

Zadaća - Web aplikacije sa informacijama o glumcima

Poslužitelj: spider.foi.hr

Korijenska mapa zadaje: /var/www/RWA-ITDP/2023/zadaca_01/{LDAP_korisnicko_ime}/

Pristup preko web preglednika: http://spider.foi.hr:{vaš_port}

Portovi: /var/www/RWA-ITDP/2023/portovi.js

Baza: RWA2023{LDAP_korisnicko_ime}.sqlite

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

Način predaje: Rješenje postaviti na moodle u radionicu u zip obliku do dogovorenog roka. Radi jednostavnosti ova zadaća se testira lokalno.

Struktura zadaje: U direktoriju smiju biti **samo datoteke i poddirektoriji vezane uz zadaću**, sve ostalo treba ukloniti (npr. skripte s vježbi). **Svugdje u zadaci treba koristiti relativne putanje.** Ako u tekstu zadaje 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/Dokumentacija: Potrebno je ispuniti dokumentaciju i predati zajedno sa zadaćom te anketu vezanu uz zadaću na Moodle sustavu. Ako **se ne ispuni anketa/dokumentacija 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 zadaje: Na kratkoj obrani na vježbama otvara se kod zadaje 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 zadaje.** **Ako student/ica ne da odgovor na većinu postavljenih pitanja smatra se da nije autor zadaje što znači 0 bodova iz zadaje, a time i potpis prema modelu praćenja.** Ako postoji sumnja o autorstvu zadaje, nastavnik zadržava pravo pozvati studenta/icu na konzultacije na dulju obranu zadaje u posebnom terminu konzultacija.

Pomoć i pitanja: Sva pitanja u vezi zadaje postavljaju se u **poseban forum za zadaću na sustavu Moodle.** Nema privatnih konzultacija u vezi zadaje 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** zadaje; 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 zadaje. Boduju se dijelovi koji su rađeni samostalno nakon vježbi. U bodovanje, uz činjenicu radi li aplikacija

prema specifikacija 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. Ako neki dio servisa ne radi po definiranoj specifikaciji dobiva se 0 bodova za taj dio, ali se mogu ostvariti bodovi za web aplikaciju koja komunicira sa takvim servisom. Pripazite na velika i mala slova, točku, zarez i sl. **Implementirani servis bez funkcionalne web aplikacije koja ga koristi ne nosi maksimalne bodove iako je ispravno implementiran.**

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

Dizajn: Dizajn stranice je proizvoljan, važno je da se stranica može normalno koristiti u preglednicima Firefox i Chrome/Chromium. 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 kreira svoju vlastitu SQLite bazu podataka. Kreirana i popunjena baza se mora također nalaziti u direktoriju zadaće. **Nastavnik neće bazu samostalno kreirati.**

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 UWT 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 za datum i sati:minute:sekunde u 24 satnom obliku za vrijeme. Izbjegavati input element tipa date i tipa time jer oni koriste američki model datuma i vremena.

Provjere unosa korisnika: Preporuka je da se sav korisnički unos validira kako ne bi bilo moguće srušiti aplikaciju krivim unosima ili napraviti napad. Dozvoljeno je korištenje regularnih izraza, ali nije obavezno. Ukoliko se koriste treba ih znati objasniti

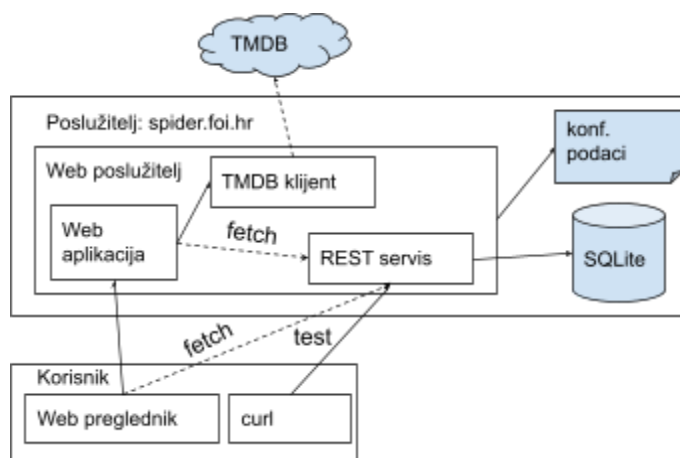
Testiranje: aplikaciju/servis treba testirati i s krivim podacima, provjeriti rubne slučajeve, a ne samo idealni scenarij korištenja. Kod testiranja servisa predlažemo curl naredbu jer možete lako više puta izvršiti. Također radi lakšeg testiranja bolje je napraviti prvo servisni dio, a tek onda aplikativni dio koji koristi REST. Naravno mogu se koristiti i drugi program (npr. Postman) za testiranje servisa.

