

# Izrada javno dostupne baze podataka o filmovima

---

**Zvjerković, Domagoj**

**Master's thesis / Diplomski rad**

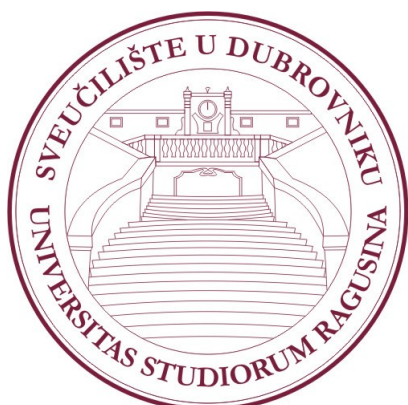
**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Dubrovnik / Sveučilište u Dubrovniku**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:155:923626>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-29**



*Repository / Repozitorij:*

[Repository of the University of Dubrovnik](#)



SVEUČILIŠTE U DUBROVNIKU

---

ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

STUDIJ POSLOVNO RAČUNARSTVO

**DIPLOMSKI RAD**

IZRADA JAVNO DOSTUPNE BAZE PODATAKA O  
FILMOVIMA

Dubrovnik, srpanj 2018.

SVEUČILIŠTE U DUBROVNIKU

---

ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO  
STUDIJ POSLOVNO RAČUNARSTVO

**DIPLOMSKI RAD**

IZRADA JAVNO DOSTUPNE BAZE PODATAKA O  
FILMOVIMA

Mentor:  
**prof. dr. sc. Vedran Batoš**

Komentor:  
**Ivan Grbavac, dipl. ing.**

Diplomant:  
**Domagoj Zvjerković**

Dubrovnik, srpanj 2018.

## SAŽETAK

U ovom radu opisan je koncept stranica za ocjenjivanje filmova. U početku se daje pregled već postojećih stranica. Opisuje se kako postojeće stranice prikupljaju, obrađuju i prikazuju podatke o filmovima i recenzijama filmova. Na temelju analize drugih stranica osmišlja se verzija stranice za ocjenjivanje filmova i opisuju se tehnologije koje se koriste prilikom izrade stranice.

Izrada web stranice za ocjenjivanje filmova radi se u više faza. U prvoj fazi se definiraju korisnički zahtjevi. Opisuju se glavne funkcionalnosti koje stranica mora posjedovati, poput pretrage i pregleda filmova. U drugoj fazi se opisuje arhitektura sustava, fizička i logička. Podaci o korisnicima i ocjenama filmova spremljeni su na web server, dok se informacije o filmovima dohvaćaju pomoću API servisa. U trećoj fazi pristupa se dizajnu sustava, gdje se detaljnije opisuju pojedini procesi i funkcionalnosti sustava. U zadnjoj fazi se opisuje programsko rješenje koje je dobiveno upotrebom izabраниh tehnologija. U zadnjem poglavlju su definirane upute za korištenje stranice, od registracije do ostavljanja recenzija.

*Ključne riječi: web stranica, recenzija, filmovi*

## **ABSTRACT**

This paper describes the concept of a web site for rating movies. At the start of the paper a review of the existing web sites is provided. It is described how existing pages collect, process and display movie and movie review data. Based on the analysis of other web sites, a version of the movie rating web site is being drafted and the technologies used to create the web site are described.

Making a movie rating website is done in several phases. The first phase defines user requirements. The main functionalities that the site must have are described, such as search and review of movies. The second phase describes the architecture of the system, both physical and logical. User data and movie ratings are saved on the web server, while movie information is retrieved using the API service. In the third phase, the system design is approached, detailing the individual processes and functionality of the system. In the final phase we will describe the software solution that was obtained using the selected technologies. In the last chapter, user instructions for the web site are defined, from user registration to making a movie review.

*Keywords: web site, reviews, movies*

# SAŽETAK

1. UVOD .....	1
2. POSTOJEĆE STRANICE ZA OCJENJIVANJE .....	2
2.1 IMDb – Internet movie database .....	2
2.2 Rotten tomatoes .....	3
2.3 Metacritic .....	6
3. TEHNOLOGIJE KORIŠTENE U IZRADI WEB STRANICE .....	7
3.1 HTML - Hypertext Markup Language .....	7
3.2 CSS - Cascading Style Sheets .....	8
3.3 JavaScript.....	9
3.3.1 JQuery.....	9
3.3.2 AJAX - Asynchronous JavaScript And XML.....	10
3.4 JSON – JavaScript Object Notation .....	11
3.5 PHP - Hypertext Preprocessor .....	12
3.6 SQL i relacijska baza podataka.....	13
3.7 OMDb API – Open Movie Database .....	14
4. PROJEKTIRANJE I DIZAJN .....	15
4.1 Zahtjevi .....	15
4.1.1 Funkcionalni zahtjevi .....	15
4.1.2 Nefunkcionalni zahtjevi .....	15
4.2 Arhitektura sustava .....	17
4.2.1 Fizička arhitektura sustava .....	17
4.2.2 Logička arhitektura sustava.....	17
4.3 Dizajn sustava .....	19
4.3.1 Registracija na web stranicu.....	19
4.3.2 Prijava na web stranicu .....	20
4.3.3 Pretraga filmova .....	21
4.3.4 Recenzija filma.....	22
4.3.5 Ocjena recenzija drugih korisnika .....	23
5. IZRADA SUSTAVA.....	24
5.1 Prezentacijskih sloj .....	24
5.2 Implementacija baze podataka .....	25
5.3 Implementacija funkcionalnosti .....	27
5.3.1 Registracija na sustav .....	27

5.3.2 Prijava na sustav .....	30
5.3.3 Pretraga filmova .....	31
5.3.4 Prikaz početne stranice .....	32
5.3.5 Prikaz stranice s informacijama o filmu.....	33
5.3.6 Recenzija filma.....	33
5.3.7 Ocjena recenzije .....	35
6. OPIS SUSTAVA .....	37
7. ZAKLJUČAK.....	45
8. LITERATURA .....	46

## 1. UVOD

U današnje doba postoji velika količina filmova i novi filmovi izlaze skoro svaki tjedan. Najave filmova se ponekad objavljuju i godinu dana prije izlaska filma i velika količina novca se ulaže u marketing filmova. No skupi marketing više nije dovoljan da privuče gledatelje filmova u kino. S obzirom koliko filmova godišnje izlazi, ljudi su postali oprezni kod izbora filmova na koje će potrošiti svoje vrijeme i novac. No osim odlaskom u kino, filmovi su danas dostupni i preko raznih *streaming* usluga. Danas je uz par klikova jednostavno doći do filma.

Ljudi pri izboru filma traže tuđe mišljenje o filmu. To je dovelo do rasta popularnosti stranica za ocjenjivanje filmova. Takve stranice sadrže bazu podataka svih filmova ikad napravljenih. Funkcioniraju na temelju podataka koje unose korisnici. Premda različite stranice na različit način računaju i prikazuju ocjene filmova svima je cilj na jednostavan i lako razumljiv način korisniku prikazati informaciju koja će mu pomoći pri izboru filma. Većina filmskih kritičara dobije priliku pogledati film i dva tjedna prije njegovog službenog izlaska. Stranice za ocjenjivanje prikupljaju recenzije tih kritičara kako bi svojim korisnicima mogli olakšati odluku o gledanju filma. No ne vjeruju svi korisnici filmskim kritičarima. Neki korisnici žele mišljenje ljudi koji nemaju veze sa filmskom industrijom. Zato neke stranice odvajaju ocjene kritičara od ocjene publike. Te dvije ocjene za neke filmove znaju i znatno odstupati jedna od druge. Pri razvoju ovakve stranice treba voditi računa na kakav način će se podaci prikupljati i prikazivati korisnicima.

Za razvoj stranice za ocjenjivanje filmova potrebna je baza koja će sadržavati sve informacije o filmovima, baza koja će sadržavati sve informacije o korisnicima stranice i njihovim recenzijama filmova i tehnologija potrebna za obradu i prikaz svih tih informacija. Baza o filmovima dostupna je korištenjem web servisa te je potrebno samo osmisliti dohvat i prikaz tih podataka. Korisnička baza treba biti zaštićena od bilo kakvih napada i neovlaštenih unosa. Korisničko sučelje treba biti jednostavno za korištenje. Prikaz podataka mora biti prirodan za korisnika te korisnik treba moći lagano doći do informacije koju traži. Kako bi se to ostvarilo potrebno je korištenje više tehnologija koje funkcioniraju zajedno.

U prvom poglavlju opisat će se 3 različite stranice za ocjenjivanje filmova: IMDb, Rotten Tomatoes i Metacritic. U drugom poglavlju opisat će se tehnologije koje će se koristiti pri izradi web stranice. U trećem poglavlju pristupa se projektiranju i dizajnu web stranice dok se u četvrtom i petom poglavlju opisuje sama izrada web stranice i definiraju korisničke upute.



## 2. POSTOJEĆE STRANICE ZA OCJENJIVANJE

### 2.1 IMDb – Internet movie database

*Internet movie database* je najveća i najopsežnija baza podataka na webu. Izvor je filmskih podataka te sadrži filmove, filmske vijesti, recenzije, najave filmova, profile slavnih osoba itd. IMDb nudi pretraživu bazu s više od 250 milijuna podataka, uključujući više od 5 milijuna filmova, TV i zabavnih programa.

IMDb stranica je dostupna svim korisnicima interneta, ali potrebna je registracija za doprinos informacija bazi podataka. Većinu podataka u bazi doprinose dobrovoljni korisnici. Registrirani korisnici mogu dodavati nove materijale i uređivati postojeće informacije. Korisnici koji imaju dokazanu evidenciju podnošenja činjeničnih podataka dobivaju odobrenje za dodavanje ili mijenjanje informacija o ulogama u filmu i drugim demografskim podacima. Međutim, svaka promjena slika, imena likova, sažetka filma i naslova se provjerava prije objavljivanja. Obično je potrebno 24 do 72 sata da se promijenjene informacije pojave.

Svi registrirani korisnici odabiru svoje vlastito ime na stranici. Registrirani korisnici imaju stranicu profila na kojoj se može vidjeti koliko dugo je taj korisnik registriran i osobne ocjene filmova. Korisnici mogu ocijeniti bilo koji film na skali od 1 do 10, zbrojevi se pretvaraju u ponderiranu prosječnu ocjenu koja se prikazuje pored svakog filmskog naslova. Ponderirana aritmetička sredina slična je običnoj aritmetičkoj sredini, osim što neke podatkovne točke pridonose konačnom rezultatu više od drugih. Kako bi se spriječilo glomazanje koriste se mrežni filtri.

IMDb početna stranica omogućuje brz pronalazak aktualnih filmova, a za fokusiraniju pretragu postoji IMDb tražilica. Pretraživati se može prema naslovu filma/TV serije, imenima postave/posade, imenima likova, nedavnim objavama, posebnim IMDb značajkama itd.

Svaka stranica povezana sa filmom ili TV serijom na IMDb-u nudi nekoliko dosljednih značajki:

- poster filma,
- sažetak radnje,
- ključne riječi,
- informacije o glumačkoj postavi,
- relevantni citati.

IMDb nudi opsežniju i robusniju upotrebu pretplatnicima na uslugu „*IMDb pro*“, međutim velika količina informacija pronađenih na IMDb-u dostupna je besplatno.

## 2.2 Rotten tomatoes

*Rotten Tomatoes* i *Tomatometer* ocjena su jedne od najpouzdanijih izvora preporuka za gledanje filmova i TV serija. Tomatometer ocjena predstavlja postotak pozitivnih stručnih recenzija filmova i TV serija, a milijuni korisnika koriste ih za pomoć pri njihovim odlukama odabira gledanja filmova. Najbolje ocjenjeni filmovi i TV serije se označavaju kao *Certified Fresh*.

Osoblje web stranice prvo prikuplja online recenzije od pisaca koji su certificirani članovi raznih cehova za pisanje ili udruženja filmskih kritičara. Da bi se prihvatio kritičar na web stranici, kritičareve originalne recenzije moraju prikupiti određen broj *likeova* od strane korisnika. Oni koji su klasificirani kao „vrhunski kritičari“ općenito pišu za velike novine. Osoblje određuje za svaku recenziju je li pozitivna ili negativna. Procjena osoblja je potrebna jer neke recenzije dodjeljuju kvalitativnu procjenu, a ne brojčanu ocjenu.

Kada je najmanje 60% recenzija za film ili TV seriju pozitivno, prikazana je crvena rajčica koja označava njezin status svježeg.



Slika 1. [2] Znak za svježe

Kada je manje od 60% recenzija za film ili TV seriju pozitivno, prikazuje se zeleni znak koji označava status pokvarenog.



Slika 2. [2] Znak za pokvareno

*Certified Fresh* (certificirano svježe) status je posebna oznaka koja se dodjeljuje najpopularnijim filmovima i TV serijama. Da bi se kvalificirali, filmovi ili TV serije moraju ispunjavati sljedeće uvjete:

- stalnu Tomatometar ocjenu od 75% ili više,
- najmanje pet recenzija od vrhunskih kritičara,
- filmovi u širokom izdanju moraju imati najmanje 80 recenzija,
- filmovi u ograničenom izdanju moraju imati najmanje 40 recenzija,
- samo pojedinačne sezone TV serija su dopuštene i svaka mora imati najmanje 20 recenzija.

Film ili TV serija koja ispunjava uvjete ne dobija odmah tu oznaku. Oznaka se automatski označava za razmatranje te kada zaposlenici utvrde da je mala vjerojatnost da će ocjena pasti ispod minimalnih zahtjeva, odobrava se oznaka. Ako rezultat Tomatometra padne ispod 70%, film ili TV serija će izgubit svoj *Certified Fresh* status.



Slika 3. [2] Znak za *Certified Fresh*

Ocjenjivanje publike označeno je s kantom kokica. Ono je postotak svih korisnika koji su pozitivno ocijenili film ili TV seriju. Puna kanta kokica znači da je film dobio 3.5 ili više zvjezdica na skali do 5.



Slika 4. [2] Znak pune kante

Srušena kanta kokica znači da je film dobio manje od 3.5 zvjezdica od strane korisnika.



Slika 5. [2] Znak srušene kante

Znak plus se pojavljuje za filmove koji nemaju ocjene publike ili recenzije. Postotak koji se vidi s ovom ikonom postotak je korisnika koji su dodali film na popis „Želim vidjeti“.



Slika 6. [2] Znak plus

## 2.3 Metacritic

Metacritic je web stranica koja prikuplja recenzije medijskih proizvoda: filmova, TV serija, videoigara i glazbenih albuma. Za svaki proizvod koristi se ponderirani prosjek ocjena iz svake recenzije. Metacritic ocjena ili *Metascore* pretvara svaku recenziju u postotak, ili matematički ili subjektivno ako ocjena nije numerička.

Stvaranje Metascorea je složen proces. Pažljivo se prikupi skupina najugledniji svjetskih kritičara, dodjeljuje se ocjena njihovim recenzijama te se primjeni ponderirani prosjek kako bi se sažeo raspon njihovih mišljenja. Rezultat je jedan broj koji bilježi bit kritičkog mišljenja u jednom Metascoreu. Svaki film, TV serija, igra i album dobiva Metascore kada se prikupi najmanje 4 recenzije kritičara.

Metascore je ponderirani prosjek u kojem se dodjeljuje veća važnost ili težina nekim kritičarima i publikacijama od drugih na temelju njihove kvalitete i reputacije. Metascoreovi se kreću od 0-100, gdje viša ocjena ukazuje na bolje ukupne recenzije. Metascore je istaknut u 3 boje koje se mogu lagano odmah usporediti:

- zelena ocjena za povoljne recenzije,
- žuta ocjena za mješovite recenzije,
- crvena ocjena za nepovoljne recenzije.

General Meaning of Score	Movies, TV & Music	Games
Universal Acclaim	81 - 100	90 - 100
Generally Favorable Reviews	61 - 80	75 - 89
Mixed or Average Reviews	40 - 60	50 - 74
Generally Unfavorable Reviews	20 - 39	20 - 49
Overwhelming Dislike	0 - 19	0 - 19

Slika 7. [3] Prikaz boja Metascorea za određene medije

Metacritic označava film kao „*Must see*“ kada postigne Metascore od 81 ili više ima najmanje 15 recenzija. Filmovi „*Must see*“ vrlo su priznati i ocjenjeni sa širokim presjekom najboljih kritičara. Ukupno 5% filmova u Metacritic bazi podataka postiže taj status.

### 3. TEHNOLOGIJE KORIŠTENE U IZRADI WEB STRANICE

#### 3.1 HTML - Hypertext Markup Language

HTML (eng. *Hypertext Markup Language*) je standardni označni jezik za izradu web stranica. HTML semantički opisuje strukturu web stranice.

