# Java Web Development, Modul 3, Test

**NAPOMENA! ZA SVAKI ZADATAK OBAVEZNO UPISATI U FAJL *URADJENO.TXT* ŠTA JE URAĐENO!**

Korišćenjem radnih okvira Spring Boot, AngularJS i Bootstrap napraviti Web aplikaciju za *agenciju za promet nekretninama*. Aplikacija treba da obezbedi rad sa sledećim entitetima:

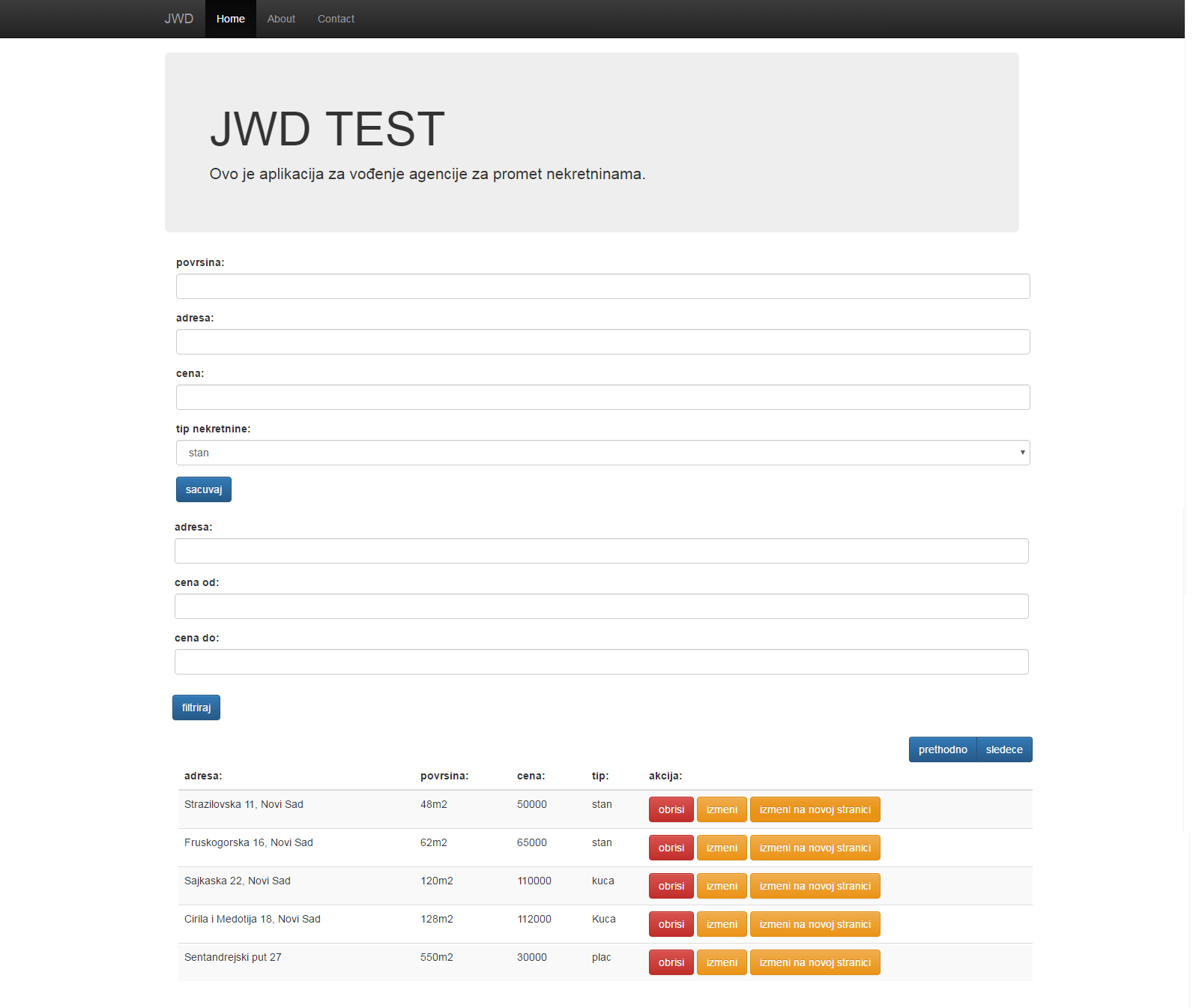
|  |
| --- |
| *Nekretnina*  – entitet koji ima polja   * *Id* – identifikator * *Površina* – celobrojna vrednost * *Cena* – celobrojna vrednost * *Adresa* – tekstualna vrednost * *Tip nekretnine* –instanca klase *Tip nekretnine*. Nekretnina može biti samo jednog tipa.   *Tip nekretnine* – na primer *stan*, *kuća*, *plac*, ... Ovaj entitet ima sledeća polja polje   * *Id* – identifikator * *Naziv*– tekstualna vrednost * *Nekretnine*– lista instanci klase *Nekretnina*. |

