

# Završni ispit iz Programiranja za veb – Septembar 3 2019/20

## 29.09.2020.

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

Implementirati aplikaciju koja omogućava kreiranje zadataka koje korisnik treba da uradi („kanbana”) i njihov progres.

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

- U MongoDB bazi podataka *pveb\_septembar3\_2020* nalazi se kolekcija *kanbans* koja sadrži informacije o karticama sa „to do” zadacima koje je potrebno uraditi. Svaka kartica je opisana narednim informacijama:
  - jedinstvenim identifikatorom unosa (*\_id*), naslovom (*title*), tekstualnim opisom zadatka (*description*), brojem koraka potrebnim za rešavanje (*numOfSubtasks*) i brojem do sada kompletiranih koraka (*completedTasks*).

Kreirati shemu korišćenjem Mongoose biblioteke na osnovu informacija iz date kolekcije. Sva polja u shemi postaviti na **obavezna** i **obezbediti da MongoDB podrazumevano dodeli vrednost polja *completedTasks* na 0**. Potom kreirati model *Kanban*.

- Korišćenjem Express i Mongoose biblioteka implementirati naredni REST API:
  - GET /api/kanban
    - Značenje: dohvaćanje svih kanban zadataka koji nisu kompletirani (tj. koji imaju makar 1 korak koji nije kompletiran). Rezultat sortirati po ukupnom broju koraka (*numOfSubtasks*) rastuće.
    - Odgovor: JSON reprezentacija rezultujućeg niza i statusni kod 200.
  - POST /api/kanban
    - Značenje: kreiranje novog kanbana na osnovu podataka koji su prosledjeni u telu zahteva. Ako je korisnik prosledio naziv kanbana koji već postoji u BP, onda je potrebno vratiti kod 400 i poruku „The kanban with the same title already exists!”. U suprotnom, vratiti kod 200 i poruku „The kanban is successfully created!”.
    - Odgovor: JSON reprezentacija objekta koji sadrži poruku i statusni kod. Statusni kod odgovora prilagoditi uspešnosti operacije.
  - PATCH /api/kanban/{id}

- \* Značenje: izmena kanbana čiji se identifikator prosleđuje kao parametar putanje *id* tako što se povećava broj kompletiranih zadataka za 1. Izvršiti proveru da li kanban sa zadatim identifikatorom postoji u BP. Ukoliko ne postoji, vratiti kod 404 i poruku „The kanban with id *id* does not exist!“. Inače, vratiti kod 200 i poruku „A progress was made on kanban with id *id*!“.
- \* Odgovor: JSON reprezentacija objekta koji sadrži poruku i statusni kod. Statusni kod odgovora prilagoditi uspešnosti operacije.

- Voditi računa da se odgovarajući elementi serverske aplikacije nalaze u odgovarajućim datotekama.
- Sve greške proslediti funkciji srednjeg sloja koja radi na nivou grešaka (definisanoj u *app.js* datoteci).