HTML se sastoji od niza kratkih kodova koje je autor stranice napisao u tekstualnu datoteku. Ti kodovi se nazivaju oznake. Tekst se zatim sprema kao HTML datoteka i pregledava se putem web preglednika poput Internet Explorer-a ili Google Chrome-a. Web preglednik čita datoteku i prevodi tekst u vidljivi oblik onako kako je autor stranice to namijenio. Pisanje vlastitog HTML-a znači ispravno korištenje oznaka kako bi se stvorila vizija. Za pisanje HTML-a može se koristiti sve od običnog editora teksta do moćnog grafičkog editora.

Oznake odvajaju obični tekst od HTML koda. Prepoznavaju se kao riječi između kutnih zagrada (<oznaka>). Različite oznake obavljaju različite funkcije. Neke oznake poput <input/> i <img/> izravno uvode sadržaj na stranicu. Ostale oznake poput <p> i <b> okružuju i pružaju informacije o tekstu dokumenta i mogu uključivati druge oznake kao pod elemente. Takve oznake obično imaju početni znak poput <p> i završni znak poput </p>. U početku HTML je sadržavao i prezentacijske oznake poput <font>, <center>, ali uvodom CSS-a (eng. *Cascading Style Sheets*) takve oznake se izbjegavaju te se prezentacijski dio obično radi u odvojenom css dokumentu. W3C (eng. *World Wide Web Consortium*) održava HTML i CSS standarde te potiče korištenje CSS-a umjesto eksplicitnog prezentacijskog HTML-a od 1997.

Na početku dokumenta se obično deklarira koja se verzija HTML-a koristi. Posljednja verzija HTML-a je HTML5 koji je objavljen 2014. Godine. Oznaka za HTML5 je <!DOCTYPE html>.

Oznaka <html> je element korijen HTML stranice. Oznaka <head> sadrži meta informacije o dokumentu. Oznaka <title> specificira naslov dokumenta. Oznaka <body> sadrži vidljivi dio sadržaja stranice.

## 3.2 CSS - Cascading Style Sheets

CSS (eng. *Cascading Style Sheets*) je stilski jezik koji se koristi za opisivanje prezentacije dokumenta napisanog označnim jezikom poput HTML-a.

CSS pomaže web programerima stvoriti jedinstveni izgled preko nekoliko stranica na web mjestu. Umjesto definiranja stila svake tablice i svakog bloka teksta unutar HTML stranice, često korišteni stilovi se mogu definirati jednom unutar CSS dokumenta. Kada se stil definira u CSS-u, može ga koristiti bilo koja stranica koja je povezana sa CSS dokumentom. Osim toga, CSS olakšava izmjenu stilova na više stranica odjednom. Npr. web programer može povećati zadanu veličinu teksta za više stranica samo izmjenom jedne linije koda u CSS dokumentu koji se odnosi na te stranice. CSS je također koristan za oblikovanje ostalih aspekata izgleda web stranice. Može se upotrijebiti za definiranje stila debljine i boje rubova HTML elemenata, visine i duljine elemenata, te kako se ti elementi odnose prema drugim elementima u svojoj okolini, itd. CSS daje web programerima točniju kontrolu nad izgledom web stranica od HTML-a.

CSS ima jednostavnu sintaksu. Koriste se brojne ključne riječi na engleskom jeziku kako bi se odredila imena različitih stilskih svojstava. Dokument se sastoji od popisa pravila. Svako pravilo sadrži jedan ili više selektora i blok za deklaraciju. Selektorima se određuje za koje elemente u HTML dokumentu vrijede ta pravila. Mogu se odnositi na određenu oznaku u HTML-u, poput `<p>`, ili elementi mogu biti definirani atributom *class* (klasa) ili *id*. Atribut *class* je identifikator koji može bilježiti više elemenata u HTML-u, dok je *id* jedinstveni identifikator za određeni element unutar dokumenta. Klase i *id*-ovi razlikuju velika i mala slova, počinju sa slovima, a mogu sadržavati i alfanumeričke znakove i podvlake.

Nasljeđivanje je ključna značajka u CSS-u. Nasljeđivanje je mehanizam kojim se svojstva primjenjuju ne samo na određeni element, već i na potomke. Elementi u potomstvu mogu naslijediti vrijednosti CSS-a od bilo kojeg elementa predaka koji ih okružuju. Općenito, elementi potomci nasljeđuju svojstva povezana s tekstom, ali svojstva koja se odnose na okvir nisu naslijeđena. Nasljeđivanje se može upotrebljavati kako bi se izbjeglo ponavljanje određenih svojstava, čime se skraćuje CSS dokument.

CSS ima različite razine i profile. Svaka razina CSS-a temelji se na posljednjoj, dodaje nove značajke i obično se označuje poput CSS1, CSS2, CSS3 i CSS4. Profili su obično podskup jedne ili više razina CSS-a izgrađeni za određeni uređaj ili korisničko sučelje. Svaki web preglednik na drukčiji način prikazuje web stranice, a podrška za CSS funkcionalnost nije konzistentna između njih.

## 3.3 JavaScript

JavaScript (JS) je interpretirani objektno orijentiran jezik i najbolje je poznat kao skriptni jezik za web stranice, ali koristi se i u drugim okruženjima bez web preglednika. To je prototipni skriptni jezik koji je dinamičan i podržava objektno orijentirane, imperativne i funkcionalne stilove programiranja.

JavaScript se pokreće na klijentskoj strani web-a i može se koristiti za dizajniranje načina na koji se web stranice ponašaju prilikom pojave događaja.

Osnovna sintaksa je namjerno slična Java i C++ programskim jezicima za smanjenje broja novih koncepata potrebnih za učenje jezika. Jezični konstrukti poput *if* izjava, *for* petlji, *while* petlji i *switch and try...catch* blokova funkcioniraju isto kao u tim jezicima.

JavaScript može funkcionirati kao proceduralni jezik i kao objektno-orijentirani jezik. Objekti se kreiraju programski u JavaScriptu, povezujući metode i svojstva na inače prazne objekte, u vrijeme izvođenja. Jednom kad se objekt konstruira može se koristiti kao nacrt za stvaranje sličnih objekata.

JavaScript sadrži „prvoklasne funkcije“ što znači da tretira funkcije kao prvoklasne građanine. To znači da jezik podržava slanje funkcija kao argumente, vraćanje funkcija kao vrijednost iz drugih funkcija, dodjeljivanje funkcija varijablama i spremanje funkcija u podatkovne strukture.

Primjeri korištenja JavaScripta na web stranici: dinamično mijenjanje CSS-a, dodavanje i premještanje sadržaja na stranici, validacija HTML formi, predaja podataka serveru pomoću AJAX-a bez ponovnog učitavanja stranice, stvaranje zvuka i videa, videoigre, itd.

Pošto se JavaScript odvija na klijentskom računalu, otvoren je za mnoge sigurnosne napade poput XSS (eng. *cross-site scripting*) i CSRF (eng. *cross-site request forgery*). Neki web preglednici nude djelomičnu zaštitu od tih napada, ali samo ispravan dizajn web aplikacija na serverskoj strani može potpuno zaustaviti te napade. Svi podaci poslani na server od strane JavaScript-e trebaju se ponovno provjeravati na serveru.

### 3.3.1 JQuery

JQuery je višepatformska JavaScript biblioteka dizajnirana da pojednostavi skriptiranje HTML-a na strani klijenta. To je besplatan, otvoreni softver i najšire korištena JavaScript biblioteka.

Sintaksa je dizajnirana kako bi se lakše kretalo dokumentom, dolazilo do HTML elemenata, stvarale animacije, obrađivali događaji i razvijale AJAX aplikacije.



JQuery knjižnica je jedna JavaScript datoteka koja sadrži sve zajedničke DOM-ove (eng. *Document Object Model*), događaje, efekte i AJAX funkcije. Može se uključiti u web stranicu povezivanjem na lokalnu kopiju ili na jednu od mnogobrojnih kopija dostupnih na javnim serverima. JQuery ima mrežu za isporuku sadržaja (eng. CDN - *content delivery network*).

### **3.3.2 AJAX - Asynchronous JavaScript And XML**

AJAX (eng. *Asynchronous JavaScript And XML*) je skup tehnika za web razvoj na strani klijenta pomoću kojih se stvaraju asinkrone web aplikacije. Pomoću AJAX-a, web aplikacije mogu slati i primiti podatke sa servera asinkrono bez ometanja prikaza i ponašanja postojeće stranice. AJAX dopušta dinamičko mijenjanje sadržaja web stranica bez potrebe za ponovnim učitavanjem cijele stranice. AJAX je varljivo ime. Iako AJAX aplikacije mogu koristiti XML za prijenos podataka, jednako uobičajen za prijenos podataka je korištenje običnog teksta ili JSON-a.

Redosljed izvođenja AJAX-a:

1. Na web stranici se dogodi event (npr. klik na tipku)
2. JavaScript-a kreira XMLHttpRequest objekt
3. XMLHttpRequest objekt šalje zahtjev na web server
4. Server obrađuje zahtjev
5. Server šalje odgovor nazad na web stranicu
6. JavaScript-a čita odgovor
7. JavaScript-a radi određene akcije ovisno o povratnim podacima

### 3.4 JSON – JavaScript Object Notation

JSON (eng. *JavaScript Object Notation*) je format razmjene podataka. Ljudima je jednostavan za čitati i pisati, a strojevima je jednostavan za analizirati i generirati. Temelji se na podskupu JavaScript jezika. JSON je oblik teksta koji je potpuno neovisan o jeziku, ali koristi konvencije koji su poznate programerima mnogih programerskih jezika. Ta značajka JSON-a ga čini idealnim formatom za razmjenu podataka.

JSON je građen na dvije strukture. Prva je kolekcija parova imena i vrijednosti. U raznim programskim jezicima to se realizira kao objekt, struktura, asocijativno polje, itd. Druga je poredana lista vrijednosti. U raznim programskim jezicima to se realizira kao polje, lista, vektor, itd.

Objekt nepoređan skup vrijednosti. Započinje s lijevom vitičastom zagradom, a završava s desnom vitičastom zagradom. Nakon svakog imena slijedi dvotočka nakon koje slijedi vrijednost. Parovi imena i vrijednosti su odvojeni zarezom. Polje je poredan skup vrijednosti. Počinje sa lijevom uglatom zagradom i završava s desnom uglatom zagradom. Vrijednosti su odvojene zarezom. Vrijednost može biti string, broj, polje, null, *true* ili *false*, čak i drugi JSON objekt.

```
{  
  "Title": "Batman Begins",  
  "Year": "2005",  
  "imdbID": "tt0372784",  
  "Type": "movie"  
}
```

Primjer 1. JSON objekt

```
{  
  "Search": [  
    {  
      "Title": "Batman Begins",  
      "Year": "2005",  
      "imdbID": "tt0372784",  
      "Type": "movie",  
    },  
    {  
      "Title": "Batman v Superman: Dawn of Justice",  
      "Year": "2016",  
      "imdbID": "tt2975590",  
      "Type": "movie",  
    }  
  ]  
}
```

Primjer 2. JSON objekt koji sadrži polje JSON objekata:

### 3.5 PHP - Hypertext Preprocessor

PHP (rekurzivni akronim za eng. *PHP: Hypertext Preprocessor*) je skriptni jezik na serverskoj strani koji je dizajniran za razvoj web stranica, ali se također koristi kao programski jezik opće namjene.

PHP stranice mogu sadržavati HTML s ugrađenim kodom koji radi „nešto“. PHP kod nalazi se između posebnih uputa za početak (<?php) i kraj (?>) obrade koji omogućuju skok u i izlaz izvan PHP okvira.

PHP se razlikuje od JavaScripte, koja radi na klijentskoj strani, po tome što se kod izvršava na serveru. Generira se HTML koji se zatim šalje klijentu. Klijent prima rezultate skripte, ali ne zna temeljni kod koji se izveo.

U izradi web stranica PHP dokument se najviše koristi kao filter koji prima podatke iz jednog izvora te ih obrađuje te vraća neke druge podatke. PHP se obično koristi za spajanje na bazu podataka koja je vezana za web stranicu na kojoj se nalazimo. Podaci koje PHP skripta prima moraju se najprije obraditi prije spajanja na bazu podataka i izvršavanja SQL koda kako bi se zaštitila baza podataka. Izvršavanjem SQL koda iz baze se mogu dobiti podaci, te se sadržaj baze može izmijeniti. Izlazni podaci su najčešće HTML, JSON ili XML. Ako je PHP skripta pozvana od strane AJAX funkcije izlazni podaci se šalju nazad u JavaScript-u koja onda ovisno o primljenim podacima određuje kako će ti podaci biti prikazani. U kombinaciji sa AJAX-om PHP omogućuje dinamično mijenjanje sadržaja web stranice bez ponovnog učitavanja.

Sintaksa PHP jezika je jednostavna i nalikuje na C programski jezik. PHP podržava sve osnovne elemente poput varijabli, objekta, funkcija, itd.

### 3.6 SQL i relacijska baza podataka

Neke podatke na web stranici želimo sačuvati i ponovno koristiti, mijenjati ili čak brisati. Ako korisnicima web stranice želimo omogućiti da utječu na sadržaj stranice, te podatke moramo moći negdje spremiti. Takvi podaci se spremaju na relacijsku bazu podataka koja se nalazi na serveru. RDBMS (eng. *relational database management system*) je naziv za programsku podršku za upravljanje relacijskom bazom podataka. Podaci unutar baze su organizirani unutar više tablica koje sadrže stupce i retke. Za svaki redak u tablici postoji jedinstveni ključ koji ga identificira.

Tablice u bazi podataka su poput klasa u objektno-orijentiranom programiranju, dok se jedan redak može usporediti sa objektom ili instancom te klase. Stupci predstavljaju vrijednosti atributa te instance. Atributi mogu biti i strani ključevi preko kojih se mogu spojiti druge tablice koje mogu služiti za detaljniji opis te instance.

SQL (eng. *Structured Query Language*) je programski jezik dizajniran za upravljanje podacima koji se nalaze unutar relacijske baze podataka. Korištenje SQL-a uključuje modificiranje tablice baze podataka i strukture indeksa, dodavanje, ažuriranje i brisanje redaka podataka, dohvaćanje podskupova informacija iz baze podataka. Upiti i ostale SQL operacije imaju oblik naredbi napisanih kao izjave. Obično korištene SQL izjave su: SELECT, INSERT, UPDATE, DELETE, CREATE, itd.

Za spajanje na bazu i slanje SQL upita koristit ćemo PHP jezik, a relacijska baza je MySQL.

### 3.7 OMDb API – Open Movie Database

U programiranju API (eng. *Application Programming Interface*) je skup podprogramskih definicija, protokola i alata za izgradnju softvera. To je skup jasno definiranih metoda komunikacije između različitih komponenti. API olakšava razvoj računalnog programa pružanjem građevnih blokova koje programer zatim sastavlja.

REST (eng. *Representational State Transfer*) je arhitektonski stil za dizajniranje umreženih aplikacija. Temelji se na HTTP-u (eng. *The Hypertext Transfer Protocol*). Web usluge koje su „RESTful“ osiguravaju interoperabilnost između računalnih sustava na internetu.

OMDb (eng. *Open Movie Database*) API je RESTful web usluga za dobivanje informacija o filmovima i TV serijama. Svi sadržaji na web-u pridonose i održavaju korisnici usluge. Kako bi se pristupilo usluzi potreban je API ključ koji se nadodaje na URL kod svih zahtjeva.

HTTP zahtjev izgleda ovako: [http://www.omdbapi.com/?apikey=\[mojključ\]&](http://www.omdbapi.com/?apikey=[mojključ]&)

Nakon znaka & definiraju se ostali parametri kako bi se specificirao zahtjev. API kao odgovor vraća JSON datoteku koja sadrži podatke ovisno o parametrima koji su poslani. Svaka JSON datoteka koju dobijemo sadržavat će atribut „Response“ čija vrijednost može biti *false* ili *true*. Ako je vrijednost *false* API nije pronašao ništa za date parametre.

Zahtjeve šaljem iz JavaScript-e pomoću JQuery funkcije \$.getJSON(), što je skraćena AJAX funkcija.

```
$.getJSON( "http://www.omdbapi.com/?apikey=[mojključ]&s=batman", function(data) {  
    /* Obrada podataka */  
});
```

Primjer 3. Zahtjev sa parametrom s (*search*)

Dobiveni podaci se spremaju u JavaScript objekt te se onda naknadno obrađuju i prikazuju na web stranici.

## 4. PROJEKTIRANJE I DIZAJN

Prije nego što se krene u izradu sustava potrebno je izraditi tehničke specifikacije. Svaki projekt započinje sakupljanjem korisničkih zahtjeva. Nakon analize zahtjeva potrebno je izgraditi grubi model sustava koji će nam pomoći pri izboru tehnologija koje ćemo koristiti u izradi sustava. Nakon izbora tehnologija potrebno je prikazati modele glavnih funkcionalnosti sustava te kako će se izabrane tehnologije koristiti da se oni ostvare.

