

Završni ispit iz Programiranja za veb – Jun2 2019/20

06.07.2020.

Ispit se polaže najviše 180 minuta. Raspodela poena je data u narednoj tabeli:

	Zadaci		Teorija										
Redni broj	1	2	1	2	3	4	5	6	7	8	9	10	Suma
Najviše poena	15	20	5	5	5	5	5	5	5	5	5	5	70*

*Poeni za teoriju se skaliraju sa ukupnih 50 na 35.

Da bi se položio završni ispit, potrebno je ostvariti najmanje: 5 poena na Node.js zadatku, 10 poena na Angular zadatku i 12 poena na teorijskom delu. Rešenja zadataka koja imaju sintaksnih grešaka nose 0 poena. Kod pod komentarima neće biti pregledan.

Teorijska pitanja

Uputstvo: U radnom direktorijumu na *Desktop*-u nalazi se datoteka *PrezimeIme_alasNalog_grupa_teorija.txt* u kojoj se nalaze teorijska pitanja. U ovoj datoteci ćete smestiti odgovore na pitanja. Obavezno preimenovati datoteku da sadrži Vaše informacije. Obavezno pratiti sve instrukcije koje su date u datoteci.

Napomena za zadatke

Ukoliko neki zahtevi u tekstovima zadataka mogu da se urade na više načina, onda:

- Ako je tekst u zahtevu **podebljan**, to znači da je potrebno da se taj zahtev implementira na opisan način.
- U suprotnom, ostavljeno je Vama da odlučite na koji način ćete implementirati. Slično, ako neki deo zahteva nije eksplicitno naveden (na primer, ako nije naglašeno da se neki formular validira ili da se vrši obrada u slučaju grešaka), onda se podrazumeva da ne morate da vodite računa o tome u Vašim rešenjima.

Zadaci

1. Napisati Node.js aplikaciju koja ispunjava naredne zahteve:

- U MongoDB bazi podataka *pueb_jul_2020* nalazi se naredna kolekcija:
 - Kolekcija *songs* sadri informacije o pesmama iz plejliste: jedinstveni identifikator pesme (*_id*), ime izvođača (*artist*), naziv pesme (*title*) i broj puta koliko je ta pesma bila poslušana (*count*).
- Kreirati shemu korišćenjem Mongoose biblioteke na osnovu informacija iz date kolekcije. **Podesiti u shemi da su *artist*, *title* i *count* obavezna polja. Podesiti u shemi da polje *count* ima podrazumevanu vrednost 1.** Na osnovu ove sheme kreirati model *Song*.
- Korišćenjem Express i Mongoose biblioteka implementirati naredni REST API:
 - GET */api/songs/{sortBy}*
 - * Značenje: Dohvatanje svih pesama. Urediti pesme rastuće prema parametru *sortBy*, koji može biti *artist*, *title* ili *count*.
 - * Odgovor: JSON reprezentacija niza pesama i statusni kod 200.
 - PATCH */api/songs/*
 - * Značenje: Uvećavanje broja slušanja jedne pesme na osnovu identifikatora pesme koji se prosleđuje kroz telo zahteva.
 - * Odgovor: JSON reprezentacija izmenjene pesme i statusni kod 200.
 - PUT */api/songs/*
 - * Značenje: Dodavanje nove pesme. Telo zahteva sadrži naziv izvođača i naziv pesme.
 - * Odgovor: JSON objekat koji sadrži poruku. Ako postoji pesma u BP sa prosleđenim izvođačem i nazivom, vratiti 309 i poruku „The song already exists! Try again!“. Ako ne postoji, uneti novu pesmu sa datim podacima, vratiti 201 i poruku „The song has successfully been scrobbled!“.
- Voditi računa da se odgovarajući elementi serverske aplikacije nalaze u odgovarajućim datotekama.

- Sve greke proslediti funkciji srednjeg sloja koja radi na nivou greka (definisanog u *app.js* datoteci).

