

Projekt pri predmetu Praktikum I:

Moj Obrtnik

Avtorji: Tomaž Zajc, Marko Zmazek, Jure Turk, Luka Gričar, Dimitar Micevski

Mentor: Mitja Gradišnik

**Kazalo vsebine :**

[1. **Uvod** 4](#_Toc516740685)

[1.1 Besedilo naloge 5](#_Toc516740686)

[1.2 Specifikacije in zahteve 5](#_Toc516740687)

[1.3 Izbrano ogrodje in jeziki 5](#_Toc516740688)

[2. **Uporabljena orodja in rešitev problema** 5](#_Toc516740689)

[2.1 Laravel 6](#_Toc516740690)

[2.2 Artisan 6](#_Toc516740691)

[2.3 Laracasts 7](#_Toc516740692)

[3. **Podatkovna baza** 7](#_Toc516740693)

[3.1 Načrtovanje podatkovne baze 8](#_Toc516740694)

[3.2 Ustvarjanje podatkovne baze 10](#_Toc516740695)

[4. **Uporabniški vmesnik** 11](#_Toc516740696)

[4.1 Materialize 11](#_Toc516740697)

[4.2 Uporabniška izkušnja in vizualno načrtovanje 11](#_Toc516740698)

[5. **Opis komponent** 12](#_Toc516740699)

[5.1 Poti 12](#_Toc516740700)

[5.2 Kontrolerji 14](#_Toc516740701)

[5.3 Migracije 15](#_Toc516740702)

[5.4 Blade 16](#_Toc516740703)

[5.5 Prijava in registracija 17](#_Toc516740704)

[5.6 E-pošta 18](#_Toc516740705)

[5.7 Naročila, storitve in avtorizacija 19](#_Toc516740706)

[5.8 Ocenjevanje in komentiranje 20](#_Toc516740707)

[6. **Varnost** 21](#_Toc516740708)

[6.1 Kodiranje gesel 21](#_Toc516740709)

[6.2 Verifikacija 21](#_Toc516740710)

[6.3 Nadzor dostopa 22](#_Toc516740711)

[7. **Testiranje** 23](#_Toc516740712)

[7.1. Registracija in vpis obrtnika 23](#_Toc516740713)

[7.2. Iskanje po storitvah 24](#_Toc516740714)

[7.3. Naročanje, ocenjevanje in komentiranje storitev 26](#_Toc516740715)

[7.4. Ustvarjanje in urejanje storitev 29](#_Toc516740716)

[8. **Zaključek** 31](#_Toc516740717)

**Kazalo slik :**

[slika: 1 Primer dokumentacije 5](#_Toc516741538)

[slika: 2 Laracasts 7](#_Toc516741539)

[slika: 3 Osnutek 8](#_Toc516741540)

[slika: 4 Prvoten ER model podatkovne baze 8](#_Toc516741541)

[slika: 5 Končen ER model naše podatkovne baze 9](#_Toc516741542)

[slika: 6 Ustvarjanje tabel v orodju MySQL slika: 7 Ustvarjanje tabel z migracijami 9](#_Toc516741543)

[slika: 8 Skripta za vstavljanje podatkov v podatkovno bazo 10](#_Toc516741544)

[slika: 9 Primer oblikovanja z materialize 11](#_Toc516741545)

[slika: 10 Domača stran 11](#_Toc516741546)

[slika: 11 Poti 12](#_Toc516741547)

[slika: 12 Navigacijska vrstica 12](#_Toc516741548)

[slika: 13 Pot 12](#_Toc516741549)

[slika: 14 Metoda v kontrolerju 12](#_Toc516741550)

[slika: 15 Prikaz strani 13](#_Toc516741551)

[slika: 16 Izbris profila 14](#_Toc516741552)

[slika: 17 Pot za profil 14](#_Toc516741553)

[slika: 18 Metoda za izbris uporabnika 14](#_Toc516741554)

[slika: 19 Predefiniranje tabele za oceno slika: 20 Tuji ključi za tabelo ocena 15](#_Toc516741555)

[slika: 21 Env datoteka 15](#_Toc516741556)

[slika: 22 Artisan ukaz 15](#_Toc516741557)

[slika: 23 phpMyAdmin 16](#_Toc516741558)

[slika: 24 Datoteka blade 16](#_Toc516741559)

[slika: 25 Programska koda za registracijo 17](#_Toc516741560)

[slika: 26 Naročilo 18](#_Toc516741561)

[slika: 27 metoda za pošiljanje e-pošte 18](#_Toc516741562)

[slika: 28 Primer e-pošte 18](#_Toc516741563)

[slika: 29 Metoda za prikaz storitev 19](#_Toc516741564)

[slika: 30 Metoda za prikaz naročil 19](#_Toc516741565)

[slika: 31 Ocenjevanje in komentiranje 20](#_Toc516741566)

[slika: 32 Geslo hesh-ano z bcrypt 21](#_Toc516741567)

[slika: 33 Pogoji 21](#_Toc516741568)

[slika: 34 E-poštni naslov že v uporabi 23](#_Toc516741569)

[slika: 35 Gesli se ne ujemata 23](#_Toc516741570)

[slika: 36 E-mail ali geslo sta napačna 24](#_Toc516741571)

[slika: 37 Razvrščanje po izbrani kategoriji in regiji 24](#_Toc516741572)

[slika: 38 Razvrščanje po najvišji oceni 25](#_Toc516741573)

[slika: 39 Razvrščanje glede na priljubljenost (število vseh skupnih naročil) 25](#_Toc516741574)

[slika: 40 Razvrščanje po najnovejšem/najstarejšem 26](#_Toc516741575)

[slika: 41 Naročanje, ocenjevanje in komentiranje onemogočeno lastniku obrti (ni gumbov) 26](#_Toc516741576)

[slika: 42 Preverjanje izpolnjenost obveznih vnosnih polj 27](#_Toc516741577)

[slika: 43 Ustrezno izpolnjena vnosna polja 27](#_Toc516741578)

[slika: 44 Sporočilo o uspešno oddanem naročilu 28](#_Toc516741579)

[slika: 45 Sporočilo o novih naročilih 28](#_Toc516741580)

[slika: 46 Prejeta naročila 28](#_Toc516741581)

[slika: 47 Razvrščanje naročil 29](#_Toc516741582)

[slika: 48 Ustvarjanje storitve 29](#_Toc516741583)

[slika: 49 Urejanje storitve 30](#_Toc516741584)

[slika: 50 Zavrnjen dostop do urejanja, saj storitev ni v lasti obrtnika 30](#_Toc516741585)

1. Uvod

# Besedilo naloge

**Moj obrtnik**

Načrtujte in izdelajte sistem, ki bo uporabnikov v pomoč pri iskanju obrtnikov ter pošiljanja poizvedb za želena obrtniška dela. V okviru sistema vzpostavite spletno stran, ki bo obiskovalcem omogočala pregled obrtnikov izbrane gospodarske kategorije ter storitev, ki jih ti obrtniki ponujajo. Vsakemu izmed obrtnikov, ki sodelujejo v spletni skupnosti omogočite, da na portalu strukturirano predstavijo sebe ter storitev, ki jih ponujajo (podprite samo obrtnike, ki so storitveno usmerjeni). Ker je vtis predhodnih odjemalcev storitev obrtnika ključen za proces odločanja o najemu obrtnika, odjemalcem portala omogočite komentiranje storitev obrtnikov. Da bi lahko uporabniki portala lažje našli ustreznega obrtnika, implementirajte tudi napredno iskanje po razpoložljivih obrtnikih. Uporabnik naj ima možnost povpraševanja pri izbranem obrtniku.

Mentor: Mitja Gradišnik

# Specifikacije in zahteve

* Iskanje obrtnikov
* Pošiljanje poizvedb (naročil)
* Pregled storitev glede na izbrane kriterije
* Predstavitev Obrtnika
* Komentiranje storitev
* Napredno iskanje
* Možnost povpraševanja pri izbranem obrtniku

# Izbrano ogrodje in jeziki

Kot Izbrano ogrodje oziroma ti. »framework«, smo uporabili Laravel. To ogrodje se nam je zdelo primerno, saj smo kot programski jezik izbrali PHP. Nobeden izmed članov skupine ni še pred začetkom predmeta Praktikum uporabljal tega orodja, zato smo imeli na začetku nekaj težav z ogrodjem, saj ga še nismo znali uporabljat. Laravel nam je bil predlagan s strani asistenta pri predmetu Osnove spletnih tehnologij, saj smo trije člani naše skupine pri predmetu Praktikum za rešitev nalog pri predmetu Osnove spletnih tehnologij uporabljali programski jezik PHP. Kot Integrirano razvojno okolje pa smo poenoteno uporabiljali Microsoft Visual Studio Code. Za ustvarjanje spletnih strani smo uporabli označevalni jezik HTML, za stilsko oblikovanje in raznorazne efekte pa stilski jezik CSS.

1. Uporabljena orodja in rešitev problema

# Laravel

Laravel je eno izmed odprto kodnih orodij za izdelavo interaktivnih spletnih strani. Ustvarjeno je bilo specifično za programski jezih PHP. Tako kot Springboot, je tudi Laravel zasnovan na MVC arhitekturi. Ko smo se odločali za ogrodje, ki ga bomo uporabili nas je Laravel prepričal s svojo preprostostjo in učinkovitostjo. Samo ogrodje ima zelo obsežno vendar preprosto dokumentacijo, ki nam je omogočala lažje razumevanje orodja in vseh funkcij ki jih ogrodje omogoča.



slika: 1 Primer dokumentacije

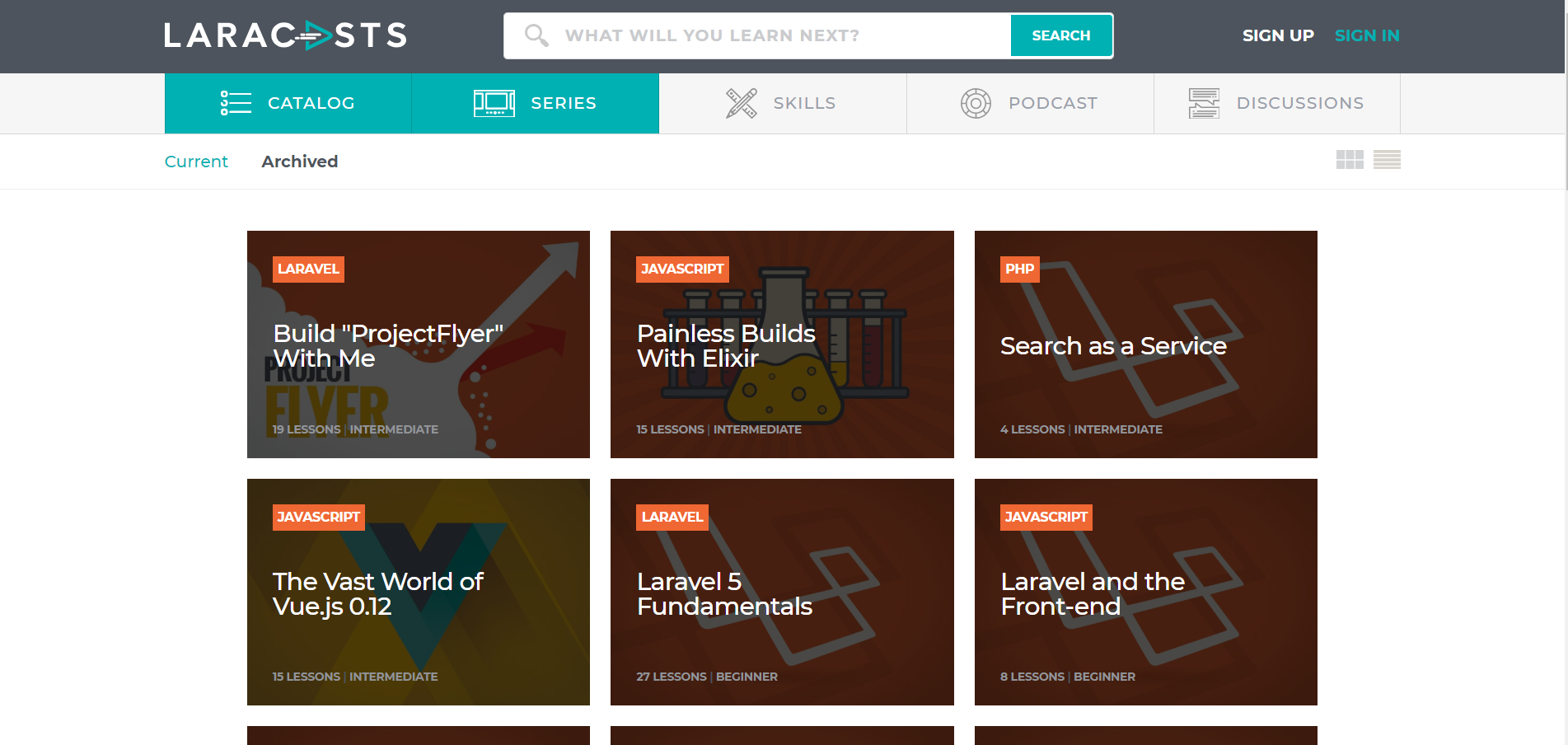
# Artisan

Artisan je implementiran vmesnik orodja Laravel, zagotavlja nam veliko število ukazov, ki nam olajšajo pri razvoju aplikacije. Interakcija z Artisanom poteka preko ukaznega poziva, omogoča pa tudi ustvarjanje novih ukazov, ki jih definiramo sami. Definitivno je Artisan učinkovito in zelo uporabno orodje, ki nam je pri našem projektu bilo v veliko pomoč. Nekaj najpogostejših ukazov ki smo jih uporabljali pri razvoju naše aplikacije:

* Php artisan migrate (migracija podatkovne baze)
* Php artisan serve (Zažene aplikacijo na strežniku)
* Php artisan make(generira izbrane elemente npr: kontroler, blade…)

# Laracasts

K našemu spoznavanju orodja Laravel je zraven dokumentacije veliko pripomogla tudi spletna stran Laracasts. Laracasts je spletna stran namenjena vsem, ki se želijo seznaniti z ogrodjem Laravel. Na njej najdemo veliko video vsebin, ki na kratek in jedrnat način predstavijo uporabo orodja Laravel in njegovih funkcionalnosti. Razen video posnetkov, ki so namenjeni za ogrodje Laravel lahko najdemo tudi ostale video vsebine, ki so povezane z ogrodjem Laravel (PHP, javascript, testiranje, razvojna okolja…).