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/>
- Traženje glumca: <https://developer.themoviedb.org/reference/search-person>
 - https://api.themoviedb.org/3/search/person?query=Spencer&include_adult=false&language=en-US&page=1
 - https://api.themoviedb.org/3/search/person?query=Bud%20Spencer&include_adult=false&language=en-US&page=1
- Svi podaci o glumcu: <https://developer.themoviedb.org/reference/person-details>
 - <https://api.themoviedb.org/3/person/18841?language=en-US>
- Putanje do slika: https://image.tmdb.org/t/p/original/{poster_path}
<https://image.tmdb.org/t/p/original/cVbBoKxRDFOdDKwdpRmxVazDWIE.jpg>

Globalna arhitektura

Zadaća se sastoji od web poslužitelja (servera) kreiranog u programu NodeJS koji se pokreće na vama dodijeljenom portu (vaše korisničko ime) koji se čita iz datoteke **portovi.js** i nesmije biti hardkodiran. Web poslužitelj pruža web aplikaciju i web servis. Web servis je REST servis omogućuje rad s vlastitom SQLite bazom podataka. Web aplikacija pruža korisničko sučelje za prikaz informacija o glumcima, rad sa korisnicima i sl. Informacije o glumcima dobivaju se iz vanjskog servisa TMDB preko posebne klase TMDB klijent. Slika 1 prikazuje globalnu arhitekturu sa glavnim komponentama i vezama. Iz Slike 1 je vidljivo da samo REST web servis ima pristup bazi podataka. Web aplikacija u komunicira s REST servisom u pozadinskom i klijentskom dijelu putem Fetch API-a. **U realizaciji je potrebno primjenjivati asinkrono programiranje te ni u jednom trenutku ne bi trebalo doći do blokiranja glavnog programa.**



Slika 1. Globalna arhitektura

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 SQLite baze generirano iz alata,
- slika ERA modela baze koja se izveze iz alata MySQL Workbench i
- **dokumentacija.html** - sadrži informacije o vama kao autoru: slika lica, ime, prezime, foi 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. Sadrži tablicu 2 i 3 ovog dokumenta, ali umjesto opisa piše da li je nešto napravljeno ili prazno ako nije napravljeno. Također sadrži dodatne informacije koji dijelovi nisu napravljeni, dodatne informacije o greškama ili dijelove koji se žele posebno istaknuti.

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. U bazi podataka mora se nalaziti minimalno 10 podataka ukupno za sve glavne entitete (npr. korisnik). **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 “admin”, ulogu “administrator” i lozinku “rwa”. Ako korisnici ne postoje u bazi, neće se bodovati dio koji ulazi u funkcionalnosti odgovarajuće uloge.

Kod modeliranja baze pročitajte cijeli tekst oba dijela web aplikacije i web servisa da biste dobili sve entitete i attribute koji vam trebaju. Vaš servis nužno ne diktira kako izgleda baza i možete u bazi imati podatke na jedan način posložene, a servis ih nudi na drugi način.

Opis rada NodeJS web poslužitelja

Kod pokretanja poslužiteljskog programa web servisa koristi se:

npm start

koji pokreće

node server.mjs nazivDatotekeKonfiguracije.csv

Prilikom pokretanja provjerava se dali je proslijeđen jedan ulazni parametar, da li postoji datoteka konfiguracije i dali su podaci u datoteci ispravni. 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 prikazani u tablici 1. Ako neki podatak nedostaje u datoteci ili ima pogrešnu vrijednost u odnosu na specifikaciju u tablici 1, 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 zahtjeve za obradu.

U tablici 1 definirani su nazivi svih parametara koji se koriste. Za svaki parametar napisani su kriteriji u kakvom obliku mora biti podatak te kratak opis čemu služi. Konfiguracijski podaci spremaju se u CSV datoteci u obliku “naziv:vrijednost”, svaki u novi red. Vrijednost ne može imati znak dvotočka.