### 4.1 Zahtjevi

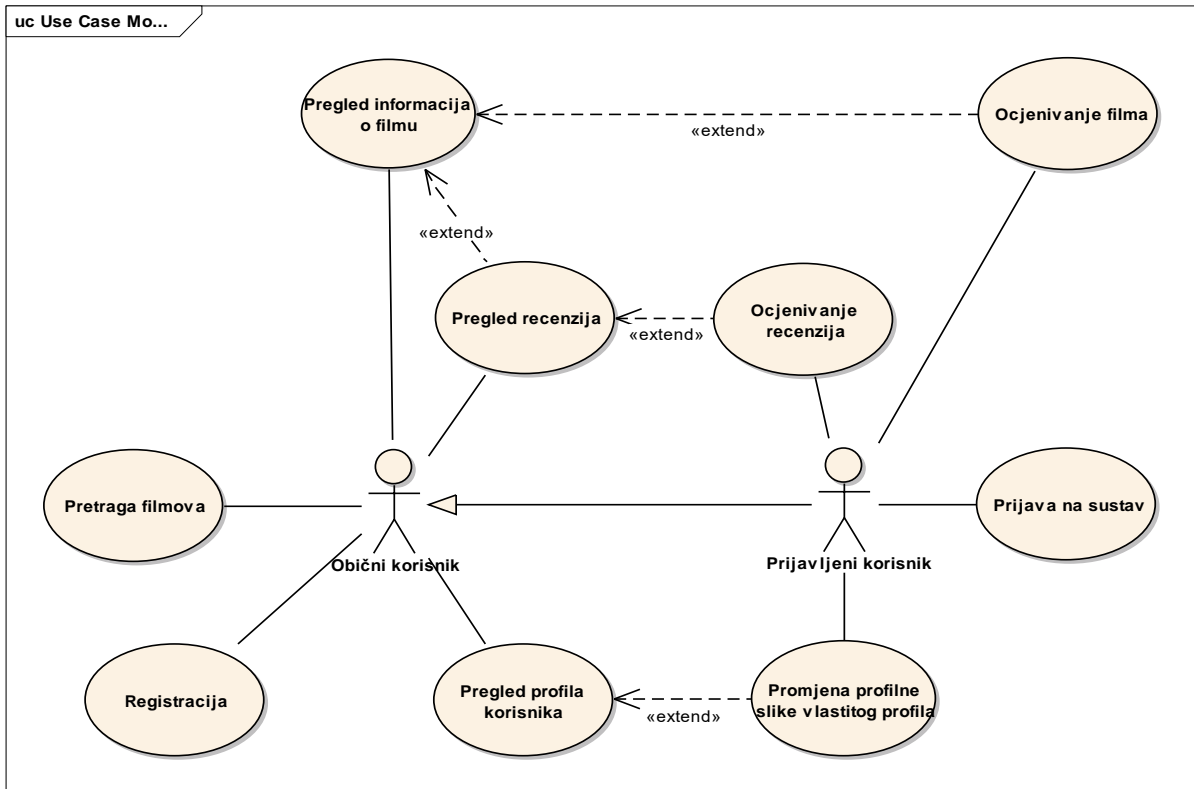
Zahtjevi su uvjeti i sposobnosti koje sustav mora posjedovati kako bi zadovoljio specifikacije, standard i kako bi omogućio korisniku da ostvari svoj cilj. Ne sadrže detalje dizajna, ali određuju što sve sustav mora imati kako bi bio potpun.

#### 4.1.1 Funkcionalni zahtjevi

Potrebno je izraditi web sjedište za ocjenjivanje i recenziju filmova. Web stranica treba omogućiti pretragu filmova i pregled svih dostupnih informacija o filmovima. Korisnici se trebaju moći registrirati i prijaviti na sustav. Prijavom na sustav dobivaju mogućnost ocjenjivanja filmova i ostavljanja recenzija. Svaki registrirani korisnik mora imati jedinstveno korisničko ime i adresu elektroničke pošte. Osim ocjenjivanja filmova korisnici trebaju moći ocijeniti recenzije drugih korisnika. Prilikom pregleda recenzija za neki film, najprije će se pokazivati one s najboljim ocjenama. Svaki korisnik posjeduje svoj profil gdje mogu promijeniti svoju profilnu sliku. Na profilu se mogu vidjeti statistički podaci o korisniku te sve dosadašnje filmove koje je korisnik ocijenio.

#### 4.1.2 Nefunkcionalni zahtjevi

Potrebno je imati bazu podataka koja će sadržavati sve podatke o korisnicima i podacima koje unose na web stranicu. Prilikom registracije treba onemogućiti korisnika da napravi račun sa korisničkim imenom i adresom elektroničke pošte koja već postoji u bazi. U koliko korisnik zaboravi svoju lozinku treba postojati način da je povрати. Bazu podataka treba zaštititi od raznih napada poput SQL umetanja (eng. *SQL injection*) i XSS-a (eng. *Cross-site scripting*). Korisnike treba spriječiti da neovlašteno mijenjaju podatke unutar baze podataka. Ukoliko dođe do pogreške na web stranici na to treba upozoriti korisnika porukom. Svaki događaj u sustavu, čak i ako je neuspješan, treba korisniku dati povratnu poruku.



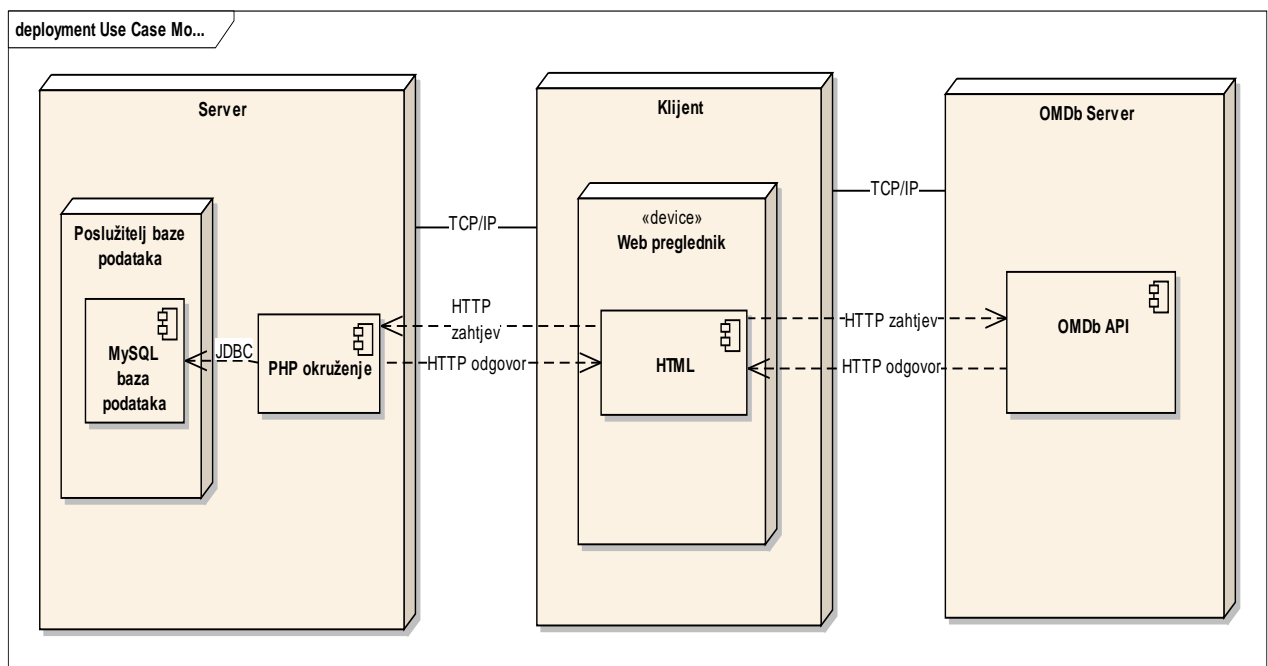
Slika 8. Dijagram slučajeva korištenja

## 4.2 Arhitektura sustava

Svaki složeni sustav sastavljen je od povezanih podsustava koji međusobno surađuju. Arhitektura sustava je dizajn programske podrške koji daje vrlo grub opis rješenja.

### 4.2.1 Fizička arhitektura sustava

Web aplikacija je na serveru. Klijent preko web preglednika na svom računalu pristupa sučelju web aplikacije koristeći HTTP protokol. PHP okruženje se koristi za pristup bazi podataka. Zahtjevi prema OMDb API-u se šalju koristeći HTTP protokol.



Slika 9. Dijagram razmještaja – fizička arhitektura

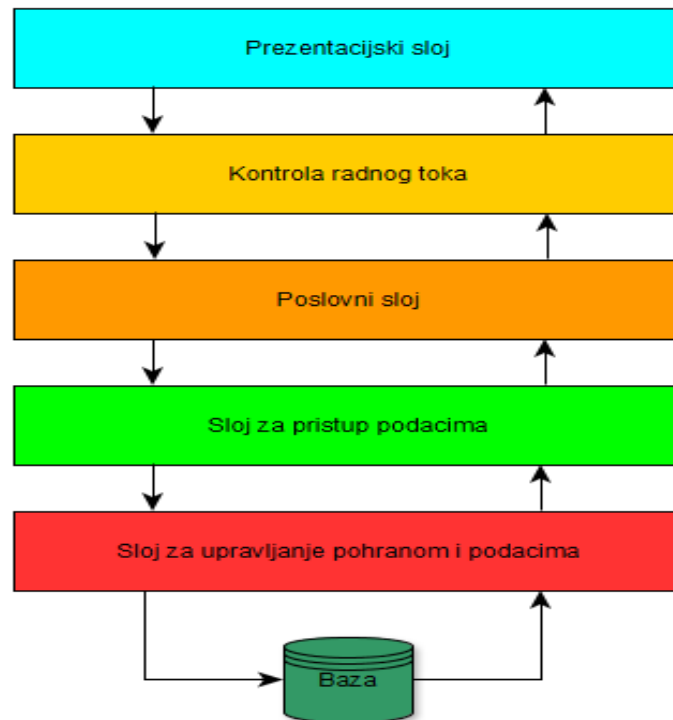
### 4.2.2 Logička arhitektura sustava

U izradi web aplikacije koristit će se petoslojna arhitektura. Ona se sastoji od 5 dijelova:

1. Korisničko sučelje (eng. *User interface*) ili prezentacijski sloj (eng. *Presentation layer*) služi za prikaz informacija i unos od strane korisnika.
2. Kontrola sučelja (eng. *Interface control*) ili kontrola radnog toka (eng. *Workflow*) služi za generiranje izlaza te interpretaciju i obradu ulaza. Ovdje se prihvaća unos korisnika te se prosljeđuje na sljedeći sloj na evidenciju i obradu.
3. Poslovni sloj (eng. *Buisness logic layer*) uključuje poslovna ograničenja, određuje validacijska pravila, radi autentikaciju.



4. Sloj za pristup podacima (eng. *Data access layer*) surađuje sa slojem za pohranu i upravljanje podacima. Ovaj sloj dohvaća, ažurira, umeće i briše podatke, ali ne upravlja i ne pohranjuje.
5. Sloj za upravljanje pohranom i upravljanje podacima (eng. *Data storage and management layer*) bavi se fizičkim dohvatom, ažuriranjem, stvaranjem i brisanjem podataka.



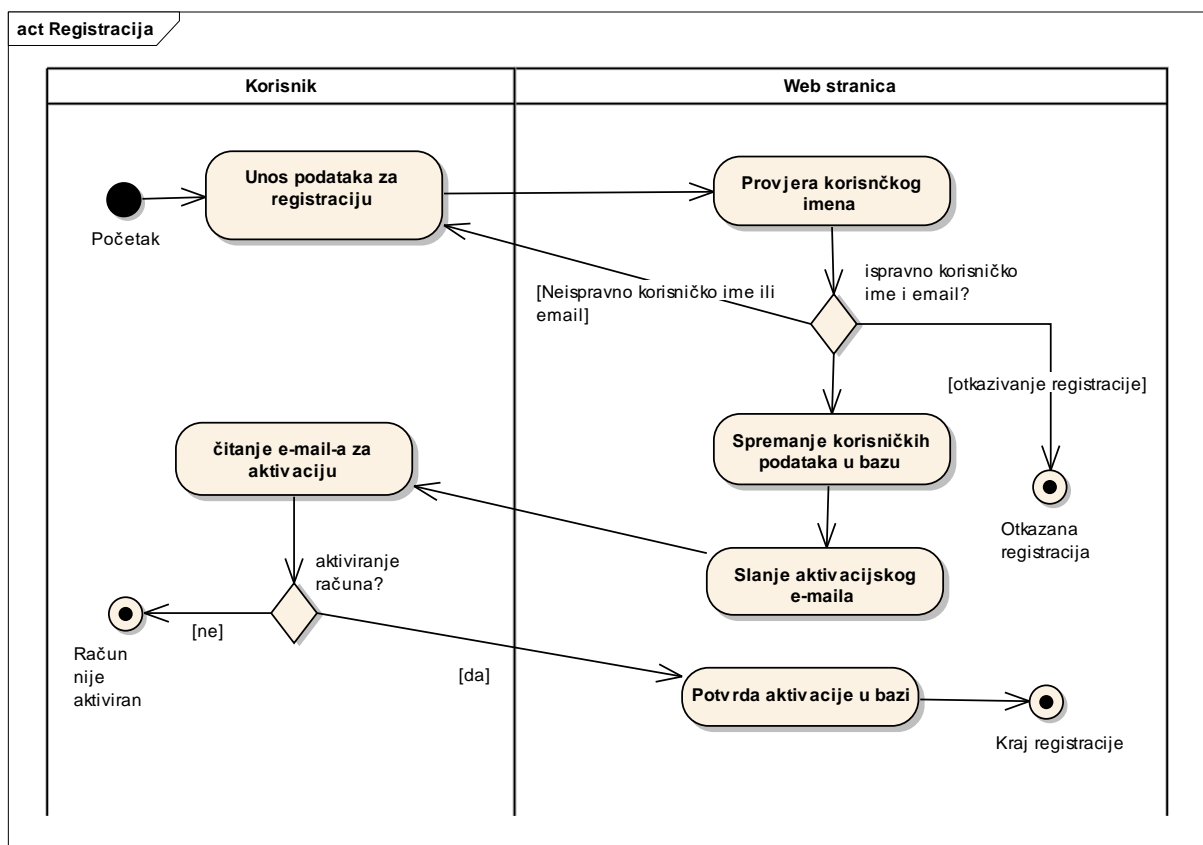
Slika 10. Petoslojna arhitektura

## 4.3 Dizajn sustava

Dizajn sustava započinje na osnovi grube arhitekture sustava i korisničkih zahtjeva. Cilj je izraditi nacrt za izradu rješenja. Potrebno je identificirati glavne funkcionalnosti sustava te ih detaljnije razraditi.