slika: 2 Laracasts

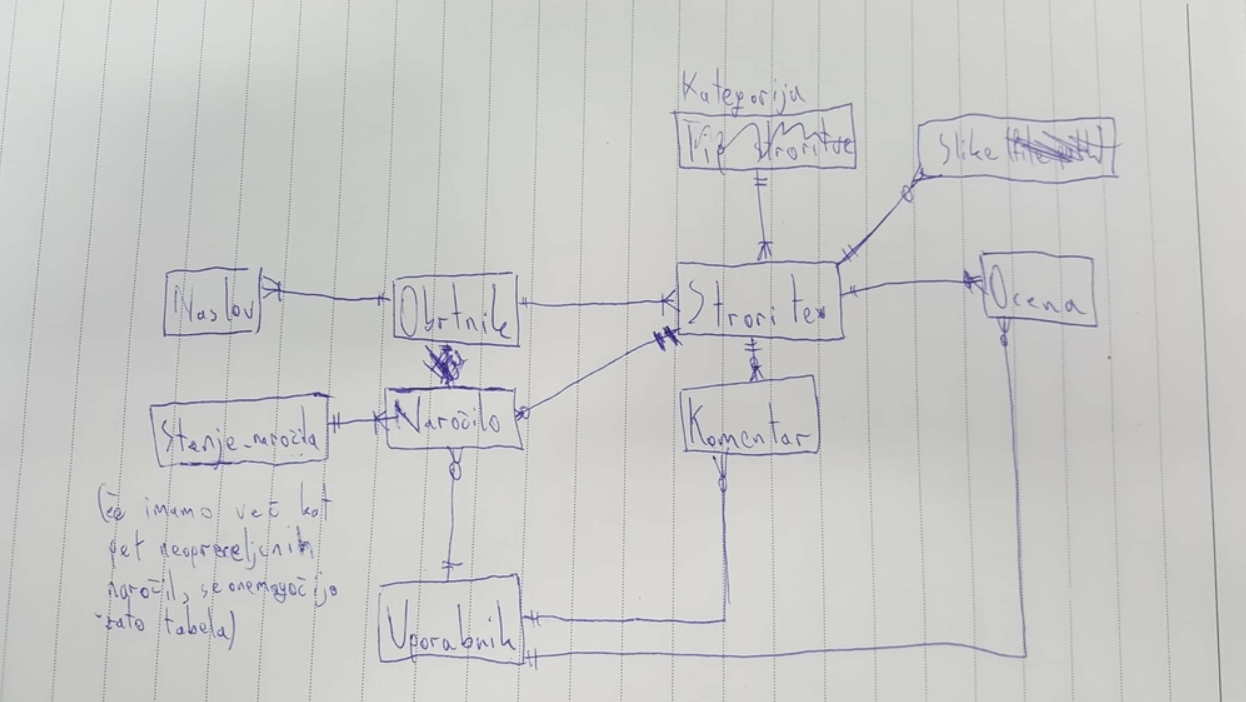
1. Podatkovna baza

Podatkovna baza se je postopoma spreminjala in dopolnjevala. Na začetku projekta smo po ustvarjenem ER modelu ustvarili skripto za ustvarjanje in povezovanje tabel, vendar smo kasneje naleteli na težave pri implementaciji podatkovne baze v naš projekt. Ker nismo našli rešitve, smo namesto vključitev podatkovne baze, generirali podatkovno bazo oziroma je to za nas naredil artisan. Z ukazom »php artisan migrate«, je orodje artisan za nas generiralo podatkovno bazo na podlagi podatkov, ki smo jih vnesli v migracije. Celoten proces [migracije](#migracije) je razložen in predstavljen v poglavju z programsko kodo.

# Načrtovanje podatkovne baze

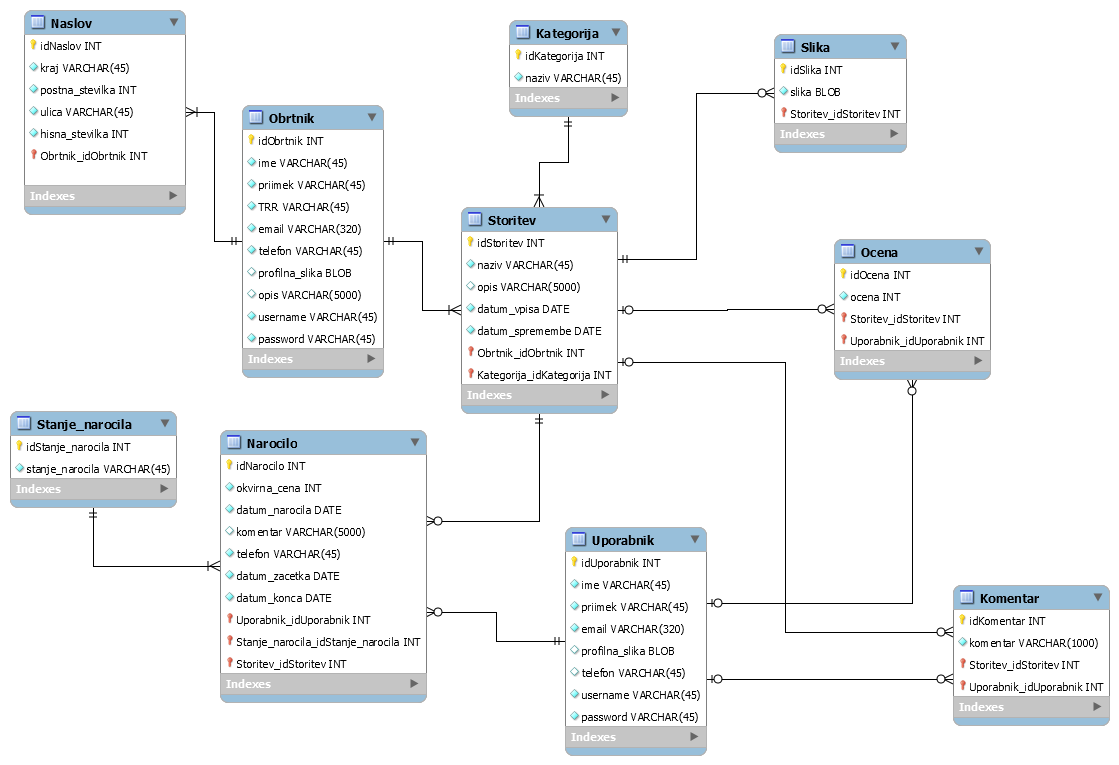
Razen manj pomembnih dodajanj in odstranjevanj atributov lahko razvoj naše podatkovne baze prikažemo v treh fazah. Pri vsaki izmed faz se je naša podatkovna baza spreminjala glede na potrebe in funkcionalnosti naše aplikacije.

**Faza 1:** Prvi korak pri načrtovanju podatkovne baze, je bila določitev entitet, njihovo poimenovanje in relacije med njimi. Tako smo na podlagi tega izdelali osnutek, ki nam je bil v pomoč pri nadaljnem modeliranju in razvoju podatkovne baze.



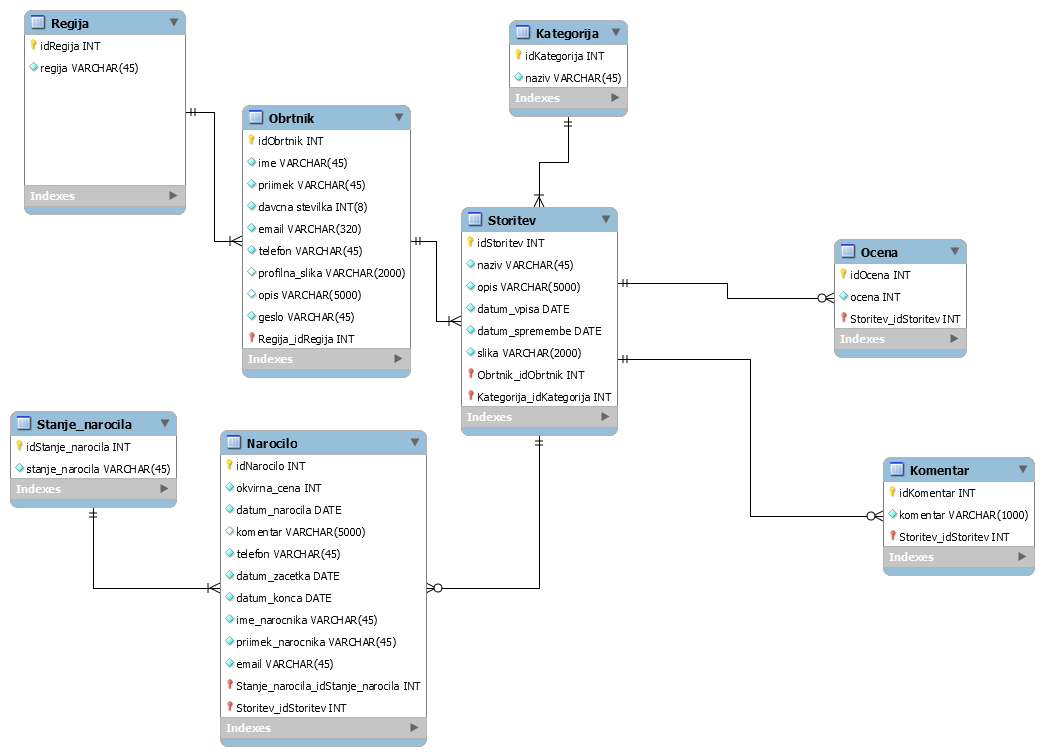
slika: 3 Osnutek

**Faza 2:**  V drugi fazi načrtovanja smo na podlagi osnutka izdelali ER model, dodali atribute in dopolnili model.



slika: 4 Prvoten ER model podatkovne baze

**Faza 3:**  V zadnji fazi načrtovanja smo iz naše podatkovne baze izločili entiteto uporabnika. Zaradi primanjkovanja časa in zahtevnosti projekta smo bili primorani odstraniti ta tip uporabnika. Vse entitete ki so bile povezane z uporabnikom smo zardi tega morali dopolniti z ustreznimi atributi, s katerimi smo lahko delno kompenzirali izločitev uporabnika.

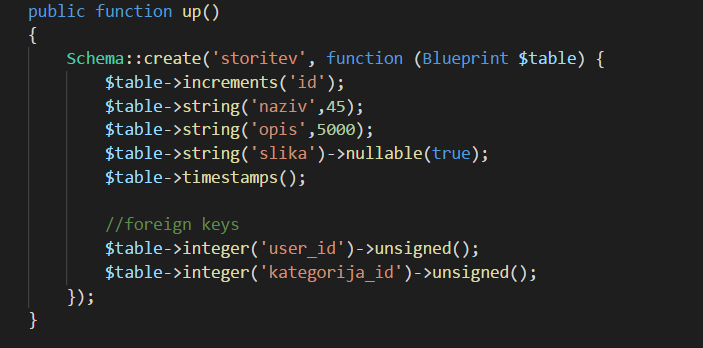
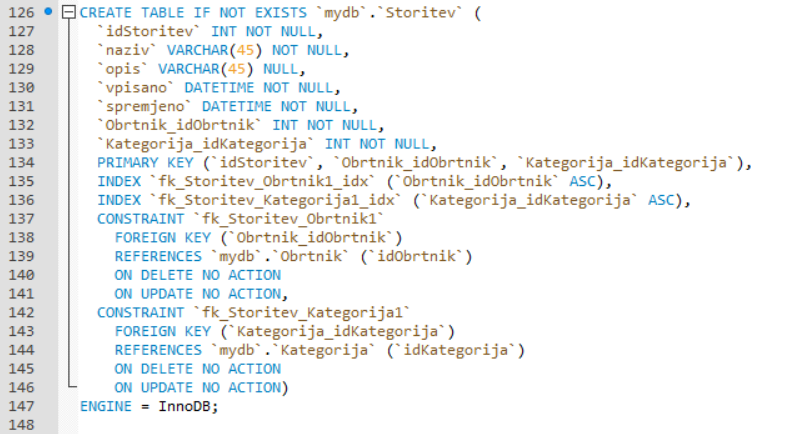


slika: 5 Končen ER model naše podatkovne baze

# Ustvarjanje podatkovne baze

Zaradi zapletov z implementacijo naše že obstoječe podatkovne baze, ki smo jo ustvarili z jezikom SQL , smo našo podatkovno bazo na nov način ustvarili z migracijami. Postopek migracij nam ni povzročal težav, saj smo s pomočjo pregledne dokumentacije in spletne strani laracasts hitro osvojili zahtevano znanje.

Primerjava obeh načinov:



slika: 6 Ustvarjanje tabel v orodju MySQL slika: 7 Ustvarjanje tabel z migracijami

Po postopku predefiniranja migracij smo z orodjem artisan ustvarili našo podatkovno bazo. Testne podatke in podatke potrebne za delovanje aplikacije smo nato v orodju phpMyAdmin z skripto vstavili v podatkovno bazo. V podatkovno bazo smo vključili tudi funkcijo ''on delete cascade'', ki ob izbrisu starševske tabele posredno izbriše tudi ostale tabele povezane z njo. Naša podatkovna baza je tako bila končana in je nemoteno delovala.

