

Završni test

Korišćenjem radnih okvira Spring Boot, React i Bootstrap razviti Web aplikaciju za vođenje evidencije o odsustvima u okviru preduzeća.

0.1 Obezbediti kreiranje minimalnog skupa test podataka (bilo iz koda, bilo SQL srkiptom)

1.1 Aplikacija treba da obezbedi rad sa sledećim entitetima, uz čije attribute su data ograničenja na nivou baze podataka:

Odeljenje:

- Id - identifikator
- Ime - tekstualna vrednost (jedinstvena, obavezna)
- Bonus slobodnih dana – celobrojna vrednost

Radnik:

- Id - identifikator
- JMBG – tekstualna vrednost (obavezna, jedinstvena)
- Ime i prezime – tekstualna vrednost (obavezna)
- E-mail - tekstualna vrednost
- Godina radnog staža – celobrojna vrednost
- Slobodnih dana – celobrojna vrednost (**ne unosi korisnik, izračunava se**)
- Odeljenje - veza sa instancom klase odeljenje (jedan radnik može pripadati samo jednom odeljenju, a jedno odeljenje može imati više radnika). Veza je dvosmerna.

Odsustvo:

- Id – Identifikator
- Datum početka - tekstualna vrednost (obavezna)
- Radnih dana - celobrojna vrednost (obavezna)
- Radnik - veza sa instancom klase radnik (jedno odsustvo može biti vezano samo za jednog radnika, a za jednog radnika može biti vezano više odsustva). Veza je dvosmerna.

Aplikacija treba da vodi računa o broju slobodnih dana za jednog radnika. Broj slobodnih dana se izračunava po sledećoj formuli:

- Svi radnici imaju minimalno 20 slobodnih dana;
- Za svakih 5 godina radnog staža, dodaje se po jedan dodatni slobodni dan;
- Na kraju, s obzirom na to u kojem odeljenju radnik radi, dodaje se bonus.

Pri naknadnoj promeni odeljenja radniku, ne treba ponovo izračunavati broj slobodnih dana.

1.2 Pomoću radnog okvira Spring Boot implementirati sledeći REST API:

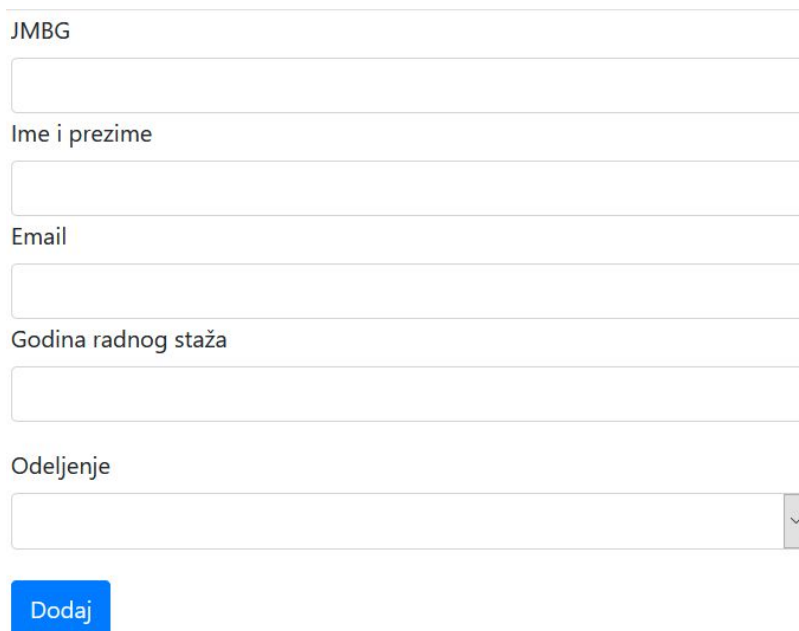
- GET /api/odseljenja - preuzimanje svih sektora
- GET /api/radnici - preuzimanje svih radnika (paginirano)
- GET /api/radnici/{id} - preuzimanje radnika po zadatom id-u
- POST /api/radnici - dodavanje novog radnika
- PUT /api/radnici/{id} - izmena postojećeg radnika
- DELETE /api/radnici/{id} – brisanje postojećeg radnika

- Voditi računa o tome da, koliko je to moguće, REST metode vraćaju preporučeni HTTP status za odgovarajući rezultat neke operacije.