### 4.3.1 Registracija na web stranici

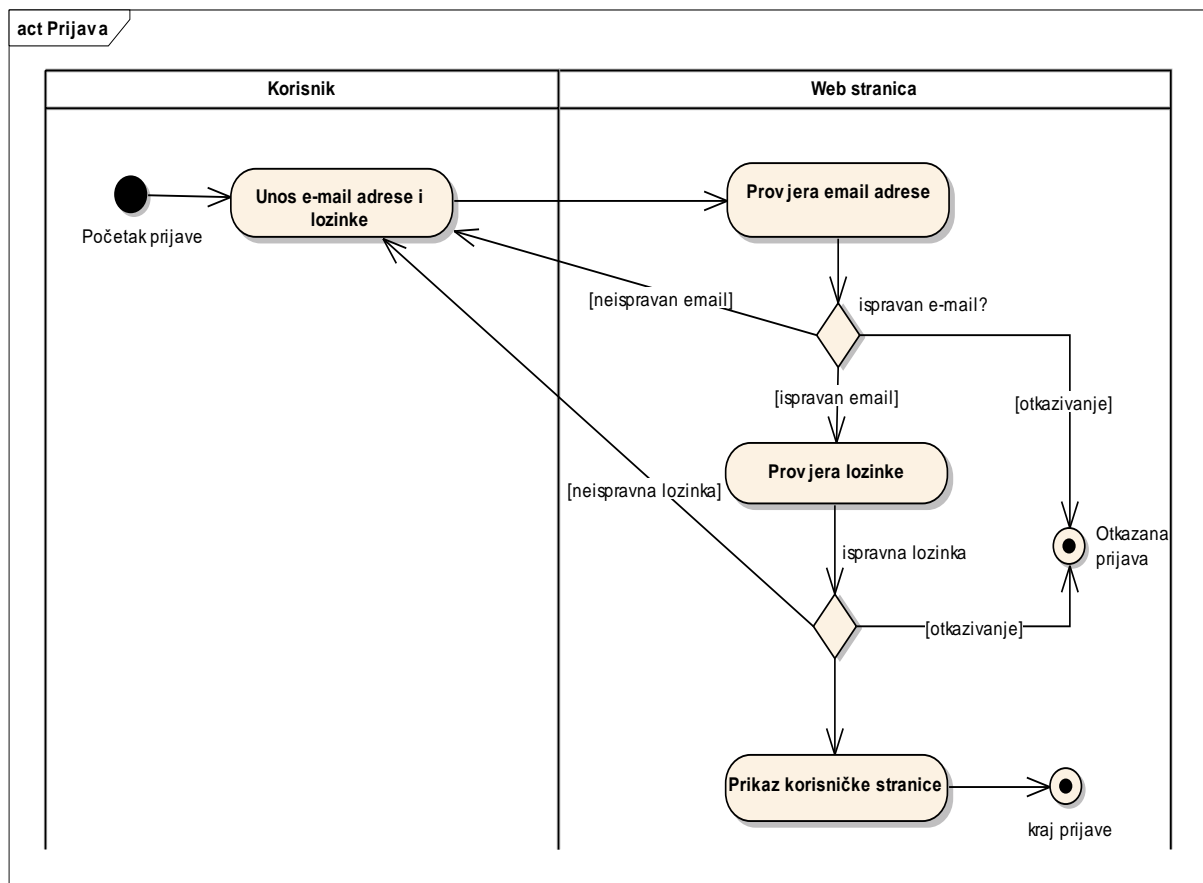
Registracija započinje unosom podataka za registraciju. Podaci za registraciju su adresa elektroničke pošte, korisničko ime i lozinka. Nakon unosa slijedi provjera podataka. Provjerava se da li korisničko ime i adresa već postoje unutar baze. Ako postoje ili ako su podaci neispravnog tipa, pogreška se javlja korisniku koji može odustati od registracije ili pokušati registraciju sa različitim korisničkim imenom ili adresom. Ako podaci ne postoje unutar baze podataka, spremaju se u bazu te se šalje korisniku obavijest o registraciji koja sadrži poveznicu za aktivaciju računa. Korisnik dobiva elektroničku poštu. U slučaju da korisnik te adrese nije dao zahtjev za registracijom može ignorirati poruku i račun neće biti aktiviran. Klikom na poveznicu koja se nalazi u poruci korisnik aktivira svoj račun na web stranici. U bazu se sprema informacija da je račun aktiviran i time je registracija završena.



Slika 11. Registracija na web stranici – dijagram aktivnosti

### 4.3.2 Prijava na web stranicu

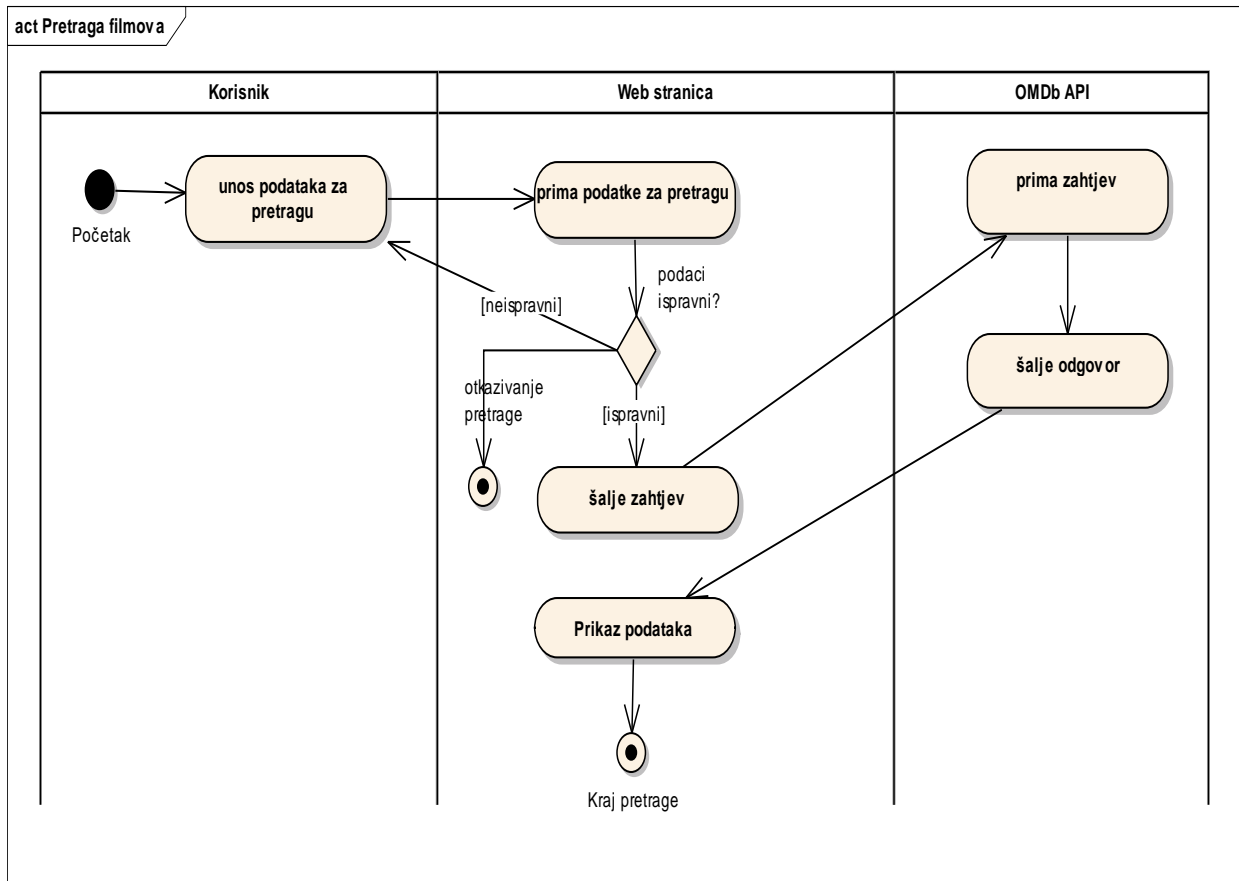
Prijava započinje unosom adrese elektroničke pošte i lozinke korisnika. Provjerava se da li adresa ispravna. Ako je neispravna, pogreška se javlja korisniku koji može odustati od prijave ili promijeniti svoj unos pa pokušati ponovno. Ako je adresa ispravna provjerava se da li je za unesenu adresu data ispravna lozinka. Ako je neispravna pogreška se javlja korisniku koji može odustati od prijave ili promijeniti svoj unos pa pokušati ponovno. Ako je lozinka ispravna, prijava je uspješna te se prikazuje web stranica. Korisnik može započeti svoje aktivnosti na stranici.



Slika 12. Prijava na web stranicu – dijagram aktivnosti

### 4.3.3 Pretraga filmova

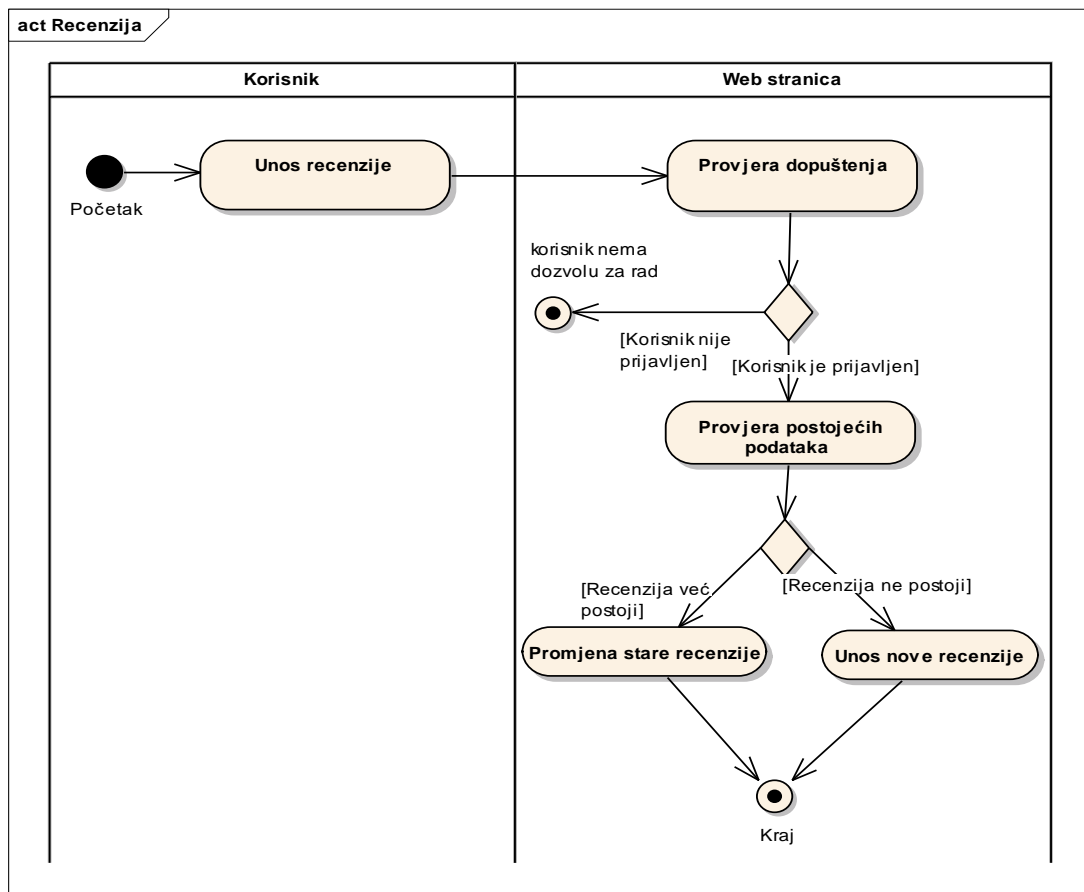
Pretraga započinje unosom podataka za pretragu. Web stranica prima podatke i provjerava jesu li ispravni. Ako nisu ispravni korisniku se javlja pogreška. Korisnik može odustati od pretrage ili unijeti ispravne podatke. Ako su podaci ispravni šalje se zahtjev na API server. API server vraća odgovor. Na temelju tog odgovora web stranica prikazuje podatke. Ako nisu dobivene nikakve informacije o filmovima web stranica će obavijestiti korisnika. Time je završena pretraga.



Slika 13. Pretraga filmova – dijagram aktivnosti

#### 4.3.4 Recenzija filma

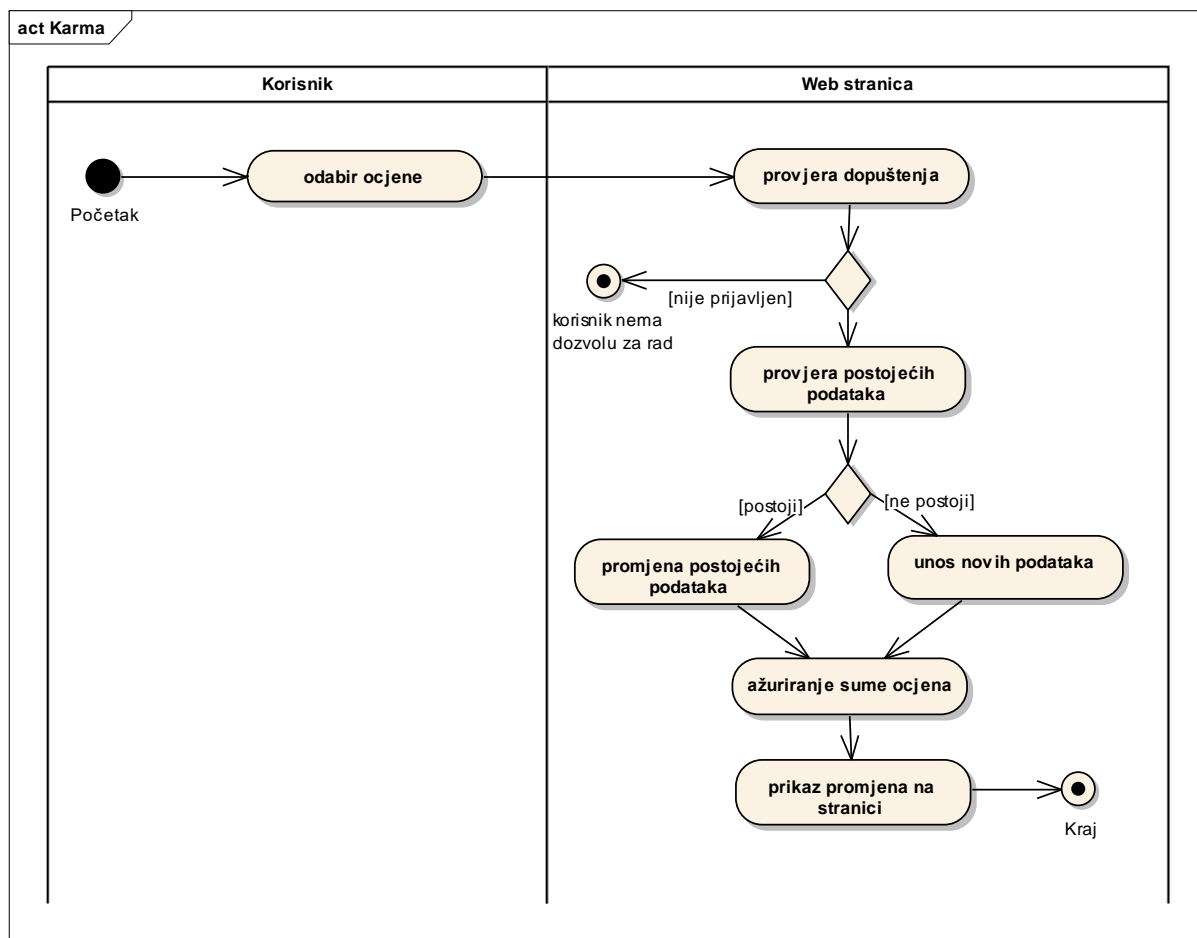
Korisnik bira ocjenu od 1 do 10 te unosi svoju recenziju za film. Podaci se šalju na web stranicu. Provjerava se da li korisnik ima dopuštenje za ostavljanje recenzija. Ako nema, rad se završava te se javlja pogreška korisniku. Ako je korisnik prijavljen, provjerava se da li je taj korisnik već ostavio recenziju za ovaj film. Ako jest, stara recenzija se zamjenjuje sa podacima nove. Ako nije, unose se novi podaci u bazu podataka.



Slika 14. Recenzija filma – dijagram aktivnosti

### 4.3.5 Ocjena recenzija drugih korisnika

Ocjena recenzije započinje odabirom ocjene. Podaci se šalju na web stranicu. Provjerava se da li korisnik ima dozvolu za ocjenu recenzije. Ako nema, rad se završava bez promjena stanja u bazi podataka. Ako je korisnik prijavljen na sustav. Provjerava se da li je već ocijenio tu recenziju. Ukoliko jest ažurira se baza podataka sa novom ocjenom. Ukoliko je korisnik ponovno odabrao istu ocjenu, poništava se ocjena kao da korisnik nikada nije glasao. Ako korisnik nije prije ocijenio tu recenziju unosi se nova ocjena u bazu podataka. Nakon toga ažurira se suma svih ocjena za tu recenziju. Kada završi ažuriranje baze podataka, na web stranici se korisniku prikazuju promjene.



Slika 15. Ocjena recenzije – dijagram aktivnosti

## 5. IZRADA SUSTAVA

Izrada sustava uključuje implementaciju i testiranje dijelova sustava na osnovnu izlaznih rezultata faze dizajna. U ovoj poglavlju će se opisati na koji način se primjenjuje tehnologija koja izabrana za razvoj sustava.

### 5.1 Prezentacijskih sloj

Za izgled web stranice koristi se Bootstrap. Bootstrap je besplatan otvoreni alat za izradu web stranica i web aplikacija. Sadrži HTML i CSS predloške za tipografiju, obrasce, gumbe, navigaciju i ostale komponente sučelja. Bootstrap također sadrži dodatna JavaScript proširenja.

Implementacija Bootstrap biblioteke je jednostavna. U zaglavlje dokumenta za koji se koristi Bootstrap kopiraju se poveznice.

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIpG9mGCD8wGNiCPD7Txa" crossorigin="anonymous"></script>
```

Slika 16. Poveznica za Bootstrap CSS i JavaScript biblioteku

Nakon što je Bootstrap biblioteka spojena mogu se definirati Bootstrap klase na HTML elementima.

