

PREDMET: IT255 Veb sistemi 1

PROJEKTNI ZADATAK

Veb prodavnica mobilnih telefona

STUDENT

Jovica Ristić 4362

ASISTENT

Miloš Milašinović

16.01.2023.

Sadržaj

[1. Uvod 3](#_Toc125075702)

[2. Analiza 3](#_Toc125075703)

[2.1. Cilj aplikacije 3](#_Toc125075704)

[2.2. Korisnici sistema 3](#_Toc125075705)

[2.3. Slucajevi koriscenja 4](#_Toc125075706)

[2.4. Funkcionalni i nefunkcionalni zahtevi 7](#_Toc125075707)

[2.5. Projektovanje i arhitektura sistema 9](#_Toc125075708)

[2.6. Implementacija sistema 10](#_Toc125075709)

[2.7. Prikaz aplikacije 23](#_Toc125075710)

[3. Zaključak 28](#_Toc125075711)

[4. Reference 28](#_Toc125075712)

# Uvod

Dokument predstavlja dokumentaciju projektnog zadatka iz pedmeta IT255.

Sam projekat predstavlja veb shop u kojem se prodaju mobilni telefoni. Aplikacija je zamisljena tako da ima dve vrste korisnika, obične korisnike (potencijalne kupce) i administratore.

Kada se aplikacija pokrene u pretraživaču se otvara početna stranica, nacigacija u tom trenutnu sadrzi opcije za pocetnu stranicu, pregled ponude i login korisnika.

Ukoliko korisnik koji jos uvek nije prijavljen otvori prikaz ponude telefona, on moze samo pregledati specifikacije telefona dok mu porucivanje istog nece biti dozvoljeno dok se ne uloguje.

Korisnik se prijavljuje tako sto otvara stranicu za login odakle ukoliko nema otvoren nalog moze da predje na stranicu za registraciju.

Kada je korisnik uspesno ulogovac on ce moci pored pregledanja telefona i da porucuje telefone.

Ukoliko je ulogovani korisnik administrator, onda on može dodavati nove mobilne telefone, pregledati, izmenjivati ili brisati podatke o telefonima koji vec postoje u bazi, odnosno koji su u ponudi, administrator takodje moze i da vidi sve porudzbine koje su napravili kupci i za svaku od porudzbina moze da oznaci da je isporucena, takodje moze i da obrise porudzbinu.

Aplikacija je izgrađena pomoću Angular okvira, za stilizovanje html stranica koriscen je Bootstrap, dok je baza podataka u json formatu a za pristup njoj napravljen je „fake“ API.

# Analiza

## Cilj aplikacije

Cilj aplikacije je da korisnicima omogući da na jedan jednostavan i efikasan način vrše pregled svih telefona u shopu i poručivanje istih. Pored toga cilj aplikacije je takođe da olakša rad zaposlenima, odnosno administratorima, koji na jednostavan način mogu da prate sve porudzbine koje su kupci napravili kao i da imaju uvid u to da li je porudzbina isporucena ili nije, dodaju nove telefone u shop i vrše određene izmene sa već postojećim laptopovima.

## Korisnici sistema

Postoje dve vrste korisnika sistema:

* Korisnici
* Administratori

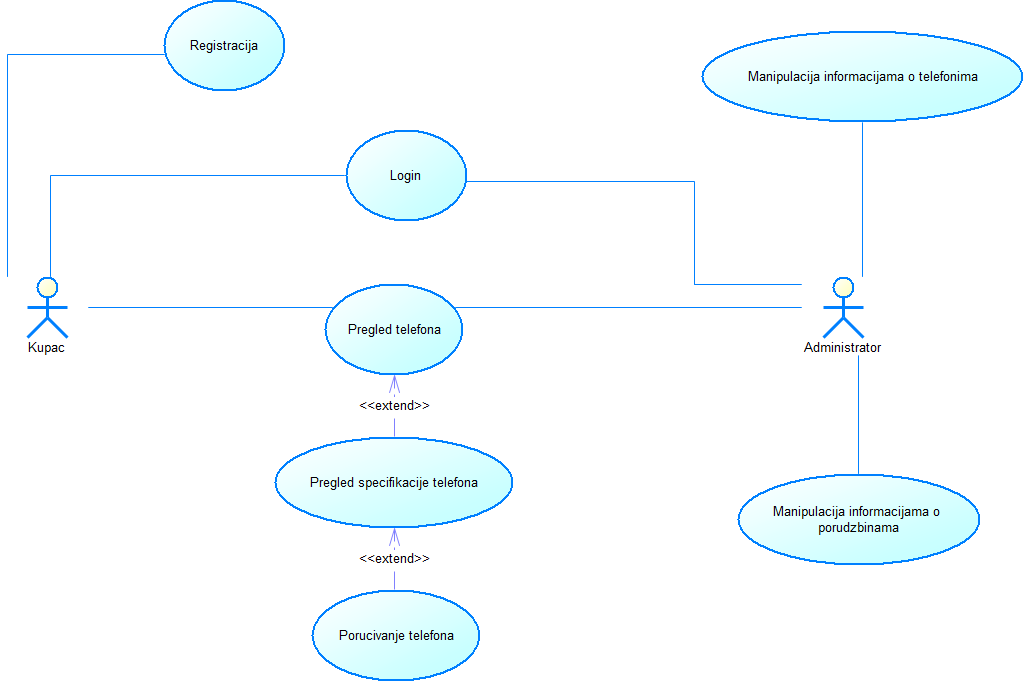
## 2.3. Slucajevi koriscenja

U ovom delu dokumenta vrsice se analiza ucesnika sistema kroz slucajeve koriscenja. Kao sto smo vec rekli imamo dva ucesnika: korisnike i administratore.

Korisnici vrse registraciju i prijavljivanje na sistem, takodje korisnici mogu da pregledaju ponudu telefona i odabirom nekog telefona mogu da poruce isti ukoliko su prijavljeni na sistem.

Administratori imaju apsolutnu kontrolu nad podacima koji se pojavljuju u sistemu, oni mogu da na sledeci nacin manipulisu podacima: mogu da dodaju nove telefone, vrse izmene nad postojecim ili ih brisu, takodje mogu da vide sve porudzbine vrse neke izmene nad njima i takodje mogu da ih brisu.

Na slici u nastavku kroz dijagram (Use-Case) mozemo videti sve slucajeve koriscenja za oba aktera sistema.



*Slika 1. Prikaz dijagrama slucajeva koriscenja*

Sada cemo dati detaljan opis slucajeva koriscenja tako sto cemo kreirati tabele i na pregledan nacin opisati scenarije upotrebe sistema.

U prvom koraku bice kroz sledecu tabelu opisan slucaj koriscenja: Registracija

|  |  |
| --- | --- |
| Slucaj koricenja: Registracija | |
| Opis: | Registracija novih korisnika |
| Aktor: | Korisnik |
| Preduslov: | 1. Korisnik pristupio sistemu |
| Tok dogadjaja: | 1. Korisnik otvara stranicu za registraciju 2. Korisnik popunjava formu svojim podacima |
| Uslovni tok: | 1. Korisnik ne popunjava uspesno formu |
| Postuslov: | 1. Korisnik se uspesno registrovao |

*Slika 2. Prvi slucaj koriscenja*

U drugoj tabeli bice opisan slucaj koriscenja: Login

|  |  |
| --- | --- |
| Slucaj koricenja: Login | |
| Opis: | Login korisnika na sistem |
| Aktor: | Korisnik, administrator |
| Preduslov: | 1. Korisnik pristupio sistemu 2. Korisnik ima otvoren nalog |
| Tok dogadjaja: | 1. Korisnik otvara stranicu za login 2. Korisnik unosi gmail i lozinku 3. Korisnik potvrdjuje login |
| Uslovni tok: | 1. Korisnik nema nalog 2. Korisnik ne unosi validne podatke |
| Postuslov: | 1. Korisnik se uspesno prijavio na sistem |

*Slika 3. Drugi slucaj koriscenja*

U trecoj tabeli bice opisan slucaj koriscenja: Pregled telefona

|  |  |
| --- | --- |
| Slucaj koricenja: Pregled telefona | |
| Opis: | Pregled svih telefona iz ponude |
| Aktor: | Korisnik, administrator |
| Preduslov: | 1. Korisnik pristupio sistemu |
| Tok dogadjaja: | 1. Korisnik otvara stranicu koja prikazuje ponudu telefona |
| Uslovni tok: | 1. Korisnik ne nalazi odgovarajuci telefon |
| Postuslov: | 1. Korisnik uspesno pronalazi zeljeni telefon |

*Slika 4. Treci slucaj koriscenja*

U cetvrtoj tabeli bice opisan slucaj koriscenja: Pregled specifikacija telefona

|  |  |
| --- | --- |
| Slucaj koricenja: Pregled specifikacija telefona | |
| Opis: | Pregled specifikacije telefona na osnovu izbora iz ponude |
| Aktor: | Korisnik, administrator |
| Preduslov: | 1. Korisnik pristupio sistemu 2. Korisnik odabrao telefon |
| Tok dogadjaja: | 1. Korisnik otvori ponudu telefona 2. Korisnik klikne na dugme detalji |
| Uslovni tok: | 1. Korisnik nije zadovoljan specifikacijama telefona |
| Postuslov: | 1. Korisnik uspesno vrsi odabir telefona |

*Slika 5. Cetvrti slucaj koriscenja*

U petoj tabeli bice opisan slucaj koriscenja: Porucivanje telefona

|  |  |
| --- | --- |
| Slucaj koricenja: Porucivanje telefona | |
| Opis: | Porucivanje telefona na osnovu izbora iz ponude |
| Aktor: | Korisnik, administrator |
| Preduslov: | 1. Korisnik pristupio sistemu 2. Korisnik ulogovan na sistem 3. Korisnik odabrao telefon |
| Tok dogadjaja: | 1. Korisnik selektuje zeljeni telefon 2. Korisnik klikne na fugme Poruci odmah 3. Korisnik popunjava formu sa potrebnim podacima 4. Korisnik klikne na dugme Poruci |
| Uslovni tok: | 1. Korisnik nije ulogovan 2. Korisnik nije validno popunio formu |
| Postuslov: | 1. Korisnik je uspesno porucio telefon |

*Slika 6. Peti slucaj koriscenja*

Sada su nam ostala jos dva scenarija koja su karakteristicna samo za administratore:

* Manipulacija informacijama o telefonima
* Manipulacija informacijama o porudzbinama

U sestoj tabeli bice opisan slucaj koriscenja: Manipulacija informacijama o telefonima

|  |  |
| --- | --- |
| Slucaj koricenja: Manipulacija informacijama o telefonima | |
| Opis: | Manipulacija informacijama o telefonima od strane admina |
| Aktor: | Administrator |
| Preduslov: | 1. Admin pristupio sistemu 2. Admin ulogovan na sistem |
| Tok dogadjaja: | 1. Admin dodaje nove telefone 2. Admin menja podatke o postojecim telefonima 3. Admin brise postojece telefone |
| Uslovni tok: | 1. Admin nije ulogovan 2. Admin ne unosi validne podatke u formu |
| Postuslov: | 1. Uspesno dodata soba 2. Uspesno promenjeni podaci o telefonu 3. Uspesno obrisan telefon |

*Slika 7. Sesti slucaj koriscenja*

U sestoj tabeli bice opisan slucaj koriscenja: Manipulacija informacijama o porudzbinama

|  |  |
| --- | --- |
| Slucaj koricenja: Manipulacija informacijama o porudzbinama | |
| Opis: | Manipulacija informacijama o porudzbinama od strane admina |
| Aktor: | Administrator |
| Preduslov: | 1. Admin pristupio sistemu 2. Admin ulogovan na sistem |
| Tok dogadjaja: | 1. Admin pregledava sve porudzbine 2. Admin menja stanje porudzbine na isporuceno 3. Admin brise postojece porudzbine |
| Uslovni tok: | 1. Admin nije ulogovan 2. Ne postoje porudzbine |
| Postuslov: | 1. Uspesno promenjeno stanje porudzbine 2. Uspesno obrisana porudzbina |

*Slika 8. Sedmi slucaj koriscenja*

## 2.4. Funkcionalni i nefunkcionalni zahtevi

U delu funkcionalnih zahteva biće prikazani svi funkcionalni zahtevi koje sistem treba da zadovolji.