1.3 Na nivou API-ja validirati sledeće stavke:

“JMBG” radnika je tekst dužine 13 karaktera

“E-mail” radnika je dobro formatiran



JMBG

Ime i prezime

Email

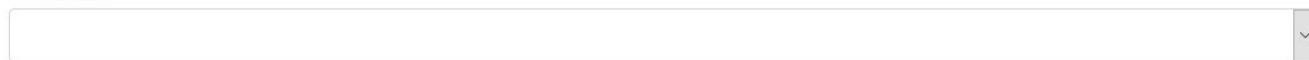
Godina radnog staža

Odeljenje


Dodaj

Slika 1.




Odeljenje



JMBG



Prethodna Sledeća

JMBG	Ime i prezime	Email	Slobodni dani	Odeljenje	Akcije
23082020821213	Milica Miljković	mm@mail.com	20	Računovodstvo	Odsustvo 
24082020821213	Darko Dedić	dd@mail.com	14	Računovodstvo	Odsustvo 
23082020821213	Marija Marić	mm2@mail.com	11	Kadrovska	Odsustvo 

Slika 2.

2.1 Obezbediti unos novog radnika na **zasebnoj stranici** (slika 1). Na stranici za prikaz i pretragu radnika postoji dugme za prelazak na stranicu za unos (slika 2). Nakon unosa vrednosti u polja prikazana na slici 1 i klika na dugme dodaj, radnik se preko API-ja dodaje u aplikaciju i potom ga program vraća na stranicu za prikaz i pretragu. **Polje “Slobodnih dana” ne unosi korisnik.**

2.2 Obezbediti brisanje radnika. Nakon klika na dugme obriši u tabeli za prikaz radnika, radnik se briše iz aplikacije i uklanja iz tabele.

2.3 Obezbediti paginirani prikaz podataka. Dugme prethodno i dugme sledeće iznad tabele za prikaz radnika (slika 2) omogućuju promenu stranice. Ukoliko se korisnik nalazi na prvoj stranici onemogućiti dugme prethodno, a ukoliko se nalazi na poslednjoj onemogućiti dugme sledeće. Paginacija se vrši na back-end delu aplikacije.

2.4 Obezbediti pretragu radnika, preko forme za filtriranje (slika 2). Prilikom pretrage korisnik može da odabere *odeljenje* (preko opcije za selekciju, *dropdown*) ili unese JMBG. Pronalaze se i prikazuju radnici koji zadovoljavaju sve kriterijume. Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi. Filtriranje se vrši na *back-end* delu aplikacije. Odraditi tako da **NE** postoji dugme na koje se inicira pretraga, već promenama vrednosti polja dolazi do svežavanje tabele.

2.5 Izmenu radnika **NE** morate raditi na front-end strani!

3. Implementirati uzimanje odsustva za jednog radnika:

- Klikom na dugme “Odsustvo” (slika 2.), prelazi se na novu stranicu za zakazivanje odsustva, sa formom za unos podataka o odsustvu. Kada korisnik unese podatke o odsustvu, podaci iz forme se šalju na backend odgovarajućom API metodom i logika za zakazivanje odsustva se implementira na nivou servisa na back-end-u. Korisnik ne može da iskoristi više dana odsustva nego što ima na raspolaganju. Ako je taj uslov zadovoljen, podaci o odsustvu se čuvaju u bazi.
- Ako je radnik iskoristio sve dane za odsustvo, na stranici za zakazivanje mu je prikazan samo tekst “Iskorišteni su svi slobodni dani”.

4. Aplikacija bi trebalo da bude obezbeđena i sa back-end, i sa front-end strane što znači da samo prijavljeni korisnik može da joj pristupi. Korisnik se prijavljuje na sistem preko *login* stranice. Za ovaj deo, dozvoljeno je koristiti postojeće materijale kao takve.