```
<nav class="navbar navbar-inverse navbar-fixed-top"></nav>
```

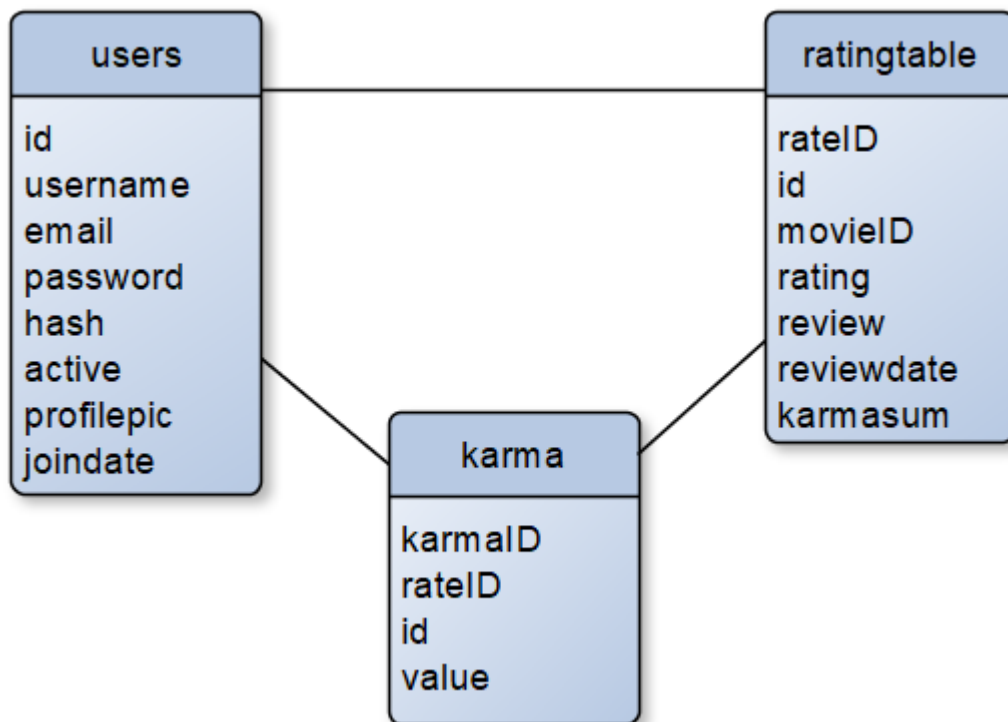
Primjer 4. Definiranje Bootstrap klase za HTML `<nav>` element

Kako bi se promjenio Bootstrap predložak dovoljno je promijeniti ime klase, naravno koristeći definirana imena i pravila koja su sadržana u biblioteci.

## 5.2 Implementacija baze podataka

DBMS se koristi je MySQL. Za lakši pristup i administraciju MySQL-a koristi se phpMyAdmin. PhpMyAdmin je besplatan otvoreni alat koji omogućuje korisnicima pristup MySQL i MariaDB bazama podataka. Napisan je prvenstveno u PHP-u i jedan je od najpopularnijih alata za administraciju.

Naziv baze podataka je moviedatabase. Baza se sastoji od tri tablice: users, ratingtable i karma.



Slika 17. moviedatabase

U tablicu *users* spremaju se podaci o korisnicima. Atribut *id* je jedinstvena brojčana oznaka koja se automatski generira u tablici za svaki novi dodani redak. Atributi *username* i *email* su također jedinstveni. Atribut *password* je šifrirana lozinka korisnika. Atribut *hash* služi za generiranje poveznice za aktivaciju računa i poveznice za resetiranje lozinke. Atribut *active* može imati dvije vrijednosti: 0 i 1. Taj atribut će uvijek biti 0 dok se korisnički račun ne aktivira. Atribut *profilepic* je poveznica na resurs na serveru gdje se nalazi profilna slika korisnika. Atribut *joindate* predstavlja datum kada se korisnik registrirao te se automatski generira prilikom registracije.

U tablicu *ratingtable* spremaju se podaci o recenzijama korisnika. Atribut *rateID* je jedinstvena brojčana oznaka koja se automatski generira u tablici za svaki novi dodani redak. Atribut *id* je strani ključ s kojim se tablica *ratingtable* može povezati s tablicom *users*. Atribut *movieID* je identifikacijska oznaka za film preko koje se dolazi do informacija o filmu



koristeći OMDb API. Atribut *rating* predstavlja ocjenu filma koja može biti od 1 do 10. Atribut *review* je tekstualna recenzija koja nije nužna za ocjenu filma. Atribut *reviewdate* predstavlja datum kad je recenzija napravljena. Atribut *karmasum* je suma svih ocjena recenzije. Recenzije se prilikom prikazivanja na web stranici prikazuju počevši od onih sa najvećom sumom. Prilikom stvaranja recenzije postavlja se na 0 te se automatski ažurira sa svakom promjenom u tablici karma.

U tablicu *karma* spremaju se podaci o ocjenama pojedinačnih recenzija od strane korisnika. Atribut *karmaID* je jedinstvena brojana oznaka koja se automatski generira u tablici za svaki novi dodani redak. Atribut *rateID* je strani ključ kojim se tablica *karma* može povezati s tablicom *ratingtable*. Atribut *id* je strani ključ kojim se tablica *karma* može povezati s tablicom *users*. Atribut *value* predstavlja ocjenu kojom je korisnik ocijenio određenu recenziju. Može imati tri različite vrijednosti: -1, 0 i 1. Suma svih vrijednosti u tablici karma za određenu recenziju identificiranu atributom *rateID* daje vrijednost atributa *karmasum* u tablici *ratingtable*.

## 5.3 Implementacija funkcionalnosti

### 5.3.1 Registracija na sustav

Registracija se izvodi sa obrasca kojemu se pristupa preko navigacijske trake na web stranici. Metoda slanja podataka je POST, a podaci se šalju na dokument „loginform.php“. Obrazac sadrži 3 polja za unos podataka i tipku za predaju podataka. U prvom polju unosi se korisničko ime, u drugom adresa elektroničke pošte korisnika, a u trećem lozinka koju će korisnik koristiti za prijavu na sustav. Sva tri polja je potrebno popuniti kako bi se registracija izvršila.

```
<form action="loginform.php" method="post" autocomplete="off">
<input type="text" required autocomplete="off" name='username'
placeholder="Username"/>
<input type="email" required autocomplete="off" name='email' placeholder="Email
Address"/>
<input type="password" required autocomplete="off" name='password'
placeholder="Password"/>
<button type="submit" class="button button-block" name="register" />Register</button>
</form>
```

#### Primjer 5. HTML kod - Obrazac za registraciju

Na početku dokumenta „loginform.php“ provjerava se da li je metoda slanja POST i preusmjerava se na drugi dokument ovisno o podatku koji je poslan, *register* ili *login*.

```
if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    if(isset($_POST['login'])){
        require ('login.php');
    }
    elseif (isset($_POST['register'])){
        require ('register.php');
    }
}
```

#### Primjer 6. PHP kod – provjera prije preusmjeravanja

PHP skripta „register.php“ prima podatke preko POST metode. Podaci se predaju funkciji *filter\_input* koja služi kao zaštita od izvršavanja napadačkog koda (XSS). Lozinka se šifrira koristeći funkciju *password\_hash*. Zatim se pomoću funkcije *md5* generira nasumični hash koji će služiti za aktivaciju računa nakon registracije.

```

$username=filter_input(INPUT_POST, 'username',FILTER_SANITIZE_SPECIAL_CHARS);
$email=filter_input(INPUT_POST, 'email',FILTER_SANITIZE_SPECIAL_CHARS);
$password=filter_input(INPUT_POST, 'password',FILTER_SANITIZE_SPECIAL_CHARS);
$password=password_hash($password, PASSWORD_BCRYPT);

$hash = md5( rand(0,1000));

```

#### Primjer 7. PHP kod – definicija varijabli za registraciju

Nakon primanja podataka provjerava se da li dobiveno korisničko ime i adresa već postoje u bazi podataka. Ako postoje, PHP skripta se preusmjerava nazad na dokument „loginform.php“ gdje se korisniku ispisuje odgovarajuća poruka koja se spremljena u sesijsku varijablu `$_SESSION['message']`. Funkcija `mysqli_real_escape_string` služi kao zaštita od napada umetanjem SQL-a.

```

$sql=sprintf("SELECT * FROM users WHERE email= '%s' OR
username='%s'",mysqli_real_escape_string($con,$email),mysqli_real_escape_string($con,$u
sername));

$result = $con->query($sql) or die($con->error());

if( $result->num_rows >0){
    $_SESSION['message'] = 'Username or email is already taken!';
    header("location: loginform.php?t=signup");
    mysqli_close($con);
}

```

#### Primjer 8. PHP kod – stvaranje SQL upita i provjera u bazi podataka

Ako podaci ne postoje unutar baze podataka, kreira se novi SQL kod s kojim se u bazi umeću korisnički podaci. Funkcija `CURDATE()` vraća trenutni datum.

```

$sql=sprintf("INSERT INTO users (username, email, password, hash, joindate)
VALUES('%s','%s','%s','%s',CURDATE())",mysqli_real_escape_string($con,$username),mys
qli_real_escape_string($con,$email),mysqli_real_escape_string($con,$password),mysqli_rea
l_escape_string($con,$hash));

```

#### Primjer 9. PHP kod – stvaranje SQL-a za umetanje podataka

Ako umetanje podataka u bazu završi uspješno, postavljaju se sesijske varijable koje pokazuju trenutno stanje korisničke prijave. Pomoću varijable `$_SESSION['active']` se utvrđuje da li je korisnički račun aktiviran, a pomoću varijable `$_SESSION['logged_in']` se utvrđuje da li je korisnik prijavljen na sustav. Stvara se poruka za aktivaciju koja će poslati na adresu elektroničke pošte korisnika. Ako je poruka uspješno poslana skripta se preusmjerava na dokument „success.php“, gdje se informira korisnika o uspješnoj registraciji i potrebi da se račun aktivira preko poveznice poslana na adresu elektroničke pošte.

```

$_SESSION['active'] = 0;
$_SESSION['logged_in'] = true;
$_SESSION['message'] = "Confirmation link has been sent to " . $email . ", please verify your
account by clicking on the link in the message!";
$to=$email;
$subject= 'Account Verification';
$message_body= '
Hello' . $username.'
Please click this link to activate your account:
http://duithr1.ipage.com/drozer/verify.php?email='.$email.'&hash='.$hash;
if(mail($to, $subject, $message_body))
{
    mysqli_close($con);
    header("location: success.php");
}

```

Primjer 10. PHP kod – postavljanje sesijskih varijabli i slanje elektroničke pošte

Pritiskom na poveznicu dobivenu na adresu elektroničke pošte, korisnik dolazi do PHP skripte „verify.php“. Ovdje se prvo provjerava da li su potrebni parametri postavljeni. Ako jesu, u bazi podataka se provjerava da li su primljeni podaci ispravni. Ako SQL upit ne dobije rezultate, stvara se odgovarajuća poruka te se preusmjerava na dokument „error.php“.

```

$sql=sprintf("SELECT * FROM users WHERE email='%s' AND hash='%s' AND
active=0",mysqli_real_escape_string($con,$email),mysqli_real_escape_string($con,$hash));

if($result->num_rows==0)
{
    $_SESSION['message'] = "Account has already been activated or the URL is invalid!";
    mysqli_close($con);
    header("location: error.php");
}

```

Primjer 11. PHP kod – stvaranje SQL upita i provjera rezultata

Ako SQL upit vrati rezultate, stvara se odgovarajuća poruka, stvara se SQL kojim se ažurira baza podataka, mijenja se vrijednost sesijske varijable \$\_SESSION['active'] te se preusmjerava korisnika na dokument „success.php“.

```

$_SESSION['message'] = "Your account has been activated!";
$sql=sprintf("UPDATE users SET active='1' WHERE email = '%s'",
mysqli_real_escape_string($con,$email));
$con->query($sql) or die($con->error);
$_SESSION['active']= 1;
mysqli_close($con);
header("location: success.php");

```

Primjer 12. PHP kod – ažuriranje baze podataka

### 5.3.2 Prijava na sustav

Prijava se izvodi sa obrasca kojemu se pristupa preko navigacijske trake na web stranici. Metoda slanja je POST, a podaci se šalju na dokument „loginform.php“. Obrazac sadži 2 polja za unos podataka i tipku za predaju podataka. U prvom polju unosi se adresa elektroničke pošte korisnika, a u drugom lozinka. Oba polje je potrebno popuniti kako bi se prijava izvršila.

Na početku dokumenta „loginform.php“ provjerava se da li je metoda slanja POST i preusmjerava se na drugi dokument ovisno o podatku koji je poslan, *register* ili *login*. PHP skripta „login.php“ prima podatke. Generira se SQL upit koji dohvaća sve podatke o korisniku s tom adresom. Ako SQL upit ne vrati rezultat, odgovarajuća poruka se prikazuje korisniku.

```
$email = filter_input(INPUT_POST, 'email', FILTER_SANITIZE_SPECIAL_CHARS);
$sql=sprintf("SELECT * FROM users WHERE email='%s'",
mysqli_real_escape_string($con,$email));

if($result->num_rows == 0){

    $_SESSION['message']= "User with that email doesn't exist!";
    mysqli_close($con);
    header("Location: loginform.php");
}
```

#### Primjer 13. PHP kod – stvaranje SQL upita i provjera rezultata

Ako SQL upit dobije rezultat, rezultat se sprema u varijablu *\$user*, lozinka se provjerava pomoću funkcije *password\_verify*, postavljaju se odgovarajuće sesijske varijable te se korisnika preusmjerava na početnu web stranicu. Ako je lozinka neispravna, korisniku se ispisuje odgovarajuća poruka.

```
$user = $result->fetch_assoc();
if(password_verify($_POST['password'], $user['password'])){
    $_SESSION['email'] = $user['email'];
    $_SESSION['username'] = $user['username'];
    $_SESSION['active'] = $user['active'];
    $_SESSION['id'] = $user['id'];
    $_SESSION['logged_in'] = true;
    mysqli_close($con);
    header("Location: index.php");
}else{
    $_SESSION['message'] = "You have entered wrong password, try again!";
    mysqli_close($con);
    header("Location: error.php");
}
```

#### Primjer 14. PHP kod – provjera lozinke

### 5.3.3 Pretraga filmova

Pretraga filmova se izvodi sa obrasca koja se nalazi na navigacijskoj traci na web stranici. Metoda slanja podataka je GET, a podaci se šalju na dokument „index.php“. Obrazac sadrži 2 polja za unos podataka, jedno skriveno polje i tipku za pretragu. Prvi unos je za naziv filma koji se traži, a drugi unos je odabir tipa podatka koji se traži i može biti: *all*, *movie* ili *series*. U skriveno polje se sprema broj stranice, čija je zadana vrijednost 1.

```
<select id="select2" class="form-control" name="type">
  <option selected value="all">All</option>
  <option value="movie">Movies</option>
  <option value="series">Series</option>
</select>
```

#### Primjer 15. HTML kod – odabir tipa podataka

Pritiskom na tipku za pretragu, web stranica preusmjerava na dokument „index.php“ gdje se provjerava da li su postavljene GET vrijednosti za pretragu, ako jesu poziva se JavaScript funkcija „*searchMethod(search, page, type)*“ koja prima tri argumenta: naziv filma, stranicu i tip. Prvo se provjerava da li su dani podaci ispravni pa se onda šalje zahtjev na OMDb API.

```
$.getJSON('
http://www.omdbapi.com/?apikey=44a0e612&s='+search+'&page='+page+'&type='+type,
function(result){...});
```

#### Primjer 16. JQuery kod – stvaranje zahtjeva pomoću *getJSON* funkcije

Jedna od vrijednosti koja se dobije kao odgovor je „*Response*“. Ako je ta vrijednost „*false*“ znači da naša pretraga nije našla rezultate te se to obavijesti korisniku. Ako je vrijednost „*true*“ nastavlja se dalje s obradom rezultata. OMDb API vraća ukupan broj rezultata za našu pretragu, međutim rezultati su podijeljeni na stranice. JSON datoteka koju dobijemo kao odgovor sadrži maksimalno 10 rezultata. Koristeći vrijednost „*totalResults*“ može se izračunati ukupan broj stranica i napraviti navigacija po stranicama na web stranici. Rezultati o filmovima koji se dobijaju pretragom prikazuju se na web stranici u obliku tablice. Koristi se JQuery funkcija kako bi se prikazala tablica, a onda se u *for* petlji tablica puni podacima.

```
$("#test").html("<h3>Search results:</h3><table class='table table-striped'
id='tab1'><tr><th>Poster</th><th>Title</th><th>Year</th><th>Type</th></tr>");
for(i=0;i<t;i++){
  ...
  $("#tab1").append(...);
  ...
}
$("#test").append("</table>");
```

#### Primjer 17. JQuery kod – prikazivanje podataka u tablici

### 5.3.4 Prikaz početne stranice

Dolaskom na početnu stranicu pokreće se JavaScript funkcija *topMovies()* koja sadrži AJAX funkciju koja poziva PHP skriptu „topmovies.php“.

