

Zadaća 1 - Razvoj web aplikacije TV serije

Poslužitelj: spider.foi.hr

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

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

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

Struktura 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 zadaća; ako se utvrdi da je zadaća prepisana od drugog/ih studenta **svako dobiva 0 bodova**.

Bodovanje: Točna skala bodova nije unaprijed poznata i određuje se tek nakon što završi predaja zadaće. Boduju se dijelovi koji su rađeni samostalno nakon vježbi. U bodovanje, uz činjenicu radi li aplikacija prema specifikaciji ulazi: kvaliteta ispunjenosti ankete, kvaliteta programskog koda, model baze podataka, dizajn aplikacije, HTML/CSS validacija, količina grešaka u implementaciji, kvaliteta provjere unosa podataka od strane korisnika (sigurnost) i sl. 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.

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

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

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

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

Baza podataka: Svaki student/ica 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 treble imati više od 35 linija programskog koda programskog koda, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli, prazna linija, linija samo s { ili }. U jednoj liniji nalazi se jedna instrukcija. Linija nema više od 120 znaka. Radi se uvlačenje koda s 4 znaka space ili 1 tab. Nazivi klasa/funkcija/varijabli su smisleni i oslikavaju svrhu klase/funkcije/varijable. **Nazivi datoteka, direktorija, klasa, funkcija, varijabli i sl. pišu se na hrvatskom jeziku.** Naravno principi čistog koda koji su spomenuti na OWT za HTML i CSS također treba koristiti.

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

Datum i vrijeme: Svi datumski podaci moraju se na ekranu prikazivati i unositi u obliku dan.mjesec.godina 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 kontrolira kako ne bi bilo moguće srušiti aplikaciju krivim unosima ili napraviti napad. Preporuka je korištenje regularnih izraza jer nosi više bodova nego varijanta s if-ovima.

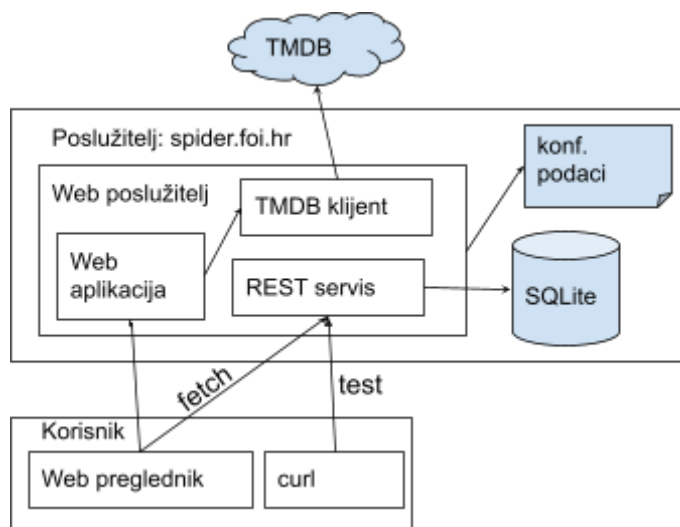
Testiranje: aplikaciju/servis treba testirati i s krivim podacima, provjeriti rubne slučajeve, a ne samo idealni scenarij korištenja. Kod testiranja servisa predlažemo curl naredbu jer možete lako više puta izvršiti. Također radi lakšeg testiranja bolje je napraviti prvo servisni dio, a tek onda aplikativni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

Korisne TMDb adrese:

- Registracija: <https://www.themoviedb.org/signup>
- API ključ: <https://www.themoviedb.org/settings/api>
- Dokumentacija: <https://www.themoviedb.org/settings/api>
- API bazični URL: <https://api.themoviedb.org/3/>
- Traženje serija: <https://developer.themoviedb.org/reference/search-tv>
 - https://api.themoviedb.org/3/search/tv?query=game&include_adult=false&language=en-US&page=1
 - https://api.themoviedb.org/3/search/tv?query=game%20of%20thrones&include_adult=false&language=en-US&page=1
- Svi podaci o seriji: <https://developer.themoviedb.org/reference/tv-series-details>
 - <https://api.themoviedb.org/3/tv/1399>
- Putanje do slika: https://image.tmdb.org/t/p/original/{poster_path}
<https://image.tmdb.org/t/p/original/jXIMScXE4J4EVHUb1JgxZnWbo4.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 kojem se može pristupiti ako imamo validan JWT. REST servis omogućuje rad s vlastitom SQLite bazom podataka. Web aplikacija pruža korisničko sučelje za rad s TV serijama, favoritima, korisnicima i sl. Informacije o serijama dobija 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 pozadinskom dijelu ne komunicira direktno sa bazom niti REST servisom već komunikacija prema REST servisu ide iz klijentskog dijela web aplikacije uz korištenje 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:

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
jwtValjanost	Broj od 15 - 3600	Sadrži broj sekundi koliko vrijedi JWT
jwtTajniKljuc	Veličina: 50 - 100 znakova Dozvoljava: velika i mala slova te brojke	Tajni ključ za JWT
tajniKljucSesija		Tajni ključ za sesiju
appStranice	Broj od 5 - 100	Sadrži brojevi 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 **“baza”**. **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"}]`

Kod svih zahtjeva **osim prijave** mora se slati validan JWT u čijem tijelu se nalazi korisničko ime prijavljenog korisnika. Ako JWT nije poslan ili nije validan o vraća se statusni kod **401** sa opisom **“potrebna prijava”**. Primjer scenarija opisan je nakon Tablice 2. Tajni ključ i trajanje za JWT se čitaju iz konfiguracijske datoteke. Nakon što je provjereno da je JWT validan iz njega se čita korisničko ime (čitanje iz sesije se ovdje ne boduje) i u bazu podataka sprema dnevnički zapis (eng. log) sa informacijama: datum, vrijeme, korisničko ime, vrsta zahtjeva (GET, POST, PUT, DELETE), traženi resurs (putanja sa parametrima) i tijelo (ako postoji u protivom se upisuje NULL).

Nakon toga obrađuje se zahtjev prema specifikaciji u Tablici 1. Na mjestima u tablici 1 gdje je prazno (bez opisa) vraća se statusni kod **501** sa opisom **“metoda nije implementirana”**. Ako metodi smije pristupi samo administrator servis vraća statusni kod **403** sa opisom **“zabranjen pristup”**. 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 se pokuša dohvatiti ispravni resurs, ali s krivim parametrima u url adresi (npr. baza/dnevnik?krivo) tada se vraća statusni kod **417** sa opisom **“neočekivani podaci”**. 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 “opis” npr. ovako: {"opis":"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 statusni kod **201** i u tijelu `{"opis":"izvršeno"}`

Tablica 2. Pregled resursa, metoda i parametara REST servisa

Resurs Parametri	GET čitanje	POST kreiranje	PUT ažuriranje	DELETE brisanje
baza/korisnici	vraća sve korisnike (200)	kreira novog korisnika (201)		
baza/korisnici /{korime}	vraća podatke određenog korisnika (200)	“zabranjeno” (405)	ažurira određenog korisnika (201)	bríše jednog korisnika (201)