2. Napisati jednostraničnu Angular aplikaciju koja ispunjava naredne zahteve**:

- Slanje svih HTTP zahteva koji su navedeni ispod implementirati u jednom **Angular servisu**. Obezbediti postojanje tačno jedne instance ovog servisa na nivou cele aplikacije.
- U korenoj komponenti aplikacije prikazati element `<nav>` koji ima klasu *nav*. Ovaj element sadrži dva elementa `<a>` koji predstavljaju veze ka: (1) početnoj stranici i (2) stranici za unos nove pesme, redom. **Dodati ispod navigacije Angular objekat za rutiranje.**
- Na početnoj strani aplikacije *http://localhost:4200/* prikazati komponentu *PlaylistComponent* koja sadrži poruku i `<select>` listu, kao na slici 1. Ova lista ima 4 opcije, pri čemu je prva opcija prazna i podrazumevana. Elementu `<select>` dodeliti klasu *form-control*.
- Svaki put kada korisnik odabere novu opciju, poslati GET zahtev serverskoj aplikaciji sa URL parametrom odabrane opcije, osim ukoliko je odabrana prazna opcija (u tom slučaju, ne treba ništa uraditi). Nakon dohvaćanja podataka o pesmama sa servera, potrebno je svaku pesmu proslediti zasebnim komponentama *SongComponent*, kao na slici 2.
- Kreirati komponentu *SongComponent* koja prihvata podatak o jednoj pesmi i prikazuje je. Ova komponenta je predstavljena `<div>` elementom koji ima klase *alert alert-info song*. U njegovom sadržaju se nalaze 3 elementa *p* koji sadrže: (1) naziv izvođača, (2) naziv pesme i (3) broj slušanja, redom. Pored njih, element `<div>` takođe sadrži i dugme `<button>` sa tekстом „+1”.
- Klikom na neko od dugmadi „+1”, poslati PATCH zahtev serverskoj aplikaciji. U telu zahteva smestiti identifikator one pesme za koju je dugme kliknuto. **Iz odgovora pročitati novu vrednost broja slušanja, i ažurirati prikaz, kao na slici 3.**
- Klikom na vezu „Add new song” iz navigacije, otvoriti stranicu *http://localhost:4200/scrobble* koja prikazuje komponentu *NewSongComponent*. Ova komponenta sadrži **reaktivni formular** koji sadrži dva polja (videti sliku 4):
 - Polje *artist* predstavlja naziv izvođača. **Validacijom reaktivnih formulara** obezbediti da ovo polje bude popunjeno.
 - Polje *title* predstavlja naziv pesme. **Validacijom reaktivnih formulara** obezbediti da ovo polje bude popunjeno.

Klikom na dugme „Add a song” tipa *submit*, izvršiti podnošenje formulara na sledeći način:

- Proveriti da li je reaktivni formular validan.
- Ukoliko nije, prikazati poruku sa klijenta kao na slici 5 i prekinuti dalje podnošenje formulara.
- Inače, poslati PUT zahtev serverskoj aplikaciji. U telu zahteva smestiti podatke iz formulara (naziv izvođača i naziv pesme). U zavisnosti od odgovora sa servera:
 - * Ukoliko je korisnik uneo izvođača i naziv pesme za pesmu koja već postoji u bazi podataka, osigurati se da klijent dobija tu informaciju (videti sliku 6). Nije potrebno obrađivati ovu grešku na klijentu!
 - * Ukoliko je korisnik uspešno uneo novu pesmu, onda prikazati **poruku sa servera** u obaveštajnom prozoru, kao na slici 7 i očititi formular tako da prikaz bude kao na slici 4. Povratkom na početnu stranicu i ponovnim biranjem kriterijuma iz liste, očekuje se da novouneta pesma bude vidljiva, kao na slici 8.
- Voditi računa o curenju memorije.

**Da bi Vam bile dostupne klase koje su navedene iznad, u datoteci *klijent/src/index.html* u sadržaju elementa `<head>` potrebno je da dodate naredne linije:

```
<link rel="stylesheet" type="text/css" href="/assets/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="/assets/styles.css">
```

Home Add new song

Choose a criterium for sorting:

Sort by song artist

Sort by song title

Sort by listen count

Slika 1: Početna stranica.

Home Add new song

Choose a criterium for sorting:

Sort by song artist

Ariana Grande	Don't Call Me Angel (feat. Miley Cyrus & Lana Del Rey)	3	+1
Lady Gaga	Rain On Me (with Ariana Grande)	6	+1
Lady Gaga	1000 Doves	9	+1

Slika 2: Prikaz početne stranice nakon odabira opcije iz *select* liste (ovde je prikazano rastuće uređenje po izvođaču).

Home Add new song

Choose a criterium for sorting:

Sort by song artist

Ariana Grande	Don't Call Me Angel (feat. Miley Cyrus & Lana Del Rey)	4	+1
Lady Gaga	Rain On Me (with Ariana Grande)	6	+1
Lady Gaga	1000 Doves	9	+1

Slika 3: Prikaz početne stranice nakon klika na dugme „+1” za uvećanje broja slušanja odgovarajuće pesme.

Home Add new song

Add a new song to playlist!

Artist name:

Song title:

Add a song

Slika 4: Prikaz stranice za dodavanje nove pesme.

Home Add new song

Add a new song to playlist!

Artist name:

Song title:

The form is not valid!

OK

Slika 5: Prikaz poruke sa klijenta u slučaju podnošenja nevalidnog formulara.

Status	Met...	Domain	File	Initiat...	Type	Transferred	Size
200	POST	localhost...	xhr_streaming/t=1593718599597	polyfl...			
200	OPT...	localhost...	/api/songs/	xhr	json	358 B	2 B
409	PUT	localhost...	/api/songs/	polyfl...	json	343 B	49 B

3 requests 51 B / 701 B transferred Finish: 10.69 s

Filter properties

JSON

message: "The song already exists! Try again!"