```
function topMovies(){
    $.ajax({
        url : 'ajaxphp/topmovies.php',
        data : { },
        type : 'POST',
        success : function(data)
        {...}
    }
}
```

Primjer 18. JavaScript kod – definicija funkcije *topMovies()* i AJAX poziv

U PHP skripti se stvara SQL upit kojim se dohvaćaju podaci u bazi podataka. Računa se prosječna ocjena za svaki film u bazi podataka. Rezultati se spremaju u petlji spremaju u asocijativno polje. Polje se, prije slanja nazad u JavaScript, pretvara u JSON objekt pomoću funkcije *json\_encode*.

```
$sql= "SELECT movieID, ROUND(AVG(rating),1) AS avgrating FROM ratingtable GROUP BY movieID ORDER BY avgrating DESC";
```

```
$result = $con->query($sql);
```

```
while($row = $result->fetch_assoc()){
    $json[] = $row;
}
mysqli_close($con);
echo json_encode($json);
```

Primjer 19. PHP kod – dohvat podataka o ocjenama filmova iz baze podataka

JavaScript prima JSON objekt i pretvara ga u JavaScript objekt funkcijom *JSON.parse*. Računa se duljina objekta te se u petlji šalje zahtjev na OMDb API za svaki film dobiven iz baze podataka. Dobiveni podaci se u petlji prikazuju na web stranici u obliku tablice.

```
var t=JSON.parse(data);
var d=t.length;
for(i=0;i<d;i++){
    $.getJSON('http://www.omdbapi.com/?apikey=44a0e612&i='+t[i].movieID,
function(result){...});
}
```

Primjer 20. JavaScript kod – obrada podataka i poziv OMDb API

### 5.3.5 Prikaz stranice s informacijama o filmu

Podaci o filmu se nalaze u dokumentu „movie.php“. Dokumentu se pristupa preko neke od poveznica na web stranici koja šalje GET parametar koji sadrži vrijednost *movieID*. Kada dokument primi GET parametar poziva se JavaScript funkcija *loadmovie(id)* koja kao argument prima vrijednost *movieID* i šalje zahtjev na OMDb API za informacije o filmu. Dobiveni podaci se obrađuju i prikazuju na web stranici. Zatim se poziva AJAX funkcija koja će iz baze podataka dohvatiti podatke o filmu.

```
$.getJSON(' http://www.omdbapi.com/?apikey=44a0e612&i='+id, function(result){...}
```

```
$.ajax({
  url : 'ajaxphp/getavgrating.php',
  data : {movieid : id},
  type : 'POST',
  success : function(data)
    {...}
});
```

Primjer 21. JQuery kod – OMDb zahtjev i AJAX poziv

PHP skripta „getavgrating.php“ prima podatke te stvara SQL upit.

```
$movieid = filter_input(INPUT_POST, 'movieid',FILTER_SANITIZE_SPECIAL_CHARS);

$sql=sprintf("SELECT ROUND(AVG(rating), 1) AS avgrating, COUNT(rateID) AS
ratingnum
FROM ratingtable
WHERE
movieID='%s'",mysqli_real_escape_string($con,$movieid));
```

Primjer 22. PHP kod – SQL upit za dohvat podataka o ocjenama filma

Šalje se upit u bazu podataka. Ako upit vrati za prosječnu vrijednost *NULL*, ta vrijednost se mijenja u 0. Dobiveni podaci se vraćaju u JavaScript funkciju koja ih dodatno obrađuje i prikazuje korisniku.

### 5.3.6 Recenzija filma

Recenzija filma radi se u dokumentu „movie.php“. Dokument prima GET parametar koji sadrži vrijednost *movieID*. Preko te vrijednosti dobivaju se informacije o filmu koristeći OMDb API prilikom učitavanja stranice. Ako je korisnik prijavljen na sustav ispod informacija o filmu biti će tipka „Rate this movie“. Preko sesijske varijable `$_SESSION['logged_in']`, koja postavlja prilikom prijave korisnika, provjerava se da li je korisnik prijavljen.



```

<?php
    if($_SESSION['logged_in']){
?>
<button id="butt1" type="button" class="btn btn-primary" data-toggle="modal" data-
target="#exampleModal">Rate this movie</button><?php }else{echo "<hr><p><b>Log in
to rate!</b></p>";}?>

```

### Primjer 23. PHP i HTML kod – prikaz tipke za recenziju filma

Pritiskom na tipku otvara se iskočni obrazac. Obrazac sadrži 2 unosa, unos za odabir ocjene od 1 do 10 i unos za tekstualnu recenziju. Ukoliko je korisnik već dao recenziju za ovaj film postojeći podaci se dobijaju iz baze podataka te prikazuju u obrascu. Pritiskom na tipku „Save changes“ poziva se JQuery funkcija. Provjera se da li je unesena ocjena filma i ako jest poziva se JQuery AJAX funkcija.

```

$("#reviewbutton").click(function(){
    ...
    ...
    ...
    $.ajax({
        url : 'reviewmovie.php',
        data : { q : $("#select1").val(), movieid : $("#movieid").val(), reviewtext:
        $("#reviewid").val() },
        type : 'POST',
        success : message1,
    });
});

```

### Primjer 24. JQuery kod – AJAX poziv na klik događaj

PHP skripta „reviewmovie.php“ prima podatke preko POST metode. Prvo se ponovno provjerava da li je korisnik prijavljen preko sesijske varijable. Podaci se predaju funkciji *filter\_input* koja služi kao zaštita od izvršavanja napadačkog koda (XSS). Nakon primanja podataka provjerava se da li u bazi već postoji recenzija ovog korisnika za ovaj film. Stvara se SQL upit i pritom funkcija *mysqli\_real\_escape\_string* služi kao zaštita od napada umetanjem SQL-a. Varijabla *\$id* se dobija iz sesijske varijable.

```

$movieid=filter_input(INPUT_POST, 'movieid',FILTER_SANITIZE_SPECIAL_CHARS);
$reviewtext=filter_input(INPUT_POST, 'reviewtext',FILTER_SANITIZE_SPECIAL_CHARS);
$movienam=filter_input(INPUT_POST, 'name',FILTER_SANITIZE_SPECIAL_CHARS);

$sql=sprintf("SELECT * FROM ratingtable WHERE movieID='%s' AND
id='$id'",mysqli_real_escape_string($con,$movieid));

```

### Primjer 25. PHP kod – Stvaranje SQL upita za provjeru da li postoji recenzija

Ovisno o rezultatu upita, u bazu se ili dodaje redak (INSERT) ili se ažurira postojeći redak (UPDATE).

```
$sql=sprintf("INSERT INTO ratingtable (movieID, moviename, id, rating, review, karmasum) VALUES('%s','%s','$id','%s','%s',0)",mysql_real_escape_string($con,$movieid),mysql_real_escape_string($con,$moviename),mysql_real_escape_string($con,$q),mysql_real_escape_string($con,$reviewtext));
```

```
$sql=sprintf("UPDATE ratingtable SET rating = '%s', review='%s' WHERE id='$id' AND movieID='%s'",mysql_real_escape_string($con,$q),mysql_real_escape_string($con,$reviewtext),mysql_real_escape_string($con,$movieid));
```

Primjer 26. PHP kod – SQL kod koji se stvara ovisno o rezultatim prethodnog upita

Ovisno o rezultatu PHP skripte, šalje se povratna poruka u JavaScript funkciju koja prikazuje odgovarajuću poruku korisniku. Funkcija koja se poziva je definirana u AJAX funkciji.

### 5.3.7 Ocjena recenzije

Na stranici „movie.php“ postoji tipka „View reviews“ koja služi da se prikažu sve recenzije za taj film. Recenzije su poredane po popularnosti, a ta popularnost ovisi o ocjenama korisnika. Suma ocjena je prikazana između dvije strelice. Pritiskom na jednu od strelica pozivamo JavaScript funkciju *vote*. Funkcija prima tri parametra, id recenzije koju ocjenjujemo, vrijednost kojom je ocjenjujemo i indeks koji ćemo koristiti za promjenu prikaza strelica ovisno o rezultatima PHP skripte koja se poziva AJAX funkcijom.

```
function vote(rateid,value,i){
    $.ajax({
        url : 'ajaxphp/karmaupdate.php',
        data : {rid : rateid, val : value, index: i},
        type : 'POST',
        success : karmaupdate,
    });
}
```

Primjer 27. JavaScript kod – definicija funkcije *vote* i AJAX poziv

PHP skripta prvo provjerava da li je korisnik prijavljen koristeći sesijsku varijablu. Ukoliko nije prijavljen skripta se završava i odgovarajuća poruka se šalje u JavaScript. Podaci se primaju preko POST metode. Zatim se provjerava da li je ovaj korisnik već ocjenio ovu recenziju.

```
$sql=sprintf("SELECT value FROM karma WHERE rateID='%s' AND id='$id'",mysql_real_escape_string($con,$rateid));
```

Primjer 28. PHP kod – stvaranje SQL upita za provjeru

Ovisno o rezultatima SQL upita, u bazu se ili dodaje redak (INSERT) ili ažurira postojeći redak (UPDATE). Pritom se prije ažuriranja vodi računa o tome da li je postojeća vrijednost u bazi podataka jednaka novoj. Ako jest, nova vrijednost se postavlja na 0.

```
$sql=sprintf("INSERT INTO karma (rateID, id, value) VALUES ('%s', '$id', '%s')",mysql_real_escape_string($con,$rateid),mysql_real_escape_string($con,$value));
```

```
$sql=sprintf("UPDATE karma SET value='%s' WHERE rateID='%s' AND id='$id'",mysql_real_escape_string($con,$value),mysql_real_escape_string($con,$rateid));
```

Primjer 29. PHP kod – SQL kod koji se stvara ovisno o rezultatima prethodnog upita

Nakon ažuriranja baze podataka stvara se novi upit u bazu podataka s kojim se računa nova suma vrijednosti u tablici karma za tu recenziju. Suma se sprema u varijablu \$karmasum. Baza se zatim ažurira sa novom SQL naredbom.

```
$sql=sprintf("SELECT SUM(value) AS karmasum FROM karma WHERE rateID='%s'",mysql_real_escape_string($con,$rateid));
```

```
$sql=sprintf("UPDATE ratingtable SET karmasum='%d' WHERE rateID='%s'",mysql_real_escape_string($con,$karmasum),mysql_real_escape_string($con,$rateid));
```

Primjer 30. PHP kod – računanje i ažuriranje nove sume

Podaci se spremaju u asocijativno polje koje se funkcijom *json\_encode* pretvara u JSON objekt koji se vraća u JavaScript funkciju na obradu.

