

# Završni ispit iz Programiranja za veb – Septembar 0 2020/21 – 1. grupa

## 27.08.2021.

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

Implementirati aplikaciju koja omogućava pregledanje događaja i stavki koje je potrebno uraditi za pripremu tih događaja.

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

- U MongoDB bazi podataka *pueb\_septembar0\_2021* nalazi se kolekcija *events*. Kolekcija *events* sadrži informacije o događajima opisane sledećim poljima:
  - jedinstvenim identifikatorom (*\_id*),
  - nazivom događaja (*title*, niska),
  - datumom održavanja (*date*, datum),
  - vremenom trajanja u minutima (*duration*, broj),
  - listom stavki koje je potrebno uraditi za pripremu tih događaja (*todo*, niz niski)

Korišćenjem Mongoose biblioteke kreirati shemu koja prati opisanu kolekciju. **Sva polja u shemi postaviti na obavezna.** Potom kreirati model *Event* za MongoDB bazu podataka.

- Korišćenjem Express i Mongoose biblioteka implementirati naredni REST API:
  - GET /api/events/month/{month}*
    - Značenje: dohvaćanje svih događaja koji se organizuju u mesecu *month*. Parametar *month* mora da predstavlja ceo broj u intervalu [0, 11] (0 označava januar, a 11 decembar). Za konkretan mesec, dovoljno je gledati da li je datum održavanja događaja u intervalu od 1. do 31. dana u mesecu. **Izvršiti projekciju podataka o događajima tako da se ne vraća niz stavki *todo*.**
    - Odgovor: JSON reprezentacija rezultujućeg niza u slučaju uspeha ili objekta greške u slučaju nevalidnosti zahteva. Statusni kod prilagoditi uspešnosti zahteva.
  - GET /api/events/{id}*
    - Značenje: dohvaćanje svih informacija o događaju (zajedno sa *todo* stavkama) čiji se identifikator prosleđuje kroz parametar *id*. Parametar *id* je obavezan i mora pripadati postojećem događaju iz kolekcije *events*.

- \* Odgovor: JSON reprezentacija rezultujućeg objekta događaja u slučaju uspeha ili objekta greške u slučaju nevalidnosti zahteva. Statusni kod prilagoditi uspešnosti zahteva.

– *PUT /api/events*

- \* Značenje: izmena događaja tako što se dodaje nova stavka koju je potrebno uraditi. Telo zahteva se sastoji od identifikatora događaja i nove stavke. Oba podatka u telu su obavezna i identifikator mora pripadati postojećem događaju iz kolekcije *events*.
- \* Odgovor: JSON reprezentacija ažuriranog objekta događaja u slučaju uspeha ili objekta greške u slučaju nevalidnosti zahteva. Statusni kod prilagoditi uspešnosti zahteva.

- Voditi računa da se odgovarajući elementi serverske aplikacije nalaze u odgovarajućim datotekama.
- Sve greške proslediti funkciji srednjeg sloja koja radi na nivou grešaka (definisanu 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. **Potpuno tipizirati zahteve**. (Pomoć: kako biste izbegli konfuziju oko imena iz raznih biblioteka, izbegavajte naziv *Event* za Vaše tipove.)
- Kreirati komponentu *EventsComponent* koja prikazuje događaje iz tekućeg meseca kao na slici 1. Komponenta se prikazuje otvaranjem stranice na adresi *http://localhost:4200/* (koristiti **Angular rutiranje**). Informacije o događajima se dohvataju slanjem odgovarajućeg zahteva servisa.

Naslov u komponenti prilagoditi tako da prikazuje naziv meseca u zavisnosti od tekućeg datuma. Jedan događaj je predstavljen elementom koji ima klasu *event* i sastoji se od narednih podelemenata, redom:

- Paragrafa sa klasom *date* koji prikazuje datum održavanja događaja (bez vremena).
- Paragrafa sa klasom *title* koji prikazuje naziv događaja.
- Paragrafa sa klasom *time* koji prikazuje vreme početka događaja.
- Paragrafa sa klasom *duration* koji prikazuje vreme trajanja događaja u minutima.
- Vezom ka stranici *http://localhost:4200/event/{id}* koja prikazuje detalje tekućeg događaja (koristiti **Angular rutiranje**). Parametar adrese *id* predstavlja identifikator tekućeg događaja.

- Kreirati komponentu *EventComponent* koja sadrži naslov događaja i listu sa stavkama koje je potrebno uraditi (slika 2) i dete komponentu *ToDoComponent* (videti ispod). Ova komponenta se prikazuje otvaranjem stranice na adresi *http://localhost:4200/event/{id}*. Informacije o događaju se dohvataju slanjem odgovarajućeg zahteva servisa.

Komponenta je predstavljena elementom sa klasom *container* i sastoji se od narednih podelemenata, redom:

- Paragrafa sa klasom *title* koji prikazuje naziv događaja.
- Paragrafa sa klasom *todo*.
- Nenumerisanom listom koja sadrži stavke koje je potrebno uraditi.
- Instancom komponente *ToDoComponent*.

- Kreirati komponentu *ToDoComponent* koja služi za dodavanje nove stavke koju je potrebno uraditi (slika 3).

Ova komponenta dobija na ulazu identifikator događaja **prosledjivanjem od roditeljske komponente** *EventComponent*.

**Reaktivni** formular se sastoji od jednog polja u kojem se unosi nova stavka. **Validacijom reaktivnih formulara** omogućiti da ovo polje bude obavezno i da stavka mora sadržati makar 10 karaktera. Ukoliko neki od ovih zahteva nije ispunjen, klikom na dugme *Add a new item* prikazati odgovarajuću poruku o grešci (slika 4). U suprotnom, poslati odgovarajući zahtev servisa za dodavanje nove stavke.

Zatim, u slučaju uspeha, resetovati formular i **proslediti roditeljskoj komponenti** *EventComponent* informacije o novom nizu stavki. Roditeljska komponenta treba da ažurira prikaz stavki. Ovo je ilustrovano na slikama 5 i 6.

- Voditi računa o curenju memorije.
- Stilski elementi zadatka se ne ocenjuju i navode se isključivo zbog kompletnosti zadatka.



\*\*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">
```

# Events in September

9/19/21

 [JavaScript conference 2021](#) 



 Starting at: 2:00 PM 



 Duration: 360 minutes 

[See details](#)

9/25/21

 [A state of modern web](#) 

 Starting at: 11:00 AM 

 Duration: 120 minutes 

[See details](#)

Slika 1: Komponenta *EventsComponent*.

 [JavaScript conference 2021](#) 

## TODO

- Code examples
- Prepare a presentation
- Practice a presentation

Todo item:

Add a new item

Slika 2: Komponenta *EventComponent*.

Todo item:

Add a new item

Slika 3: Komponenta *ToDoComponent*.

localhost:4200 says

The form is invalid

OK

## TODO

- Code examples
- Prepare a presentation
- Practice a presentation

Todo item:

Add a new item

Slika 4: Primer nevalidnog popunjavanja formulara i prikaz greške.



## TODO

- Code examples
- Prepare a presentation
- Practice a presentation

Todo item:

Add a new item

Slika 5: Primer validnog popunjavanja formulara.



## TODO

- Code examples
- Prepare a presentation
- Practice a presentation
- Submit a presentation to the conference organizers

Todo item:

Add a new item

Slika 6: Prikaz u komponentama *ToDoComponent* i *EventComponent* nakon što se uspešno dodala nova stavka.