o filmu.   |



|          |                       |  |
|----------|-----------------------|--|
| Korisnik | Film                  | Prikazuje sve podatke o filmu zajedno s posterom i poveznicom na "Galerija slika" za taj film i podatkom koji korisnik je dodao film u bazu.   |
| Korisnik | Filmovi pretraživanje | Može upisati ključne riječi po kojima se pretražuju filmovi sa servisa TMDB (preko api/tmdb). Broj stranica nalazi se u podatku " <b>total_pages</b> ". Ako pronade film koji mu se sviđa može kliknuti na dodaj u bazu. Film se svim dobivenim podacima dodaje u bazu kao prijedlog, ali čeka odobrenje admina da bude vidljiv u "Filmovi pregled". Nije potrebno spremati podatke o "production_companies", "production_countries" i "spoken_languages". Ako već postoji prijedlog za neki film tada nije moguće kliknuti dodaj u bazu.  |
| Korisnik | Galerija slika        | Korisnik vidi galeriju slika postavljenih od strane svih korisnika za odabrani film. Može pretraživati galeriju slika po datumu postavljanja slike tako da unese od <b>i/ili</b> do datum. Može kliknuti na sliku te ga se prebacuje na "Film".  |
| Korisnik | Slika                 | Može postaviti novu sliku (samo jpg, jpeg, png ili gif maksimalne veličine 500KB) za neki film koji postoji u bazi. Slika se sprema fizički na poslužitelj unutar direktorija <b>slike/{id_filma}/{korime}</b> . Direktorij označen sa id_filma predstavlja id filma pod kojim je film spremljen u bazi, a direktoirj korime je korisničko ime korisnika koji je dodao sliku. Slika se sprema pod nazivom <b>{datum_hash}.{ekstenzija}</b> . Datum je datum postavljanja slike u obliku "godina-mjesec-dan", a hash je generiran na temelju vremena i korisničkog imena. Ekstenzija je originalna ekstenzija prenesene datoteke.   |
| Admin    | Filmovi prijedlozi    | Vidi popis filmova koji su poslani kao prijedlog za dodavanje, ako odobri film postaje vidljiv svim korisnicima u pogledu "Filmovi pregled". Kod odobravanja filma preuzima se slika postera s TMDB koja se sprema lokalno na poslužitelj u direktorij <b>poster</b> . Ako ne odobri film tada se on briše i može se opet kasnije dati kao prijedlog.  |
| Admin    | Žanrovi               | Vidi pregled svih žanrova u bazi podataka, može ažurirati naziv žanra, može obrisati sve žanrove za koje <b>nema</b> filmova u bazi. Može povući listu žanrova s TMDB servisa te može dodati u bazu sve ili samo neke žanrove. Ako pri dodavanju, već postoji neki žanr s istim id-om, tada se takav preskače i uopće se ne šalje zahtjev na servis za dodavanjem. Korisniku se javlja poruka na ekranu koji žanrovi su bili preskočeni jer već postoje.   |
| Admin    | Filmovi ažuriranje    | Može ažurirati podatke nekog filma koji je <b>odobren</b> .  |
| Admin    | Korisnici             | Vidi popis korisnika, može blokirati korisnika te se takav korisnik više ne može prijaviti. Može ažurirati podatke o svim korisnicima osim podataka o e-mailu, korisničkom imenu i lozinki.  |
| Admin    | Konfiguracija         | Administrator može pregledavati i ažurirati podatke konfiguracije datoteke kroz sučelje za obje aplikacije. Ima poseban gumb "učitaj nove podatke" te se radi kopija starih podataka, novi se spremaju u datoteku i šalje se zahtjev na servis da se učitaju novi podaci. Ako servis vrati da je uspješno tada se učitavaju novi podaci u web aplikaciju i ako je sve ok javlja se poruka o uspjehu. Ako servis vrati grešku ili ne prođe provjera u web aplikaciji tada se vraćaju podaci na staru vrijednost iz kopije, kopija se briše i automatski se šalje novi zahtjev na servis da ponovo učita podatke. Korisniku se javlja poruka o grešci i ispis novih podataka (za slučaj da želi provjeriti i pokušati ponovo). |



### Konfiguracijski podaci

U tablici 3 definirani su nazivi svih parametara koje koriste aplikacije. Za svaki parametar napisani su kriteriji u kakvom obliku mora biti podatak te kratak opis čemu služi. Na kraju je dana informaciju za koju aplikaciju treba taj parametar. Konfiguracijski podaci spremaju se u CSV datoteci u obliku "naziv=vrijednost", svaki u novi red.

Tablica 3. Pregled konfiguracijskih parametara

| Naziv konfiguracije | Vrijednost   | Opis  | Potrebno za |     |
|---------------------|--|---|-------------|-----|
|                     |  |   | REST        | APP |
| rest.korime         | Veličina: 15-20 znakova<br>Dozvoljena slova i brojeke<br>Obavezno: 2 slova, 2 broja                        | Sadrži korisničko ime za pristup rest servisu   | DA          | DA  |
| rest.lozinka        | Veličina: 20 - 100 znakova<br>Dozvoljen bilo koji znak<br>Obavezno: 3 slova, 3 brojke i 3 specijalna znaka | Sadrži lozinku za pristup rest servisu  | DA          | DA  |
| app.broj.stranica   | Broj od 5 - 100  | Sadrži brojčani podatak koji govori koliko se podataka odjednom prikazuje u tablicama |             | DA  |
| tmdb.apikey.v3      | Nema restrikcija   | Sadrži API ključ v3 za pristup TMDB servisu oko 32 znaka                              | DA          |     |
| tmdb.apikey.v4      | Nema restrikcija   | Sadrži API ključ v4 za pristup TMDB servisu oko 211 znakova                           | DA          |     |