

Opis

Napraviti aplikaciju koja omogućava rezervaciju i administraciju avio-karata.

Postoje dve vrste korisnika, na osnovu razlicitih privilegija pruzicemo korisniku drugacije mogucnosti:

- **Obican korisnik** koji moze da pretrazuje i rezervise karte i manipulise rezervacijama.
- **Administrator** koji moze da kreira nove korisnike i vrsi CRUD operacije nad kartama i kompanijama.

Sistem treba da se sastoji iz dva dela:

- Za frontend mozete koristiti **Angular 2+**.
- Za backend **Spring Boot** ili **JBoss**.

Entiteti

Korisnik:

- ID:Integer - mora biti jedinstven, ne prikazuje se na frontendu
- Username:String - mora biti jedinstven
- Password :String - mora imati bar 6 karaktera i sadrzati slova i brojeve, u bazi se cuva **enkriptovan**. Mozete koristiti [BCrypt](#).
- Tip-korisnika:Enum - User ili Admin
- Bookings:List - lista rezervacija (samo kod **obicnog** korisnika)

Avionska kompanija:

- ID:Integer - mora biti jedinstven, ne prikazuje se na frontendu
- Name: String - mora biti jedinstveno i mora se sastojati od alfanumerickih karaktera

Grad:

- ID:Integer - mora biti jedinstven, ne prikazuje se na frontendu
- Name:String - ime grada, mora biti jedinstveno

Avionska karta:

- ID:Integer - mora biti jedinstven, ne prikazuje se na frontendu
- Kompanija:Kompanija - kompanija za koju je vezana karta
- One-way: Boolean - da li je karta u jednom pravcu ili povratna
- Depart: Date - datum polaska
- Return: Date - datum povratka (samo kod povratnih karata)

- Flight: Let - let za koji je karta vezana
- Count: Long - broj (≥ 0) dostupnih karata

Let:

- ID: Integer - mora biti jedinstven, ne prikazuje se na frontendu
- Ticket: List - lista karta za ovaj let
- Origin: Grad - grad iz kog se polece
- Destination: Grad - grad u koji se putuje

Rezervacija:

- ID: Integer - mora biti jedinstven, ne prikazuje se na frontendu
- IsAvailable: Boolean - da li je rezervacija dostupna ili je istekla (ako je prosao datum polaska)
- Flight: Let - rezervisani let
- Ticket: Karta - rezervisana karta

Atributi koji su u entitetima naznaceni da mogu da budu prisutni samo kod jednog korisnika su opcioni i mogu da budu null u bazi.

Entiteti moraju minimalno sadrzati navedene attribute, dodatni atributi su dozvoljeni.

Dozvoljeno je koriscenje bilo koje relacione baze.

Autentikacija i autorizacija

Backend mora da bude stateless - ne treba da zna koji je korisnik ulogovan, sto znaci da mora da postoji **autorizacija** zahteva.

Za svaki zahtev, server proverava da li korisnik koji ga je poslao ima pravo da pristupi tom endpointu.

Autentikaciju i autorizaciju implementirati pomocu [JWT](#) tokena.

Token moze da traje beskonacno dugo i nije potrebno implementirati rok trajanja.

Login stranica

Stranica za login treba da ima polja **username** i **password** i dugme za login.

Ukoliko korisnik sa takvim kredencijalima postoji, potrebno je proslediti ga na glavnu stranicu. U suprotnom ostati na istoj stranici i obavestiti klijenta o greski.

Korisnik koji nije ulogovan ne moze da pristupi sadrzaju ostalih stranica.

Glavna stranica

U gornjem desnom uglu treba da stoje **Username**, **tip** korisnika, ikonica koja predstavlja **rezervisane karte** sa indikatorom broja karata (samo za obicne korisnike) i **logout** dugme.

Klik na ikonicu vodi na stranicu sa rezervacijama za tog korisnika.

Ukoliko je admin:

- Forma za registraciju novog korisnika sa poljima za **username**, **password** i **tip korisnika**
- Forma za kreiranje nove karte sa svim potrebnim poljima
Lista mogucih kompanija, letova ili tipova korisnika treba da bude u combobox-u.
Nju cete popuniti podacima koje dobijete sa servera.

Neki mehanizam da korisnik odabere da li ce mu se prikazati:

- Samo karte u jednom pravcu,
- Samo povratne karte,
- Sve karte.

Tabela karata

Nalazi se na glavnoj stranici i sadrzi polja:

- One-way - da li je karta u jednom pravcu
- Origin - grad iz kog se polece
- Destination - grad u koji se putuje
- Depart - datum polaska
- Return - datum povratka (samo kod povratnih karata)
- Company - naziv avio-kompanije koji je link ka stranici avio-kompanije
- Count - broj dostupnih karata

Ako je korisnik **administrator** treba da postoji dugme za brisanje karte i dugme koje vodi na stranicu za izmenu karte.

Obican korisnik ima dugme u tabeli za rezervaciju karte, ukoliko je dostupna, i broj karata koji bi da rezervise. Nije moguće rezervisati više od broja dostupnih karata.

Broj dostupnih karata ne može biti manji od 0.

Tabela mora imati paginaciju.

Paginacija je funkcionalnost gde ne prikazujemo sve rezultate na jednom mestu, već ih raspoređujemo po "stranicama". Kroz stranice se prolazi sa dugmetom za prethodnu ili za sledeću stranu (ako postoje). Ako na BE postoji 25 karata, imaćemo tačno 3 puta sansu da

kliknemo dugme za sledecu stranicu. Treci put ce se prikazati samo 5 karata ako svaka stranica ima 10 rezultata prikazanih.