Tablica 1. Pregled konfiguracijskih parametara

Naziv konfiguracije	Vrijednost	Opis
tajniKljucSesija	Veličina: 50 - 100 znakova	Tajni ključ za sesiju
appStraniceNje	Broj od 5 - 100	Sadrži brojčani podatak koji govori koliko se podataka odjednom prikazuje u tablicama
tmdbApiKeyV3	Nema restrikcija	Sadrži API ključ v3 za pristup TMDB servisu oko 32 znaka
tmdbApiKeyV4	Nema restrikcija	Sadrži API ključ v4 za pristup TMDB servisu oko 211 znakova

Opis rada REST servisa

REST servis predstavlja aplikativno programsko sučelje (API) prema bazi podataka. REST servis nudi API za resurse prema specifikaciji u Tablici 2 na bazičnoj putanji “rest”. Zadnja dva retka u tablici 2 znače da možete imati dodatne resurse preko kojih se komunicira s bazom poštujući specifikaciju u tablici 2. **Sva komunikacija je u JSON obliku osim ako nije drugačije definirano u tablici.** Kod slanja/primanja jednog podatka kao što je dohvat, ažuriranje ili dodavanje korisnika šalje se samo taj jedan objekt npr. `{"ime": "pero", "prezime": "kos", ...}` Kod slanja/primanja više podataka kao što je dohvat svih korisnika šalje se niz objekata ili prazan niz ako nema podataka npr. `[{"ime": "pero", "prezime": "kos"}, {"ime": "maro", "prezime": "marić"}]`

Svim dijelovima servisa osim prijave i registracije novog korisnika smije pristupiti samo prijavljeni korisnik (tj. ako postoji validna sesija). Ako neprijavljeni korisnik pokušava pristupiti zabranjenom dijelu vraća se statusni kod **403** sa opisom “**zabranjen pristup**”. Ako metodi smije pristupiti samo administrator servis vraća statusni kod **403** sa opisom “**zabranjen pristup nedovoljna prava**”. U tablici 2 dijelovi koji su označeni **plavo znače da samo administrator smije pristupiti**. Ako se pokuša dohvatiti resurs koji nije u specifikaciji tada servis vraća statusni kod **404** sa opisom “**nema resursa**”. Ako je bilo kakvi drugi problem zbog kojeg se zahtjev ne može izvršiti (npr. brisanje nije uspješno) ili su zaprimljeni podaci neispravni vraća statusni kod **400** i prikladna poruka greške koja opisuje problem. **Sve poruke grešaka se vraćaju unutar tijela odgovora pod atributom “pogreska” npr. ovako:** `{"pogreska": "zabranjen pristup"}`

U svim zahtjevima gdje je uspješno odrađen zahtjev koji nešto dohvaća vraća se statusni kod **200** i **odgovarajući podaci**. U svim zahtjevima gdje je uspješno odrađen zahtjev koji ništa ne vraća npr. kreiranje, ažuriranje i brisanje vraća se u tijelu `{"izvršeno": "ok"}`

Tablica 2. Pregled resursa, metoda i parametara REST servisa

Resurs Parametri	GET čitanje	POST kreiranje	PUT ažuriranje	DELETE brisanje
rest/korisnici	vraća sve korisnike (200)	kreira novog korisnika (200)	“zabranjeno” (405)	“zabranjeno” (405)
rest/korisnici /{korime}	vraća podatke određenog korisnika (200)	“zabranjeno” (405)	ažurira određenog korisnika (200)	“zabranjeno” (405)
rest/korisnici /{korime}/prijava	“zabranjeno” (405)	Provjerava podatke prijave, ako su ispravni kreira sesiju. (200)	“zabranjeno” (405)	“zabranjeno” (405)
rest/glumci?str={broj stranice}	Vraća podatke svih glumaca iz baze po stranicama (200)	Dodaje novog glumca (200)	“zabranjeno” (405)	“zabranjeno” (405)

rest/glumci/{id}	Vraća podatke traženog glumca (200)	“zabranjeno” (405)	“zabranjeno” (405)	“zabranjeno” (405)
rest/{vlastiti resurs}	Vraća sve podatke (200)	Dodaje novi podatak (200)	“zabranjeno” (405)	“zabranjeno” (405)
rest/{vlastiti resurs}/{id}	Vraća podatke jednog (200)	“zabranjeno” (405)	Ažurira podatke jednog(200)	Briše podatke jednog (200)

Opis rada Web aplikacije

U aplikaciji postoje 3 uloge korisnika (admin > korisnik > gost). Svaka uloga više razine ima pristup pogledima niže razine. U tablici 3 dan je opis funkcionalnosti po ulogama s aspekta korisnika. U nekim pogledima (navedeno u tablici 3) ako se dohvaća više podataka treba implementirati stranicenje koje radi na poslužiteljskoj strani, a klijentska strana dobiva samo podatke te stranice. Maksimalan broj elemenata po stranici definiran je s konfiguracijskim parametrom **appStranicenje**.