baza/korisnici /{ <i>korime</i> }/prijava NE TRAŽI VALIDAN JWT	Vraća JWT token u zaglavlju ako postoji sesija (201). Ako ne postoji sesija javlja “zabranjen pristup” (401)	Provjerava podatke prijave, ako su ispravni kreira sesiju. (201)		
baza/favoriti	Vraća sve podatke serija spremljenih u favorite prijavljenog korisnika (200)	Dodaje novu seriju u favorite prijavljenog korisnika (201)		
baza/favoriti/{ <i>id</i> }	Vraća sve podatke tražene serije u favoritima prijavljenog korisnika (200)	“zabranjeno” (405)	“zabranjeno” (405)	Briše seriju iz favorita (201)
baza/dnevnik ?stranica= <i>broj</i> &sortiraj=[<i>d</i> <i>m</i>] [&datumOd= <i>datum</i>] [&datumDo= <i>datum</i>] [&vrijemeOd= <i>vrijeme</i>] [&vrijemeDo= <i>vrijeme</i>]	<i>vraća podatke loga za neku stranicu (<i>broj</i>), opcionalno filtrirano po vremenskom periodu (moguće sve kombinacije datuma i vremena od ili do), te podatke dobiva sortirano silazno po “datumu i vremenu” (<i>d</i>) ili uzlazno po metodi (<i>m</i>) (200)</i>			

Primjer scenarija pristupanja resursu baza/favoriti:

1. Korisnik se prijavljuje preko POST metode s krivim korisničkim imenom i lozinkom
 - a. Server vraća odgovor 400
2. Korisnik se prijavljuje preko POST metode s ispravnim korisničkim imenom i lozinkom
 - a. Server kreira sesiju i vraća odgovor 201
3. Korisnik traži resurs npr. GET baza/favoriti bez JWT
 - a. Server provjerava JWT valjanost i vraća odgovor 401
4. Korisnik se prijavljuje preko GET metode i ne šalje nikakve podatke
 - a. Server provjerava se postoji li sesija
 - i. Ako ne vraća odgovor 401
 - ii. Ako da kreira JWT, dodaje ga u zaglavlje i vraća odgovor 201
5. Korisnik traži resurs npr. GET baza/favoriti sa JWT
 - a. Server provjerava JWT valjanost JWT
 - i. Ako je nevaljani vraća odgovor 401
 - ii. Ako je valjani vraća podatke sa odgovorom 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 prijavljenosti. Ne smije se dozvoliti neovlašteni pristup stranicama.

Na stranicama koje koriste REST servis (favoriti, korisnici) treba napraviti tako da se stranica NE osvježava i da se koristi Fetch API za komunikaciju sa servisom. Može se koristiti i na drugim stranicama, ali nije obavezno.

Tablica 3. Pregled funkcionalnosti Web aplikacije

Uloga	Stranica/e	Opis
Gost	Početna	Prikazuje tražilicu serija po nazivu. 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 nadenih serija (total_results). Servis ako ima puno podataka uvijek vraća uvijek jednaki broj serija 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 naziv (name) serije, opis (overview) i gumb na detalje serije.
Gost / Korisnik	SerijaDetalji	Prikazuje detalje odabrane serije. Prikazuju se naziv (name), opis (overview), broj sezone, broj epizoda, popularnost, slika (poster_path), poveznica na vanjsku stranicu (homepage). Ako je korisnik prijavljen dolje se nalazi gumb spremi koji sprema barem ove podatke (plus tmdb id) serije u bazu ako ne postoji kao i informacije o sezonama (minimalno: naziv (name), opis (overview), slika (poster_path), broj sezone, broj epizoda u sezoni, tmdb id sezone). Te nakon toga dodaje seriju u favorite tom korisniku. Ako serija postoji u bazi tada ju samo dodaje u favorite korisniku.
Gost	Dokumentacija	Prikazuje stranicu dokumentacija.html opisanu ranije.
Gost	Prijava, Odjava	Korisnik se može prijaviti ili odjaviti.
Korisnik	Profil	Vidi svoje podatke (ime, prezime, korisničko ime, e-mail, ...) i može ih ažurirati. Korisničko ime i e-mail se <u>ne mogu ažurirati</u> .
Korisnik	Favoriti	Vidi popis serija koje ima u favoritima u bazi. Klikom na seriju može vidjeti detalje svog favorita.

