

User management

Domaći 3

1. Pregled zadatka

Zadatak je napraviti web aplikaciju za upravljanje korisnicima i njihovim permisijama. Potrebno je napraviti podršku za CRUD operacije nad korisnikom i obezbediti autorizaciju zahteva korisnika uz pomoć definisanog seta permisija.

Permisije koje je potrebno obezbediti i koje se mogu dodeliti korisniku su: `can_create_users`, `can_read_users`, `can_update_users`, `can_delete_users`. Zahtevi za `create/read/update/delete` korisnika će biti mogući samo ukoliko autentifikovani korisnik ima odgovarajuću permisiju za taj zahtev.

Autentifikaciju korisnika je potrebno implementirati korišćenjem JWT-a.

2. Zahtevi

Korisnik kao entitet sadrži ime, prezime, email adresu, lozinku i set permisija. Email adresa je jedinstvena i nije moguće kreirati više korisnika sa istom email adresom. Lozinku čuvati upotrebom nekog hash algoritma po izboru, nikako kao plain tekst. Svi atributi, osim permisija, su obavezni.

Autentifikacija korisnika se vrši slanjem email adrese i lozinke. Ukoliko su kredencijali validni, odgovor je JWT kog korisnik treba poslati uz svaki sledeći zahtev kako bi bio autentifikovan.

Ukoliko korisnik nema permisiju za određeni zahtev, potrebno je vratiti status code 403 Forbidden i korisnika obavestiti o nedozvoljenoj aktivnosti.

Nakon login-a na frontendu, dobijeni token je potrebno čuvati u `localStorage-u` i slati ga pri svakom sledećem zahtevu.

Na frontendu je potrebno implementirati 4 stranice:

1. Stranicu za login
2. Stranicu za prikaz korisnika - tabela sa svim korisnicima u kojoj se prikazuje ime, prezime, email i permisije za svakog od korisnika

- a. Stranicu za dodavanje korisnika - forma koja sadrži polja za ime, prezime, email i permisije koje će novi korisnik imati. Sva polja su obavezna i u slučaju greške pri kreiranju, potrebno je obavestiti korisnika.
3. Klikom na korisnika u tabeli otvara se stranica za edit koja sadrži polja za ime, prezime, email i permisije sa vrednostima koje odabrani korisnik ima.

Frontend mora biti svestan permisija i na osnovu njih da dozvoli prikaz/akciju.

Ponašanje frontend-a u zavisnosti od permisije:

Permisija	Ponašanje kada korisnik ima permisiju	Ponašanje kada korisnik nema permisiju
can_read_users	- Dozvoljen pristup stranici sa tabelom svih korisnika.	- Prikazati poruku da korisnik nema permisiju
can_create_users	- Prikaz linka koji vodi na stranicu za dodavanje korisnika. - Dozvoljen pristup stranici za kreiranje user-a.	- Link za pristup stranici za dodavanje korisnika je isključen ili nevidljiv. - Zabranjen pristup stranici za kreiranje user-a.
can_update_users	- Email adresa korisnika u tabeli je klikabilna. - Dozvoljen pristup stranici za edit korisnika.	- Email adresa korisnika u tabeli nije klikabilna. - Zabranjen pristup stranici za edit korisnika.
can_delete_users	- U tabeli za svakog korisnika postoji dugme za njegovo brisanje.	- Dugme za brisanje korisnika je isključeno ili nevidljivo.

Ukoliko korisnik nema nijednu permisiju, nakon uspešnog login-a, obavestiti ga alertom.

3. Tehnički zahtevi

Backend (5 poena):

- Spring ili JBoss
- Relaciona baza podataka

Frontend (5 poena):

- Angular 2+

4. Predaja zadatka

Studenti koji slušaju vežbe utorkom imaju rok za predaju 13.12. do 23:59. Studenti koji slušaju vežbe četvrtkom imaju rok za predaju 15.12. do 23:59.

Domaći se šalje na:

- santic@raf.rs za grupe: 406, 422, 402a, 402b

- nredzic@raf.rs za grupe: 401, 405

Subject mail-a mora da bude u obliku: "Domaci3 <Ime> <Prezime> <Indeks>". Npr. "Domaci3 Stefan Antić RN19/16". Loš format subject-a rezultuje sa -50% od ostvarenih poena. U telu mejla navesti grupu kojoj pripadate po zvaničnom spisku kao i sam rad koji može biti u obliku:

- Zip
- Link ka Drive-u
- Link ka GitHub privatnom repozitorijumu.