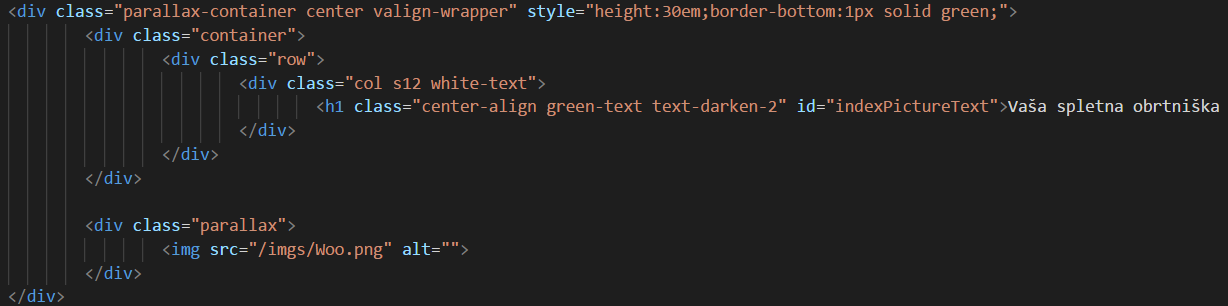


slika: 8 Skripta za vstavljanje podatkov v podatkovno bazo

1. Uporabniški vmesnik

# Materialize

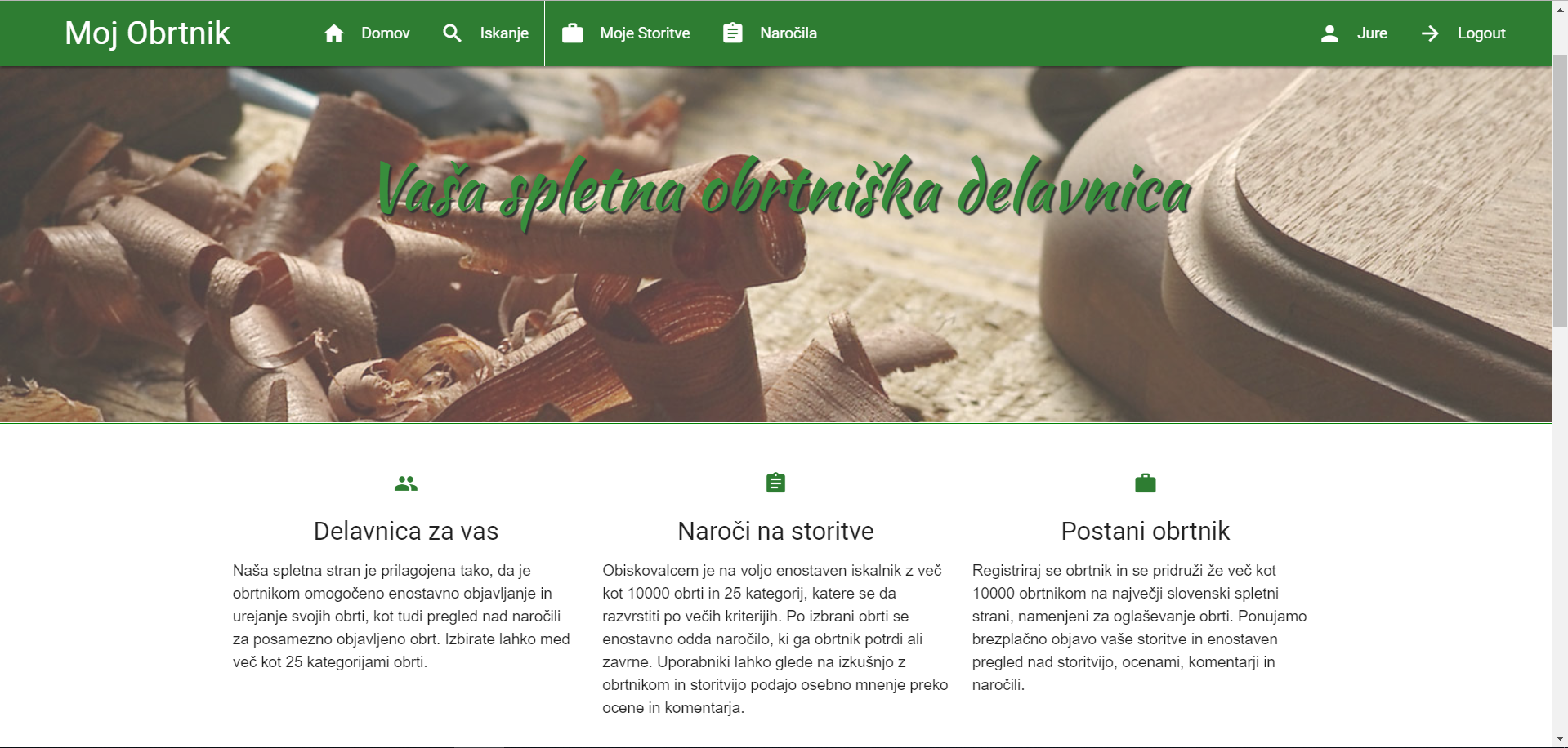
Pri oblikovanju uporabniške vmesnika smo uporabljali stilski jezik CSS. Kot alternativo, smo namesto orodja bootstrap uporabili ogrodje materialize. Materialize je css ogrodje oz. ''CSS framework'', ki temelji na tehnologiji material design. Materialize omogoča odzivnost spletne strani glede na velikost zaslona, na katerem gledamo spletno stran. Med primerjavo obeh orodij smo prišli do zaključka, da materialize omogoča nekoliko majn funkcij kot bootstrap vendar med njima ni bistvenih razlik, ki bi jih lahko izpostavili. Načeloma sta si orodji bootstrap in materialize zelo podobni in delujeta na enaki ideologiji, le da materialize sledi načelom Material design-a, katerega razvoj podpira in financira google.



slika: 9 Primer oblikovanja z materialize

# Uporabniška izkušnja in vizualno načrtovanje

Pri izgledu naše aplikacije smo sledili dobrim praksam, uporabniške izkušnje. Naša aplikacija omogoča enostavno interakcijo z uporabnikom. Pri načrtovanju aplikacije smo želeli naš produkt čimbolj približati uporabniku, saj bo le tako aplikacija uporabnika pritegnila. Tako je naša aplikacija vizualno privlačna in uporabniku omogoča hitro in funkcionalno interakcijo.



slika: 10 Domača stran

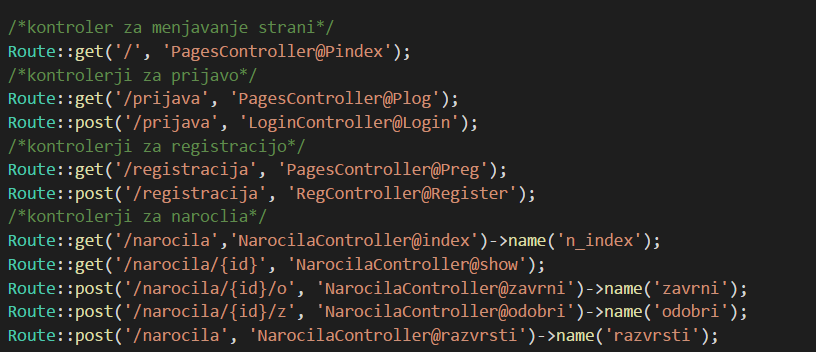
1. Opis komponent

V tem poglavju, so predstavljene glavne komponente naše aplikacije, le te so podrobneje predstavljene in obrazložene z izseki programske kode. Predstavljene komponente:

* Poti
* Kontorolerji
* Migracije
* Blade
* Prijava in registracija
* E-pošta
* Naročila, storitve in autorizacija
* Prijava in registracija

# Poti

Poti oziroma ''Routes'', so pomemben in potreben element kode saj se uporabljajo za koordinacijo med kontrolerji in usmerjajo aplikacijo po poteh ki so definirane.



slika: 11 Poti

*Primer:*

**Korak 1**: Uporabnik v uporabniškem v vmesniku izbere funkcijo iskanje.



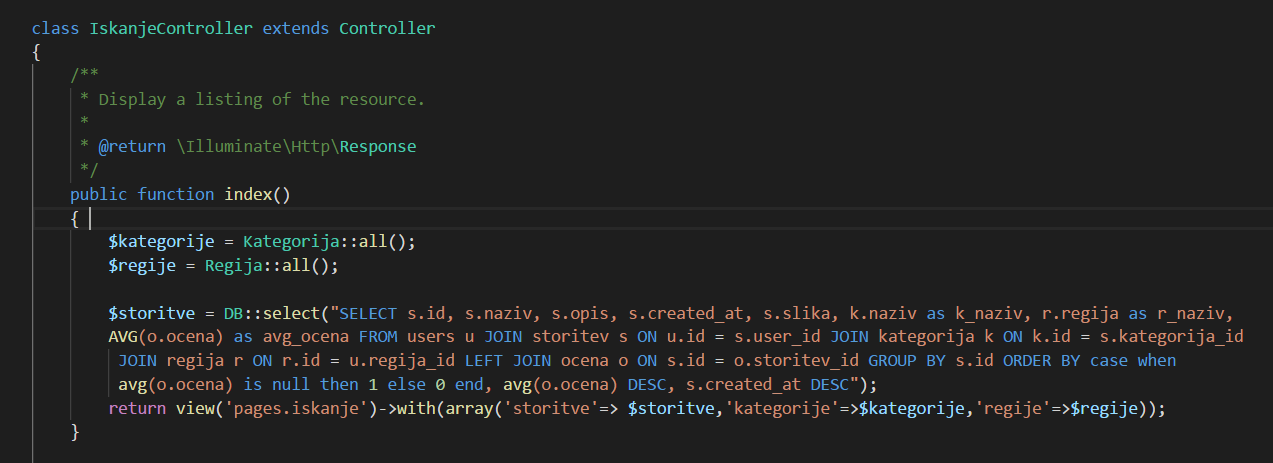
slika: 12 Navigacijska vrstica

**Korak 2:** Ustrezna pot navedena v datoteki proži metodo, v kontrolerju ki je definiran v poti.



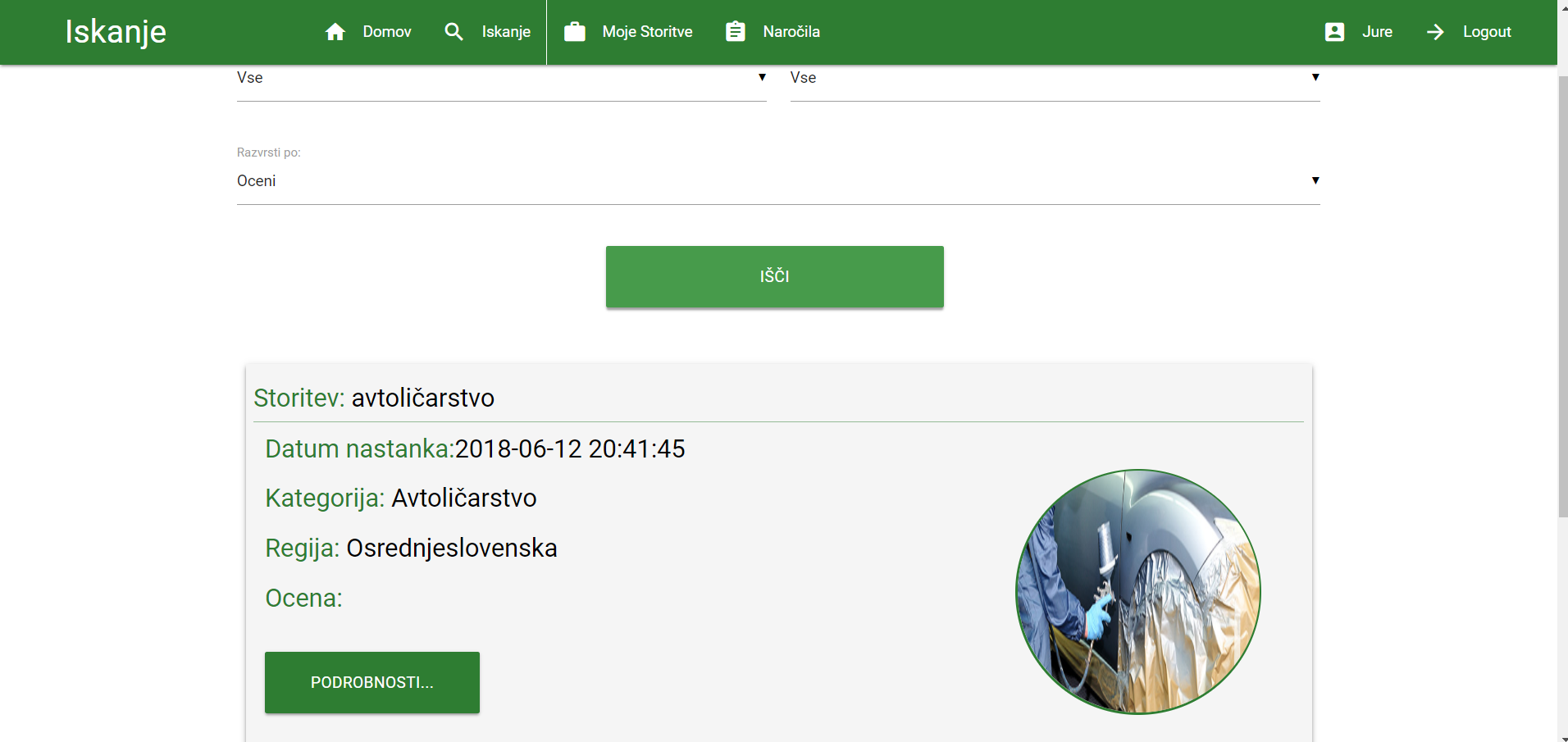
slika: 13 Pot

**Korak 3:** prožijo se metode v navedenem kontrolerju.



slika: 14 Metoda v kontrolerju

**Korak 4:** Metoda iz podatkovne baze izbere vse storitve in jih vrne prikazane na strani ''pages.iskanje''.



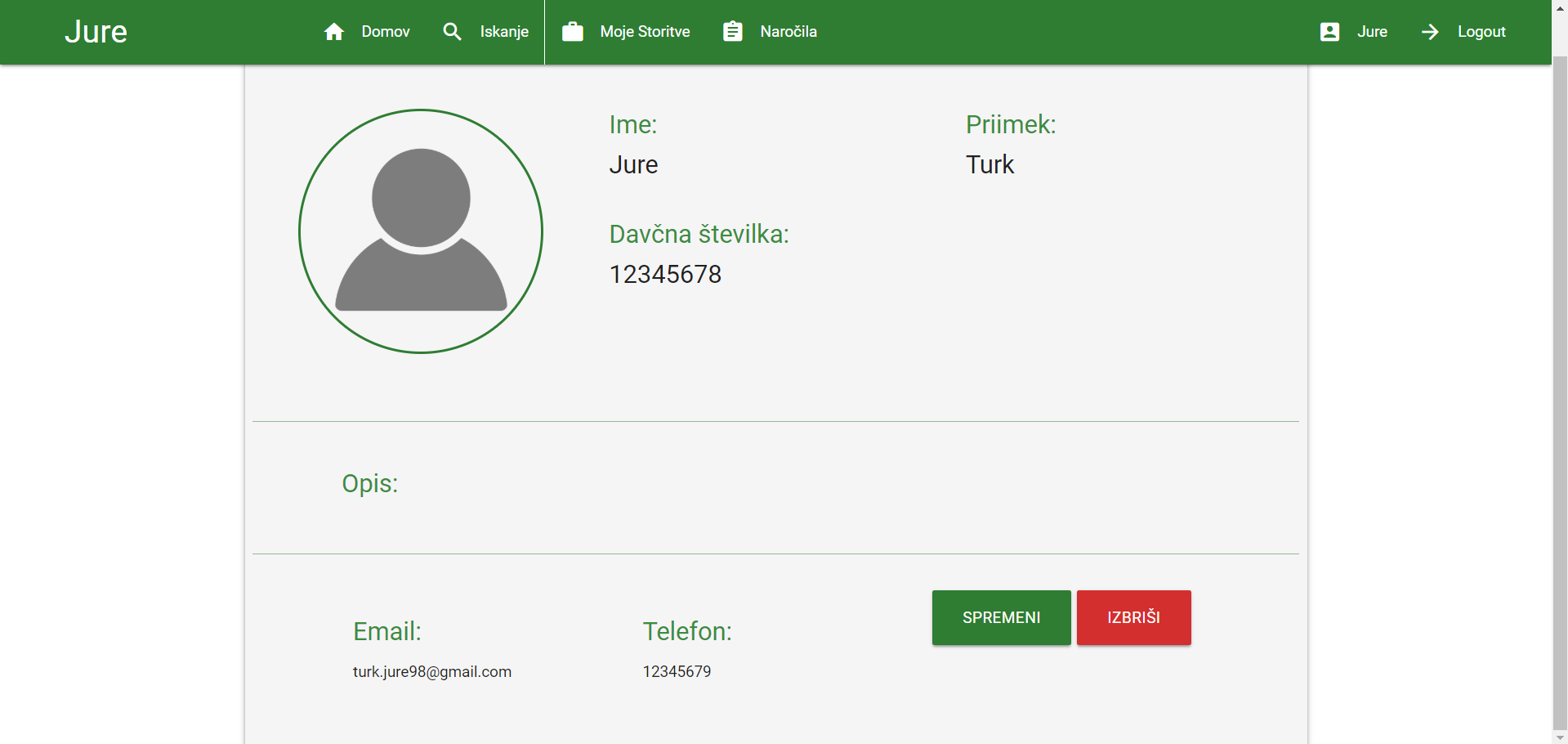
slika: 15 Prikaz strani

# Kontrolerji

Kot je že prikazano v zgornjem primeru, lahko kontrolerji vračajo različnen ''view'' oziroma pogled v obliki spletne strani. V njih pa vključujemo vse preostale funkcionalnosti.