Korisnik	FavoritDetalji	Prikazuje sve informacije o seriji i informacije o svakoj sezoni serije koje ima u bazi. Slike se prikazuju kao slike ne samo putanje iz baze.
Admin	Registracija	Gost se ne može samostalno registrirati kao novi korisnik već to radi admin. Kod dodavanja mora obavezno unjeti email, korisničko ime i lozinku. Opcionalno može unijeti ime, prezime i još barem 3 druga podatka po vlastitom odabiru. Korisnik admin mora biti unaprijed dodan u bazu. Kod registracije šalje se e-mail korisniku sa informacijom da je registrian i koje mu je korisničko ime i lozinka. Lozinka se u bazu ne smije spremati u čitljivom obliku i treba koristiti sol kod spremanja.
Admin	Korisnici	Vidi podatke svih korisnika i može obrisati nekog korisnika, pri čemu se brišu svi njegovi favoriti. Glavni korisnik admin se ne može obrisati i to treba blokirati.
Admin	Dnevnik	Vidi pregled dnevnika u obliku tablice. Podaci su inicijalno sortirani po datumu silazno. Može sortirati podatke uzlazno po metodi (GET, POST, PUT, DELETE). Može podatke filtrirati pod datumuOd, vremenuOd, datumuDo i/ili vremenuDo. Bilo koja kombinacija filtriranja je moguća, ako ništa nije uneseno dobiju se svi podaci, a svaki filter radi veću restrikciju. Potrebno je napraviti straničenje.

Zadaća 2 - Razvoj web aplikacije TV serije

Poslužitelj: spider.foi.hr

Korijenska mapa zadatka: /var/www/RWA/2023/zadaca_02/{LDAP_korisnicko_ime}/

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

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

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 zadata se testira lokalno. Zip sadrži kada se raspakira sljedeću osnovnu strukturu:

- {LDAP_korisnicko_ime}
 - server
 - **aplikacija** - direktorij sadrži REST servis prema opisu zadatka 1 i ostale pozadinske stvari koje poziva angular aplikacija
 - **angular** - direktorij sadrži kompajlirane angular datoteke koje poslužuje web aplikacija odmah na korijenskoj putanji /
 - **dokumentacija** - direktorij sadrži html stranice dokumentacije koje poslužuje web aplikacija na putanji /**dokumentacija** - moguće je dokumentaciju ugraditi u samu angular aplikaciju.
 - **konfiguracija.csv** - konfiguracijska datoteka koju koriste servis i aplikacija kod pokretanja
 - **RWA2023{LDAP_korime}.sqlite** - sama SQLite baza podataka sa podacima
 - **package.json** - npm konfiguracija - koristi se za pokretanje servera sa npm start
 - **angular** - sadrži originalni kod klijentske aplikacije - koji se koristi za ručno kompajliranje i bodovanje te se može koristiti za pokretanje live servera sa *ng serve*
 - **.angular** - ne treba ga predavati, ali ako postoji mora biti prazan
 - **.vscode** - ne treba ga predavati, ali ako postoji mora biti prazan
 - **node_modules** - ne treba ga predavati, ali ako postoji mora biti prazan
 - **projekti** - sadrži isključivo angular projekt zadatka 2, a ne druge angular projekte
 - **angular.json** - konfiguracija mora sadržavati samo jedan projekt zadatka 2
 - **tsconfig.json** - konfiguracija za TypeScript
 - **package.json** - konfiguracija za npm
 - **package-lock.json** - obrisati

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

ostvarenih bodova. Poslužitelj **spider.foi.hr** prepoznaje velika i mala slova (eng. case sensitive) kod naziva direktorija, naziva datoteka i ekstenzija.

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

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

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

Pomoć i pitanja: Sva pitanja u vezi zadaće postavljaju se u **poseban forum za zadaću na sustavu Moodle**. Nema privatnih konzultacija u vezi zadaće osim ako nastavnik na vaše pitanje u forumu ne kaže da dodete na konzultacije zbog specifičnosti problem. Preporučujemo da ne šaljete kod drugima, ne pokazujete kod drugima i ne radite zadaću zajedno iako ste cimeri ili se poznajete već 10 godina. **Zabranjeno je prepisivanje** zadaća; ako se utvrdi da je zadaća prepisana od drugog/ih studenta **svako dobiva 0 bodova**.