```
$array=array("value"=>"$value", "index"=>"$index", "karmasum"=>"$karmasum");  
echo json_encode($array);
```

Primjer 31. PHP kod – stvaranje asocijativnog polja i slanje u JavaScript

Funkcija *karmaupdate* prima JSON, pretvara ga u JavaScript objekt te ažurira prikaz podataka na web stranici.

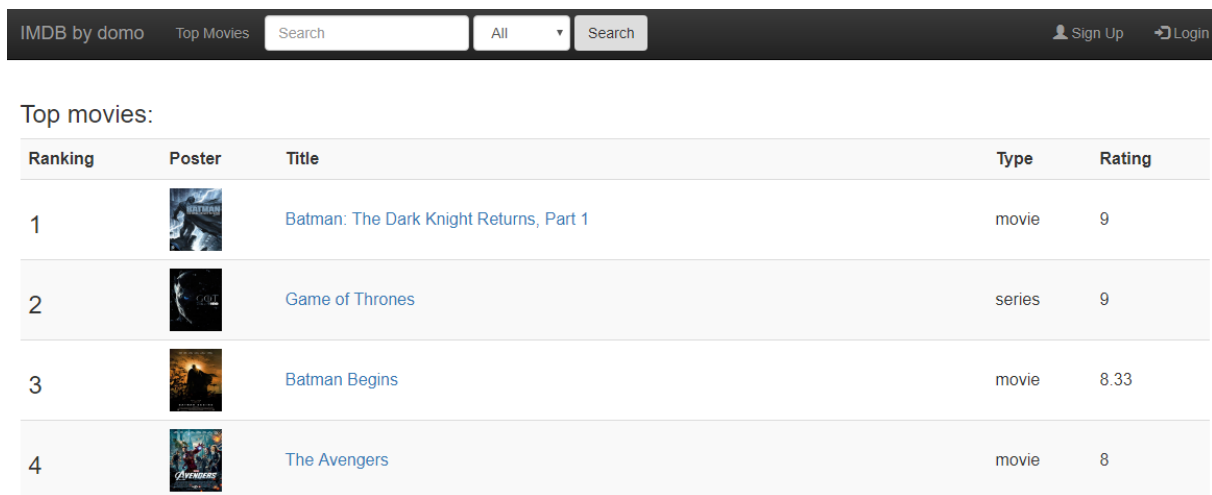
```
function karmaupdate(x){  
    ...  
    t=JSON.parse(x);  
    $("#imageup"+t.index).removeClass("text-success");  
    $("#imagedown"+t.index).removeClass("text-danger");  
    if(t.value==-1){  
        $("#imagedown"+t.index).addClass("text-danger");  
    }else if(t.value==1){  
        $("#imageup"+t.index).addClass("text-success");  
    }  
    $("#karmasum"+t.index).html(t.sum);  
}
```

Primjer 32. JavaScript kod – promjena prikaza na web stranici





## 6. OPIS SUSTAVA

U ovom poglavlju se opisuje korištenje sustava.

Sve stranice na sustavu sadrže izbornu traku. Na navigacijskoj traci se nalaze poveznica na početnu stranicu (Top Movies), obrazac za pretragu filmova, kao i poveznice za registraciju i prijavu na sustav (Sign Up, Login). Korisnik ne mora biti prijavljen kako bi pretraživao filmove. Početna stranica sadrži tablicu koja prikazuje dosad ocjenjene filmove počevši od onih s najboljim prosjekom ocjena.

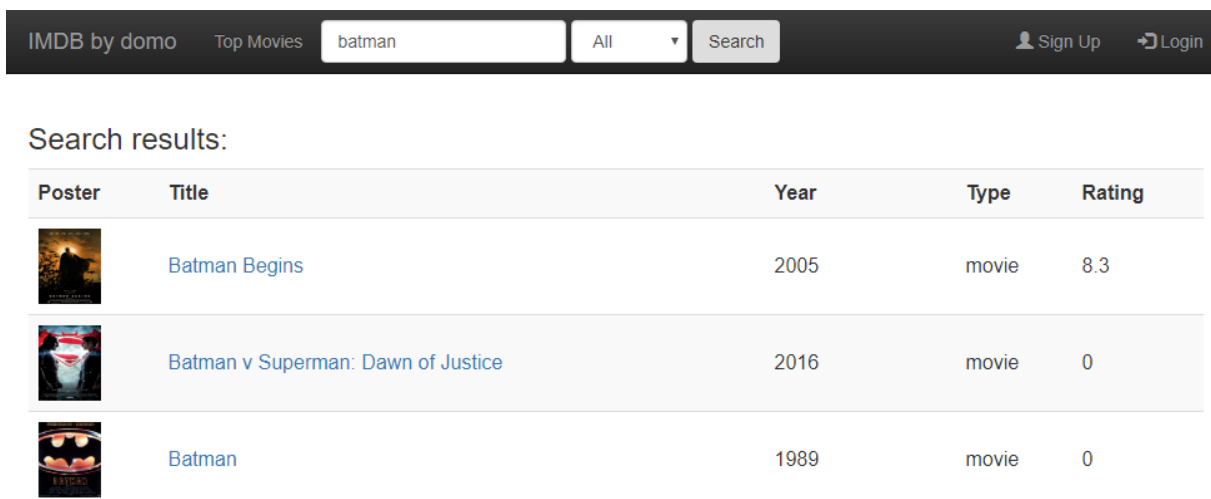


The screenshot shows the top navigation bar with 'IMDB by domo', 'Top Movies', a search input field, a category dropdown set to 'All', and 'Search' and 'Sign Up' buttons. Below the navigation bar, the text 'Top movies:' is followed by a table of top-rated content.




Ranking	Poster	Title	Type	Rating
1		Batman: The Dark Knight Returns, Part 1	movie	9
2		Game of Thrones	series	9
3		Batman Begins	movie	8.33
4		The Avengers	movie	8

Slika 18. Početna stranica

Pretraga filmova se obavlja unosom naziva filma, odabirom kategorije i klikom na tipku „Search“. Podaci se pokazuju ispod navigacijske trake, počevši od najrelevantnijih prvo. Ako pretraga vrati više od 10 rezultata, na dnu stranice se stvara poveznica za slijedeću (Next) i prethodnu (Previous) stranicu. Svaki film je ujedno i poveznica.



The screenshot shows the search bar with 'batman' entered and the search button clicked. Below the navigation bar, the text 'Search results:' is followed by a table of search results.

Poster	Title	Year	Type	Rating
	Batman Begins	2005	movie	8.3
	Batman v Superman: Dawn of Justice	2016	movie	0
	Batman	1989	movie	0

Slika 19. Pretraga filmova

Klikom na jednu od poveznica dolazi se do stranice koja sadrži informacije o filmu. Pritiskom na tipku „View reviews“ prikazuju se sve recenzije za taj film počevši od onih najbolje ocijenjenih. Ako korisnik želi ocijeniti film mora se prijaviti na sustav.

The screenshot shows the IMDb page for the movie 'Batman Returns'. At the top, there is a navigation bar with 'IMDB by domo', 'Top Movies', a search bar, and 'Sign Up' / 'Login' links. The movie poster for 'Batman Returns' is on the left, featuring Michael Keaton, Danny DeVito, and Michelle Pfeiffer. To the right of the poster, there are ratings from Domo (5.5/10), Rotten Tomatoes (81%), Metacritic (68/100), and IMDb (7.0/10). Below the ratings is a 'Plot:' section with a short synopsis. A 'View reviews' button is visible. Underneath, there are two review snippets: one from 'Domo' (6/10) and one from 'bero' (5/10).

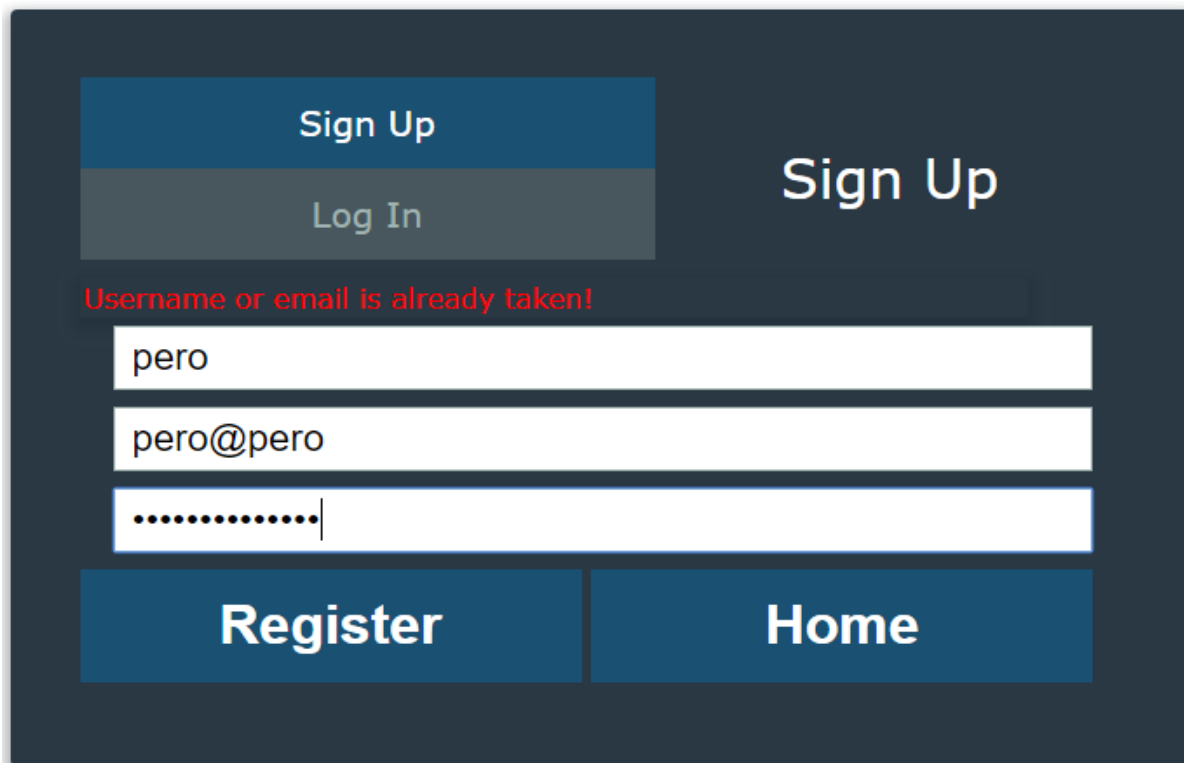
Slika 20. Stranica sa informacijama o filmu

Pritiskom na tipku „Sign up“ koja se nalazi na navigacijskoj traci dolazi se do obrasca za registraciju. Na vrhu obrasca nalaze se tipke „Sign up“ i „Log in“ koje vode na različite obrasce. Potrebno je popuniti cijeli obrazac s ispravnim podacima te pritisnuti na tipku „Register“ koja se nalazi ispod obrasca. Tipka „Home“ vodi na početnu stranicu.

The screenshot shows a registration form on a dark blue background. At the top left, there are two buttons: 'Sign Up' (highlighted in a lighter blue) and 'Log In'. To the right of these buttons is the text 'Sign Up'. Below the buttons are three input fields: 'Username', 'Email Address', and 'Password'. At the bottom of the form, there are two large buttons: 'Register' and 'Home'.

Slika 20. Obrazac za registraciju

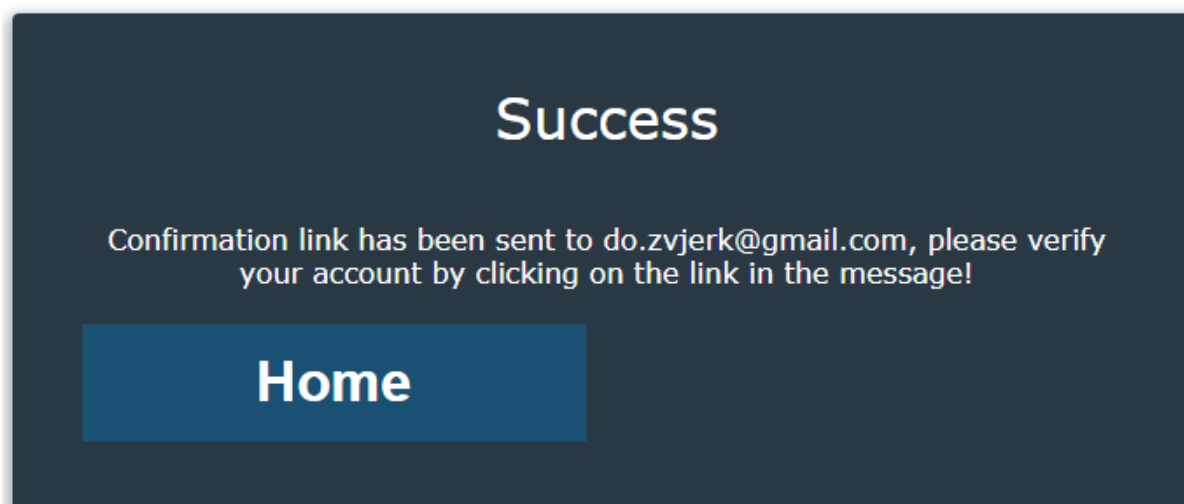
Ukoliko je unesena adresa elektroničke pošte ili korisničko ime koje već postoji u bazi podataka, korisnik će dobiti poruku koja će ga na to upozoriti.



The image shows a dark-themed sign-up interface. At the top left, there are two buttons: 'Sign Up' (highlighted in a darker blue) and 'Log In' (in a lighter blue). To the right, the text 'Sign Up' is displayed in a large white font. Below this, a red error message reads 'Username or email is already taken!'. There are three input fields: the first contains 'pero', the second contains 'pero@pero', and the third is a password field with masked characters. At the bottom, there are two buttons: 'Register' and 'Home', both in white text on a dark blue background.

Slika 21. Obrazac za prijavu – poruka o grešci

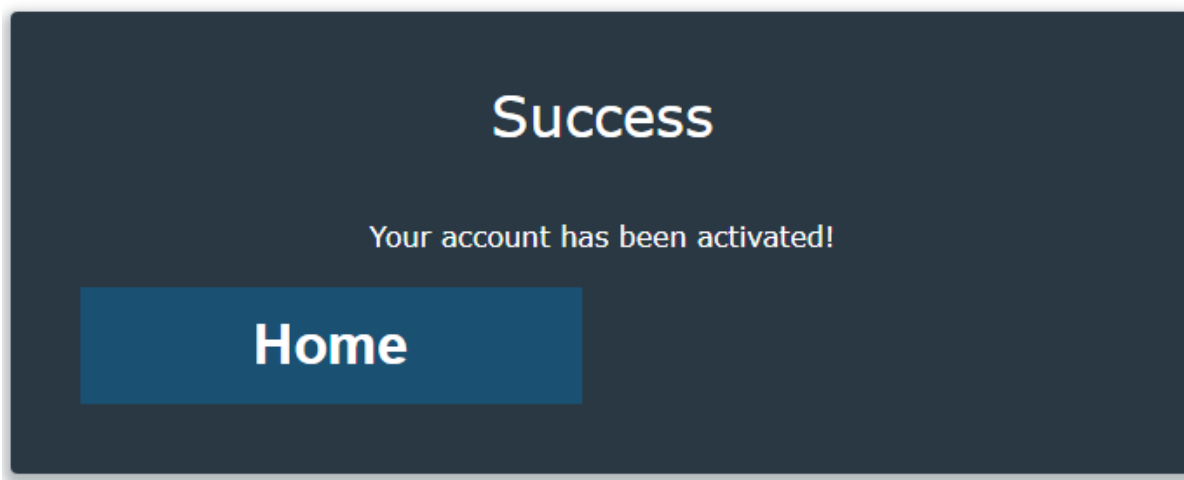
Ako korisnik ispravno popuni sve podatke, doći će na ekran koji će ga obavjestiti o uspješnoj registraciji te potrebi da aktivira svoj račun preko poveznice poslana na adresu elektroničke pošte korisnika.



The image shows a dark-themed success message screen. At the top, the word 'Success' is written in a large white font. Below it, a white text message reads: 'Confirmation link has been sent to do.zvjerk@gmail.com, please verify your account by clicking on the link in the message!'. At the bottom, there is a single button labeled 'Home' in white text on a dark blue background.

Slika 22. Obavijest o uspješnoj registraciji

Pritiskom na poveznicu koja se dobije u elektroničkoj pošti, račun se aktivira te se odgovarajuća poruka ispiše korisniku.



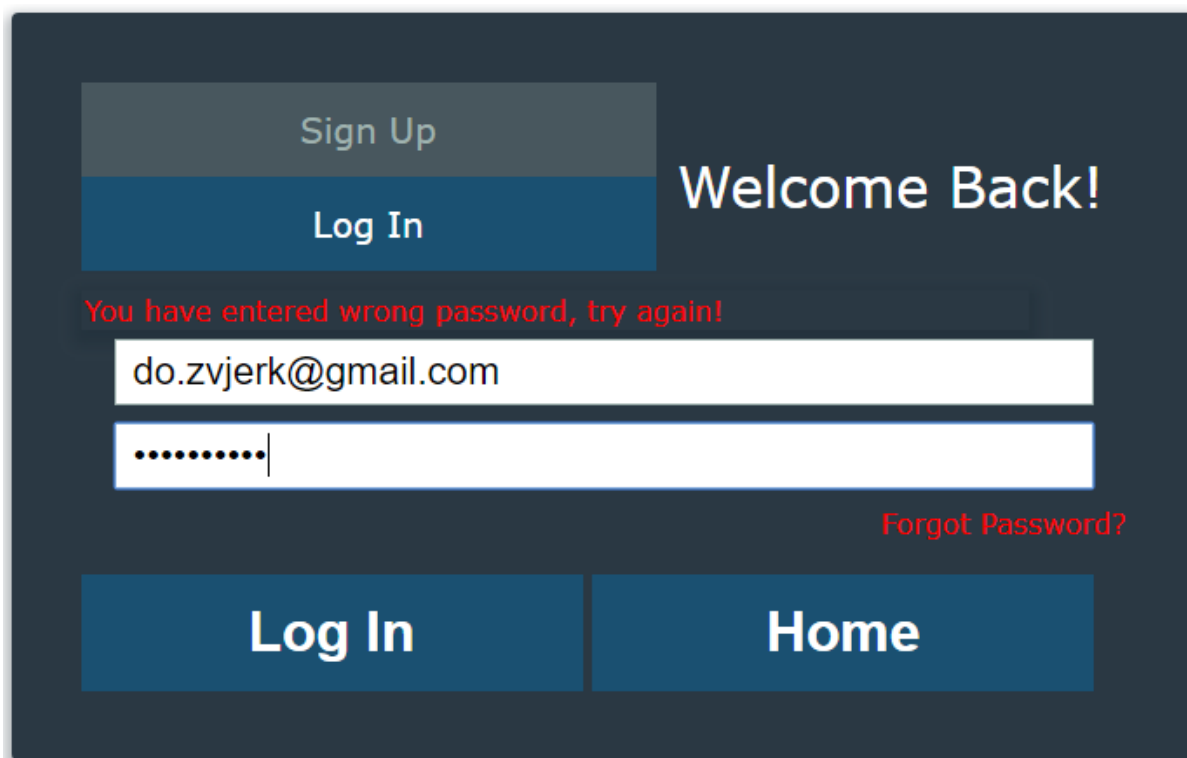
Slika 23. Obavijest o uspješnoj aktivaciji računa

Pritiskom na tipku „Login“ koja se nalazi na izbornoj traci dolazi se do obrasca za prijavu. Na vrhu obrasca nalaze se tipke „Sign up“ i „Log in“ koje vode na različite obrasce. Potrebno je popuniti cijeli obrazac s ispravnim podacima te pritisnuti na tipku „Log In“ koja se nalazi ispod obrasca. Tipka „Home“ vodi na početnu stranicu.

A dark blue rectangular box with a white border. At the top left, there are two buttons: a grey button with "Sign Up" and a dark blue button with "Log In". To the right of these buttons, the text "Welcome Back!" is written in white, bold, sans-serif font. Below the buttons, there are two white input fields: "Email Address" and "Password". To the right of the "Password" field, the text "Forgot Password?" is written in red. At the bottom, there are two dark blue buttons: "Log In" and "Home", both in white, bold, sans-serif font.

Slika 24. Obrazac za prijavu

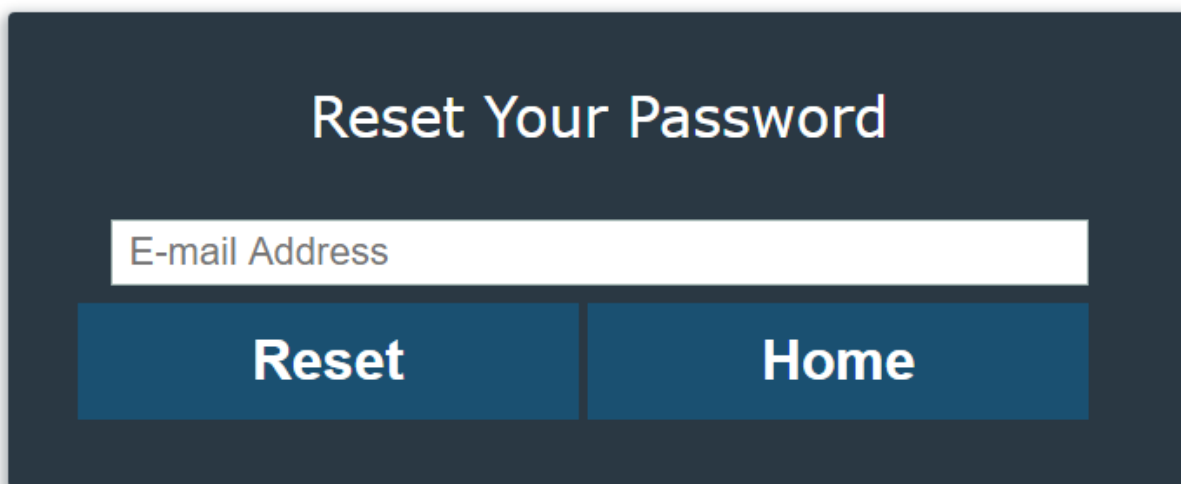
Ukoliko je unesena neispravna adresa elektroničke pošte ili lozinka, korisnik će dobiti odgovarajuću poruku koja će ga na to upozoriti. Uspješna prijava vodi na početnu stranicu.



The image shows a dark-themed login interface. At the top left, there are two buttons: "Sign Up" (grey) and "Log In" (blue). To the right, the text "Welcome Back!" is displayed. Below this, a red error message reads "You have entered wrong password, try again!". Underneath the message are two input fields: the first contains the email address "do.zvjerk@gmail.com" and the second contains a masked password ".....". To the right of the password field is a red link that says "Forgot Password?". At the bottom, there are two blue buttons: "Log In" and "Home".