*Primer:*

**Korak 1:** Uporabnik želi izbrisati svoj profil, zato izbere gumb za izbris svojega profila.



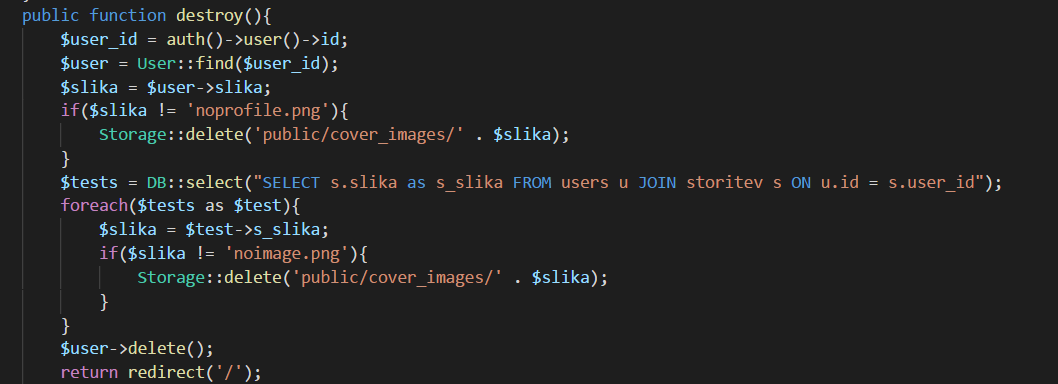
slika: 16 Izbris profila

**Korak 2:** Ustrezna pot, kliče metodo v kontrolerju.



slika: 17 Pot za profil

**Korak3:** Metoda izbriše uporabnika iz podatkovne baze.

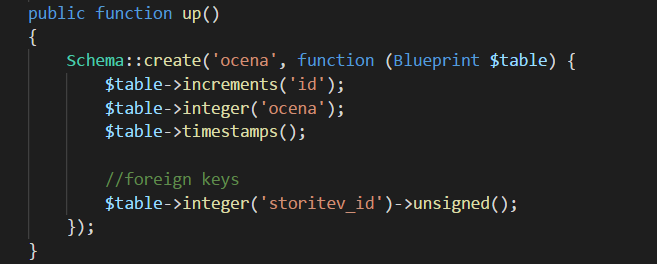
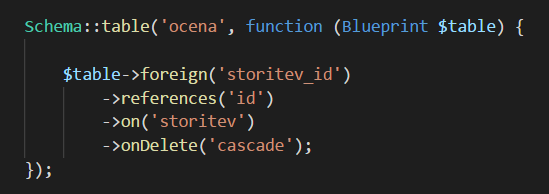


slika: 18 Metoda za izbris uporabnika

# Migracije

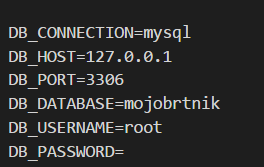
Ker smo imeli težave pri implementaciji podatkovne baze, smo podatkovno bazo ustvarili na novo s pomočjo migracij. Migracije delujejo tako, da v terminalu s pomočjo artisan-a ukažemo aplikaciji da generira podatkovno bazo definirano v migracijah. Definirati je potrebno vsako tabelo, njene atribute, ter tuje ključe.

**Korak 1:** V datoteko z migracijami definiramo vse tabele, ki jih potrebujemo in definiramo tuje ključe, ter druge tabele na katere tuji ključi referencirajo.

slika: 19 Predefiniranje tabele za oceno slika: 20 Tuji ključi za tabelo ocena

**Korak 2:** V datoteki ENV, navedemo vse atribute potrebne za povezavo z podatkovno bazo. Ko to storimo moramo tudi ustvariti podatkovno bazo z imenom navedenim v ENV datoteki. To smo storili



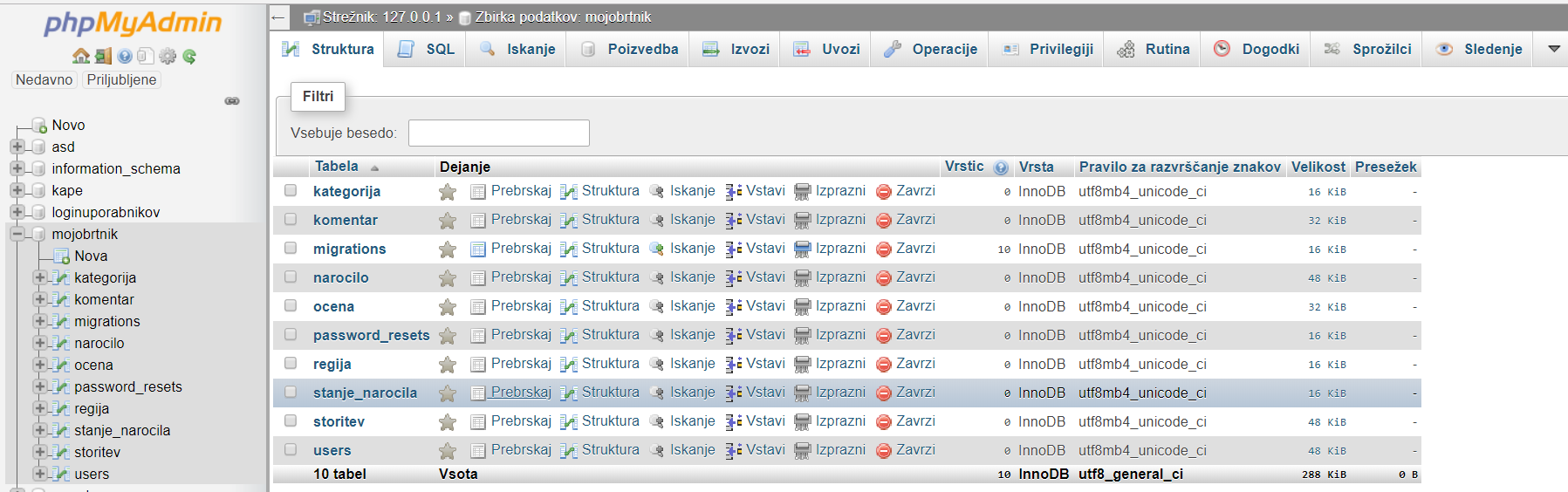
slika: 21 Env datoteka

**Korak 3:** Z ustreznim ukazom v ukaznem pozivu migriramo tabele .



slika: 22 Artisan ukaz

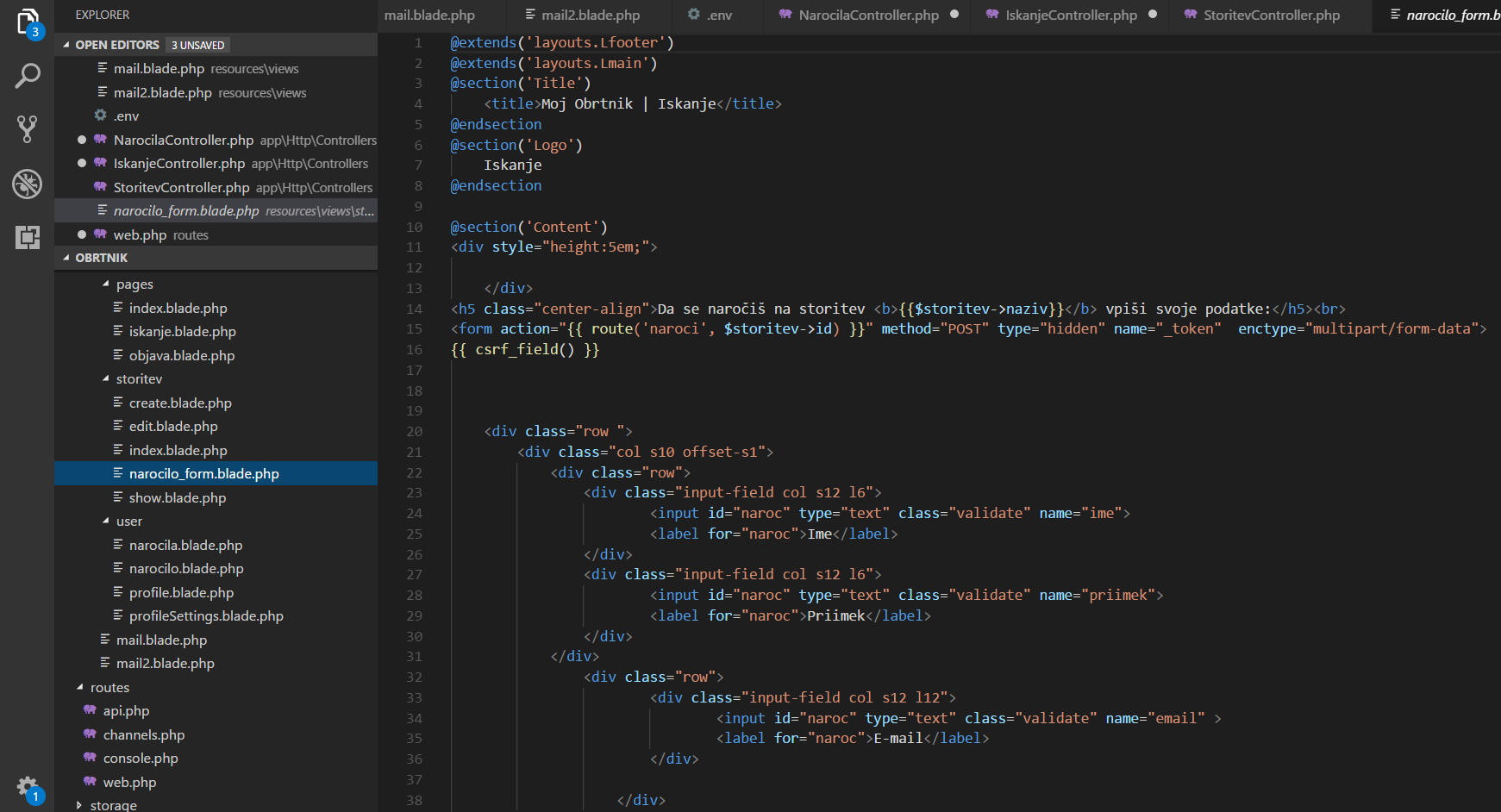
**Korak 4** : S podatkovno bazo lahko upravljamo v orodju phpMyAdmin . Tukaj smo vnašali testne podatke, in nasplošno izvajali ostale interakcije z podatkovno bazo.



slika: 23 phpMyAdmin

# Blade

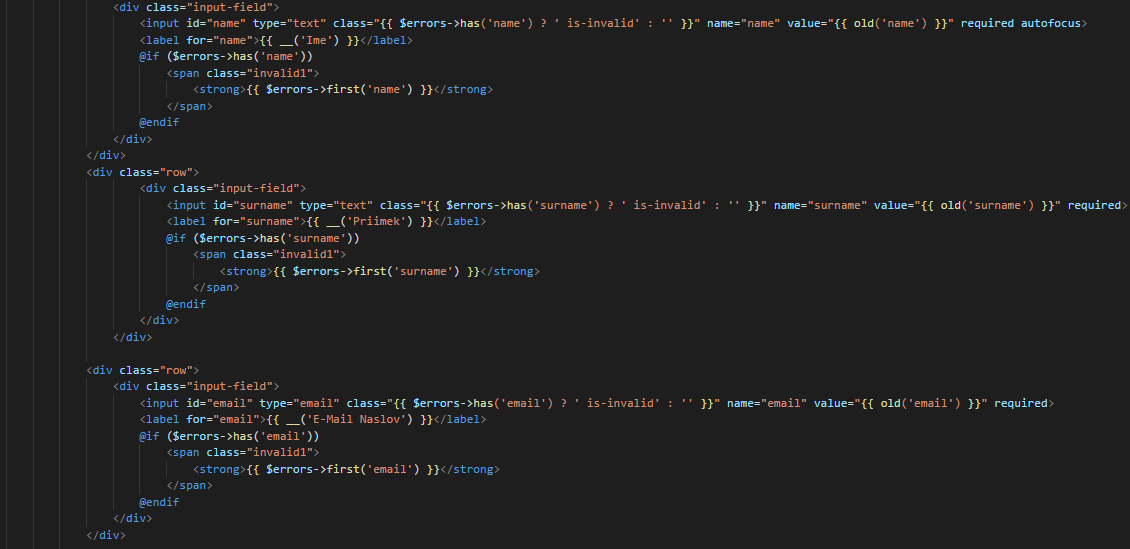
Če se ravnamo po MVC modelu, komponenta blade predstavlja View oz. pogled. Blade je statična komponenta, ustvarjena z HTML in oblikovana z CSS ter Materialize. Znotraj blade datoteke vključimo dinamične komponente



slika: 24 Datoteka blade

# Prijava in registracija

Pri izdelavi funkcionalnosti prijave in registracije smo morali poskrbeti predvsem za varnost, saj se nam ta komponenta zdi ključna iz vidika nadzora dostopa. Ločili smo funkcionalnosti obrtnika od neregistriranega uporabnika. Uvedli smo različne varnostne protokole, ki zagotavljajo da ne pride do zapletov oziroma nedovoljene uporabe funkcionalnosti, ki za uporabnika niso namenjene. Prijavo in registracijo smo izdelali manualno, vendar smo med postopkom izdelave tih dveh funkcionalnosti ugotovili, da ogrodje Laravel omogoča integracijo že vgrajenega sistema prijave in registracije, proti kateremu se naš sistem prijave in registracije ni moral primerjati iz vidika varnosti, in tehnične dovršenosti. Ko smo uvedli nov sistem prijave in registracije, smo ga morali tudi prilagoditi zahtevam naše aplikacije, dodali smo nova vnosna polja za vnos podatkov ki so potrebni pri registraciji za našo aplikacijo in tako prilagodili postopek registracije.