Home Add new song

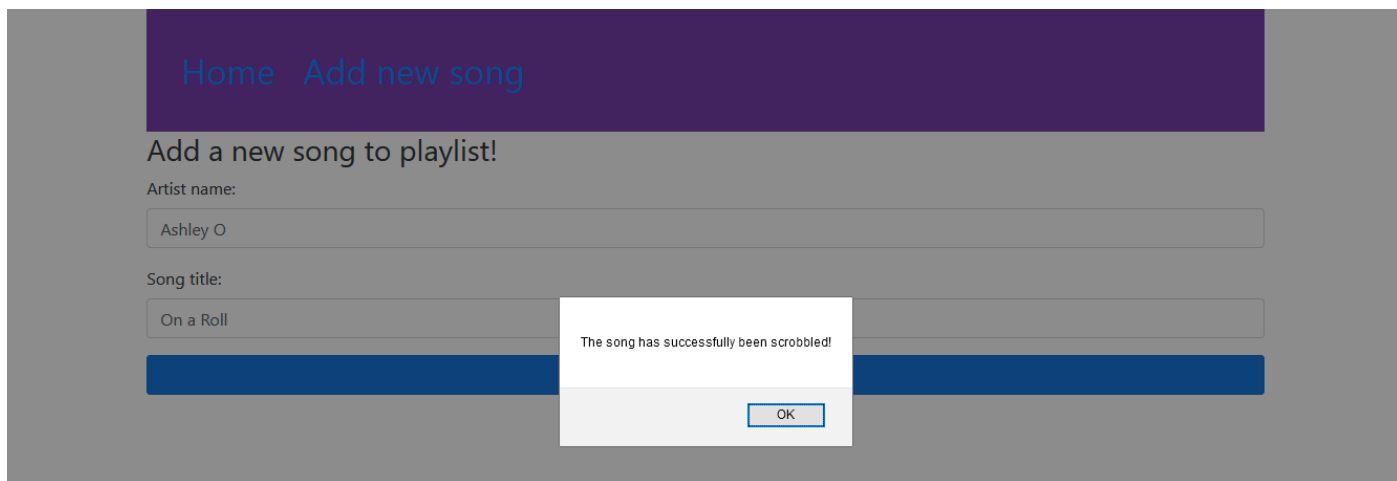
Add a new song to playlist!

Artist name:

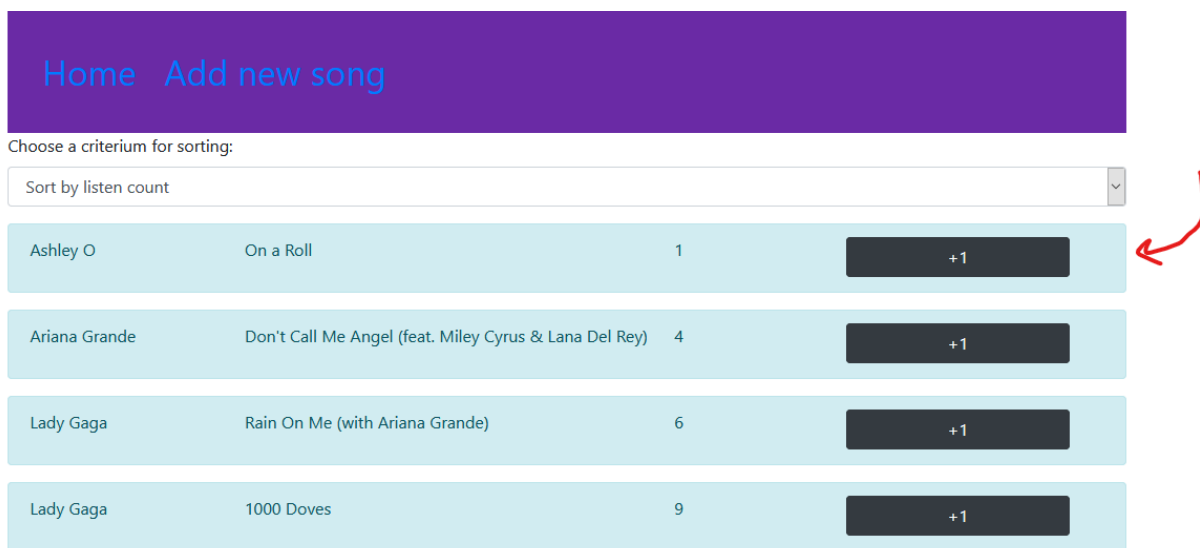
Song title:

Add a song

Slika 6: Prikaz poruke koja je dobijena sa servera u *alatima za razvoj veb pregledača* u slučaju da korisnik pokuša da unete pesmu koja već postoji. Na klijentu nije potrebno vršiti obradu ove greške: dovoljno je da klijent dobije ispravnu poruku od servera.



Slika 7: Prikaz poruke koja je dobijena sa servera u slučaju uspešno unete pesme.



Slika 8: Prikaz početne stranice nakon uspešnog unošenja nove pesme i odabira kriterijuma za uređenje. Ovoga puta, prikazano je uređenje po broju slušanja.