

Zadaća 2 - Razvoj web aplikacije Filmoteka 2. dio

Poslužitelj: spider.foi.hr

Korijenska mapa zadatka: {LDAP_korisnicko_ime}

Pristup preko web preglednika do zadatka: http://localhost:port

Portovi APP: /var/www/RWA/2022/portovi.js - ako se testira na spideru

Portovi REST: /var/www/RWA/2022/portovi_rest.js - ako se testira na spideru

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

Način predaje: Rješenje postaviti na moodle u radionicu u **zip** obliku do dogovrenog roka. Radi jednostavnosti ova zadaća se testira lokalno. Zip sadrži kada se raspakira sljedeću osnovnu strukturu:

- {LDAP_korisnicko_ime}
 - server
 - servis - direktorij sadrži REST servis api prema opisu zadatka 1
 - **package.json** - npm konfiguracija - koristi se za pokretanje servisa sa *npm start*
 - aplikacija - direktorij sadrži web aplikaciju koja poslužuje: rest api koji nema u servis dijelu, angular aplikaciju i dokumentaciju
 - **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.
 - **package.json** - npm konfiguracija - koristi se za pokretanje aplikacije sa *npm start*
 - **konfiguracija.csv** - konfiguracijska datoteka koju koriste servis i aplikacija kod pokretanja
 - **konstante.js** - sadrži konstante koje koriste u projektima servis i aplikacija
 - **baza.sqlite** - sama SQLite baza podataka sa podacima
 - **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

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 dođete 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**. Popis dostupnih modula dobiva se sa: **npm -g list**

JavaScript klijentska strana: Koristi se Angular okvir.

Baza podataka: U ovoj zadaći 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.

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

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

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

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

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

Korisne TMDb adrese:

- Registracija: <https://www.themoviedb.org/signup>
- API ključ: <https://www.themoviedb.org/settings/api>
- Dokumentacija: <https://www.themoviedb.org/settings/api>
- API bazični URL: <https://api.themoviedb.org/3/>
- Lista žanrova: <https://developers.themoviedb.org/3/genres/get-movie-list>
https://api.themoviedb.org/3/genre/movie/list?api_key=xxx
- Traženje filmova: <https://developers.themoviedb.org/3/discover/movie-discover>
https://api.themoviedb.org/3/discover/movie?api_key=xxx&language=en-US&sort_by=popularity.desc&include_adult=false&include_video=false&page=1&with_keywords=love
- Svi podaci o filmu: <https://developers.themoviedb.org/3/movies>
https://api.themoviedb.org/3/movie/500?api_key=xxx&language=en-US
- Putanje do slika: https://image.tmdb.org/t/p/w600_and_h900_bestv2/{poster_path}
https://www.themoviedb.org/t/p/w600_and_h900_bestv2/1yppMeTNQwDrzaUH4dRCx4mr8We.jpg

Zadaci općenito

1. Na stranicu dokumentacija kreirati tablicu u kojoj jasno pišu resursi iz prvog dijela (REST servis) su napravljene, a koje nisu prema opisu tablice zadatke 1. Ako je nešto napravljeno djelomično treba to jasno napisati koji dio je napravljen. **Ako ovog dijela nema prva aplikacija nosi 0 bodova.**
2. Na stranicu dokumentacija kreirati tablicu u kojoj jasno pišu pogledi iz drugog dijela (APP) koji su implementirani, a koje nisu prema opisu tablice zadatke 1. Ako je nešto napravljeno djelomično treba to jasno napisati koji dio je napravljen. **Ako ovog dijela nema druga aplikacija nosi 0 bodova.**
3. Na stranicu dokumentacija kreirati tablicu po koracima (počevši od zadaci općenito) ovog dokumenta u kojoj jasno pišu elementi opisa 2 zadatke koji su napravljeni, koji nisu. **Ako ovog dijela nema prva i druga aplikacija nosi 0 bodova.**
4. Može se napraviti još jedna tablica sa dodatnim opisima ako se želi nešto naglasiti. Nije obavezno.
5. U obje aplikacije složiti package.json tako da se sa **npm start** može pokrenuti pojedini server. Kod kreiranja start skripte obavezno definirati putanju na NODE_PATH gdje piše putanja do globalnih modula kako je to rađeno na predavanjima i vježbama. Student/ica kod testiranja tada može lako promijeniti tu putanju.
6. Kod predavanja izbrisati svoje ključeve za TMDb servis iz konfiguracijske datoteke svaki student ubacuje svoje ključeve u datoteku kod testiranja.

Zadaci REST servis

7. REST servis funkcionalno ostaje isti međutim potrebno je napraviti sljedeće izmjene:
 - a. Da radi sa SQLite bazom podataka
 - b. Da koristi port koji se čita iz konfiguracijske datoteke pod nazivom: **rest.port**
8. REST servis može dodatno imati parameter kod resursa api/filmovi koji omogućuje dohvat samo odobrenih ili ne odobrenih filmova. Nije obavezno.
9. REST servis ne korigira više username i password već prima JWT i tako validira input. Varijanta sa username i password nosi manje bodova.
10. **Za dodatne bodove može se prebaciti cijeli REST servis u TypeScript. Kod pokretanja sa npm start obavezno se mora napraviti recompilacija TypeScript-a i korpitanje svih bitnih statičkih sadržaja da aplikacija normalno radi.**

Zadaci Web aplikacija

11. Klientski dio Web Aplikacije HTML, CSS i JS na strani klijenta prebaciti u Angular sa TypeScript-om.
12. Informacije o portovima na servise koje koristi Angular aplikacija čita iz environment datoteke. Hard kodirana varijanta nosi manje bodova.
13. Dizajn treba prebaciti korištenjem Sass preprocesora u Sass ili Scss sintaksi. **Čisti CSS ili bootstrap varijante nose 0 bodova.**
14. Dodati reCaptcha v3 sustav na minimalno 3 obrazaca od kojih su obavezni: prijava, registracija i jedan po izboru.
15. Pozadinski dio web aplikacije poslužuje Angular web aplikaciju. Moguće je pokretanje angular aplikacije preko live servera sa ng serve, ali ta varijanta nosi manje bodova.
16. Pozadinski dio web aplikacije poslužuje dokumentaciju - moguće je dokumentaciju ugraditi u samu angular aplikaciju.
17. Pozadinski dio web aplikacije nudi REST api podršku za klijentsku stranu sa resursima koji ne postoje u REST servisu npr. autentifikacija korisnika i generiranje JWT-a.
18. Pozadinski dio port se mora čitati iz konfiguracijske datoteke iz podatka: **app.port**
19. **Izbaciti TOTP i aktivaciju** mailom kod prijave. Inicijalno mora postojati korisnik iz svake uloge koji nema TOTP i već je aktiviran i treba navesti podatke na stranici prijava radi lakšeg testiranja.
20. Angular aplikacija smije direktno komunicirati sa REST servisom, ali mora obavezno slati JWT. Varijanta bez JWT-a nosi manje bodova.
21. **Za dodatne bodove može se prebaciti cijeli pozadinski dio web aplikacije u TypeScript. Kod pokretanja sa npm start obavezno se mora napraviti recompilike TypeScript-a i korpitanje svih bitnih statičnih sadržaja da aplikacija normalno radi.**