Funkcionalni zahtevi su prikazani u dva sektora:

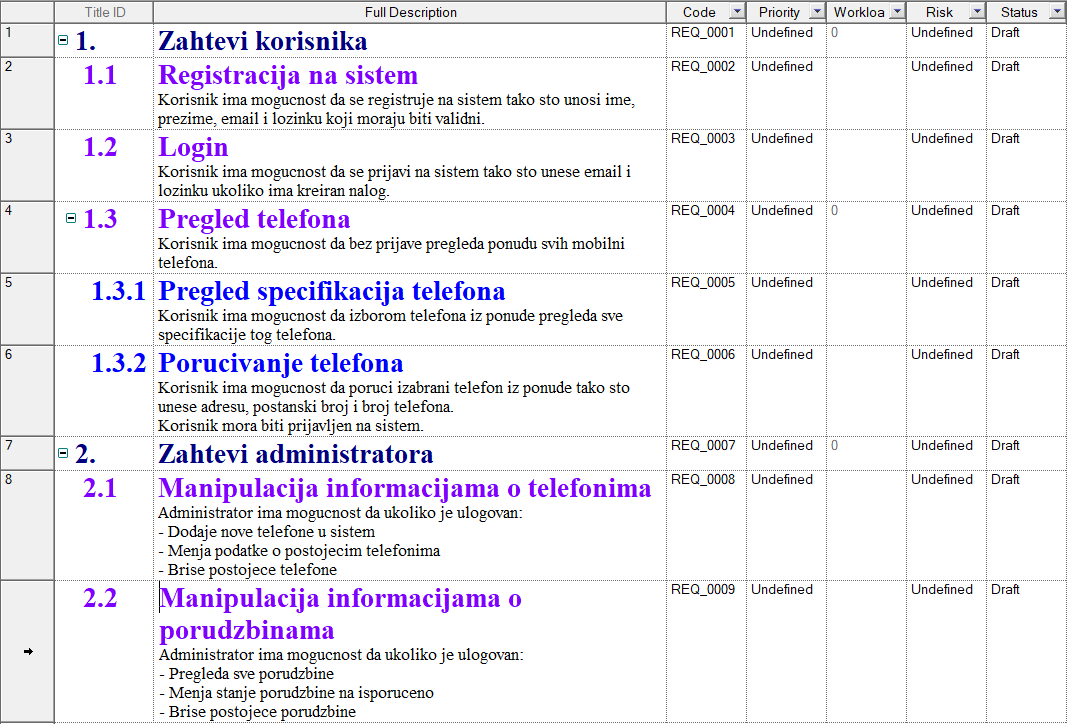
- funkcionalni zahtevi koje mora da zadovolji korisnik (kupac)

- funkcionalni zahtevi koje mora da zadovolji administrator

Za izradu i prikaz funkcionalnih zahteva se koristi Requirements model koji zadovoljava sve standarde za pisanje istih, generisan u PowerDesigner alatu.

Najpre prikazujemo funkcionalne zahteve korisnika koji su identifikovani kao:

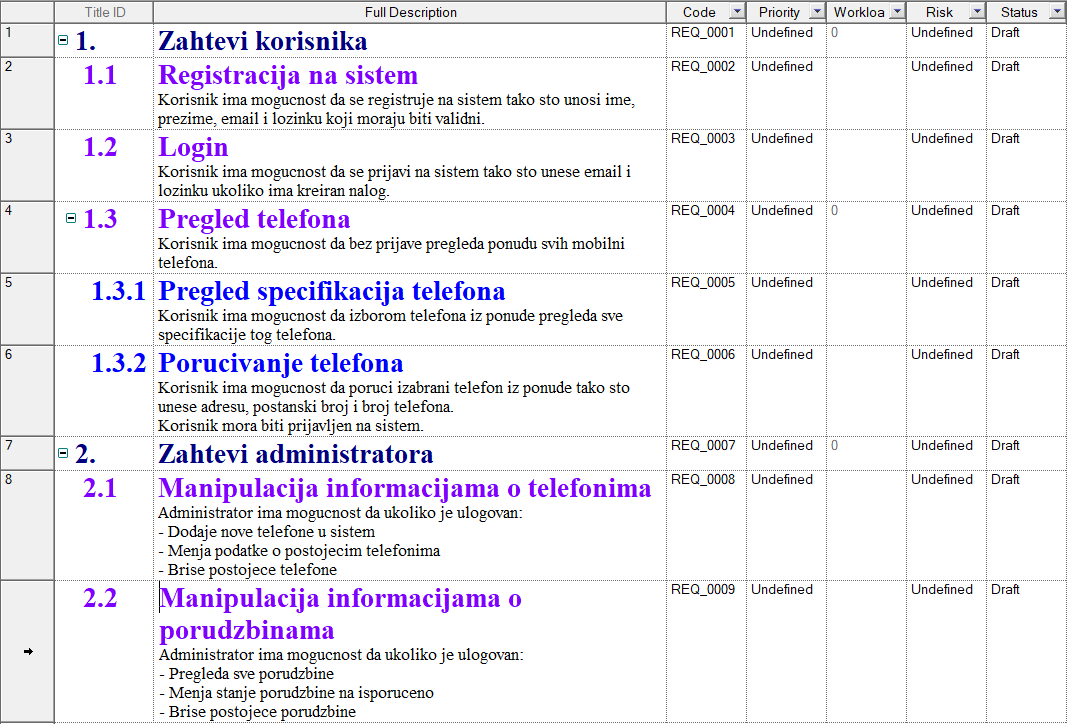
* Registracija na sistem
* Login
* Pregled telefona
* Pregled specifikacija telefona
* Porucivanje telefona



*Slika 9. Funkcionalni zahtevi korisnika*

Zatim pikazujemo funkcionalne zahteve administratora koji su identifikovani kao:

* Manipulacija informacijama o telefonima
* Manipulacija informacijama o porudzbinama

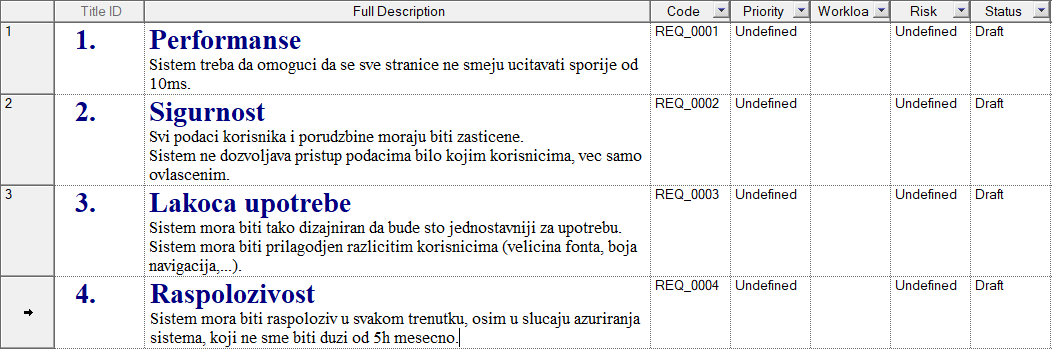


*Slika 10. Funkcionalni zahtevi administratora*

U delu nefunkcionalnih zahteva bice prikazani svi nefunkcionalni zahtevi koje ovaj sistem treba za zadovolji. To su zahtevi koji su vezani za ceo sistem i najcesce se odnose na neka ogranicenja funkcionalnosti i servisa koje pruza sistem.

Nefunkcionalni zahtevi koji se odnose na ovaj sistem koji se razvija identifikovani su kao:

* Performance
* Sigurnost
* Lakoca upotrebe
* Raspolozivost



*Slika 11. Neunkcionalni zahtevi sistema*

## 2.5. Projektovanje i arhitektura sistema

Izrada projekta ce se bazirati na Model-View-Controller (MVC) arhitekturi.



*Slika 12. MVC arhitektura*

MVC arhitektura predstavlja softversku arhitekturu, koja je najcesce koriscenja uz implementaciju korisnickog interfejsa, pa samim tim i najpogodnija je za arhitekturu web aplikacija. Ova arhitektura razdvaja logiku aplikacije u 3 odvojena dela, sto dovodi do modularnosti i bolje komunikacije medju modulima, samim tim i aplikacija je fleksibilnija.

* *Model* – definise sta bi aplikacija trebala da sadrzi. Ukoliko se stanje nekog od podataka promeni, onda je na modelu da obavesti pogled (*View*) i ponekad kontroler (*Controller*) o nastaloj promeni.
* *View* – definise kako bi aplikacija trebalo da bude prikazana;
* *Controller* – sadrzi logiku koju salje *Model*-u ili *View*-u kao odziv na korisnikovu akciju.

Za cuvanje podataka kreirana je “baza podataka” u json formatu, i napravljen je fake API za pristup tim podacima.

Za pokretanje fake API-ja potrebno je pozicionirati se u folderu u kom se nalazi I izvrsiti komandu “npm run json:server”.

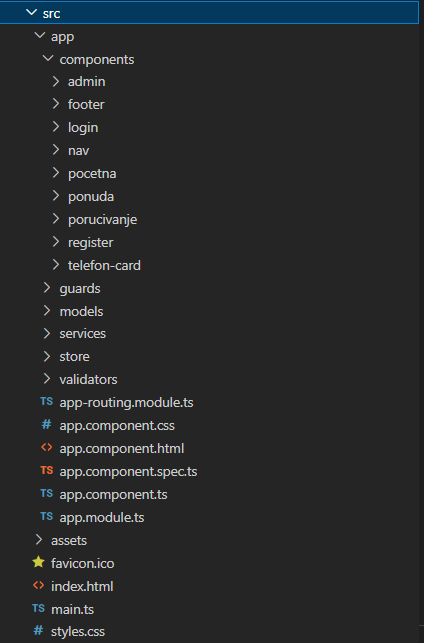
## 2.6. Implementacija sistema

Nakon opisa zahteva i demonstracije predlozene arhitekture sistema, sledi njegova implementacija iz njegovog predlozenog modela. Projekat je realizovan kroz:

* Microsoft Visual Code razvojno okruzenje
* Front-end je uradjen u Angular okviru uz pomoc Bootstrap tehnologije
* Umesto serverskog dela napravljen je fake API koji nam koristi pri realizaciji servisa vezanih za bazu podataka.

Sledi pregled datoteka Angular klijentske strane aplikacije, sledecom slikom je prikazan sadrzaj glavne komponente sistema.

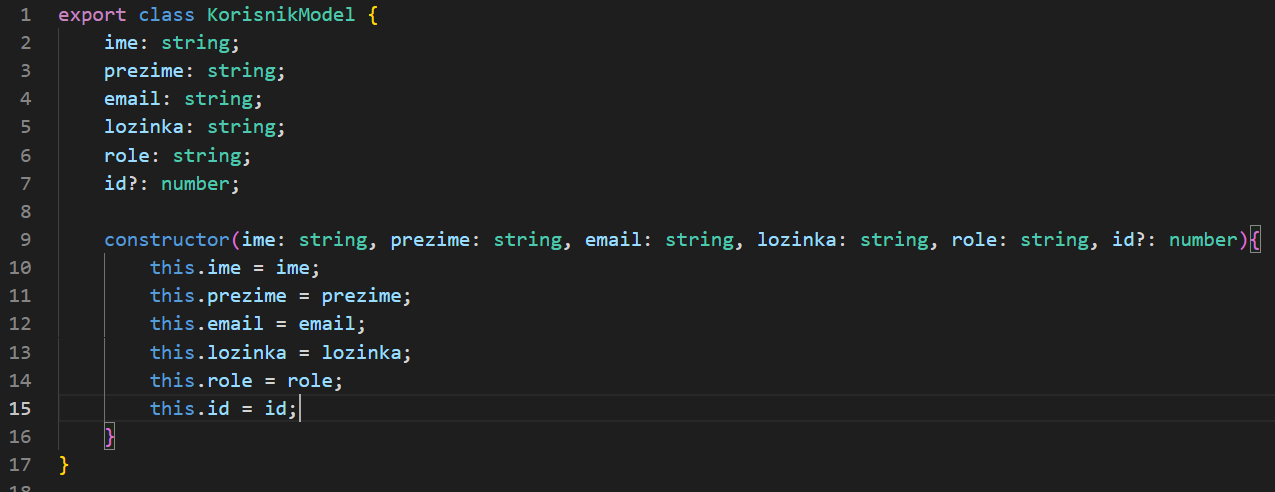
Na narednoj slici mozemo videti sadrzaj src i app foldera.



