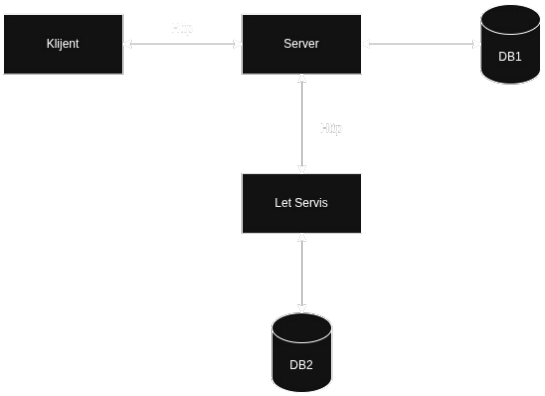
Distribuirani računarski sistemi
2025/2026

Avio letovi

Potrebno je implementirati platformu koja simulira osnovne funkcionalnosti aplikacije za rad sa letovima. Sistem se sastoji od pet komponenata:

1. **Korisnički interfejs**
2. **Servis za obradu zahteva i podataka (Server)**
3. **Baza podataka 1**
4. **Servis za rad sa letovima**
5. **Baza podataka 2**



1. Klijent

Korisnički interfejs je React ili Angular web aplikacija koja komunicira sa **Serverom** putem **REST API** poziva i **WebSocket-a** za real-time događaje.

2. Server

Python Flask aplikacija koja opslužuje klijenta i komunicira direktno sa bazom podataka. Zadatak je da obradi sve REST API zahteve koje dobije od klijenta, kao i emitovanje real-time događaja.

3. Baza podataka 1

Baza podataka predstavlja mesto za čuvanje podataka u sistemu. Podaci koje baza čuva su podaci o registrovanim korisnicima i drugi podaci po potrebi. U okviru projekta potrebno je doneti odluku o tipu baze podataka — SQL ili NoSQL, u zavisnosti od prirode i strukture podataka.

4. Let servis

Python Flask aplikacija dobija novokreirane letove i čuva ih. Vrši njihove izmjene, brisanja i obrade.

5. Baza podataka 2

Ova baza podataka čuva isključivo podatke o letovima i avio kompanijama. U okviru projekta potrebno je doneti odluku o tipu baze podataka — SQL ili NoSQL, u zavisnosti od prirode i strukture podataka.

Specifikacija zadatka

Autentifikacija

Korisnik se na aplikaciju prijavljuje unošenjem kredencijala:

- Email
- Lozinka

Neophodno je implementirati autentifikaciju zasnovanu na JWT (JSON Web Token). Na serverskoj strani pratiti neuspešne pokušaje prijave (pogrešan imejl ili lozinka) na sistem i ukoliko korisnik ima tri neuspešna pokušaja privremeno mu blokirati pristup npr. na 15 minuta (za testiranje na npr. 1 minut).

Omogućiti korisniku odjavu sa sistema.

Upravljanje korisnicima

Novi korisnički nalog se kreira prilikom registracije korisnika na platformu, pri čemu se unose:

- Ime
- Prezime
- Email
- Datum rođenja
- Pol
- Država
- Ulica
- Broj
- Stanje na računu

Novom korisniku nakon otvaranja korisničkog naloga se dodeljuje uloga **KORISNIK**. Korisnik koji želi da postavlja nove letove i avio kompanije na platformu mora da bude verifikovan kao **MENADŽER**, odnosno **ADMINISTRATOR** treba da mu postavi ulogu menadžera. Nakon dodjele uloge menadžera, korisnik dobija mail da mu je promijenjena uloga.

Administrator može da izlista sve korisnike aplikacije, briše korisničke naloge i mijenja uloge korisnicima. User, bilo da je KORISNIK ili MENADŽER, može da izmijeni sve podatke o svom korisničkom profilu, kao i da doda sliku za svoj korisnički profil.
Podatke o registrovanim korisnicima obradjivati na Serveru i čuvati u DB 1.

Rad sa letovima

Novi let na platformu može da postavi MENADŽER, pri čemu unosi sledeće podatke.