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

\*\*Da bi Vam bile dostupne klase koje su navedene ispod, 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">
```

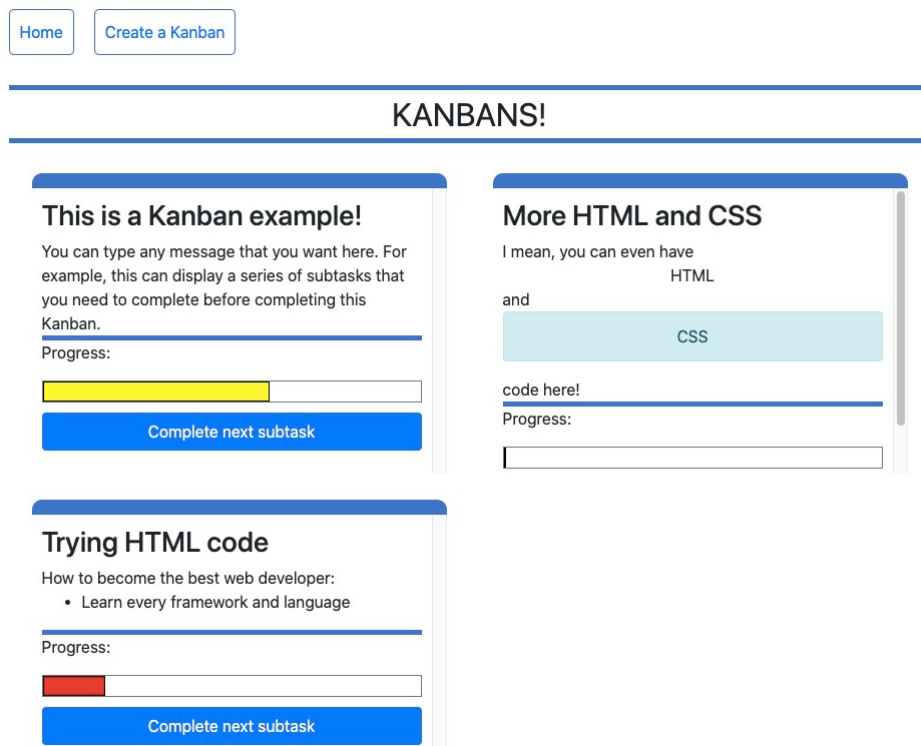
- Slanje svih HTTP zahteva koji su navedeni ispod implementirati u jednom **Angular servisu**. Obezbediti postojanje tačno jedne instance ovog servisa na nivou cele aplikacije.
- Omogućiti da se na svim stranicama aplikacije prikazuje:
  - Navigacioni element `<nav>` koji sadrži 2 veze: prva je ka početnoj stranici *http://localhost:4200/*, a druga je ka stranici *http://localhost:4200/add-a-kanban*. **Koristiti Angular rutiranje**.
  - Naslovni element `<header>` sa klasom *header*.

Prikaz je dat na slici 1.

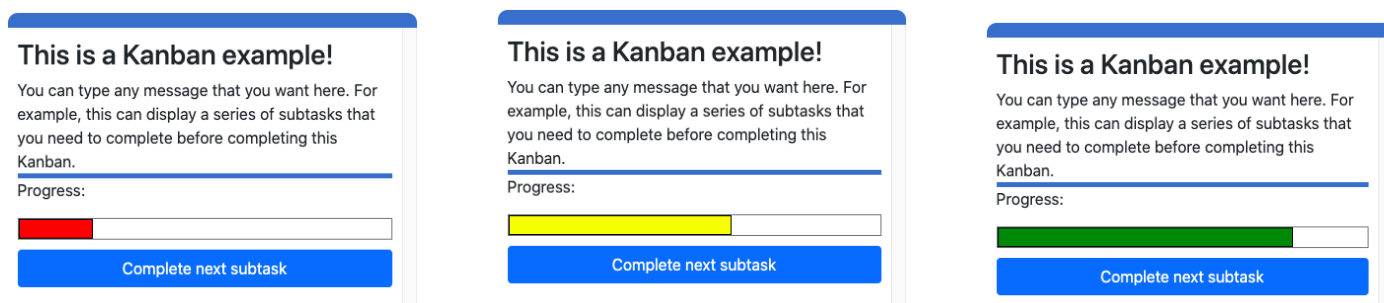
- Kreirati komponentu *KanbanListComponent* koja prikazuje listu kanban zadataka (slika 1) na početnoj stranici aplikacije sa adresom *http://localhost:4200/*. Učitavanjem početne stranice, poslati GET zahtev serverskom delu za dohvaćanje svih nekompletiranih kanbana.
- Kreirati komponentu *KanbanComponent* koja prikazuje jedan kanban iz liste (slika 1):
  - Svi podaci o kanbanu su smešteni u jedan grupišući element `<div>` sa klasom *kanban*. U okviru njega se nalazi: (1) naslov kanbana u elementu `<h3>`, (2) opis kanbana u grupišućem elementu `<div>`, (3) paragraf sa tekстом „Progress“, (4) linija progressa koja je opisana ispod i (5) dugme „Complete next subtask“.
  - Obezbediti da se opis kanbana prikazuje kao HTML sadržaj elementa `<div>`. Pomoć: u tu svrhu se može iskoristiti *vezivanje atributa innerHTML* za opis koji je dohvaćen od servera.
  - Liniju progressa prikazati kao element `<div>` sa klasama *border* i *border-secondary* u čijem sadržaju se nalazi drugi element `<div>`. Ovaj drugi element `<div>` je fiksne visine (*height*) *20px* i ivice (*border*) *1px solid black*. Obezbediti da širina (*width*) drugog `<div>` elementa bude jednaka procentu kompletiranih zadataka (*completedTasks*) u odnosu na ukupan broj zadataka (*numOfSubtasks*). U odnosu na ovaj procenat postaviti i boju pozadine (*background-color*). Ako je procenat < 30%, onda je boja crvena (*red*); inače, ako je procenat između 30% i 70%, onda je boja žuta (*yellow*); inače je zelena (*green*). Primer promene boje i širine linije progressa kanbana je dat na slici 2.