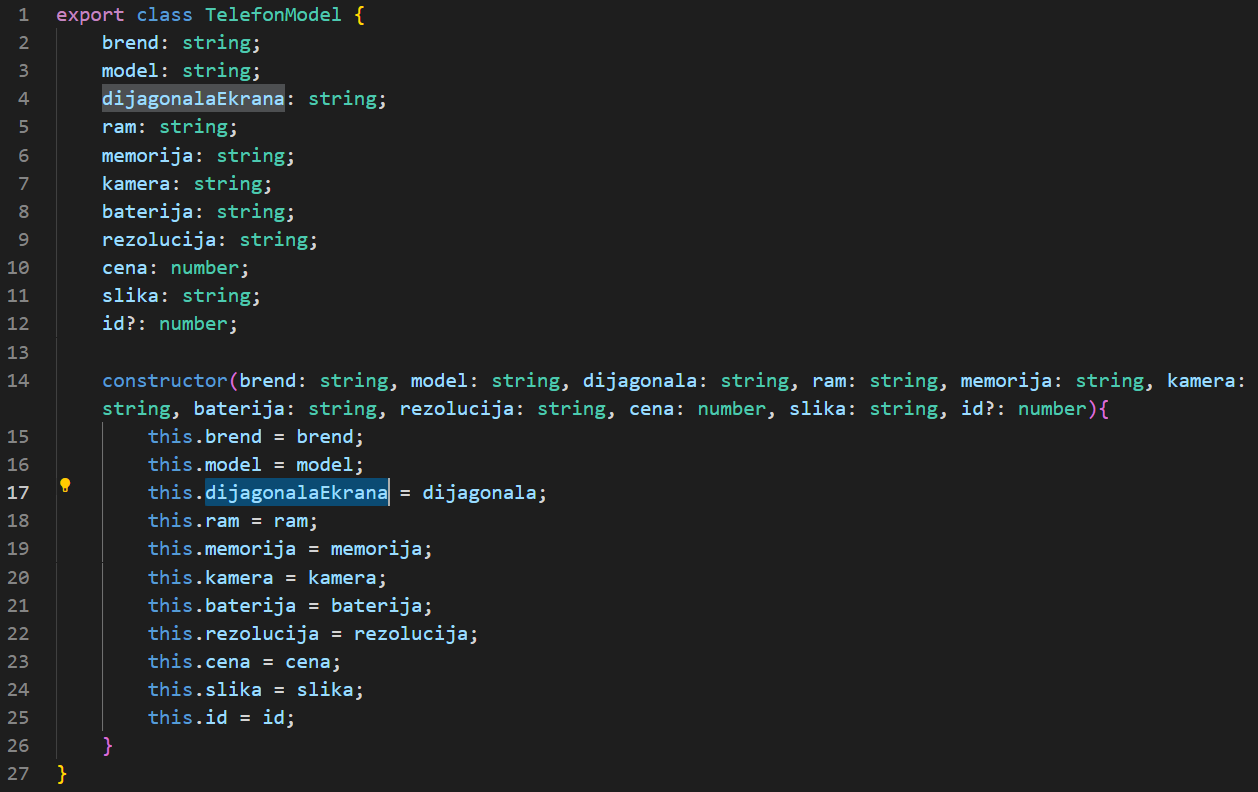
*Slika 13. src i app folderi*

Mozemo videti da su klase lepo rasporedjene po folderima radi bolje preglednosti.

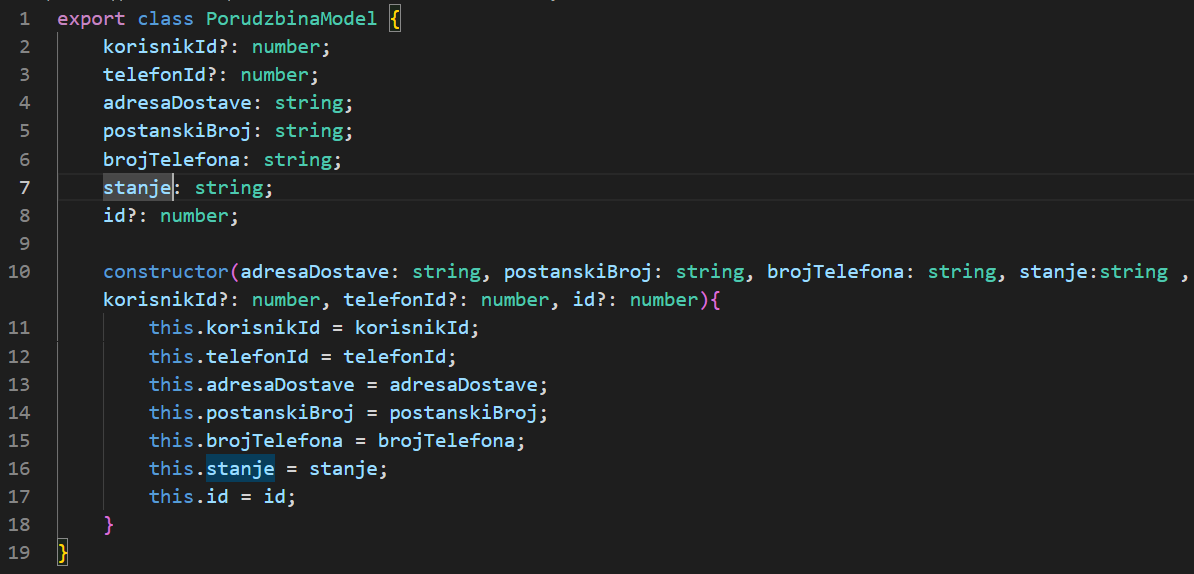
Mozemo kranuti od foldera models gde su smestene modelske klase projekta. Od modela imamo korisnike, telefone i porudzbine. U nastavku ce biti prikazane ove tri modelske klase.



*Slika 14. Modelska klasa Korisnik*

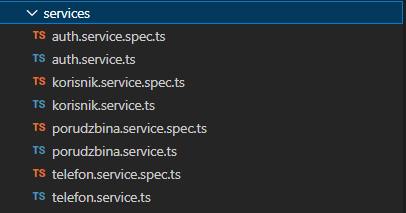


*Slika 15. Modelska klasa Telefon*



*Slika 16. Modelska klasa Porudzbina*

Sledecom slikom je dat prikaz foldera services gde se nalaze frontend servisi koji su korisceni kroz ovaj projekat.



*Slika 17. Folder „services“*

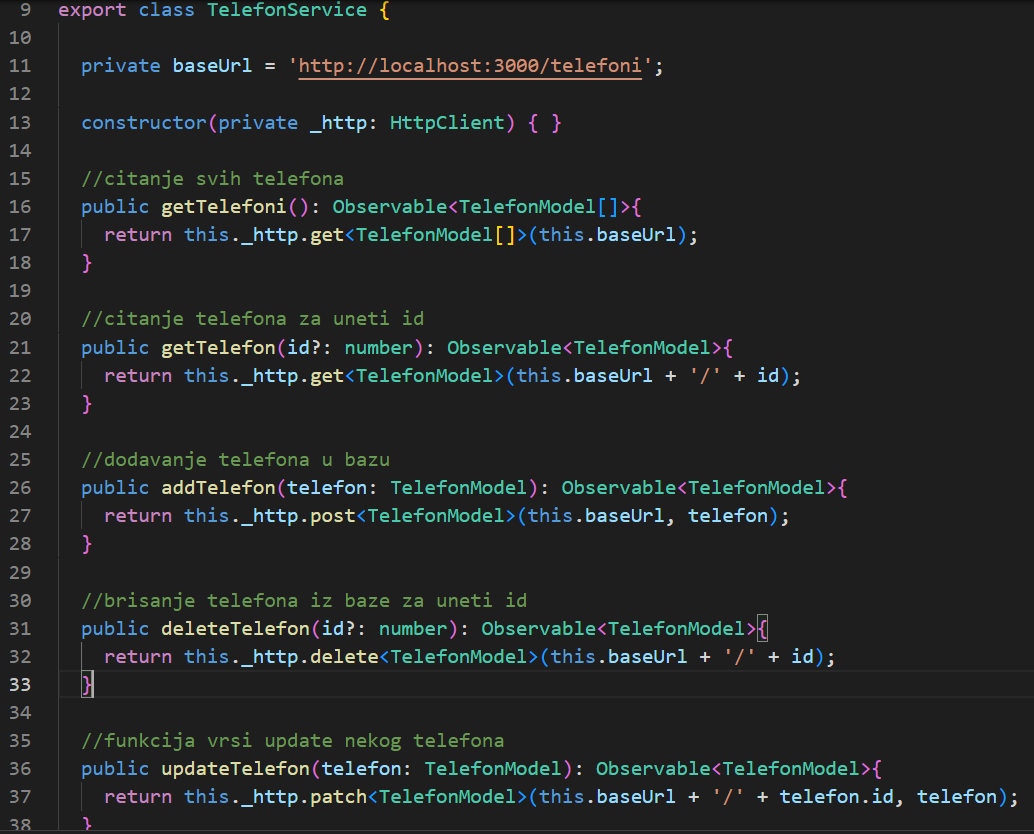
Mozemo videti da imamo 4 servisa:

* Korisnik service koji se koristi za pristup serverskoj strani, tacnije za pristup korisnicima u bazi i vrsenje CRUD operacija
* Telefon service koji se koristi za pristup serverskoj strani, tacnije za pristup telefonima u bazi i vrsenje CRUD operacija
* Porudzbina service koji se koristi za pristup serverskoj strani, tacnije za pristup porudzbinama u bazi i vrsenje CRUD operacija
* Auth service koji se koristi za registraciju i prijavu korisnika na sistem, odjavljivanje korisnika sa sistema, kao i za proveru da li je korisnik ulogovan i za proveru da li je ulogovan administrator

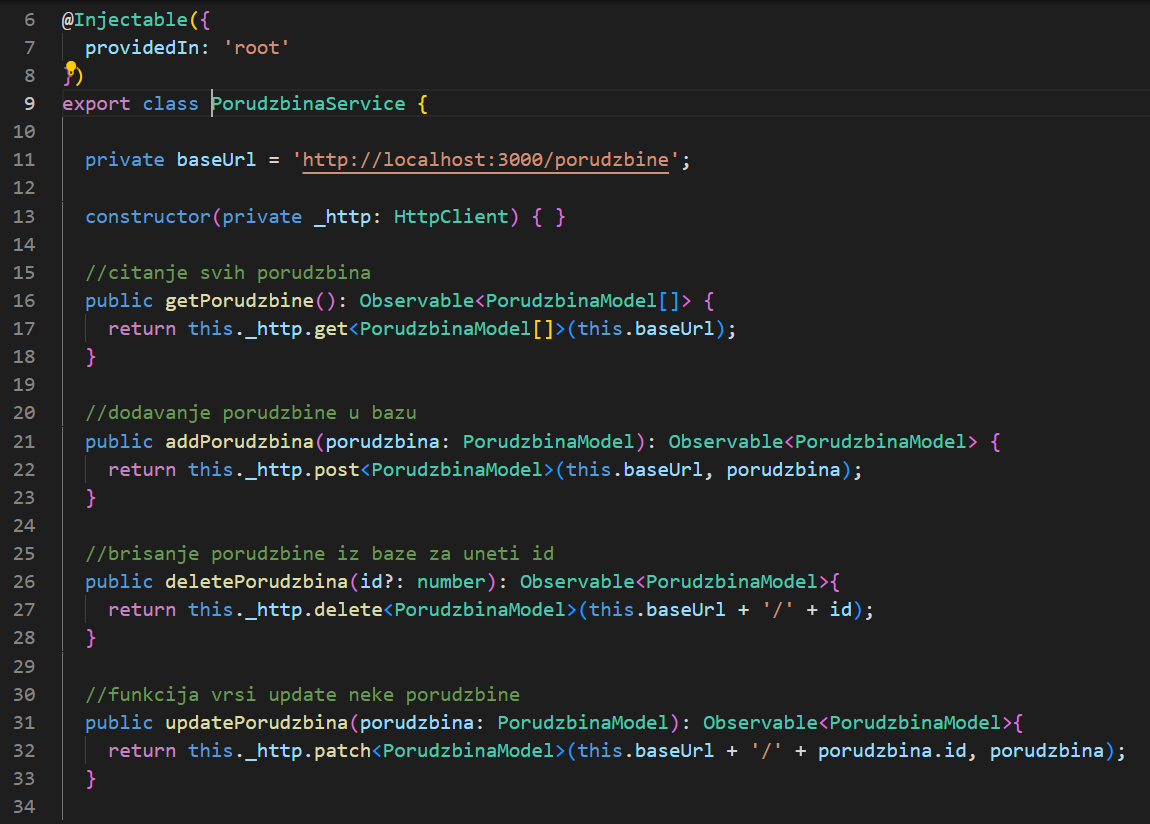
U nastavku ce biti prikazane slike ovih servisa.