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

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

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

Okviri/moduli/biblioteke: U zadaći na strani klijenta i strani poslužitelja se smiju koristiti samo okviri/moduli/biblioteke obrađeni na nastavi. Konkretno za NodeJs **kod izvršavanja** mogu se koristiti oni moduli koji su instalirani na poslužitelju globalno i **ne smiju se instalirati dodatni moduli na razini klijenta**. **Iznimka modul za kreiranje QR koda ako postoji i ako se radi dio za dodatne bodove.** Popis dostupnih modula dobiva se sa: **npm -g list**

JavaScript klijentska strana: Koristi se isključivo Angular okvir.

Baza podataka: Koristi se SQLite baza koja mora biti pripremljena i predana za jedno sa projektom. Baza je ista kao u zadaći jedan i dozvoljena je modifikacija baze u odnosu na zadaću 1 ako treba.

Čisti kod: U zadaći se treba voditi principima čistog koda što ukratko znači sljedeće. Kod organizirati u funkcije, klase i module. Moduli sadrže klase, a klase sadrže funkcije koje čine smislenu cjelinu. Funkcije ne bi trebale imati više od 35 linija programskog koda programskog koda, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli, prazna linija, linija samo s { ili }. U jednoj liniji nalazi se jedna instrukcija. Linija nema više od 120 znaka. Radi se uvlačenje koda s 4 znaka space ili 1 tab. Nazivi klasa/funkcija/varijabli su smisleni i oslikavaju svrhu klase/funkcije/varijable. **Nazivi datoteka, direktorija, klasa, funkcija, varijabli i sl. pišu se na hrvatskom jeziku.** Naravno principi čistog koda koji su spomenuti na OWT za HTML i CSS također treba koristiti.

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

Datum i vrijeme: Svi datumski podaci moraju se na ekranu prikazivati i unositi u obliku dan.mjesec.godina 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 kontrolira kako ne bi bilo moguće srušiti aplikaciju krivim unosima ili napraviti napad. Preporuka je korištenje regularnih izraza jer nosi više bodova nego varijanta s if-ovima.

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

Korisne TMDB adrese:

- Registracija: <https://www.themoviedb.org/signup>
- API ključ: <https://www.themoviedb.org/settings/api>
- Dokumentacija: <https://www.themoviedb.org/settings/api>
- API bazični URL: <https://api.themoviedb.org/3/>
- Traženje serija: <https://developer.themoviedb.org/reference/search-tv>
 - https://api.themoviedb.org/3/search/tv?query=game&include_adult=false&language=en-US&page=1
 - https://api.themoviedb.org/3/search/tv?query=game%20of%20thrones&include_adult=false&language=en-US&page=1
- Svi podaci o seriji: <https://developer.themoviedb.org/reference/tv-series-details>
 - <https://api.themoviedb.org/3/tv/1399>
- Putanje do slika: https://image.tmdb.org/t/p/original/{poster_path}
<https://image.tmdb.org/t/p/original/jXIMScXE4J4EVHUb1JgxZnWbo4.jpg>