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 prijave. Ne smije se dozvoliti neovlašteni pristup stranicama.

Na stranicama koje koriste REST servis preporuča se napraviti tako da se stranica NE osvježava i da se koristi Fetch API za komunikaciju sa servisom sa klijentske strane.

Tablica 3. Pregled funkcionalnosti Web aplikacije

Uloga	Stranica/e	Opis
Gost	Početna	Prikazuje tražilicu glumaca uz pomoć servisa TMDB. Na početku nema ničega već samo jedno tekstualno polje, tek kad se unese barem 3 znaka počinju se slati zahtjevi prema TMDB servisu. Servis u svom odgovoru sadrži informaciju o broju stranica (total_pages) i broju nađenih glumaca (total_results). Servis ako ima puno podataka uvijek vraća uvijek jednaki broj glumaca po stranici osim zadnje stranice. Potrebno je prilagoditi broj vraćenih serija temeljem parametra appStranicenje. Na primjer ako servis vrati 20 zapisa i ima 10 stranica, a u appStranicenje piše 5 treba prikazati samo 5 zapisa i 40 stranica. Ako u appStrnicenje piše 40 treba prikazati 40 zapisa i 5 stranica. Kod prikaza se prikazuje ime i prezime (original_name) glumca, mala slika (profile_path) glumca. Ako je korisnik prijavljen može kliknuti na ime i prezime koje ga vodi na stranicu “Detalji”. Ako je prijavljen korisnik admin on vidi još i gumb “spremi” koji kada se klikne sprema informacije o glumcu u bazu podataka.
Korisnik	Detalji	Prikazuje detalje odabranog glumca ako ga ima u bazi podataka. Ako podataka o glumcu nema ispisuje poruku “Podataka nema” i jedan gumb “Zatraži dodavanje”. Klikom na gumb “Zatraži dodavanje” u bazu se dodaje zapis da se traži dodavanje nekog glumca u bazu. Potrebno je barem spremiti id glumca, id korisnika koji traži dodavanje i status “dodan” je po inicijalno “0”.

		Ako ima podataka o glumcu prikazuju se ime i prezime (original_name), slika glumca (profile_path), biografija (biography), lista alternativnih imena (also_known_as), vrsta (known_for_department), popularnost (popularity), datum rođendan (birthday), mjesto rođenja (place of birth), datum smrti (deathday), link na vanjsku stranicu (homepage) i popis naslova (title) filmova/serija za koje je poznat (known_for).
Gost	Dokumentacija	Prikazuje stranicu dokumentacija.html opisanu ranije u ovom dokumentu.
Gost	Prijava, Odjava, Registracija	Korisnik se može prijaviti ili odjaviti ili registrirati. Prijava je dvorazinska korisničko ime i lozinka + jednokratni kod (TOTP). Gost se može samostalno registrirati kao novi korisnik te kod registracije mora obavezno unijeti email, korisničko ime i lozinku. Lozinka se u bazu ne smije spremati u čitljivom obliku i treba koristiti sol kod spremanja. Prilikom registracije ako je uspješna registracija prikazati korisniku njegov tajni TOTP ključ uz napomenu da si ga spremi.
Korisnik	Profil	Vidi svoje podatke (ime, prezime, korisničko ime, e-mail, ...) i može ih ažurirati. Ne prikazuju se lozinka i TOTP tajni ključ. Korisničko ime, e-mail, TOTP tajni ključ i lozinka se <u>ne mogu ažurirati</u> .
Admin	Korisnici	Vidi podatke svih korisnika.
Admin	Zahtjevi	Vidi podatke o zahtjevima za dodavanjem glumca gdje je statusni kod "dodan" jednak "0". Kod popisa vidi korisničko ime korisnika koji je zatražio dodavanje, vidi ime i prezime glumca koji se traži i gumb "Spremi". Kada klikne gumb "spremi" sprema informacije o glumcu u bazu podataka, a status "dodan" postaje automatski "1".

Dodatne mogućnosti:

1. Za dodatne bodove korištenje TypeScript-a kroz cijeli projekt.
2. Za dodatne bodove korištenje Angular okvira za klijentsku stranu.