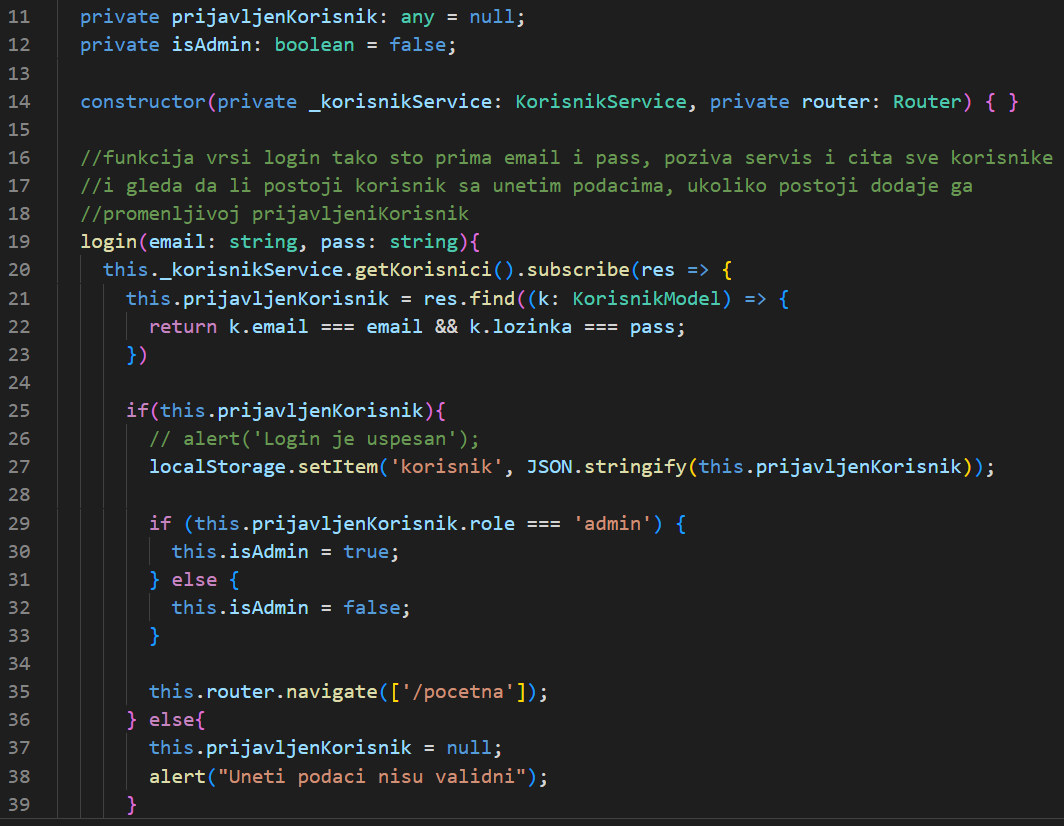
*Slika 18. KorisnikService*

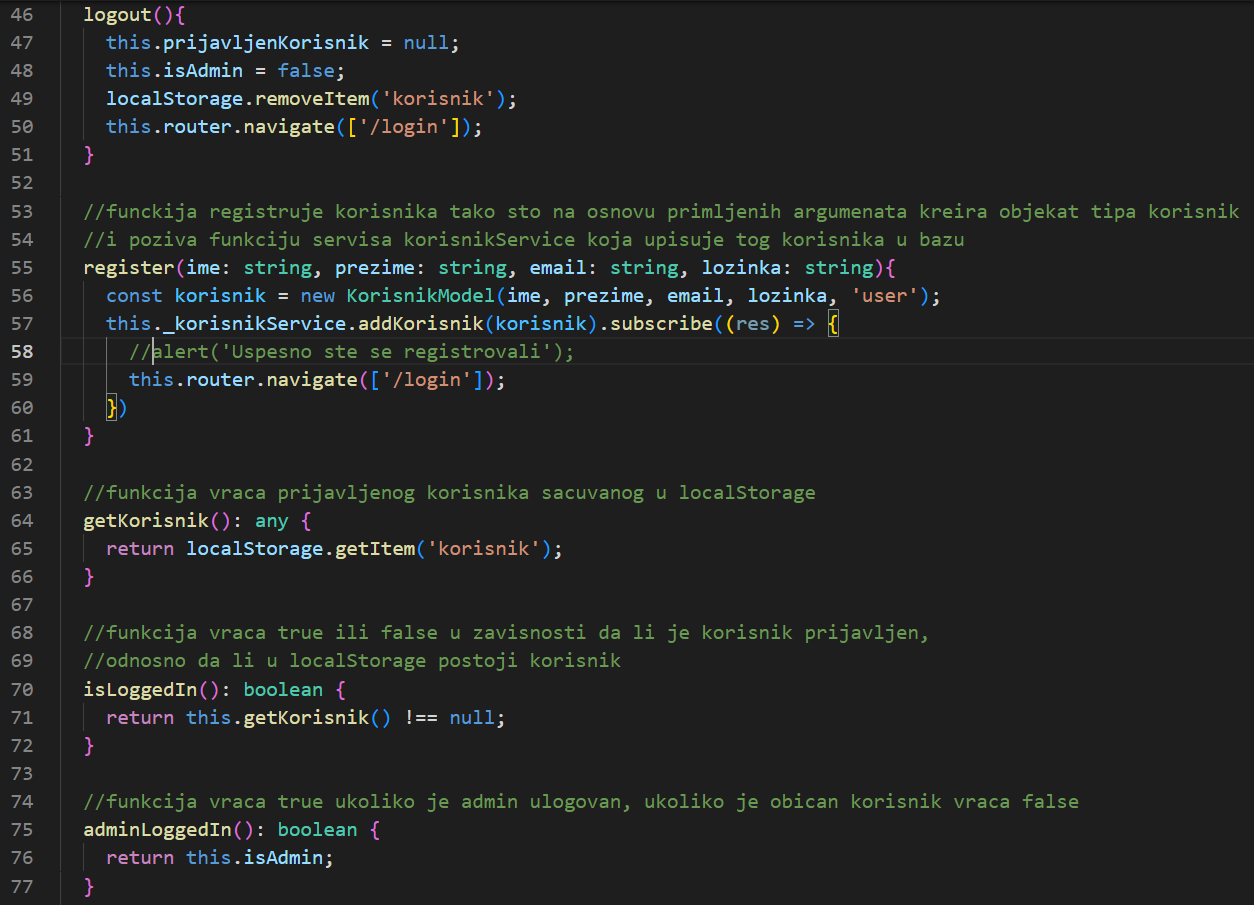


*Slika 19. TelefonService*



*Slika 20. PorudzbinaService*



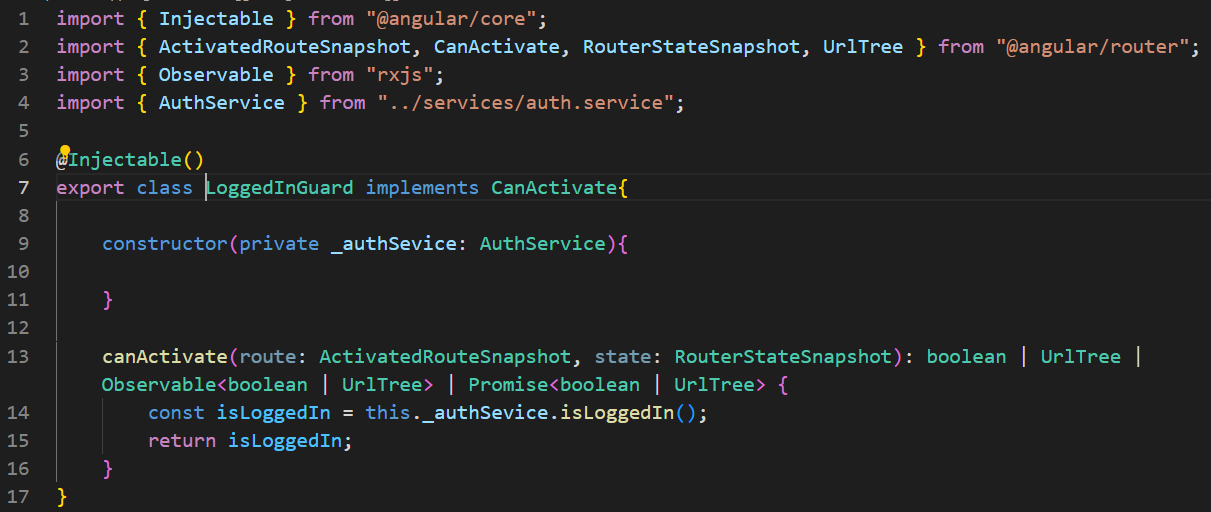


*Slika 21. AuthService*

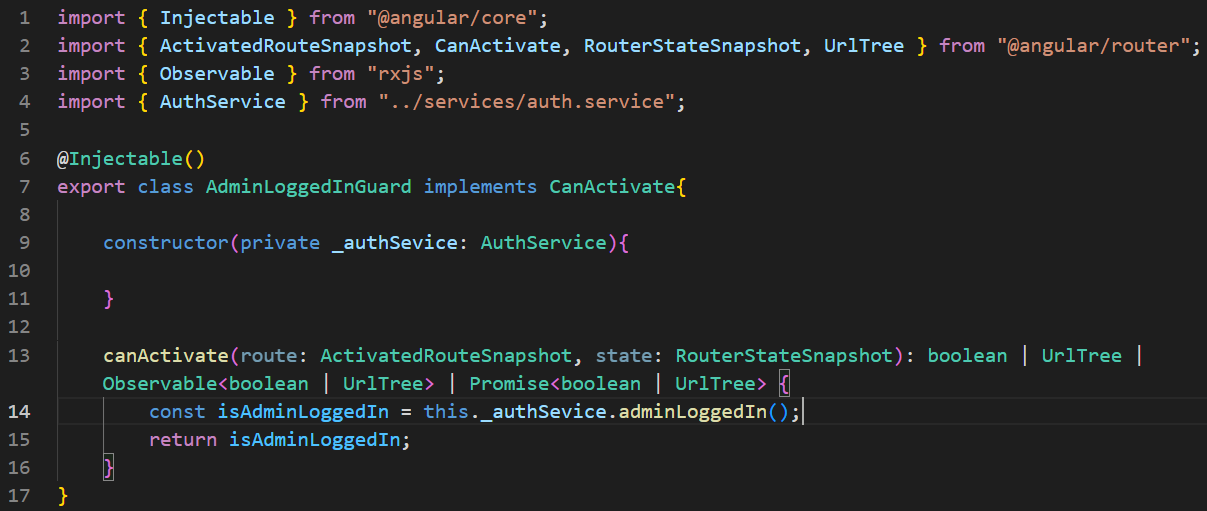
U folderu „guards“ imamo jos dva servisa, to su „guard“ servisi koji implementiraju CanActivate interfejs, i na osnovu ovih servisa se vrsi dozvola „setanja“ kroz navigaciju:

* LoggedInGuard – proverava da li je korisnik ulogovan na sistem ili ne i na osnovu toga dozvoljava otvaranje nekih stranica
* AdminLoggedInGuard – proverava da li je ulogovan korisnik admin ili ne i na osnovu toga dozvoljava otvaranje nekih stranica koje su samo za administratore

U nastavku su slike ova dva „guard“ servisa:



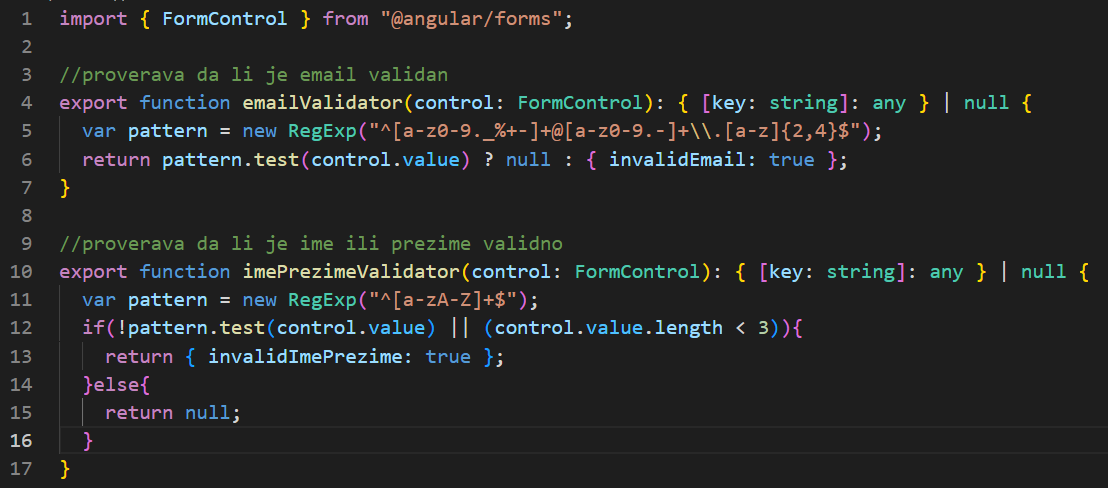
*Slika 22. LoggedInGuard servis*



*Slika 23. AdminLoggedInGuard servis*

Zatim imamo folder „validators“ gde imamo jedan fajl „validators.ts“ koji sadrzi dve funkcije, koje primaju kao parametar neko FormControl polje i proveravaju da li je njihova vrednost validna. Odnosno prva funkcija proverava da li je validna vrednost koju je korisnik uneo u formi za polje email, dok druga proverava da li uneta vrednost za ime ili prezime ima vise od tri karaktera i da li sadrzi samo slova.