Zadaci

1. Na stranicu dokumentacija ažurirati tablice iz prve zadaće oko informacija koji dijelovi su napravljeni za servis odnosno aplikaciju. Ako je nešto napravljeno djelomično treba to jasno napisati koji dio je napravljen. **Ako ovog dijela nema bodovi vezani uz web aplikaciju i/ili web servis nose 0 bodova.**
2. Na stranicu dokumentacija kreirati tablicu po koracima (počevši od zadaci općenito) ovog dokumenta u kojoj jasno pišu elementi opisa 2 zadaće koji su napravljeni, koji nisu. **Ako ovog dijela nema zadaća nosi 0 bodova.** Može se napraviti još jedna tablica sa dodatnim opisima ako se želi nešto naglasiti. Nije obavezno.
3. Složite package.json tako da se sa **npm start** može pokrenuti server i isto tako jedan package.json, tsconfig.json i angular.json da se može pokrenuti angular projekt.
4. Kod predavanja zadaće izbrisati svoje ključeve za TMDB servis iz konfiguracijske datoteke svaki student ubacuje svoje ključeve u konfiguracijsku datoteku kod testiranja.
5. Klientski dio Web Aplikacije HTML, CSS i JS na strani klijenta prebaciti u Angular sa TypeScript-om.
6. Pozadinski dio web aplikacije posluhuje kompajliranu verziju Angular aplikacije. Moguće je pokretanje Angular aplikacije preko live servera sa ng serve, ali ta varijanta nosi manje bodova.
7. Pozadinski dio web aplikacije posluhuje dokumentaciju - moguće je dokumentaciju ugraditi u samu angular aplikaciju.
8. Pozadinski dio web aplikacije nudi podršku u obliku REST servisa. REST servis sadrži sve metode kako je to bilo u prvoj zadaći za rad s bazom podataka. Dodatno svi ostali pozadinski dijelovi prve zadaće (npr. komunikacija sa TMDB servisom) koji nemaju direktne veze s bazom se sada tretiraju kao proširenje REST servisa jer klijentski dio aplikacije sada mora slati sve preko fetch zahtjeva. Angular aplikacija nikad ne komunicira direktno sa TMDB servisom već preko pozadinskog dijela kao u prvoj zadaći.
9. Sada svi pozadinski dijelovi osim inicialnog posluživanja Angular aplikacije i prijave traže validan JWT.
10. Informacije o bazičnoj adresi i portu za pozadinski dio koji koristi Angular aplikacija kod slanja fetch zahtjeva čitaju se iz environment datoteke. Hard kodirana varijanta nosi manje bodova.
11. Potrebno je doraditi dizajn stranice ako u prvoj zadaći niste vodili o tome previše računa. Dizajn treba prebaciti korištenjem Sass preprocesora u Sass ili Scss sintaksi. **Čisti CSS ili bootstrap varijante nose 0 bodova.**
12. Dodati reCaptcha v3 sustav na minimalno 3 obrazaca od kojih su obavezni: prijava, registracija i ažuriranje profila.
13. Potrebno je doraditi ažuriranje profila sa opcijom da se uključi dvorazinska autentifikacija. Prilikom prvog uključivanja dvorazinske autentifikacije potrebno je kreirati tajni ključ koji se sprema u bazu i prikazuje korisniku utekstualnom obliku ili u obliku **QR koda (za dodatne bodove)**. Korisnik može isključiti dvorazinsku autentifikaciju ili ponovo uključiti, ali više nema prikaza tajnog ključa.
14. Potrebno je doraditi prijavu da se autentifikacija radi u dvije razine. Prvo korisničko ime i lozinka te nakon toga TOTP ako je uključena dvorazinska autentifikacija za tog korisnika.

15. Potrebno je doraditi prijavu da omogućuje mogućnost prijave sa OAuth putem GitHub računa. Takav korisnik nema TOTP.
16. Korisnik koji se prijavi preko GitHub računa ne radi ništa s bazom podataka. Kod takvog korisnika spremanje favorita se radi u preglednikov local storage. Jednako tako pregled favorita i favoritDetalji se čitaju iz local storage u pregledniku. Stranica profila nije dostupna.
17. Inicijalno mora postojati korisnik iz svake uloge koji nema TOTP i treba navesti podatke na stranici prijava radi lakšeg testiranja.
18. Za dodatne bodove može se prebaciti cijeli pozadinski dio u TypeScript. Kod pokretanja sa npm start obavezno se mora napraviti recompile TypeScript-a i kopiranje svih bitnih statičnih sadržaja da aplikacija normalno radi.