slika: 25 Programska koda za registracijo

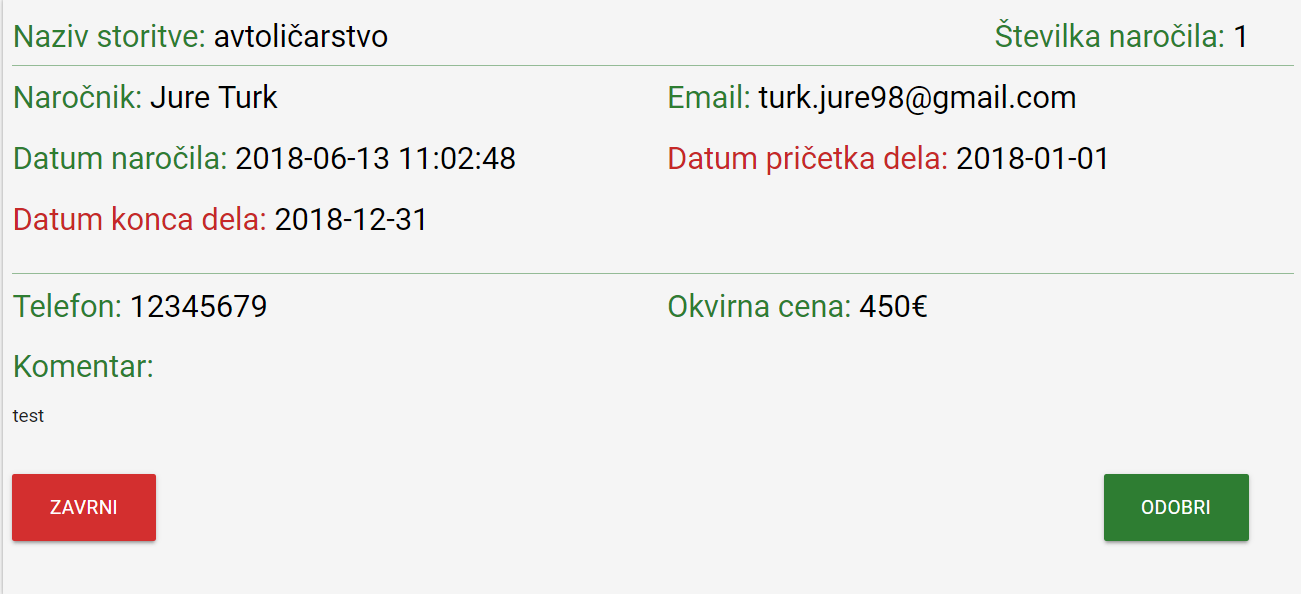
Poskrbeli smo tudi za primerno Hash-anje gesel, saj smo tako zagotovili dodatne varnostne ukrepe glede varnosti aplikacije, več informacij o varnosti je dosegljivo v poglavju  [varnost](#varnost).

# E-pošta

V naš projekt smo uvedli tudi pošiljanje E-pošte, saj je to način komunikacije z uporabnikom. Ko obrtnik odobri oziroma zavrne naročilo, uporabnik ki je naročilo oddal dobi po e-pošti obvestilo o tem, ali je njegovo naročilo bilo sprejeto oziroma zavrnjeno.

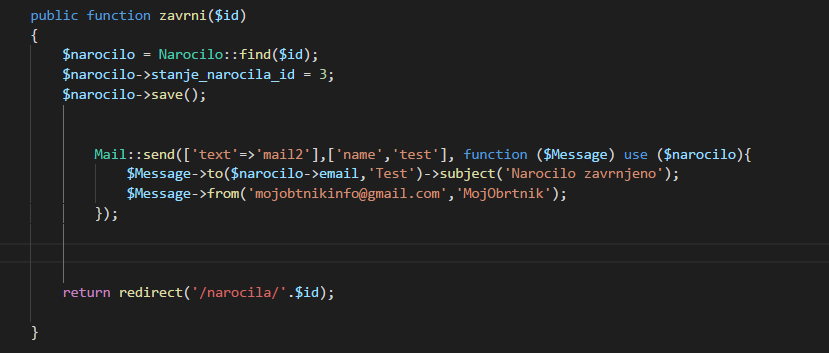
*Primer delovanja e-pošte:*

**Korak 1:** Obrtnik odobri naročilo, ki mu je bilo podano s strani uporabnika.



slika: 26 Naročilo

**Korak 2:** Metoda proži funkcijo ki pošlje e-pošto, ista metoda je bila implementirana za obe funkcionalnosti aplikacije(odobri in zavrni), le podloga v kateri je določeno ali je uporabnikovo naročilo bilo sprejeto ali zavrnjeno se v metodi spremeni da uporabnik dobi ustrezno e-pošto.



slika: 27 metoda za pošiljanje e-pošte

**Korak 3:** Uporabnik prejme ustrezno e-pošto.



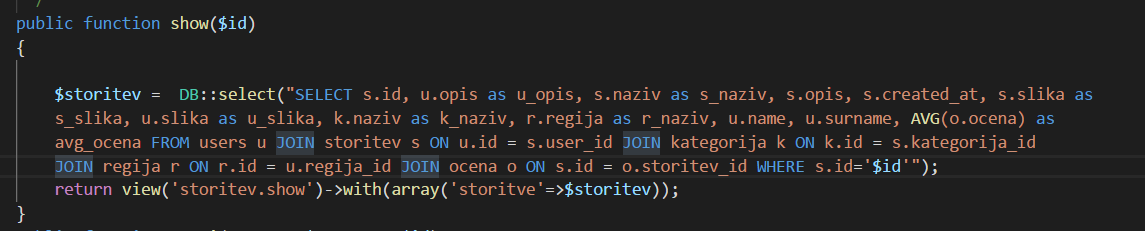
slika: 28 Primer e-pošte

# Naročila, storitve in avtorizacija

Ena izmed glavnih funkcionalnosti naše aplikacije je prikaz storitev in naročil. Storitve se prikažejo navadnemu uporabniku aplikacije. Naročila pa le obrtniku kateremu pripada določeno naročilo, tukaj je pomemben element naše aplikacije tudi verifikacijo.

**Storitve**

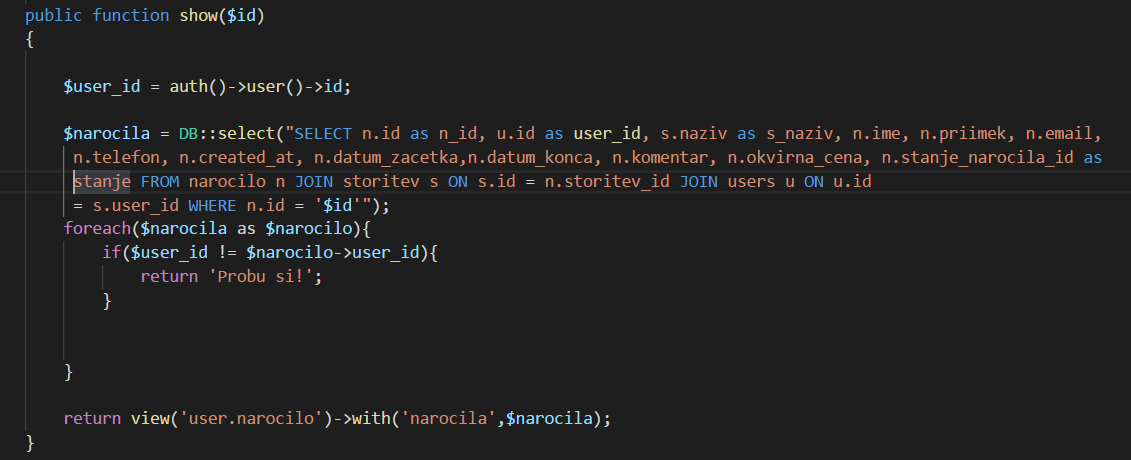
Metoda ki iz podatkovne baze prikaže vse storitve uporabniku. Da



slika: 29 Metoda za prikaz storitev

**Naročila**

Spodaj navedena metoda prikaže naročila, ki pripadajo določenemu uporabniku.



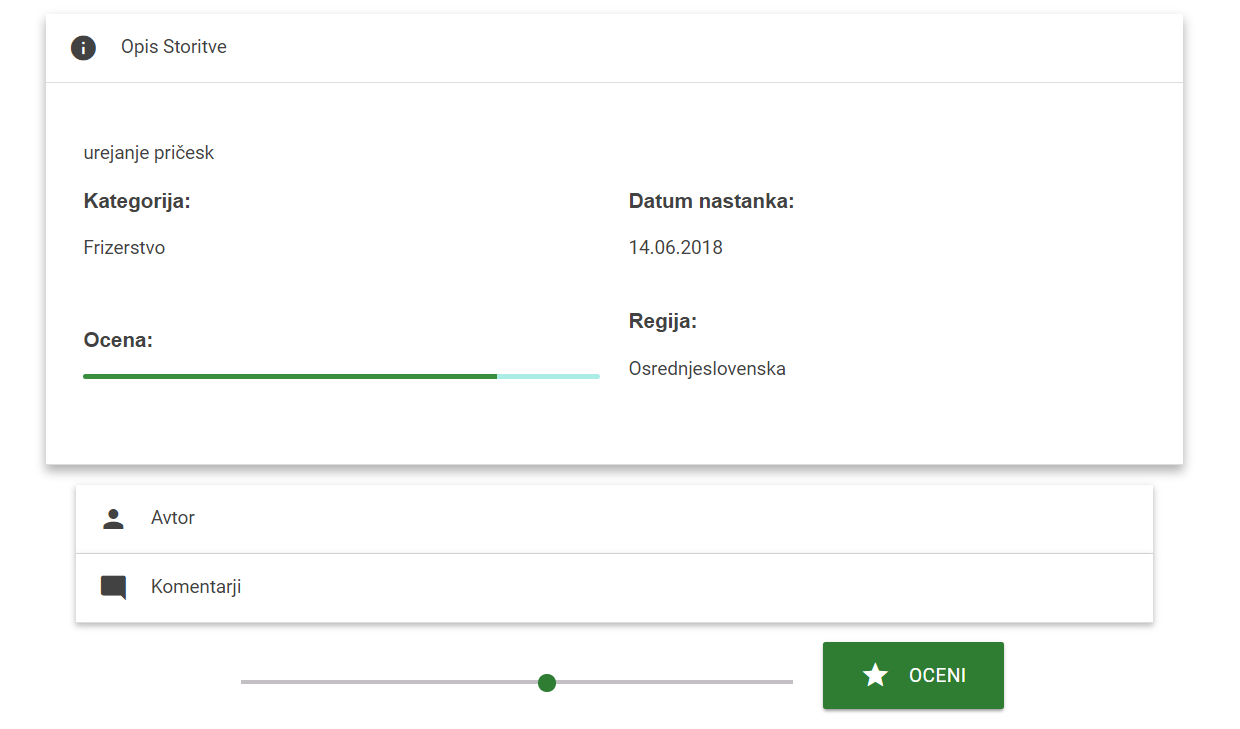
slika: 30 Metoda za prikaz naročil

**Avtorizacija**

Pri metodi za naročila z prvo vrstico kode v metodi ( $user\_id = auth()->user()->id; ) dobimo id uporabnika, ki se veže na njegova naročila, tako ne more drug uporabnik dostopati do naročil, ki niso njegova.

# Ocenjevanje in komentiranje

Kot obvezno funkcionalnost, smo morali omogočiti komentiranje storitev posameznega obrtnika, da dobijo koristniki storitev informacije o kvaliteti storitev, ki jo posamezni obrtnik ponuja. Kot varnostni protokol smo uvedli funkcijo, da je obrtniku onemogočeno ocenjevanje in komentiranje svojih storitev.



slika: 31 Ocenjevanje in komentiranje

1. Varnost

# Kodiranje gesel

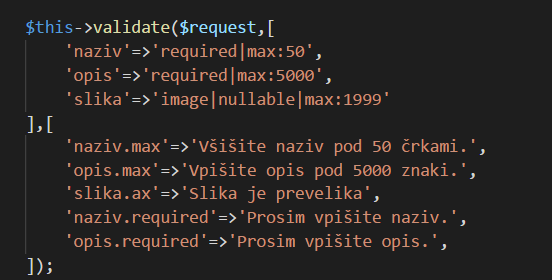
Pri shranjevanju gesel smo poskrbeli za njihovo enkripcijio, tako so gesla heshana z funkcijo bcrypt, ki je veliko bolj varna kot enkripcija z md5. Hashing z funkcijo bcrypt je tudi bolj odporen na ''brute force '' napade saj je funkcija bcrypt adaptivna in se tako lahko poveča njena iteracija zaradi katere je veliko bolj učinkovita.



slika: 32 Geslo hesh-ano z bcrypt

# Verifikacija

Za verifikacijo je poskrbljeno z pogoji pri vnosnih poljih, če so polja neizpolnjena ali nepopolno izpolnjena, aplikacija ne dovoli nadaljevanja dokler niso izpolnjeni vsi pogoji. Ta metoda je implementirana pri vseh vnosnih poljih.



slika: 33 Pogoji

Nekaj glavnih funkcij ki smo jih uporabili pri verifikaciji:

* Omejitev dolžine nizov
* E-poštni naslov mora bit v pravilnem formatu
* Pri registraciji se more bit ponovitev gesla ujemat z prej navedenim geslom
* e-poštni naslov in geslo se pri prijavi ujemata z registriranim, v nasprotnem primeru prijava ni mogoča
* Slika in opis sta lahko neizpolnjena, saj ju lahko obrtnik doda kasneje z urejanjem profila