U nastavku je slika ovih validatora.



*Slika 24. Validatori*

U folderu „store“ imamo sve potrebne fajlove za realizovanje sablona arhitekture podataka pod nazivom Redux. Odnosno folder „store“ sadrzi sva stanja, akcije, efekte, reducere i selektore.

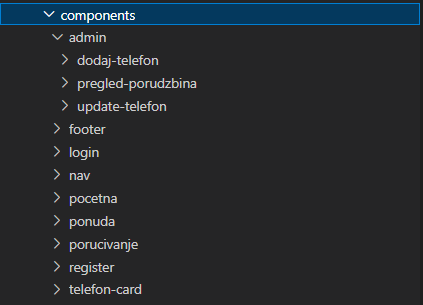
U nastavku je slika ovog foldera.



*Slika 25. Folder „store“*

Folder „components“ sadrzi sve komponente aplikacije, takodje u njemu se nalazi i folder „admin“ koji sadrzi one komponente koje su karakteristicne samo za administratore.

U nastavku je data slika ovog foldera.



*Slika 26. Folder „components“*

Mozemo videte da unutar components foldera imamo za svaku komponentu po jedan folder koji sadrzi 4 fajla a to su:

* Jedan html fajl koji definise sve ono sto korisnik vidi na stranici
* Jedan css fajl koji sluzi za stilizovanje html-a
* Jedan TypeScrip fajl koji je oznacen @Component anotacijom
* Jedan spec.ts fajl

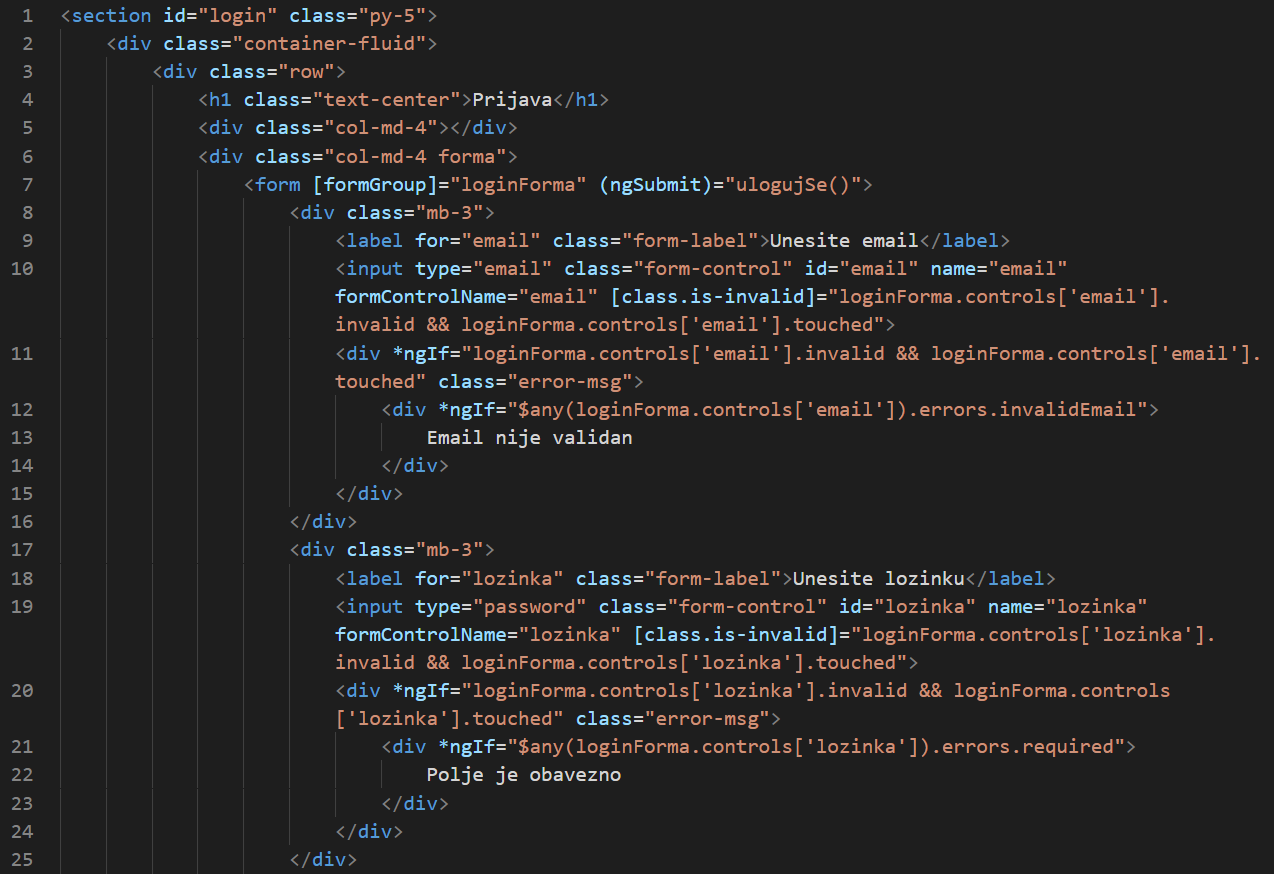
Kako se sve komponente kreiraju na slican nacin u nastavku cemo prikazati sadrzaj jedne komponente, a to ce biti komponenta login koja omogucava login korisnicima.

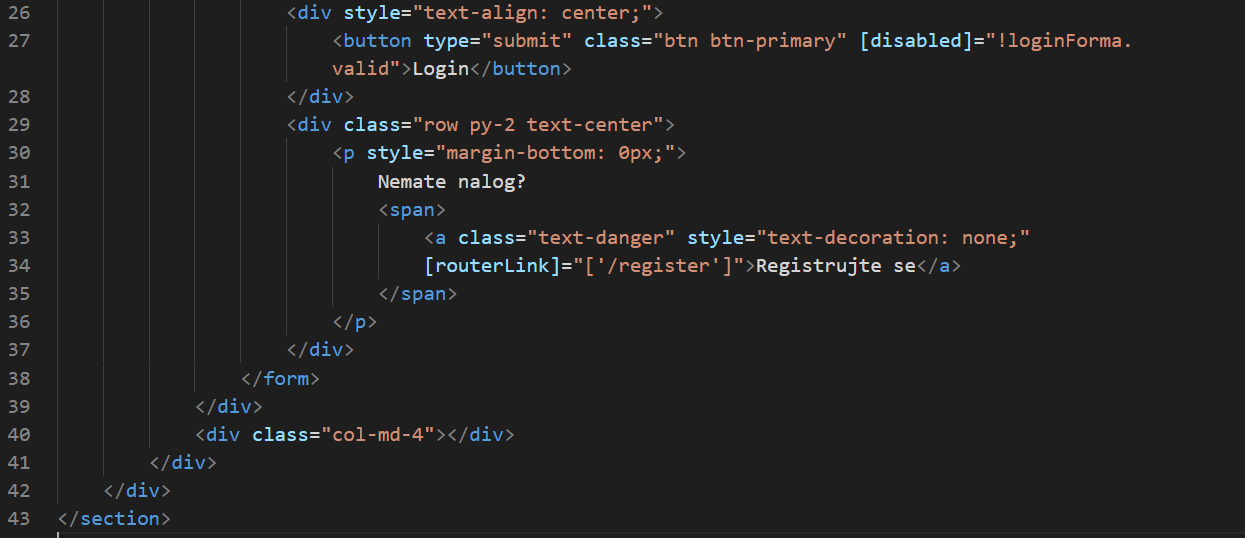
U nastavku cemo prikazati najpre html fajl koji sadrzi sve ono sto korisnik vidi na stranici. Takodje mozemo videte da je npr. forma povezana sa nekim formGroup objektom, svako polje forme povezano sa nekim FormControl objektom, takodje vidimo da se proverava da li su za odredjeno polje uneti validni podaci I ukoliko nisu prikazuje se neka poruka.

Mozemo videte I da se klikom na dugme za submit forme poziva funkcija “ulogujSe()”.

Svi ti objekti I funkcije koje se koriste na html stranici definisani su u TypeScript fajlu za tu komponentu I moze im se pristupiti iz html stranice.

U nastavku ce biti prikaza slika html fajla komponente login.





*Slika 27. Html login stranica*

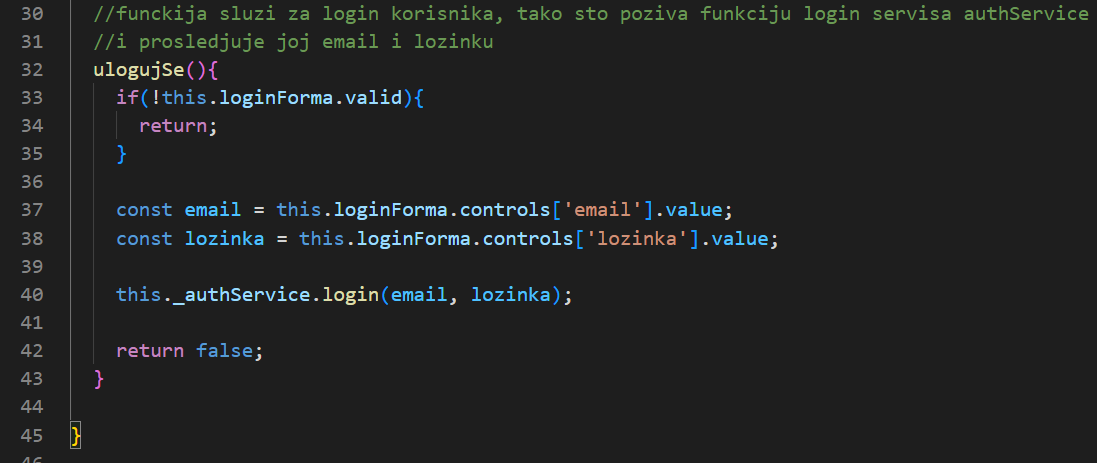
Zatim mozemo videte da svaka komponenta pa tako i ova ima i jedan css fajl u kome se stilizuje html stranice te komponente, ali kako je u ovom projektu koriscen bootstrap I manje modifikacije pisanjem css-a, konkretno za ovu komponentu nemamo nista u css fajlu jer je sve reseno pomocu bootstrapa.

Sada cemo preci na TypeScript fajl, koji nam he ustvari I glavni zaduzeni za kreiranje komponente. Imamo jednu klasu LoginComponent koja je obelezena dekoratorom @Component koji nam I kreira komponentu, unutar njega definisemo vrednost za kljuc “selector” sto ce nam predstavljati kasnije sam html tag pomocu kog se ova komponenta dodaje na stranicu, odnosno njen html kod. Zatim kao vrednost za “remplateUrl” stavljamo putanju do nase html stranice one komponente, i kao vrednost za “styleUrls” stavljamo putanju ka css fajlu.

Zatim mozemo videte da su bas unutar ove klase definisani svi oni objekti i funkcije koji se koriste u html fajlu. Takodje vidimo da je unutar konstruktora pomocu umetanja zavisnosti umetnut objekat servisa “AuthService” koji smo unapred kreirali.

U nastavku cu prikazati sliku ovog fajla.