  - Klikom na dugme „Complete next subtask“, poslati PATCH zahtev serverskom delu sa identifikatorom tog kanbana. U slučaju uspešne operacije na serveru, ažurirati prikaz liste kanbana. U slučaju da se klikom na ovo dugme tekući kanban kompletirao, očekivano ponašanje je da on ne bude nadalje prikazan (slika 3).
- Kreirati komponentu *NewKanbanComponent* koja se prikazuje na stranici *http://localhost:4200/add-a-kanban* (slika 4). Komponenta sadrži **reaktivni formular** koji sadrži naredna polja: naslov (jednolinijsko tekstualno polje), opis (jednolinijsko tekstualno polje) i broj koraka (polje za unos broja).

**Validacijom reaktivnog formulara** obezbediti da: (1) sva polja u formularu moraju biti popunjena i (2) broj koraka mora biti najmanje 5. Formular se podnosi klikom na dugme „Create a Kanban“. Ukoliko podaci iz formulara nisu validni prilikom podnošenja, prikazati **alert** prozorčić sa porukom kao na slici 5. Ukoliko su uneti podaci korektni, šalje se POST zahtev serverskom delu aplikacije za unos novog kanbana. U slučaju uspešnog unosa novog kanbana, potrebno je prikazati **poruku sa servera** kao na slici 6 i resetovati formular. *Nije potrebno obrađivati greške u HTTP odgovoru sa servera.*

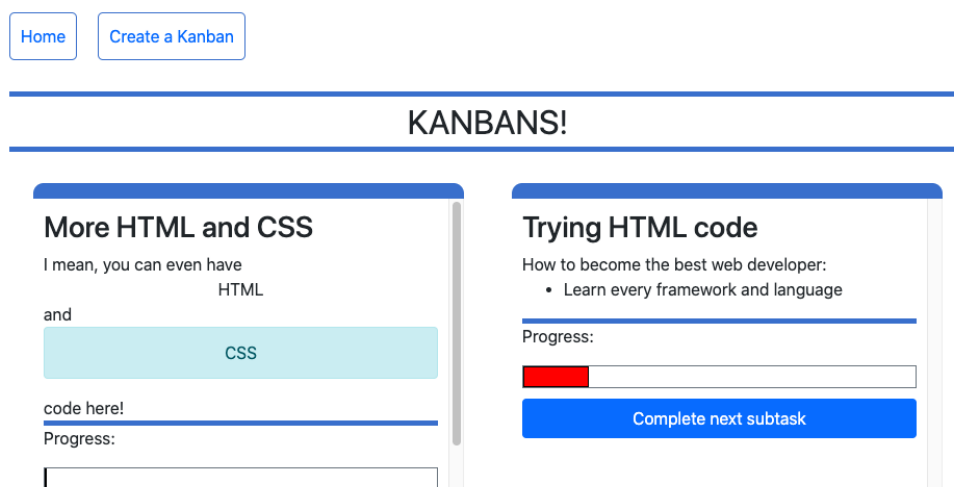
- Voditi računa o curenju memorije.



Slika 1: Početna stranica – komponente *KanbanListComponent* i *KanbanComponent*.



Slika 2: Progres jednog kanbana – komponenta *KanbanComponent*.



Slika 3: Prikaz kada se jedan kanban kompletira.

[Home](#)[Create a Kanban](#)

---

## KANBANS!

---

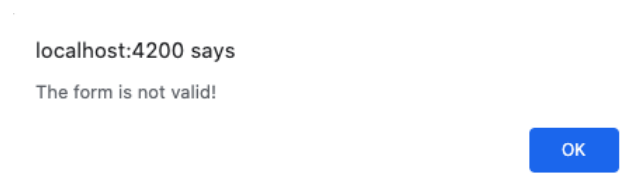
Title

Description

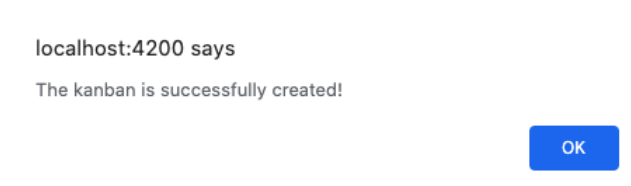
Number of subtasks

Create a Kanban

Slika 4: Stranica „Create a Kanban” – komponenta *NewKanbanComponent*.



Slika 5: Prikaz poruke sa klijenta o nevalidnom formularu prilikom podnošenja formulara za kreiranje novog kanbana.



Slika 6: Prikaz poruke sa servera o uspešnom kreiranju novog kanbana.