- Naziv leta
- Avio kompanija koja vrši let (referenca na tabelu)
- Dužina leta u km.
- Trajanje leta (za testiranje svesti na npr. 1 minut)
- Vrijeme polaska
- Aerodrom polaska
- Aerodrom dolaska
- Podatak ko je kreirao let
- Cijena karte

Kada menadžer kreira let takav let se u realnom vremenu prikazuje administratoru (koristiti **Web Socket**) koji treba da prihvati ili odbije let. Ukoliko se prihvati takav let on postaje vidljiv korisnicima. Ukoliko administrator želi da odbije let mora da ostavi razlog u vidu teksta i tada se vraća menadžeru na izmjenu. Ako je administrator ranije odobrio let, a pritom let nije počeo ili se završio, tada mu se nudi opcija da otkaže let. Korisnicima koji su kupili karte javiti mailom da je nihov let otkazan.

Let može da briše administrator dok korisnici i menadžeri mogu samo da pregledaju i rezervišu letove.

Letovi se mogu prikazati u tri tab-a. Prvi sadrži letove koji još nisu počeli. Drugi prikazuje letove koji su u toku i takođe tajmer koji prikazuje koliko je ostalo do kraja leta. Treći tab treba da sadrži otkazane i završene letove. Omogućiti pretragu po imenu leta i avio kompaniji (kreirati dropdown sa avio kompanijama).

Korisnici takodje treba da imaju račune na nivou aplikacije kako bi ih koristili za kupovinu letova. Uplate na račun omogućiti preko običnog input polja.
Kada korisnik želi da kupi kartu za let potrebno je da odabere let koji još nije krenuo i klikom na dugme omogućava asinhronu obradu kupovine (dodati neki sleep kako bi simulirali duze trajanje obrade kupovine). Obrada se izvršava asinhrono što omogućava nastavak rada na aplikaciji. Obrada se vrši na Let servisu i podaci o kupljenom letu se čuvaju u DB2. Korisnici mogu da vide spisak svojih letova i podatke o njima. Korisnici mogu da ostave ocjenu od 1 do 5 za let koji su koristili, ali tek nakon njegovog završetka. Administratori mogu da vide spisak svih ocjena korisnika i za koji je let ta ocjena ostavljena.

Generisanje izvještaja

Administrator ima mogućnost da kreira PDF izvještaj o letovima za svaki od pomenutih tabova. Kreirani izvještaj mu stiže na mail.

Napomene

- Neophodno je imati validaciju kako na klijentskoj strani, tako i na strani servera.
- Lozinke se moraju čuvati hešovane, nikako u čitljivom formatu.
- Prilikom odabira tipa baze podataka prvo istražite prednosti i mane SQL i NoSQL (nikako ne koristiti SQLite).
- Istražiti načine organizacije projekta kako u React/Angular, tako i u Flask-u. (struktura foldere)
- Ukoliko se odlučite za React, koristiti Vite.js alat.
- Koristiti ORM za komunikaciju sa bazom podataka (npr. SQLAlchemy za SQL baze).
- Koristiti ODM za komunikacuju sa bazom podataka (npr. PyMongo za MongoDB).
- Koristiti DTO klase za komunikaciju izmedju frontend-a i backenda ili izmedju dva servisa.
- U okviru projekta neophodno je na smislenom mestu upotrebiti procese.
- Koristiti virtuelno okruženje (venv, pipenv…).

Na odbranu projekta doći sa pripremljenim podacima za testiranje.

Način ocenjivanja
1. Aplikacija je funkcionalna i postoji Flask aplikacija – **51 poen**
    a. Flask aplikacija + Baza podataka
2. Zaseban UI projekat (React, Angular…) koji komunicira sa Engine-om – **10 poena**
3. Implementacija keš baze podataka – **14 poena**
4. Koriscenje procesa prilikom implementacije – **10 poena**
5. Dokerizacija aplikacije (moraju sve komponente) - **10 poena**
6. Pokretanje na više računara – **5 poena**

BROJ OSVOJENIH POENA SE MNOŽI SA 0.9