# Nadzor dostopa

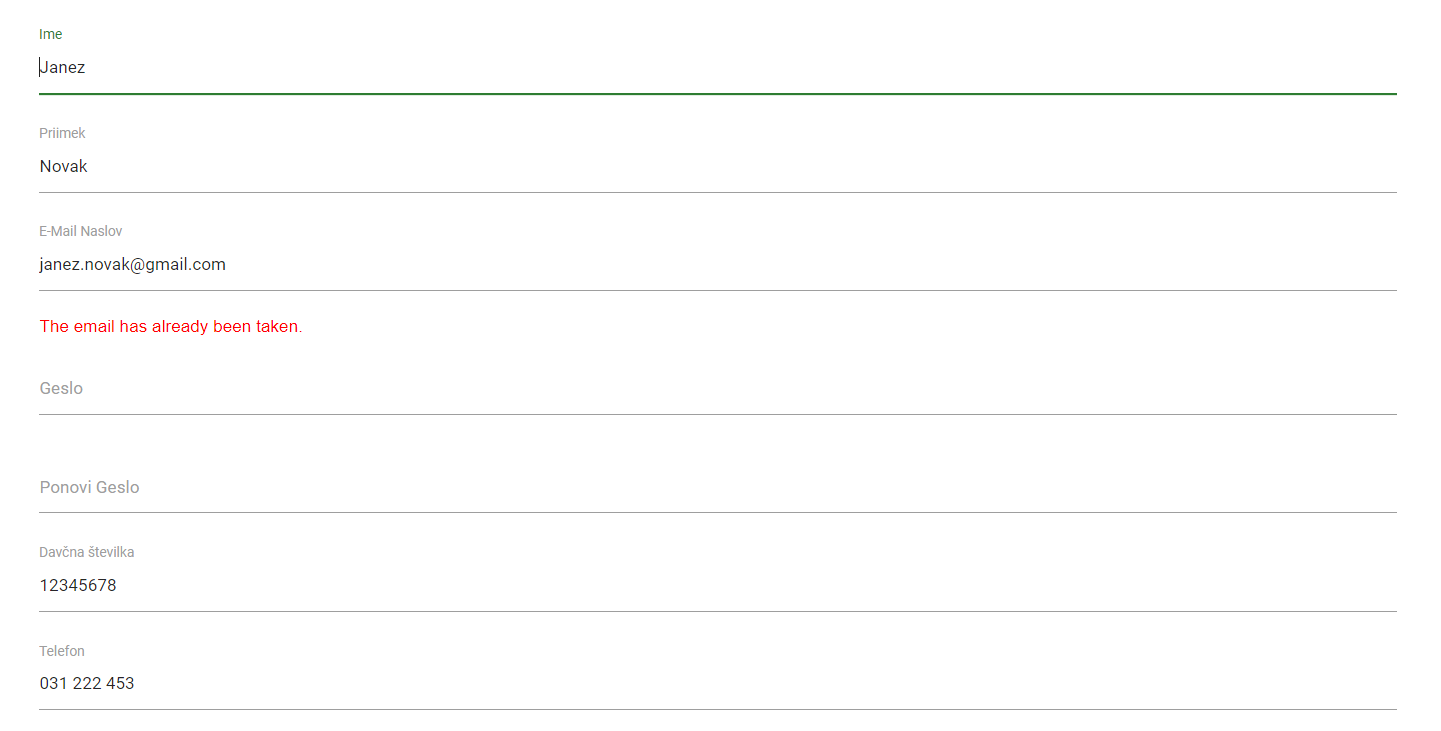
Pri nadzoru dostopa smo z autorizacijo omogočili naslednje funkcije

* Obrtnik ne more pri sebi naročati storitev
* Obrtnik ne more sprejemati tujih naročil
* Neprijavljeni uporabnik ne more dostopati do funkcionalnosti ki so dostopne le s prijavo

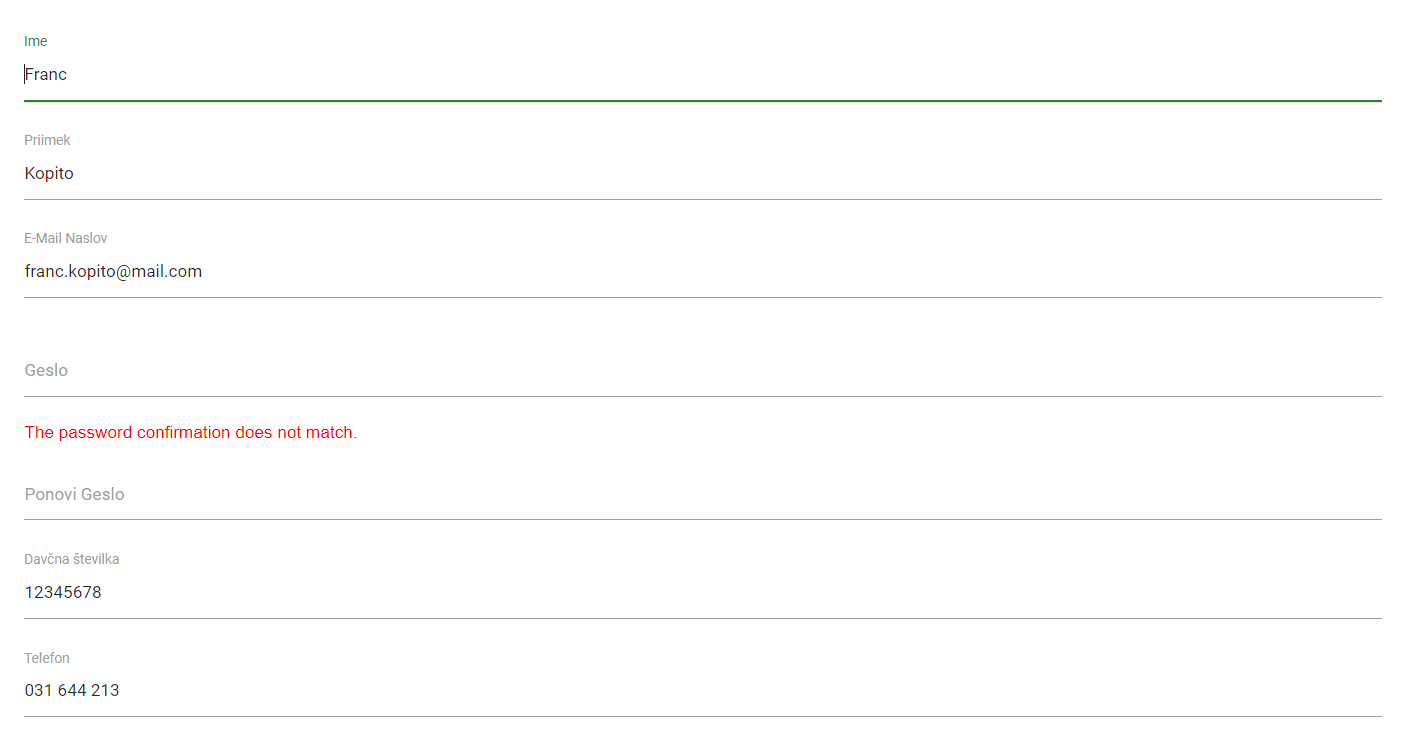
1. Testiranje

# 7.1. Registracija in vpis obrtnika

Pri registraciji uporabnika se preveri, če so vsa obvezna polja izpolnjena, če se vneseni gesli ujemata in če je e-mail naslov še prost. V primeru, da kateri od zgornjih kriterijev ni izpolnjen, se registracija zavrne in izpiše ustrezno opozorilo.

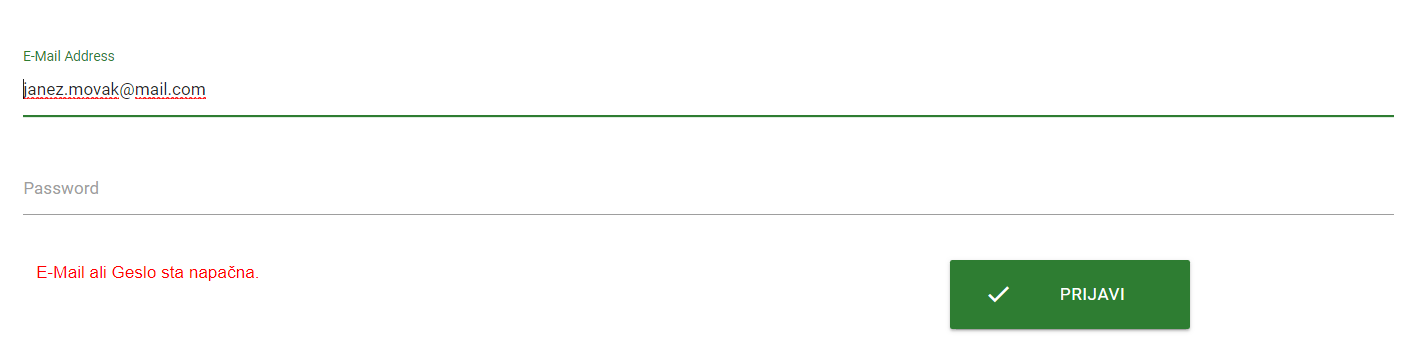


slika: 34 E-poštni naslov že v uporabi



slika: 35 Gesli se ne ujemata

Pri vpisu se preverja, če sta polji za e-pošto in geslo obe izpolnjeni in če se podatki ujemajo z tistimi v podatkovni bazi spletne strani.

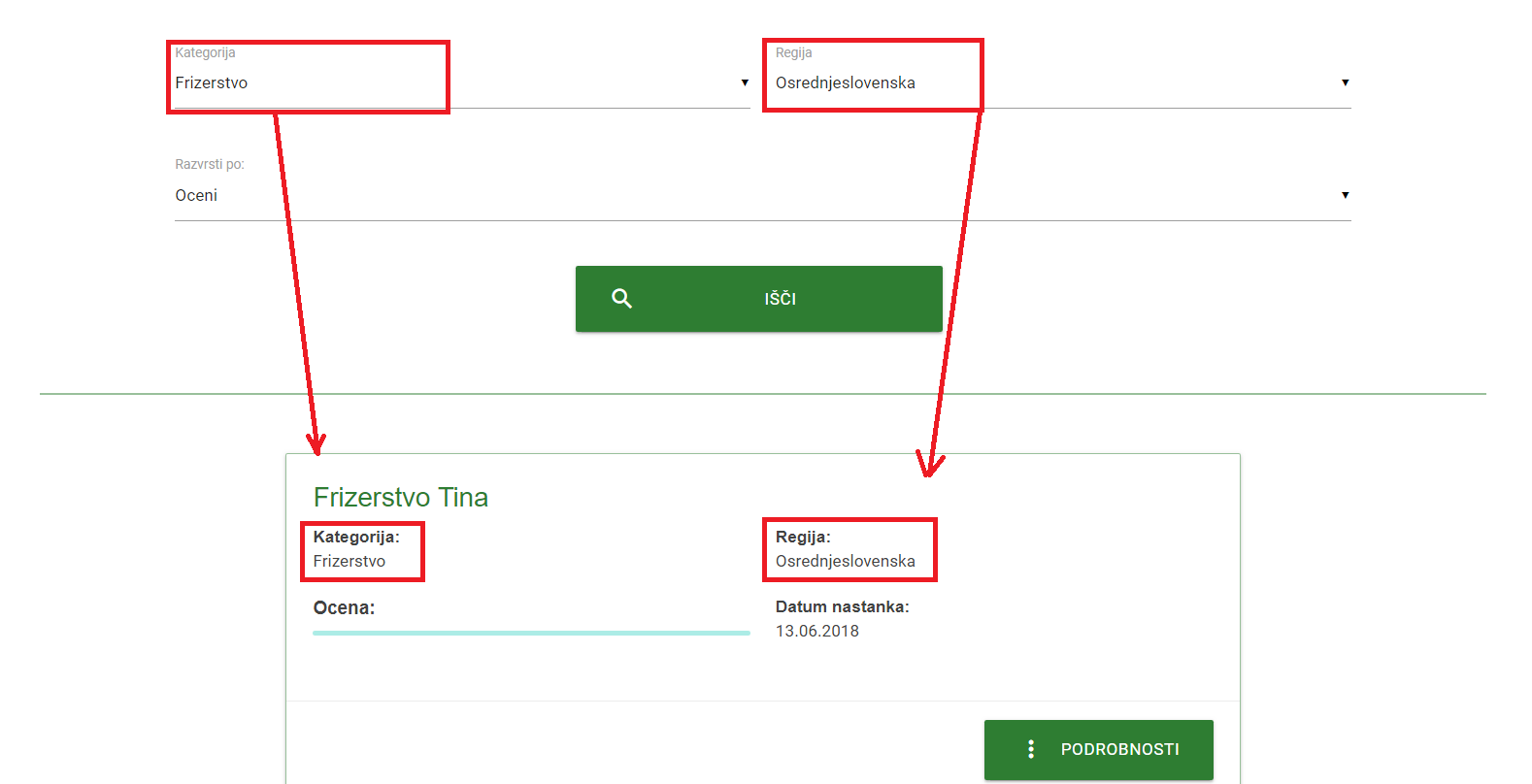


slika: 36 E-mail ali geslo sta napačna

# 7.2. Iskanje po storitvah

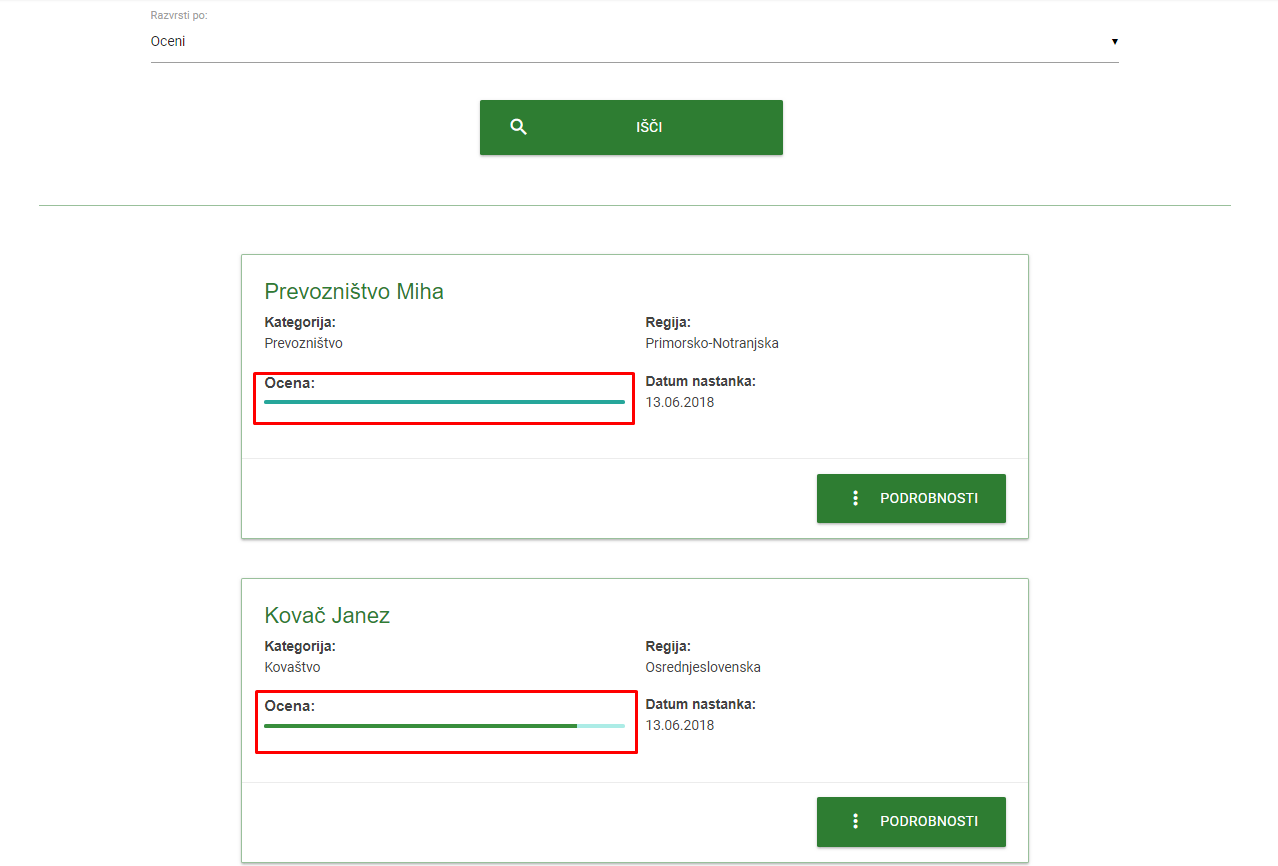
Pri iskanju smo preverili, če delujejo vsi razvrstitveni kriteriji. Na voljo je:

* iskanje glede na kategorijo, pri katerem se prikažejo samo storitve, katere ustrezajo izbrani kategoriji
* iskanje glede na regijo, pri katerem se prikažejo samo storitve, katere ustrezajo izbrani regiji



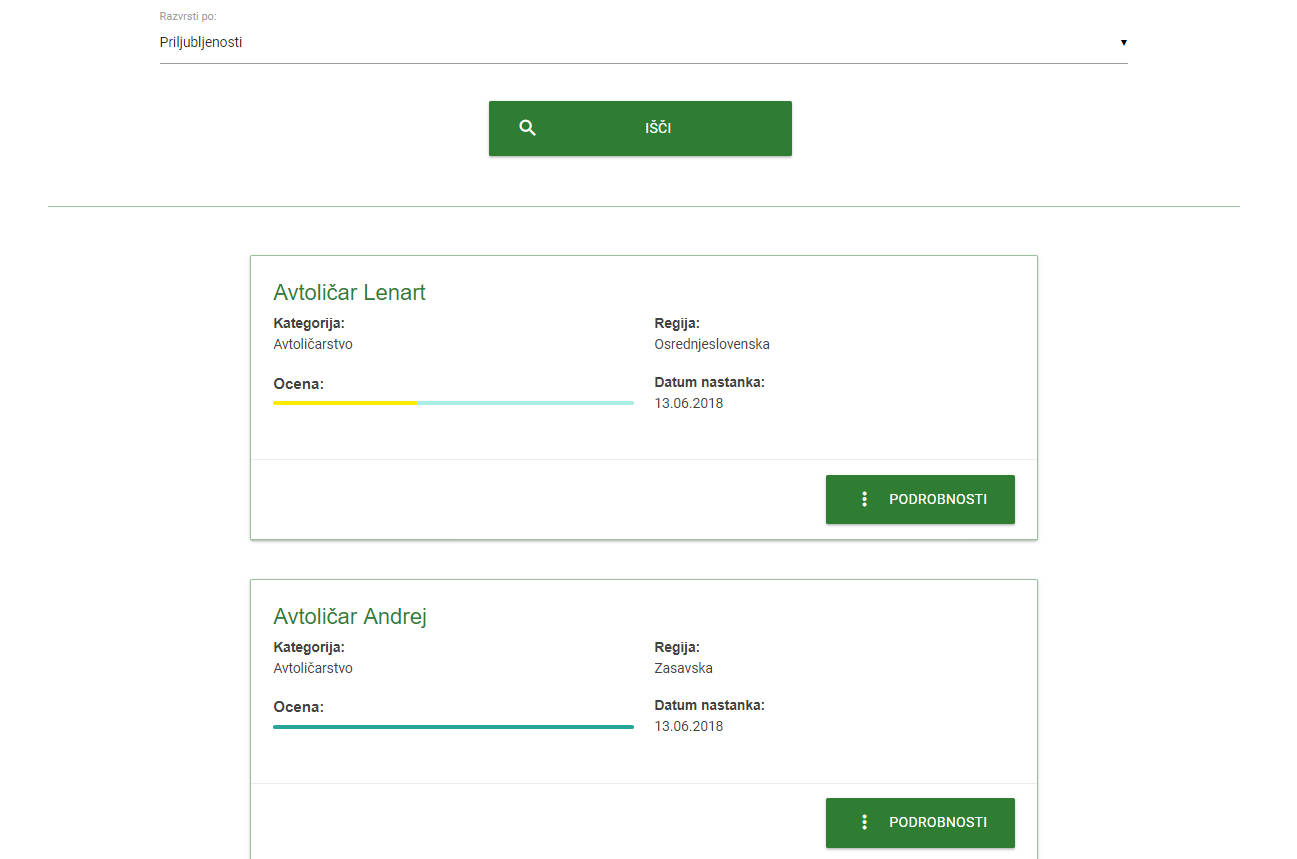
slika: 37 Razvrščanje po izbrani kategoriji in regiji

* razvrščanje glede na oceno, pri katerem se storitve razvrstijo tako, da je storitev z najvišjo oceno pri vrhu in storitev z najmanjšo oceno pri dnu



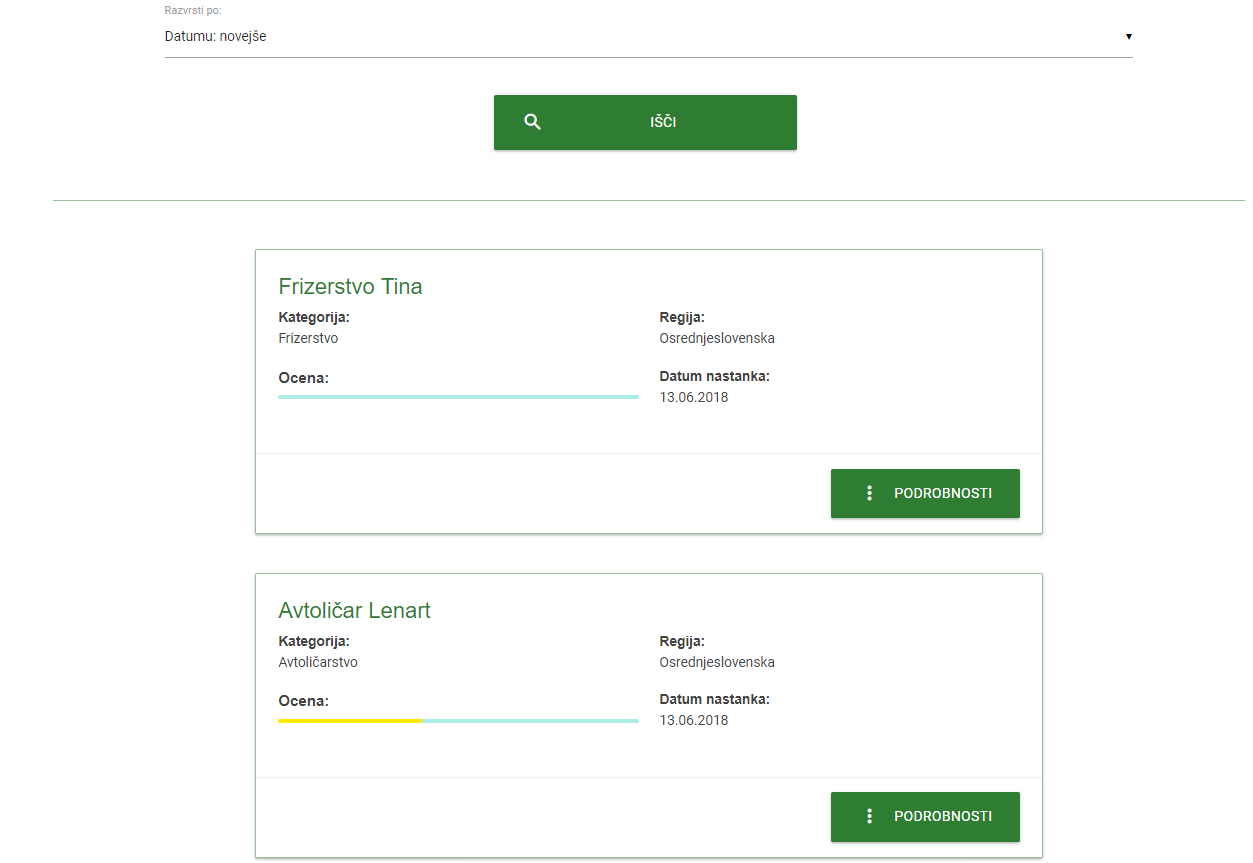
slika: 38 Razvrščanje po najvišji oceni

* razvrščanje glede na priljubljenost, pri katerem se storitve razvrstijo tako, da je storitev z najvišjim številom vseh naročil pri vrhu in storitev z najmanjšim številom naročil pri dnu



slika: 39 Razvrščanje glede na priljubljenost (število vseh skupnih naročil)

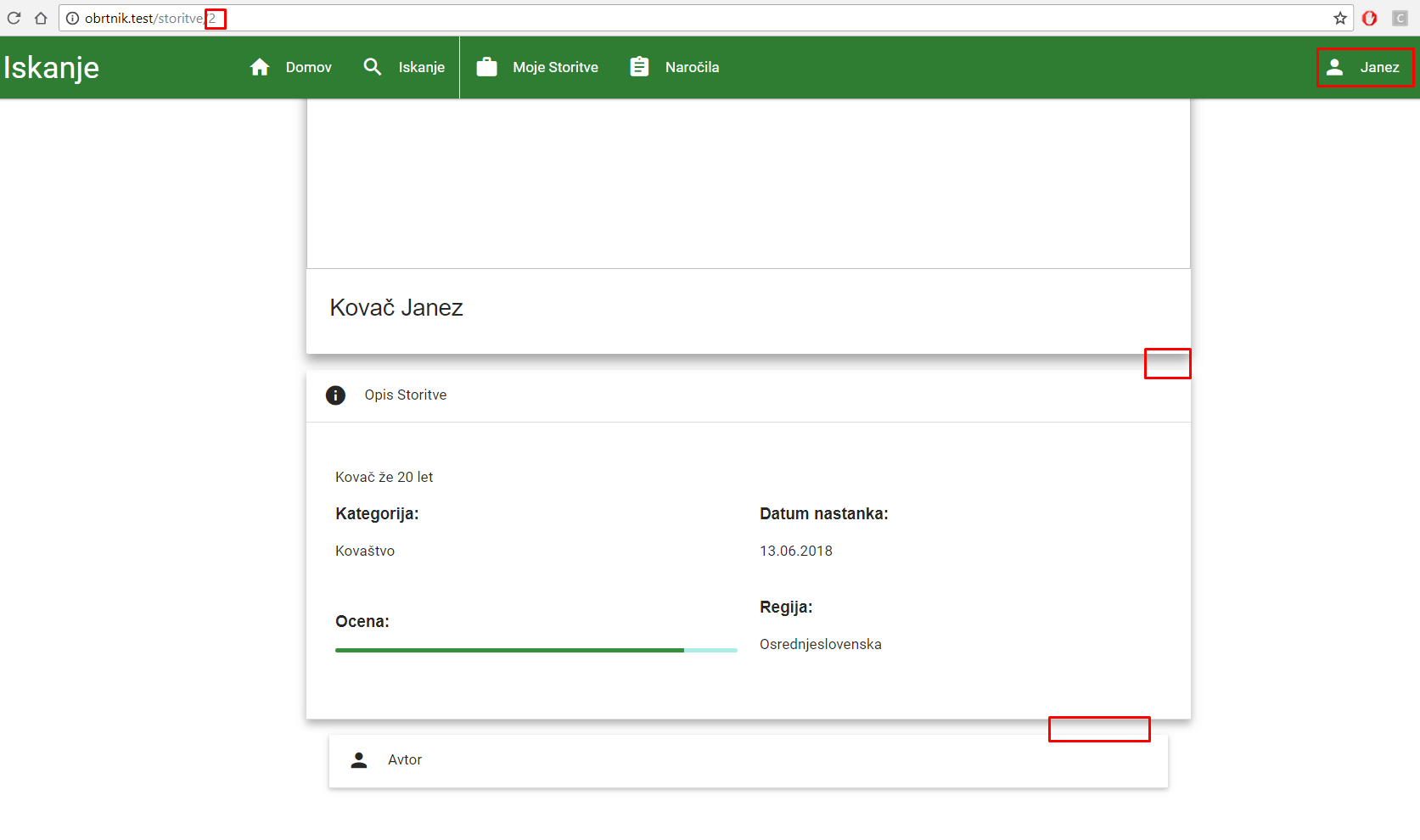
* razvrščanje glede na datum (najstarejši in najnovejši), pri katerem se storitve razvrstijo po datumu objave storitve