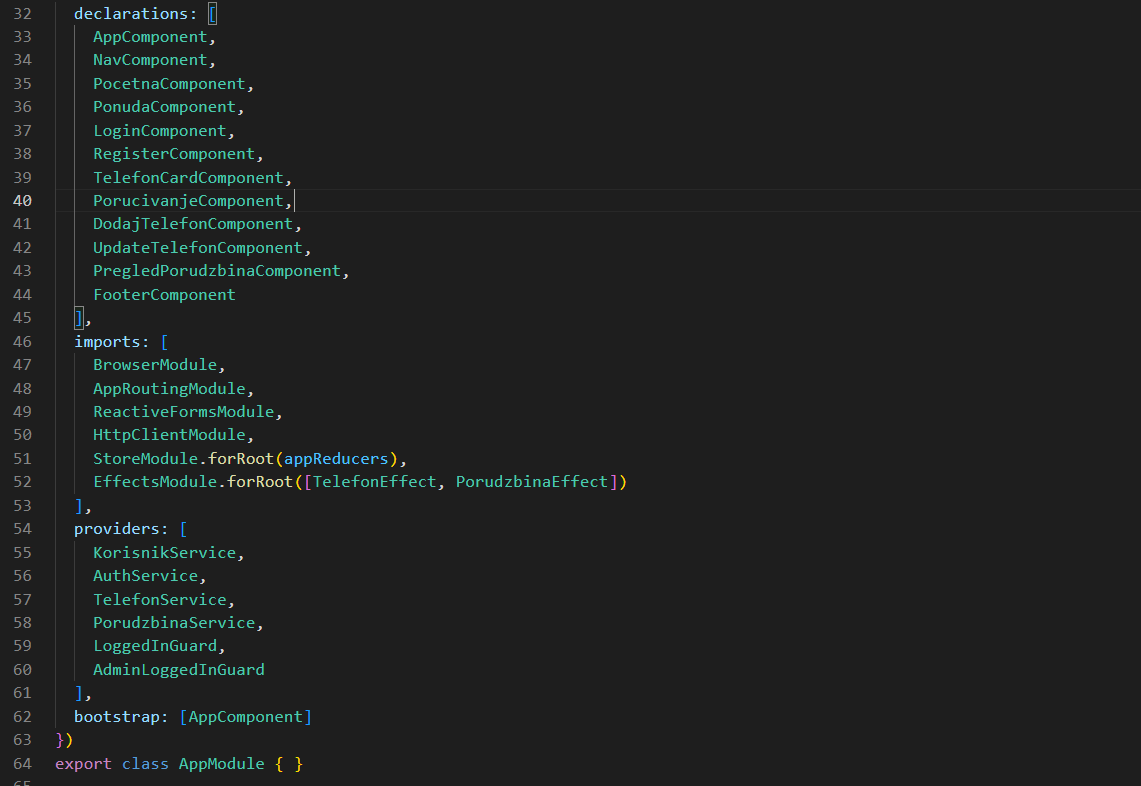


*Slika 28. Kod login komponente*

Sada cu prikazati klasu „AppModule“ koja je oznacena dekoratorom @NgModule, u ovom fajlu moze se videti importovanje modula kao i deklarisanje kreiranih komponenata. Pored toga potrebno je u providers polje dodati i sve servise koji su napravljeni u projektu kako bi mogli da budu umetnuti u klasama komponenata.

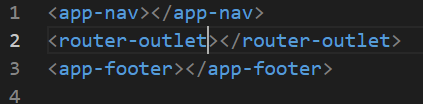
U nastavku je slika ovog fajla.





*Slika 29. „AppModule“ klasa*

Na sledecoj slici mozemo videti sadrzaj glavne „app.component.html“ html stranice, mozemo videti da je na stranici dodata „app-nav“ komponenta, odnosno bice na pocetku stranice prikazan njen html sadrzaj a ispod toga imamo „router-outlet“ i „app-footer“ komponentu.



*Slika 30. app.component.html*

Kako vidimo da se koristi rutiranje u nastavku cemo prikazati sadrzaj „app-router-module.ts“ fajla koji sadrzi sve rute aplikacije.

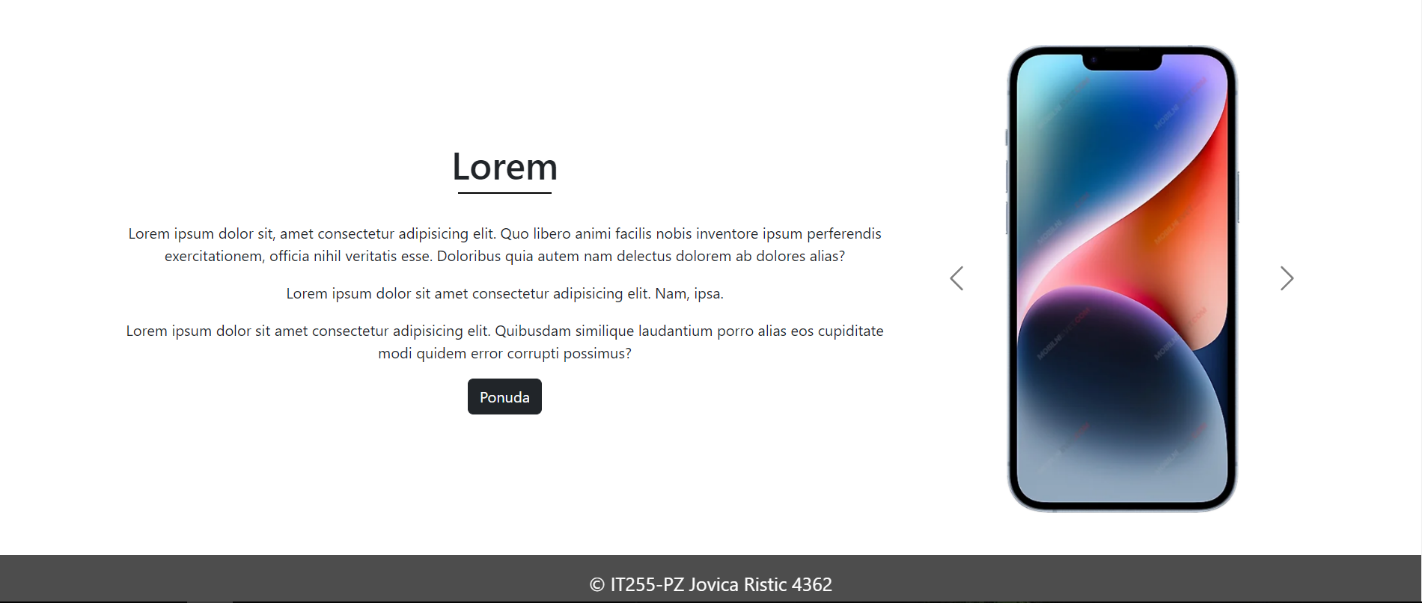


*Slika 31. rute aplikacije*

## 2.7. Prikaz aplikacije

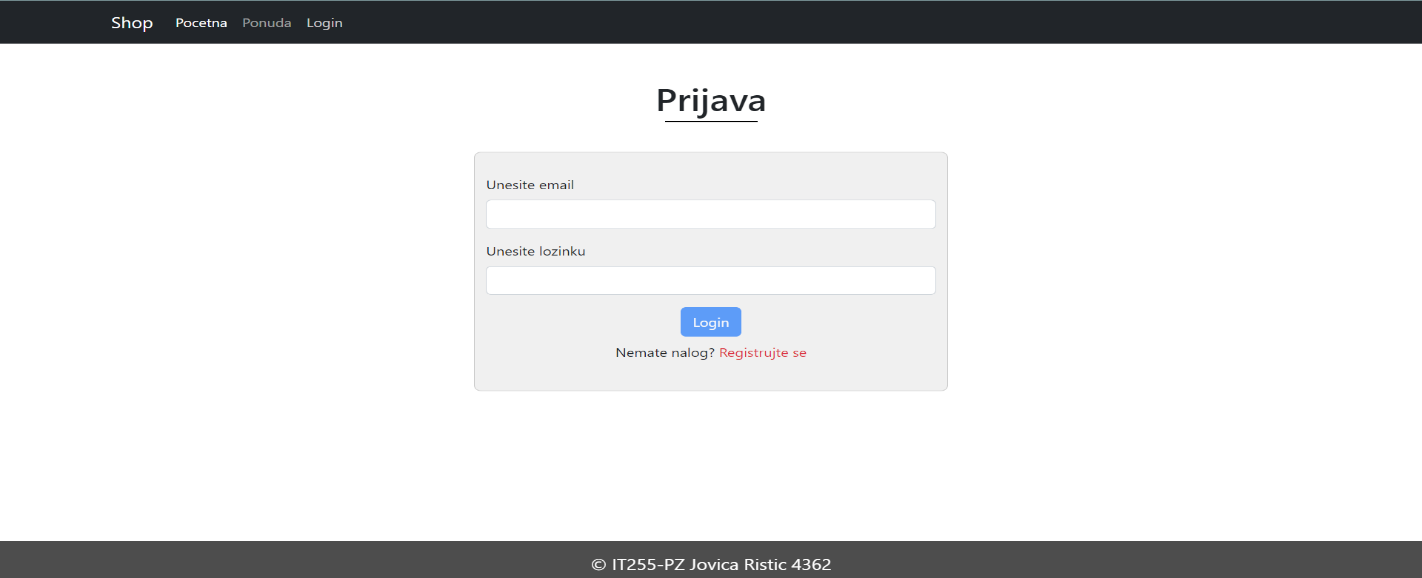
Kada pokrenemo aplikaciju imamo pocetnu stranicu koja izgleda kao na sledecoj slici.



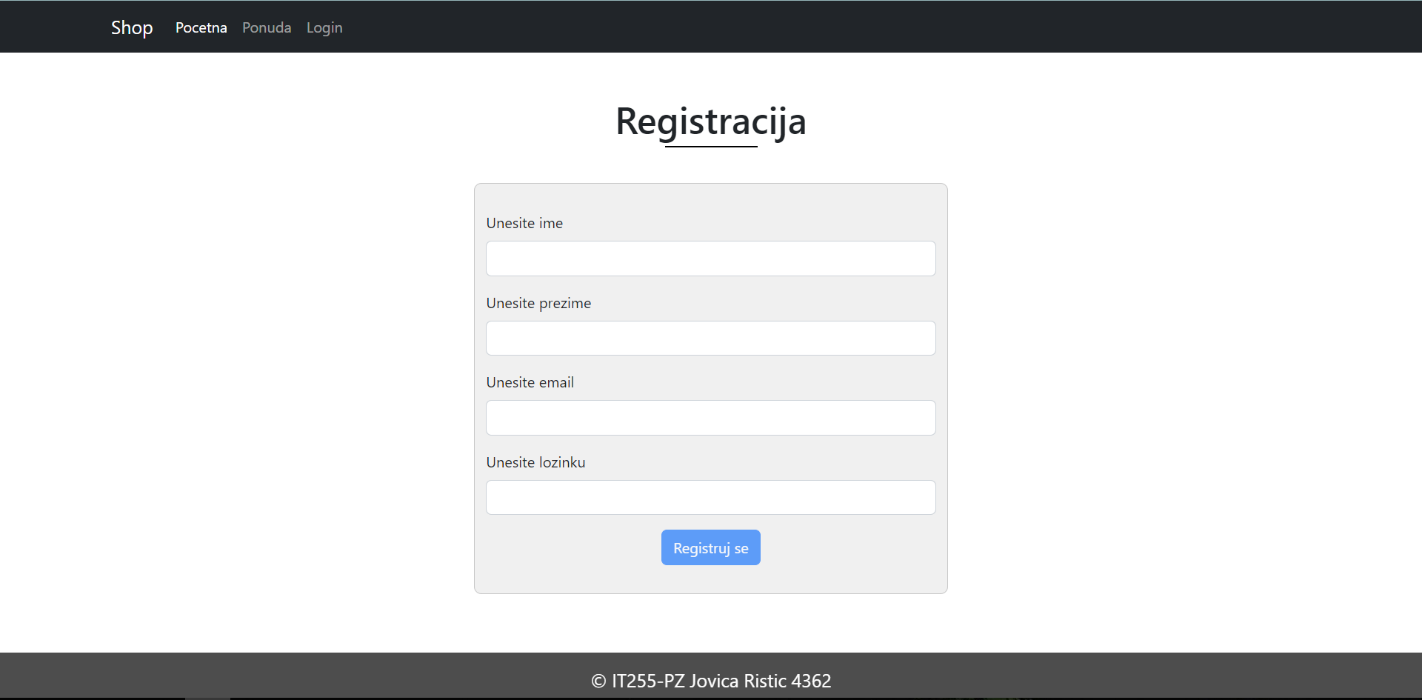


*Slika 32. Pocetna stranica*

Zatim u nastavku mozemo videti stranice za prijavljivanje korisnika ili za registraciju ukoliko nemaju nalog.

