

# Završni ispit iz Programiranja za veb – Jun1 2019/20

## 25.06.2020.

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	20	15	5	5	5	5	5	5	5	5	5	5	70*

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

### 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\_jun\_2020* nalazi se naredna kolekcija:
  - Kolekcija *payments* sadrži informacije o uplatama za nekog korisnika: jedinstveni identifikator zadatka (*\_id*), ime korisnika (*name*), prezime korisnika (*surname*), njihovu adresu elektronske pošte (*email*) i niz iznosa njihovih uplata (*amounts*).

Kreirati shemu korišćenjem Mongoose biblioteke na osnovu informacija iz date kolekcije. Na osnovu ove sheme kreirati model *Payment*.

- Korišćenjem Express i Mongoose biblioteka implementirati REST API koji je opisan u narednoj tabeli:

Metod	Parametri	Značenje	Rezultat
GET	/api/payments/{email}	Dohvatanje jedne uplate na osnovu email-a.	JSON reprezentacija jedne uplate ili <i>null</i> ukoliko prosleđeni email ne postoji.
GET	/api/payments/{id}	Dohvatanje jedne uplate na osnovu identifikatora.	JSON reprezentacija jedne uplate koja sadrži samo informacije o identifikatoru ( <i>_id</i> ) i uplatama ( <i>amounts</i> ). Vratiti 404 ako ne postoji identifikator.
PATCH	/api/payments	Dodavanje novog iznosa uplate u niz uplata.	JSON opis uspešnosti izmene. Vratiti 404 ako ne postoji identifikator.

- 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\*\*:

- Slanje svih HTTP zahteva koji su navedeni ispod implementirati u jednom **Angular servisu**. Obezbediti postojanje tačno jedne instance ovog servisa na nivou cele aplikacije.

- Na početnoj strani aplikacije `http://localhost:4200/` prikazati polje za unos adrese elektronske pošte. Prikazati odgovarajuću poruku, kao na slici 1. Polje ima klasu `form-control`, dugme ima klase `btn btn-success btn-block`, a poruka se prikazuje u paragrafu `<p>`. Klikom na dugme „Pretraži”, poslati GET zahtev serverskoj aplikaciji sa URL parametrom unetog email-a. U zavisnosti od rezultata sa servera:
  - Ako je server vratio `null`, prikazati poruku u paragrafu `<p>` kao na slici 2.
  - Inače, prikazati informacije kao na slici 3. Ime i prezime i email se prikazuju u paragrafu `<p>`. Element „Prikaži sve uplate” ima klase `btn btn-dark`. Polje u formularu ima klasu `form-control`, a dugme ima klase `btn btn-primary`.
- Ukoliko je pronađena uplata na osnovu email-a, potrebno je prikazati i **reaktivni formular** koji sadrži dva polja (videti sliku 3):
  - Polje `id` je sakriveno (koristiti HTML atribut `hidden`) i njegova vrednost predstavlja identifikator uplate koja je dohvaćena sa servera. Prilikom obrade odgovora GET zahteva opisanog u prethodnom koraku, potrebno je **dinamički postaviti vrednost ovog polja** (sva `AbstractControl` polja reaktivnih formulara imaju metod `setValue` kojim se ovo može uraditi). **Validacijom reaktivnih formulara** obezbediti da ovo polje bude popunjeno.
  - Polje `newAmount` predstavlja novi iznos koji korisnik uplaćuje. **Validacijom reaktivnih formulara** obezbediti da ovo polje bude popunjeno, kao i da ima najmanju vrednost od `1000`.

Klikom na dugme „Dodaj novu uplatu” tipa `submit`, izvršiti podnošenje formulara na sledeći način:

- Proveriti da li je reaktivni formular validan.
- Ukoliko nije, prikazati poruku kao na slici 4 i prekinuti dalje podnošenje formulara.
- Inače, poslati PATCH zahtev serverskoj aplikaciji. U telu zahteva smestiti podatke iz formulara (identifikator i novi iznos). U slučaju uspeha, prikazati **poruku sa servera** kao na slici 5.
- Klikom na „Prikaži sve uplate”, otvara se stranica `http://localhost:4200/payment/{id}` na kojoj se, slanjem GET zahteva sa URL parametrom identifikatora uplate, dohvata informacija o uplati čiji je identifikator prosleđen u URL-u stranice sa prethodne stranice. Na stranici prikazati svaki iznos u paragrafu `<p>`. **Koristiti odgovarajući Angular filter za prikaz iznosa u dolarima**. Takođe, na stranici prikazati i vezu za povratak na `http://localhost:4200/`. Prikaz je dat na slici 6 (karakter `<` se u HTML-u zapisuje kao `&lt;`).
- 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 narednu liniju:

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

## Pregled prijava

Unesite email:



Unesite email u polje iznad i pritisnite dugme za pretragu

Slika 1: Početna stranica.

## Pregled prijava

Unesite email:



Ne postoji uplata sa datim email-om

Slika 2: Prikaz početne stranice ukoliko ne postoji podatak o uplati sa zadatim email-om.

## Pregled prijava

Unesite email:

Pretraži

Ime i prezime: Petar Perić

Email: petar.peric@mail.com

Prikaži sve uplate

Novi iznos za uplatu:

Dodaj novu uplatu

Slika 3: Prikaz početne stranice ukoliko postoji podatak o uplati sa zadatim email-om.

Pregled prijava

Unesite email:

Pretraži

Ime i prezime: Petar Perić

Email: petar.peric@mail.com

Prikaži sve uplate

Novi iznos za uplatu:

Dodaj novu uplatu

Formular nije validan! Ispravite podatke.  
OK

Slika 4: Poruka koja se prikazuje u slučaju podnošenja nevalidnog formulara.

Pregled prijava

Unesite email:

Pretraži

Ime i prezime: Petar Perić

Email: petar.peric@mail.com

Prikaži sve uplate

Novi iznos za uplatu:

Dodaj novu uplatu

Successfully added the amount  
OK

Slika 5: Dobijena poruka sa servera koja se prikazuje u slučaju podnošenja ispravnog formulara. Na slici je prikazana poruka ako je server uspešno izvršio operaciju.

## Pregled prijava

[<-- Vratite se nazad](#)

\$15,000.00

\$4,500.00

\$5,000.00

\$14,500.00

\$4,000.00

Slika 6: Prikaz stranice sa prikazom svih iznosa.