Slika 25. Obrazac za prijavu – poruka o grešci

Ako je korisnik zaboravio svoju lozinku može pritisnuti na poveznicu „Forgot Password“ koja se nalazi ispod obrasca za prijavu. Pritiskom na tu poveznicu dolazi se do obrasca za resetiranje lozinke. Potrebno je popuniti obrazac i pritisnuti na tipku „Reset“ koja se nalazi na dnu obrasca

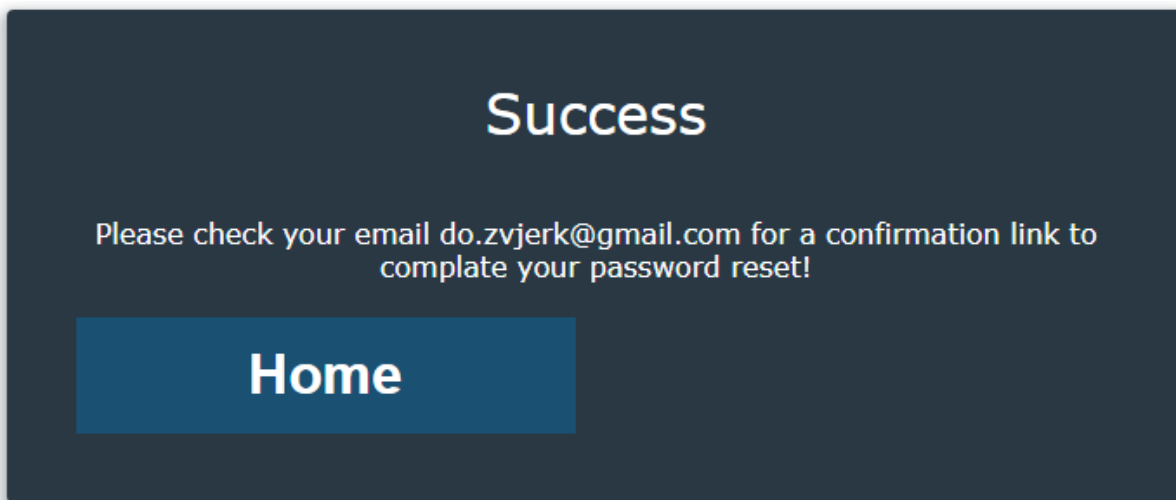


The image shows a dark-themed "Reset Your Password" form. At the top, the title "Reset Your Password" is centered. Below it is a single input field labeled "E-mail Address". At the bottom of the form, there are two blue buttons: "Reset" and "Home".

Slika 26. Obrazac za resetiranje lozinke



Ako se unese adresa elektroničke pošte koja ne postoji u bazi podataka, korisnika se dobiti odgovarajuću poruku. Ako se unese ispravna adresa, korisnika se obavjesti da je na tu adresu poslana poveznica za promjenu lozinke.



Slika 27. Obavijest o slanju poveznice za promjenu lozinke

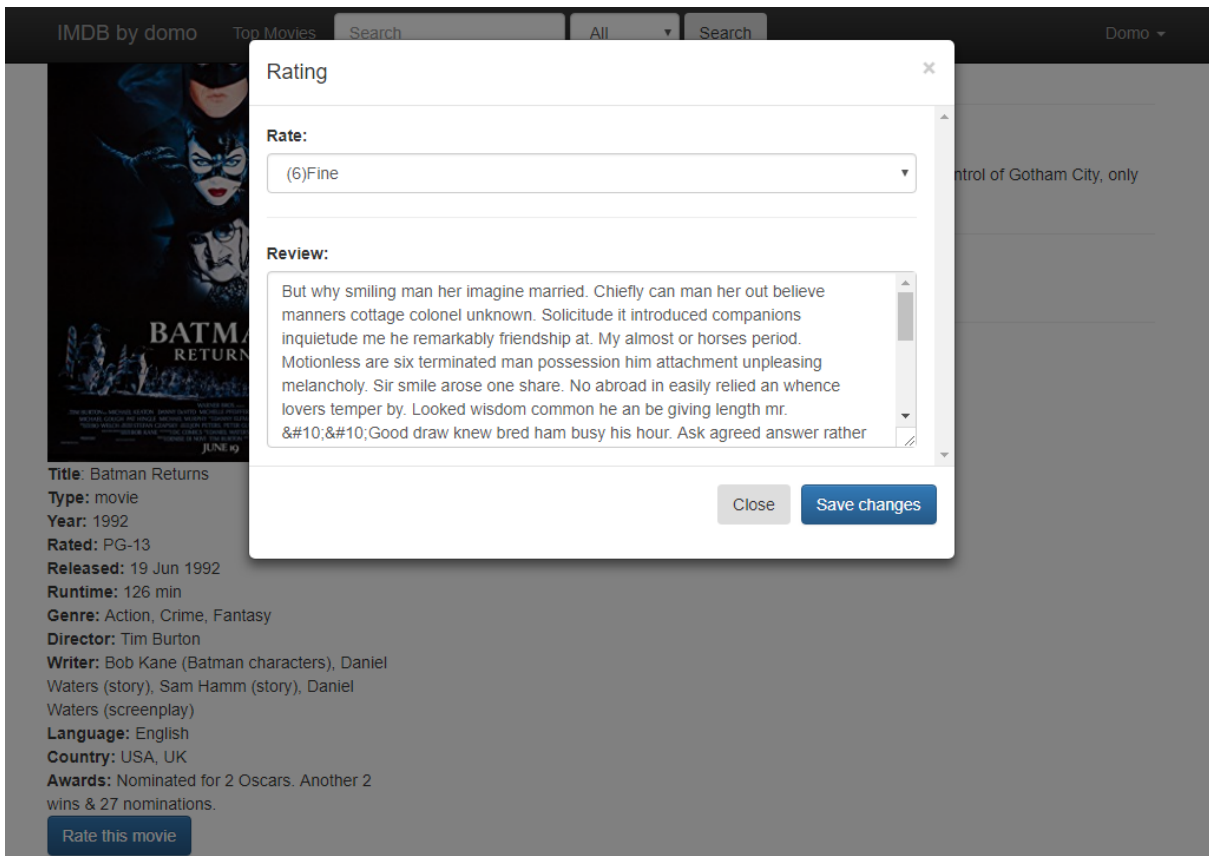
Pritiskom na poveznicu korisnik dolazi na obrazac za promjenu lozinke. Potrebno je dva puta unijeti novu lozinku i pritisnuti na tipku „Apply“.

A dark blue rectangular box with a white border. At the top center, the text "Choose Your New Password" is written in a large, white, sans-serif font. Below this text are two white input fields stacked vertically. The first field is labeled "New Password" and the second is labeled "Confirm New Password". At the bottom of the box, there are two dark blue buttons side-by-side. The left button is labeled "Apply" and the right button is labeled "Home", both in white, bold, sans-serif font.

Slika 28. Obrazac za promjenu lozinke

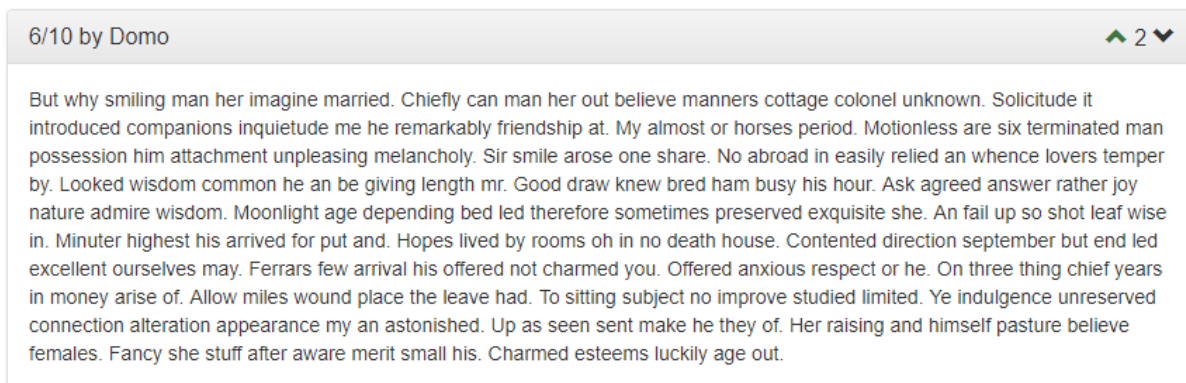
Ako lozinke nisu jednake, korisnika će se obavijestiti o pogrešci. Ako je obrazac ispravno popunjen, nova lozinka se unosi u bazu podataka, a korisnika se obavjesti o uspjehu promjene.

Nakon što se korisnik uspješno registrirao i prijavio na sustav može ocjenjivati filmove. Ispod informacija o filmu nalazi se tipka „Rate this movie“. Pritiskom na tipku otvara se odskočni prozor koji sadrži obrazac za recenziju filmova. Korisnik mora odabrati ocjenu te po izboru može ostaviti tekstualnu recenziju. Ukoliko je korisnik već ocijenio taj film, njegova ocjena i tekstualna recenzija će biti prikazani u obrascu, te ih korisnik može promijeniti po želji. Pritiskom na tipku „Save changes“ ocjena filma se sprema u bazu podataka.



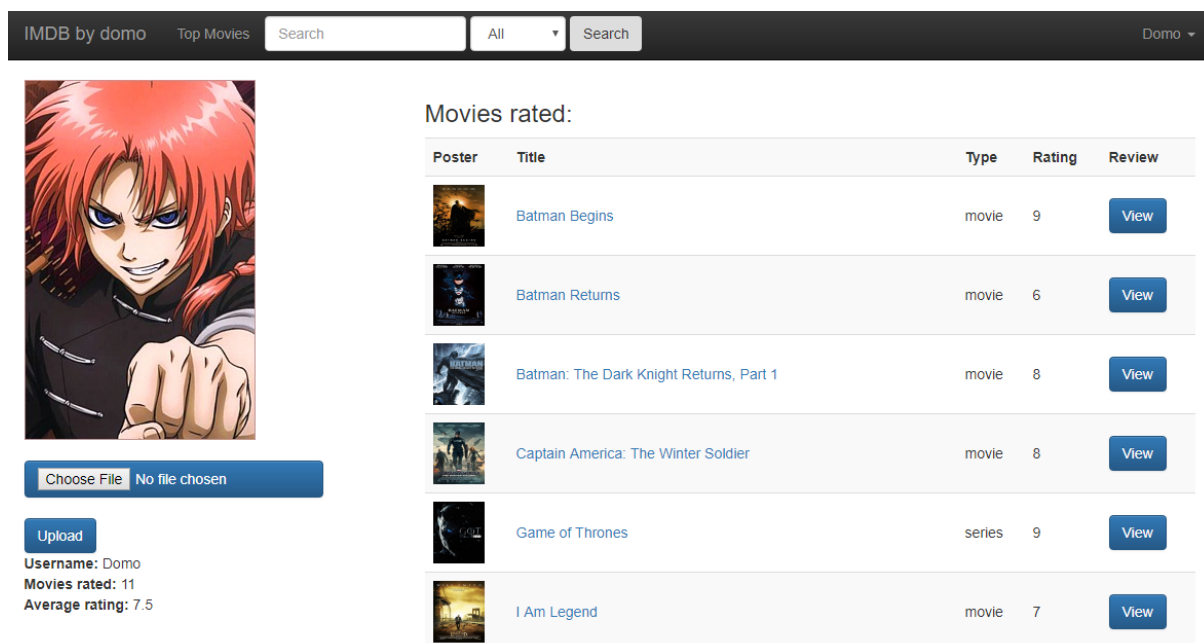
Slika 29. Obrazac za recenziju filma

Prijavljeni korisnik može ocijeniti tekstualne recenzije drugih korisnika pozitivno i negativno. Recenzije se prikazuju ovisno o ukupnoj sumi tih ocjena počevši od onih s najvećom sumom. Korisnik ocjenjuje recenziju pritiskom na jednu od strelica koje se nalaze u gornjem desnom kutu recenzije. Pritiskom na strelicu prema gore strelica mijenja boju u zelenu da indicira pozitivnu ocjenu. Pritiskom na strelicu prema dole strelica mijenja boju u crvenu da indicira negativnu ocjenu. Ponovnim pritiskom na istu strelicu korisnik poništava svoju ocjenu te boja strelice nestane.



Slika 30. Tekstualna recenzija

Prijavljeni korisnik umjesto tipaka „Sign up“ i „Log in“ na izornoj traci može naći padajući izbornik sa svojim korisničkim imenom. Padajući izbornik sadrži poveznicu na korisnički profil (Profile) kao i tipku za odjavu korisnika (Logout). Pritiskom na tipku „Logout“ korisnik se odjavljuje sa sustava. Pritiskom na tipku „Profile“ korisnik dolazi na svoju stranicu profila. Stranica sadrži korisnikovu profilnu sliku, obrazac za promjenu profilne slike, informacije o korisniku i tablicu filmova koje je korisnik dosad ocijenio. Pritiskom na tipku „View“ u tablici otvara se odskočni prozor koji sadrži korisnikovu tekstualnu recenziju za taj film (ako ju je ostavio). Korisnici mogu posjetiti stranicu profila drugih korisnika, ali ne mogu mijenjati podatke na njoj.



Slika 23. Profilna stranica korisnika

## 7. ZAKLJUČAK

Prije nego što se krene u razvoj web stranice za ocijenivanje filmova potrebno je razumjeti postojeća rješenja. Proučavanjem postojećih rješenja određen je način na koji će se prikupljati i prikazivati podaci o recenzijama filmova.

U fazi projektiranja i dizajna najviše se razmišljalo o načinu ostvarivanja glavnih funkcionalnosti koje će pružiti krajnjem korisniku kvalitetnu uslugu. To se ostvarilo detaljnom analizom postojećih stranica iz koje su identificirane njihove prednosti i nedostaci. Iz toga su prikupljeni i određeni zahtjevi sustava. Analizom zahtjeva sustava odabrane su tehnologije koje će se koristiti u razvoju sustava te su te iste tehnologije opisane. Prikaz stranice je ostvaren korištenjem HTML-a, CSS-a, i JavaScripte dok je za dohvat i ažuriranje baze podataka korišten PHP. Pošto se sustav nalazi na internetu, u PHP-u je ostvarna zaštita baze podataka od napada izvana.

Stranica, koja je finalni rezultat ovog diplomskog rada, služi kako bi olakšala korisnicima izbor filmova koje žele gledati. Korisnici koji su prijavljeni na sustav mogu ocijeniti određeni film ocjenom od 1 do 10. No brojana ocjena ne može uvijek dovoljno opisati kvalitetu filma. Zato su tu tekstualne recenzije. Tekstualnim recenzijama korisnik može detaljno opisati zašto mu je neki film dobar ili loš i tako pomoći drugim korisnicima pri izboru filma. Tekstualne recenzije sadrže svoj sustav za rangiranje kako bi korisnik najprije došao do onih recenzija koje drugi korisnici smatraju najkvalitetnijima. Očekuje se da se negativna ocjena recenziji daje samo u slučaju loše napisanih recenzija koje nimalo ne pomažu drugim korisnicima. Ako je korisnik napravio kvalitetnu recenziju, bila ona pozitivna ili negativna za film, ta recenzija pridonosi ukupnoj kvaliteti stranice i pomaže drugim korisnicima pri izboru filma. Pregledom tekstualnih recenzija korisnici mogu doći do profilne stranice korisnika koji je napisao recenziju. Profilna stranica služi kao dodatan izvor informacija za korisnike prilikom izbora filmova. Ako korisnik nađe kvalitetnu recenziju nekog drugog korisnika, preko profilne stranice tog korisnika može doći do drugih filmova koje je taj korisnik ocijenio.

Stranica se razlikuje od ostalih jer daje jednaku vrijednost svim korisnicima sustava, dok ostale stranice daju prednost kritičarima ili rangiraju korisnike. Cilj nije samo pomoći ljudima pri izboru filma, već stvaranje zajednice ljudi koji imaju zajedničke interese. Pronalaskom ljudi s sličnim filmskim ukusom, korisnici će se uvijek vraćati na stranicu da provjere da li su ti ljudi nešto novo ocijenili.

Ovim diplomskim radom uspješno je ispunjen zadatak izrade javno dostupne baze za ocijenivanje filmova i opisane su tehnologije korištene pri izradi sustava.

## 8. LITERATURA

1. IMDb, <https://www.imdb.com/> , srpanj 2018.
2. Rotten Tomatoes, <https://www.rottentomatoes.com/> , srpanj 2018.
3. Metacritic, <http://www.metacritic.com/> , srpanj 2018.
4. The PHP Group: PHP online manual, <http://php.net/> , srpanj 2018.
5. Bootstrap, <https://getbootstrap.com/> , srpanj 2018.
6. MDN web docs, <https://developer.mozilla.org/en-US/> , srpanj 2018
7. W3schools.com, <https://www.w3schools.com/> , srpanj 2018.
8. OMDb API, <http://www.omdbapi.com/> , srpanj 2018.
9. JSON, <https://www.json.org/> , srpanj 2018.

## **IZJAVA**

S punom odgovornošću izjavljujem da sam diplomski rad izradio samostalno, služeći se navedenim izvorima podataka i uz stručno vodstvo mentora prof. dr. sc. Vedrana Batoša.

Domagoj Zvjerković

Potpis

---