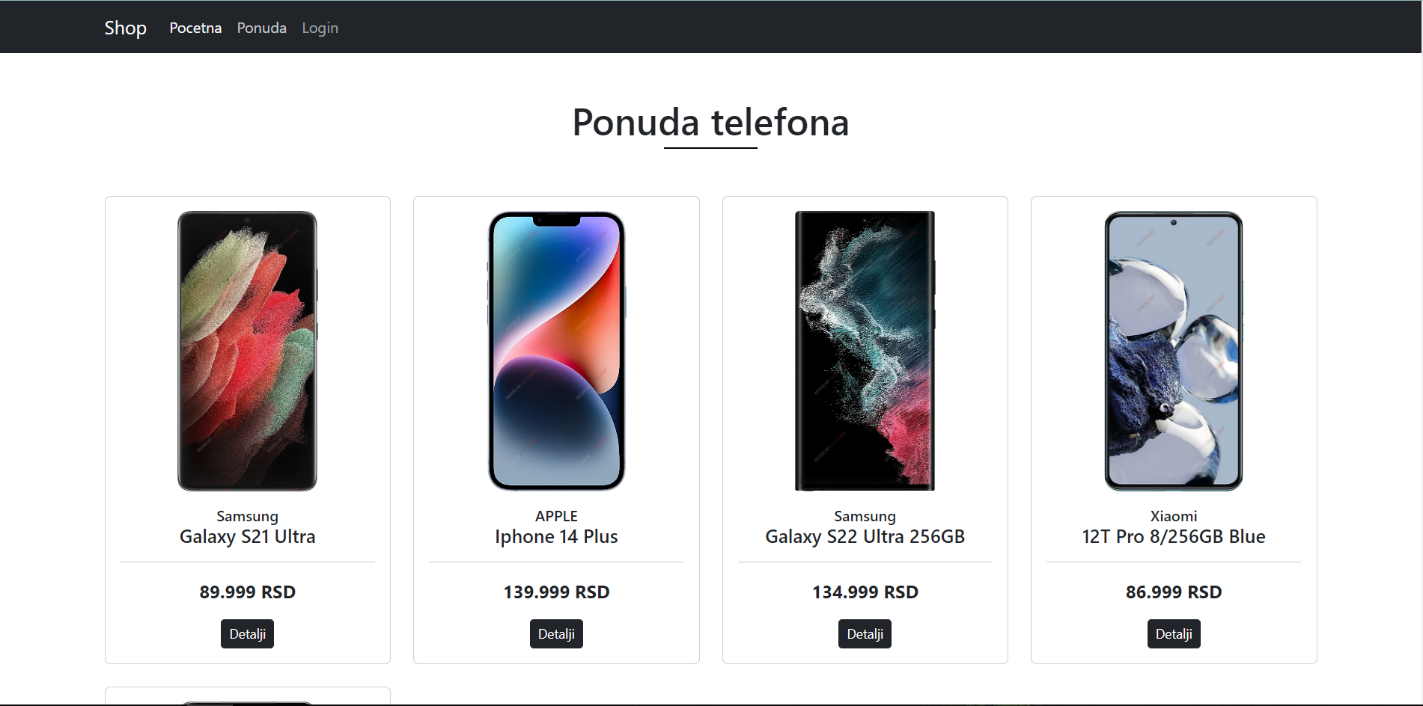


*Slika 33. Login stranica*



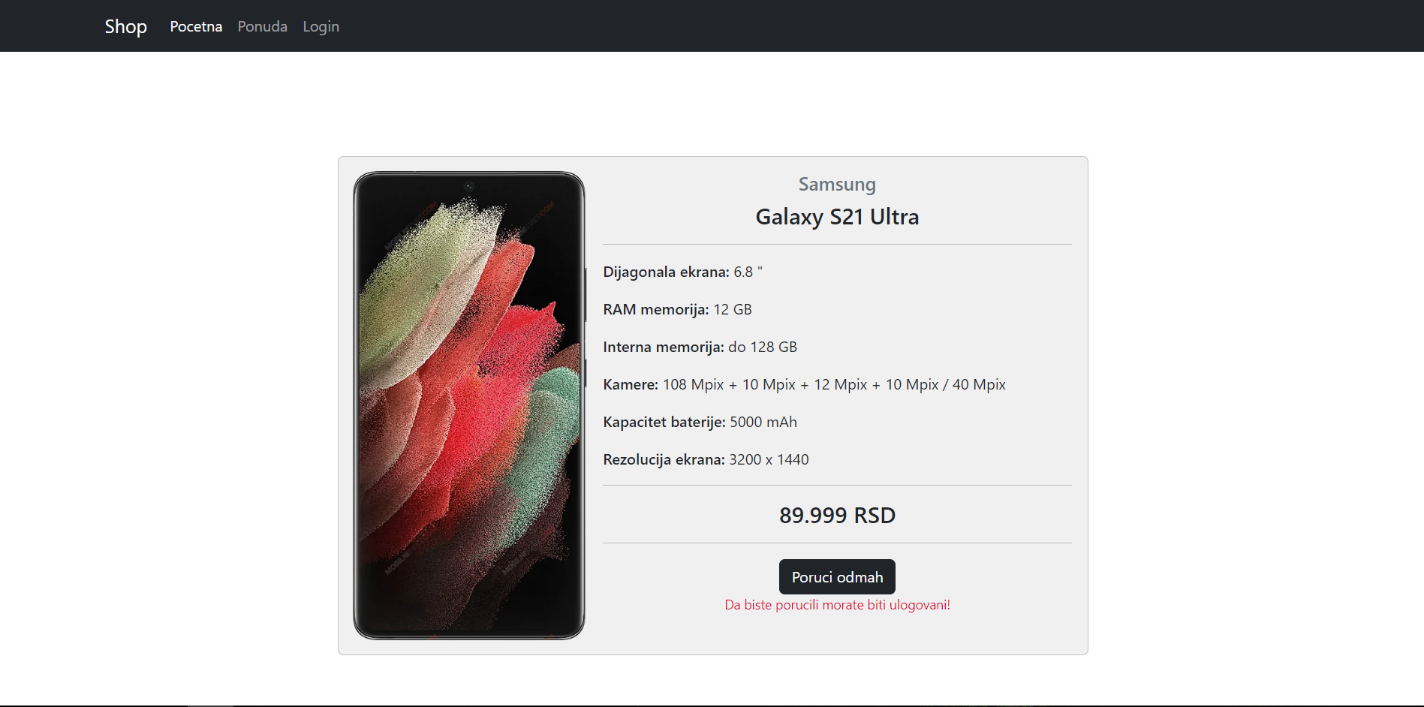
*Slika 34. Registracija stranica*

Zatim u nastavku ce biti prikazana slika stranice sa ponudom telefona.



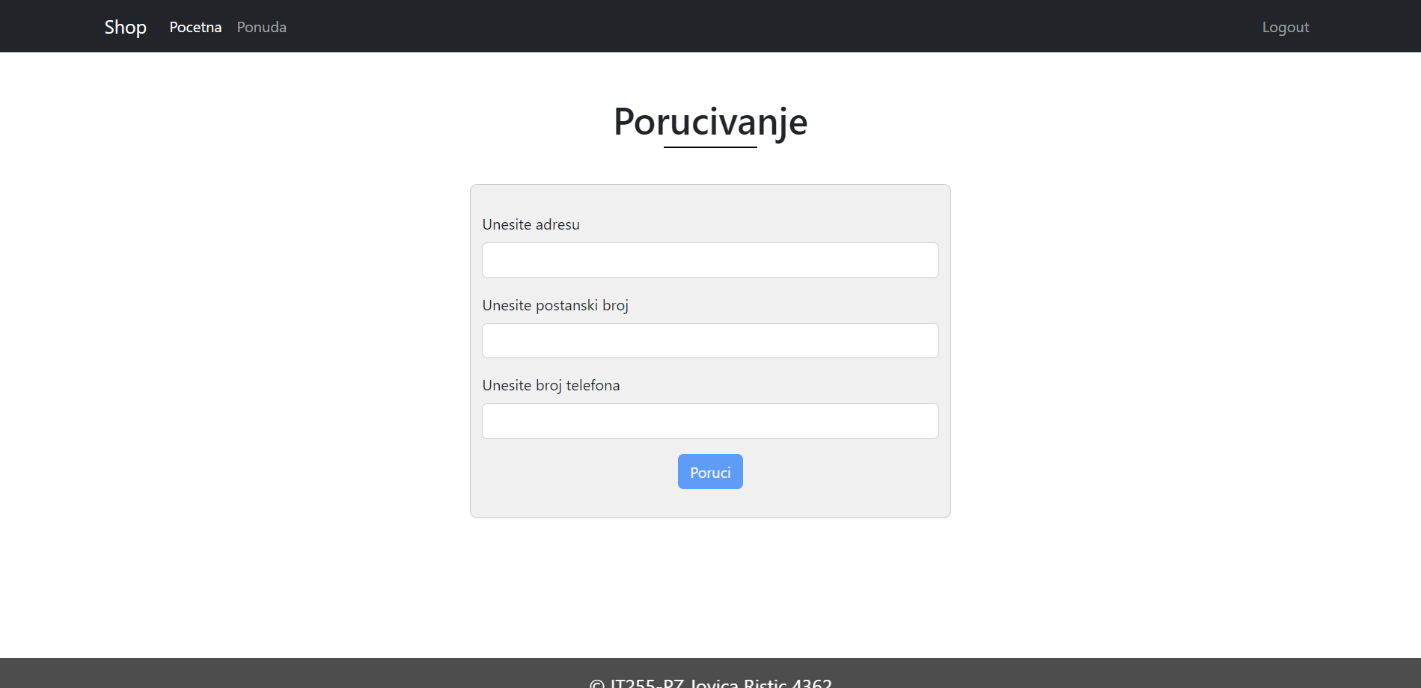
*Slika 35. Ponuda stranica*

Klikom na dugme detalji otvara se stranica sa detaljima za odredjeni telefon. U nastavku je slika stranice.



*Slika 36. Detalji stranica*

Stranica za porucivanje izgleda kao na sledecoj slici.



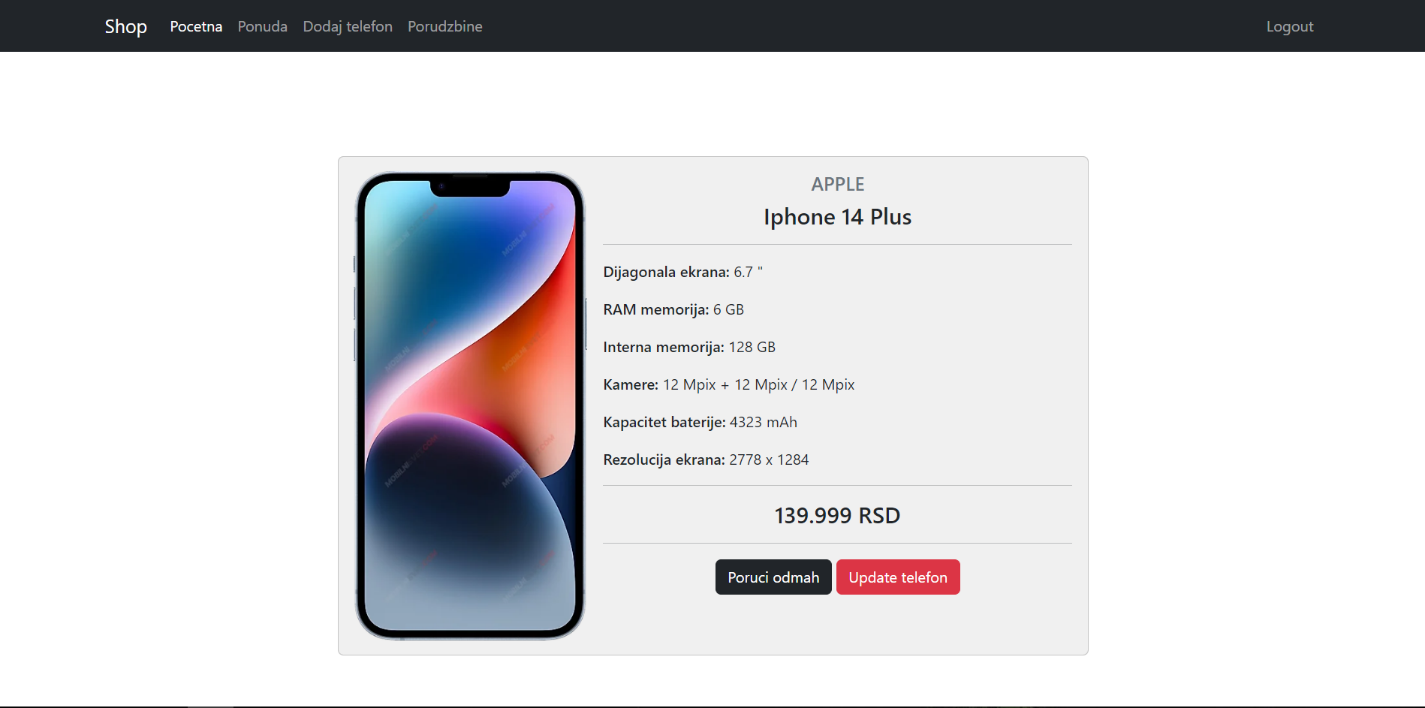
*Slika 37. Stranica za porucivanje*

Ukoliko je ulogovan administrator, on ce na stranici sa ponudom telefona imati za svaki telefon dugme obrisi, klikom na ovo dugme brise se telefon iz baze, i uklanja se sa stranice dostupnih telefona. U nastavku je slika stranice Ponuda za administratore.