U nasem slucaju imacemo tabelu karata sa dva dugmeta ispod za sledece i prethodne rezultate. Tabela ce na osnovu toga da prikazuje odredenih 10 karata. Svaki put kad se klikne neko od dugmica, potrebno je pozvati BE i popuniti tabelu sa odgovorajucih 10 karata. Na FE ne sme biti ucitano vise od 10 karata sto znaci da se paginacija vrsi na BE.

Da biste postigli ovakav efekat predlazem da napravite pomocnu klasu PaginationResponse koja bi imala jedan atribut tipa Object (tu biste smestili vas model) i jedan atribut tipa PageInfo koji je takodje model koji poseduje informacije o tome koliko ima stranica ukupno, koja je stranica u pitanju, koliko podataka ima po stranici.

Svaka stranica ima footer

U njemu stoji danasnji datum i vreme u formatu dan/mesec/godina sat:minut:sekund.
Vrednost treba da se osvezava u realnom vremenu, kao sat koji otkucava.
Za ovu funkcionalnost koristiti [DataPipe](#).

Forme

Sve forme treba da mogu da se posalju i klikom misa na submit, i pritiskom tastera "enter" na tastaturi.

Stranica za izmenu karte

U gornjem desnom uglu treba da stoje **Username**, **tip** korisnika, ikonica koja predstavlja **rezervisane karte** sa indikatorom broja karata (samo za obicne korisnike) i **logout** dugme.

Sadrzi:

- Formu za izmenu karte sa svim potrebnim poljima, dugme za slanje i za ponistavanje operacije.

Stranica avio-kompanije

U gornjem desnom uglu treba da stoje **Username**, **tip** korisnika, ikonica koja predstavlja **rezervisane karte** sa indikatorom broja karata (samo za obicne korisnike) i **logout** dugme.

Sadrzi:

- Naziv kompanije
- Tabelu svih karata za tu kompaniju (uz mogućnost brisanja i izmene za **administratora** i rezervisanje i pretrage za **user-a**)
- Dugme za brisanje kompanije (Ako je korisnik **administrator**)
Brisanjem kompanije brisu se i sve karte vezane za obrisanu kompaniju
Kao i obrisane karte iz svih letova.
- Forma za izmenu naziva kompanije (Ako je korisnik **administrator**)
Izmena naziva kompanije povlači sa sobom izmenu u svim entitetima povezanim sa tom kompanijom.
- Formu za kreiranje kompanije samo sa poljem za naziv (Ako je korisnik **administrator**)

Stranica rezervacija

U gornjem desnom uglu treba da stoje **Username**, **tip** korisnika, ikonica koja predstavlja **rezervisane karte** sa indikatorom broja karata (samo za obične korisnike) i **logout** dugme.

Sadrzi:

- Tabelu svih rezervacija u kojoj se nalaze:
 - Kolone iz tabele karata
 - Kolona da li je rezervacija istekla
 - Dugme za brisanje rezervacije, rezervacija se može obrisati **do 24 sata pre datuma polaska**Brisanjem rezervacije, treba osvežiti listu rezervacija tog korisnika u bazi.

Error handling

Error handling i validacija su obavezni i svaka funkcija mora da prijavi grešku ukoliko ulazni parametri ili interno stanje ne dozvoljavaju izvršenje operacije.

I **backend** i **frontend** validacije su obavezne!

- Voditi računa za unose koji su **prazne vrednosti**.
- Mesto polaska i destinacija ne mogu biti isto mesto.
- Vreme polaska ne može biti nakon vremena povratka.
- Korisnika obavestiti kad god napravi grešku.
- Ne može da se registruje korisnik sa username-om koji već postoji (slično i za druge entitete koji moraju imati jedinstvena polja).
- Neko Postman-om pokuša da rezervise više karata nego što postoji. Broj karata ne sme biti negativan.
- Kada se desi greška na backend-u, frontend bi na osnovu odgovora trebao obavestiti korisnika o tome da se desila greška.

Napomene:

Ukoliko koristite Google Chrome, on kesira stranice pa se može desiti da ste napravili izmenu a da stranica i dalje izgleda isto.

Zato kada napravite izmenu, otvorite stranicu u incognito prozoru.

Sve nazive (User, Admin, One-Way, Ticket, Booking...) možete nazvati proizvoljnim imenima u vašim projektima.

Napravite određeni broj predefinisanih karata/ kompanija/ letova/ gradova/ korisnika radi lakšeg testiranja i demonstracije.

Frontend ne treba da čuva nikakve podatke, već treba da ih dohvati sa backend-a, tako da se podaci ne izgube posle refresha stranice.

Predaja

Ne zaboravite da prijavite ispit, da bi profesor mogao da unese poene u bazu, bez obzira da li ste položili teoretski deo ili ne!

Rok za predaju je 07.09. u ponoć. Odbrana u subotu, 11.09. u 10:00 u RAF1.

Projekat nosi 40 poena, a minimalno je potrebno ostvariti 20 poena na projektu.
Za uslov je potrebno implementirati sve do isprekidane linije, uz error handling.

Na Git, Drive, .zip je dovoljno poslati samo kod, bez .jar, node_modules, .idea i drugih nepotrebnih foldera i fajlova koji su u vezi sa framework-om, bibliotekom ili okruženjem koje koristite. Pogotovo na Git nije ni preporučljivo komitovati te foldere.

Domaci se šalje na **amicic@raf.rs**

Naslov imejla i treba da glasi: Projekat <Ime> <Prezime> <Indeks>

Telo imejla treba da bude jedno od sledećih:

- Zip
- Link ka zip-u
- Link ka Drive-u
- Link ka GitHub/GitLab repou

Ako saljete Git, stavite da je repozitorijum private i dodajte me kao collaborator-a

<https://github.com/alemicic>