1. Pomoću radnog okvira Spring Boot implementirati sledeći RESTful API:

|  |
| --- |
| **GET /api/nekretnine** – preuzimanje svih neketnina  **GET /api/nekretnine /<id>** – preuzimanje nekretnine sa zadatim id-om  **POST /api/nekretnine** – dodavanje nekretnine u aplikaciju  **PUT /api/nekretnine/<id>** – izmena podataka o nekretnini sa zadatim id-om  **DELETE/api/ nekretnine/<id>** – brisanje nekretnine sa zadatim id-om iz aplikacije  **GET /api/tip** – preuzimanje svih tipova nekretnina iz aplikacije  **GET /api/tip/<id\_tipa>/nekretnine /** – preuzimanje svih nekretnina sa zadatim tipom |

Čuvanje podataka u bazi implementirati pomoću Spring Data JPA

1. Pomoću radnih okvira AngularJS i Bootstrap implementirati klijentsku aplikaciju koja izgleda kao na slici 1.



Slka 1 – Izgled stranice za unos i prikaz nekretnina

2.1) Obezbediti unos nove nekretnine. Nakon unosa vrednosti u polja prikazana na slici 2 i klika na dugme *sacuvaj* prikazano na toj slici, nekretnina se preko API-ja dodaje u aplikaciju i izmena se prikazuje u tabeli na slici 3.



Slika 2 – polja za unos nekretine



Slika 3 – tabela za prikaz nekretnina

2.2) Obezbediti brisanje nerketnine. Nakon klika na dugme *obriši* u tabeli za prikaz nekretnina (slika 3), ta nekretnina se briše iz aplikacije i uklanja iz tabele.

2.3) Obezbediti izmenu nekretnine. Nakon klika na dugme *izmeni*podaci o neretnini se prikazuju u poljima za nekretnine (slika 2). Nakon izmene ovih vrednosti i klika na dugme *sacuvaj*, izmenjeni podaci se čuvaju u aplikaciji i izmenjena nekretnina se prikazuje u tabeli.

2.4) Obezbediti izmenu nekretnine u zasebnoj stranici. Nakon klika na dugme *izmeni u novoj stranici*, podaci o nekretnini se prikazuju u zasebnoj strnici koja omogućuje čuvanje izmena. Ukoliko korisnik odabere da sačuva podatke, izmenjeni podaci se čuvaju u aplikaciji i prelazi se na stranicu za unos i prikaz nekretnina (slika 1). Dizajn zasebne stranice za izmenu nekretnine prepušten je polazniku kursa.

2.5) Obezbediti pretragu nekretnina. Prilikom pretrage korisnik može da unese *adresu*, *donju granicu cene* i *gornju granicu cene*. Adresa ne mora da bude potpuna, već se za unetu vrednost adrese proverava da li je podstring adrese u bazi. Pronalaze se i prikazuju nekrentine koje zadovoljavaju sva tri kriterijuma (i adresa sadrži deo adrese koju je korisnik uneo i cena je veća od donje granice cene i cena je manja od gorne granice cene). Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi (na primer ukoliko korisnik unese deo adrese i gornju granicu cene biće prikazane sve nekretnine čija adresa sadrži unetu vrednost i jefitnije su zadate cene, a donja granica se neće uzimati u obzir). Polja za filtriranje prikazana su na slici 4. Filtriranje se vrši na back-end delu aplikacije.



Slika 4 – polja za filtriranje nekretnina

2.6) Obezbediti paginirani prikaz podataka. Dugme *prethodho* i dugme *sledece* iznad tabele za prikaz nekretnine (slika 3) 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 na back-end delu aplikacije.