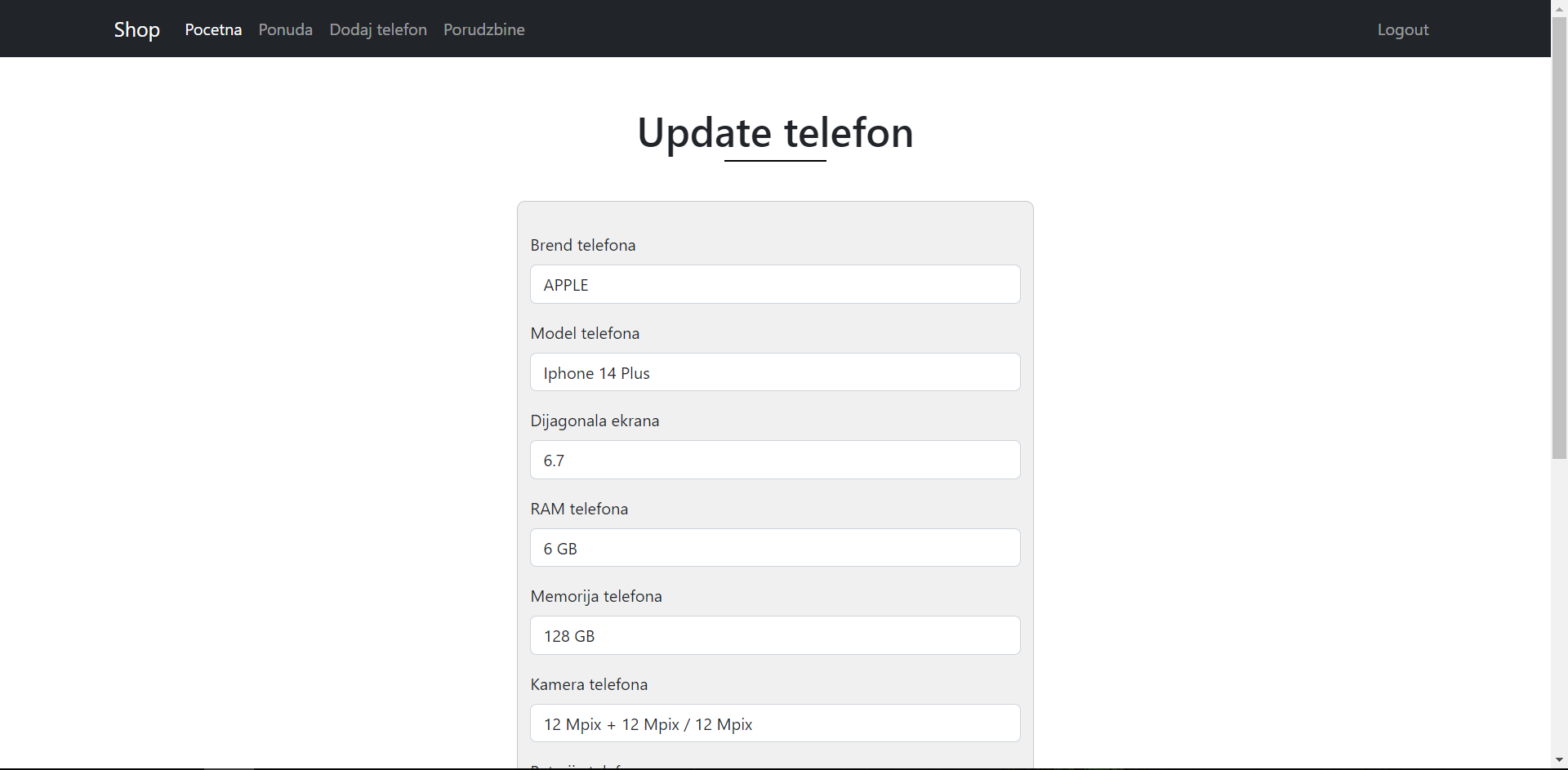
*Slika 38. Stranica ponuda za administratore*

Takodje ukoliko administrator otvori stranicu sa detaljima za neki telefon imace dodatno dugme za update telefona. U nastavku je slika stranice detalji za administratore.



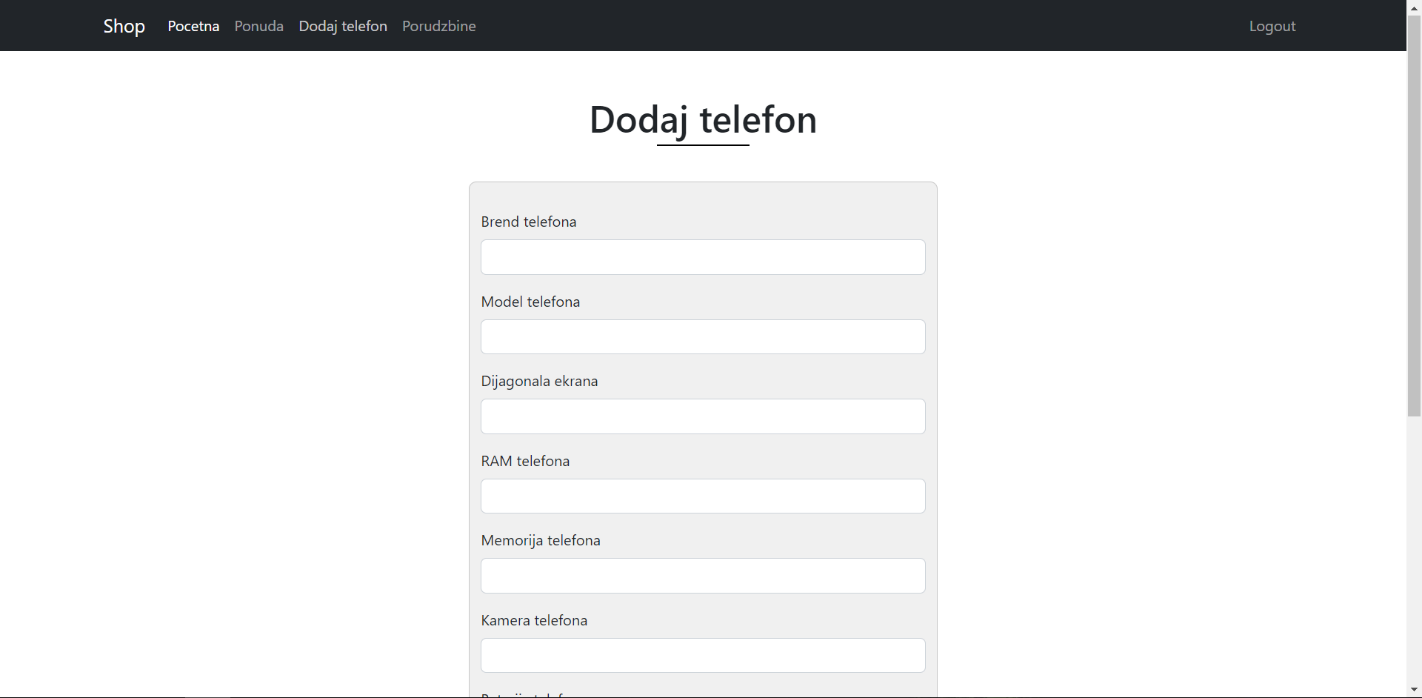
*Slika 39. Stranica detalji za administratore*

Ukoliko administrator klikne na dugme „Update telefon“ otvara se stranica sa formom za izmenu podataka nekog telefona. U nastavku je prikazana slika ove stranice.



*Slika 40. Stranica update telefon za administratore*

Administratori imaju i stranicu sa formom za dodavanje novog telefona u bazu i on ce automatski biti prikazan na stranici sa ponudom. U nastavku je slika stranice za dodavanje novog telefona.

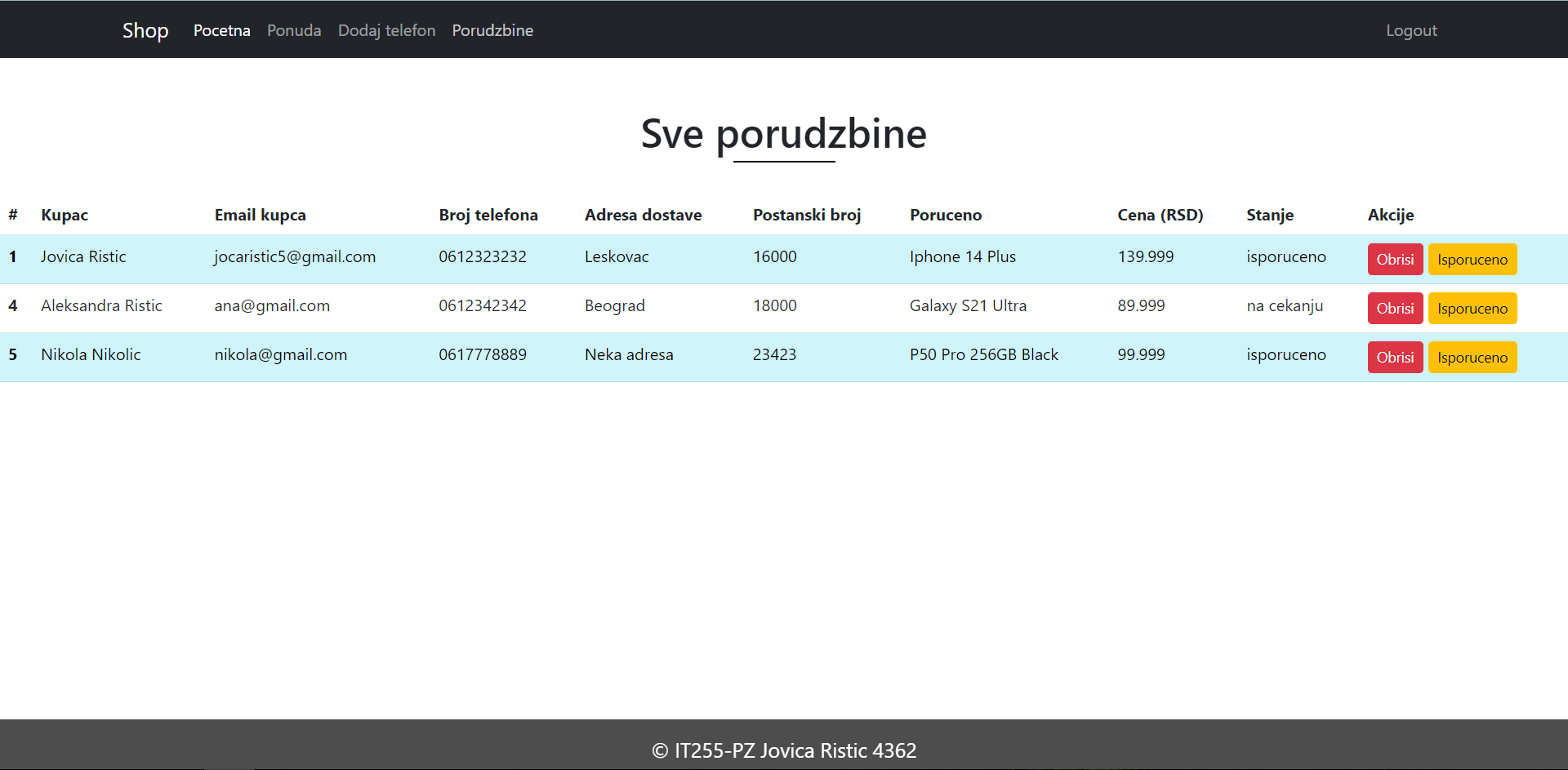


*Slika 41. Dodaj telefon stranica za administratore*

Na kraju administratori imaju jos jednu dodatnu stranicu na kojoj su prikazane sve porudzbine koje su korisnici, odnosno kupci napravili. Administrator moze da promeni stanje porudzbine na isporuceno klikom na dugme „Isporuceno“, cime ce se vrednost stanja promeniti sa „na cekanju“ na „isporuceno“ i promenice se boja pozadine za tu porudzbinu koja je isporucena.

Takodje administratori na ovoj stranici za svaku porudzbinu imaju dugme „Obrisi“ i klikom na njega brisu porudzbinu iz baze, pa se ona uklanja i sa ove stranice.

U nastavku je slika ove stranice.



*Slika 42. Stranica porudzbine za administratore*

# Zaključak

Kroz izradu ove aplikacije, obnovljeno je gradivo koje se predavalo u okviru predmeta IT255 – Veb sistemi 1, i bolje se upoznalo sa Angular okvirom i Bootstrap-om. Napravljena je jedna mini aplikacija koja predstavlja online shop mobilnih telefona, naravno aplikacija ima dosta mesta za njeno unapredjivanje ali mislim da je ovim projektom pokriveno sve sto je radjeno na ovom predmetu.

# Reference

1. Stack Overflow, <https://stackoverflow.com/questions/tagged/angular>

[2] LAMS, [http://lams.metropolitan.ac.rs:8080/lams/index.do#](http://lams.metropolitan.ac.rs:8080/lams/index.do)