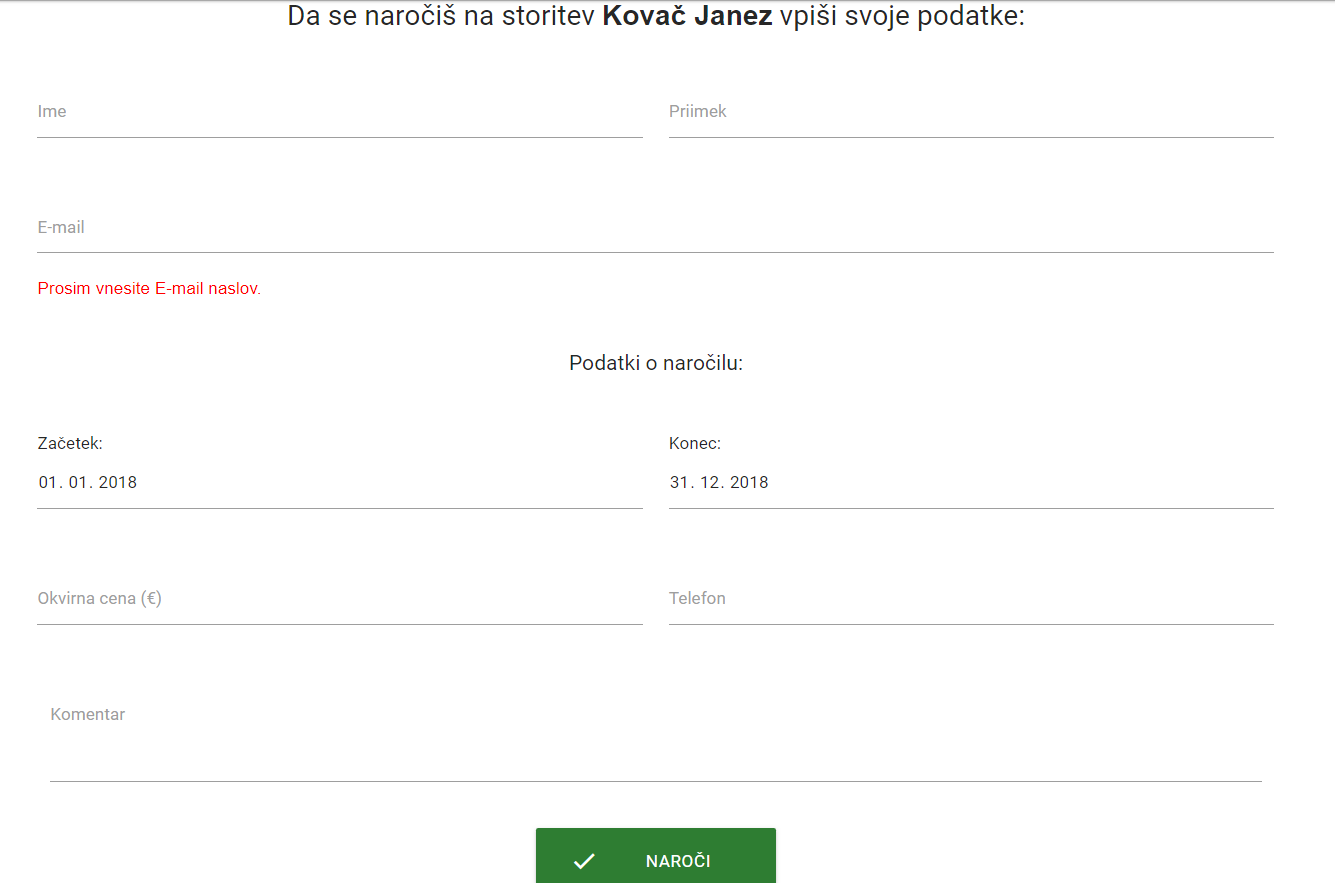
slika: 40 Razvrščanje po najnovejšem/najstarejšem

# 7.3. Naročanje, ocenjevanje in komentiranje storitev

Naročanje, ocenjevanje in komentiranje storitev je omogočen vsem obiskovalcem in registriranim obrtnikom, razen lastniku storitve, saj bi bilo naročanje na lastno obrt nesmiselno. Pri naročanju se preverja tudi izpolnjenost vseh obveznih polj.

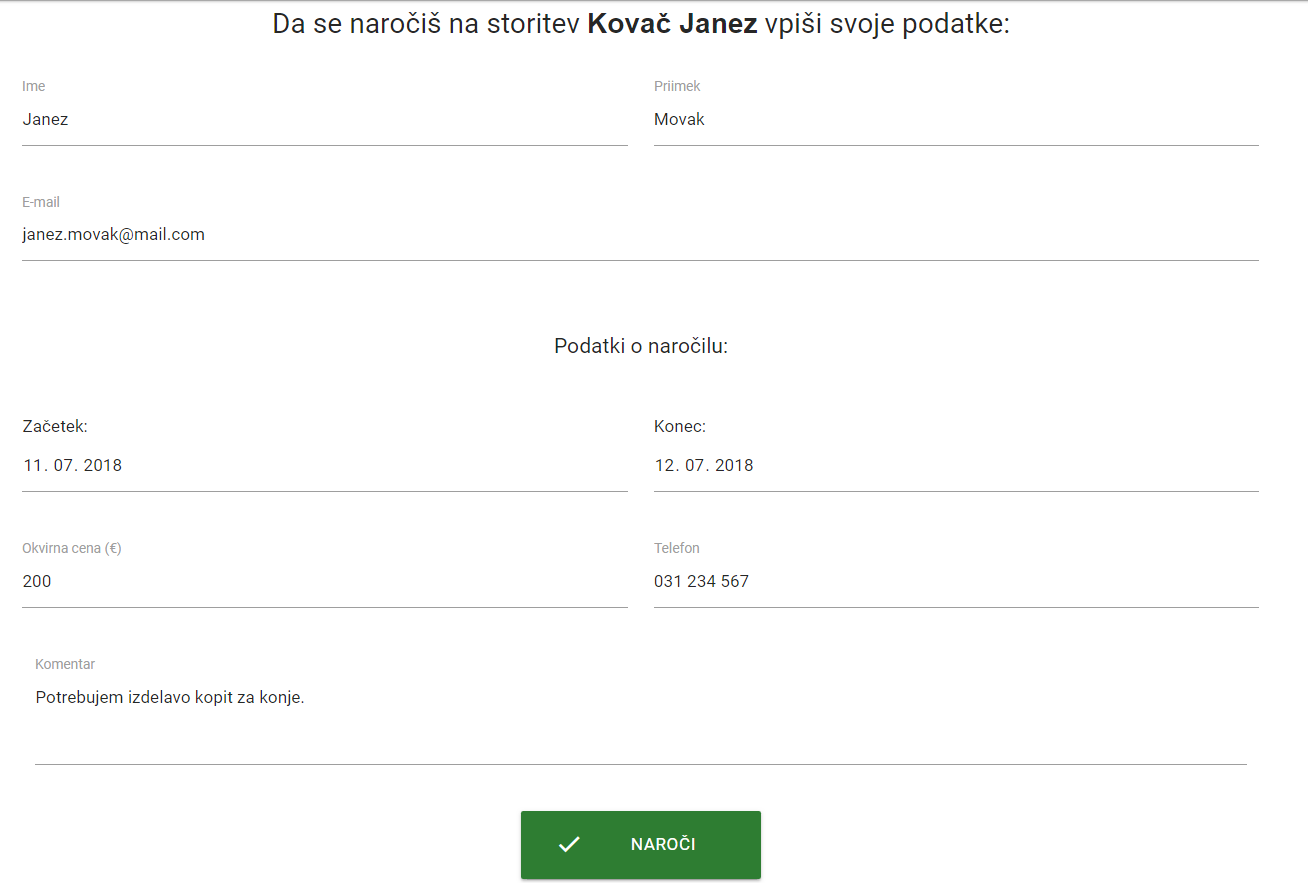


slika: 41 Naročanje, ocenjevanje in komentiranje onemogočeno lastniku obrti (ni gumbov)

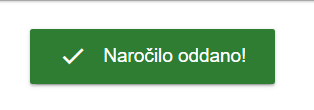


slika: 42 Preverjanje izpolnjenost obveznih vnosnih polj

Po uspešnem naročanju se izpiše potrdilo, da je bilo naročilo oddano lastniku storitve.

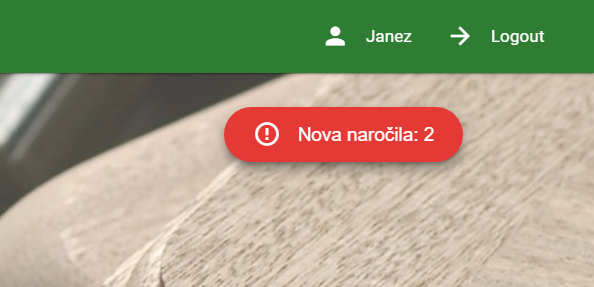


slika: 43 Ustrezno izpolnjena vnosna polja

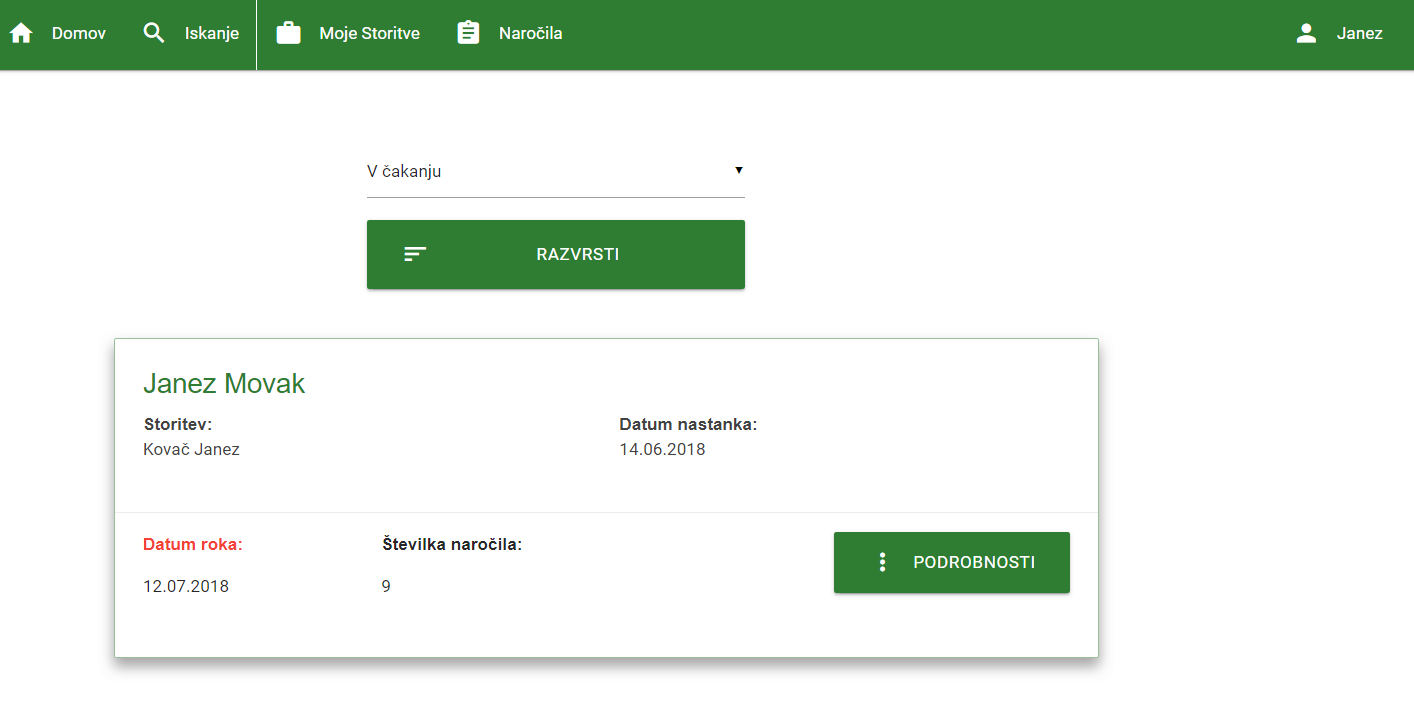


slika: 44 Sporočilo o uspešno oddanem naročilu

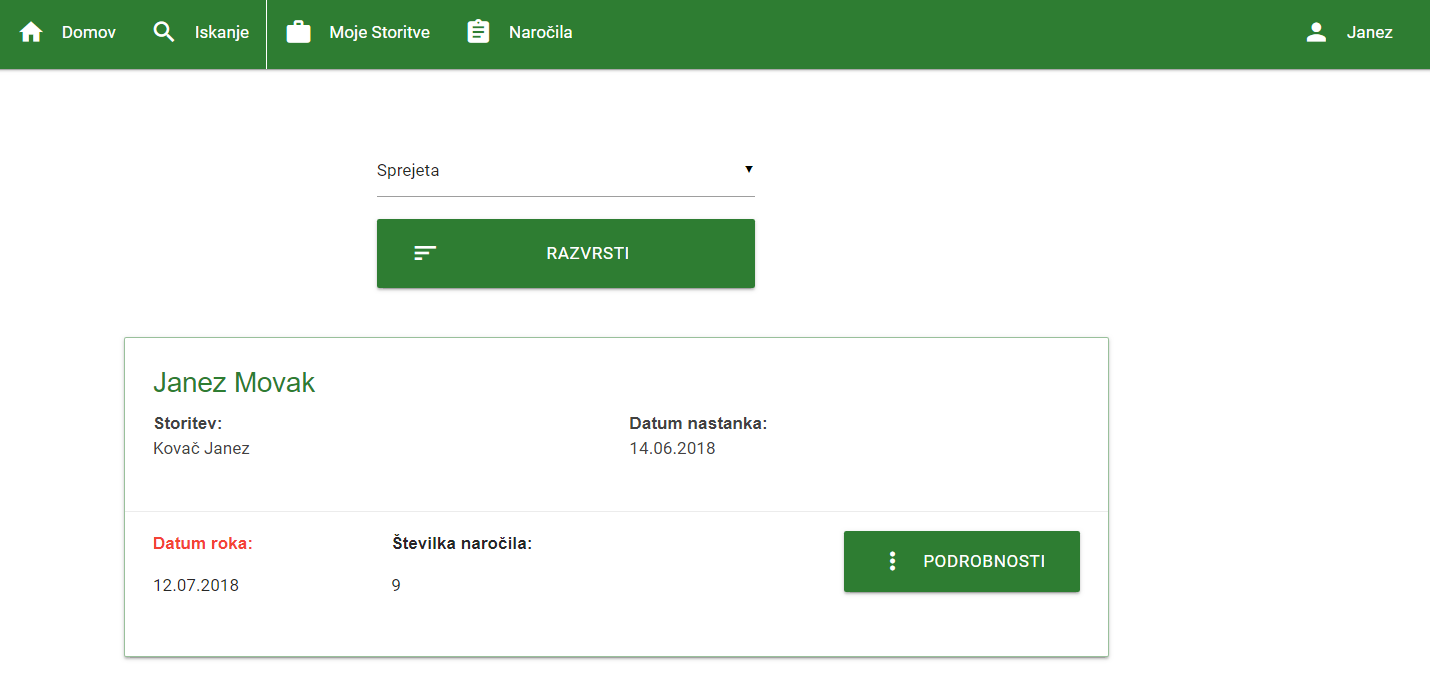
Takoj ko obrtnik prejme novo naročilo, se mu naslednjič ko obišče spletno stran prikaže sporočilo o novih naročilih.



slika: 45 Sporočilo o novih naročilih

Obrtnik v zavihek »Naročila« prejme vsa naročila, katera lahko potrdi in zavrne. V tem zavihku lahko prverja vso zgodovino čakajočih, zavrnjenih in potrjenih naročil. 

slika: 46 Prejeta naročila



slika: 47 Razvrščanje naročil

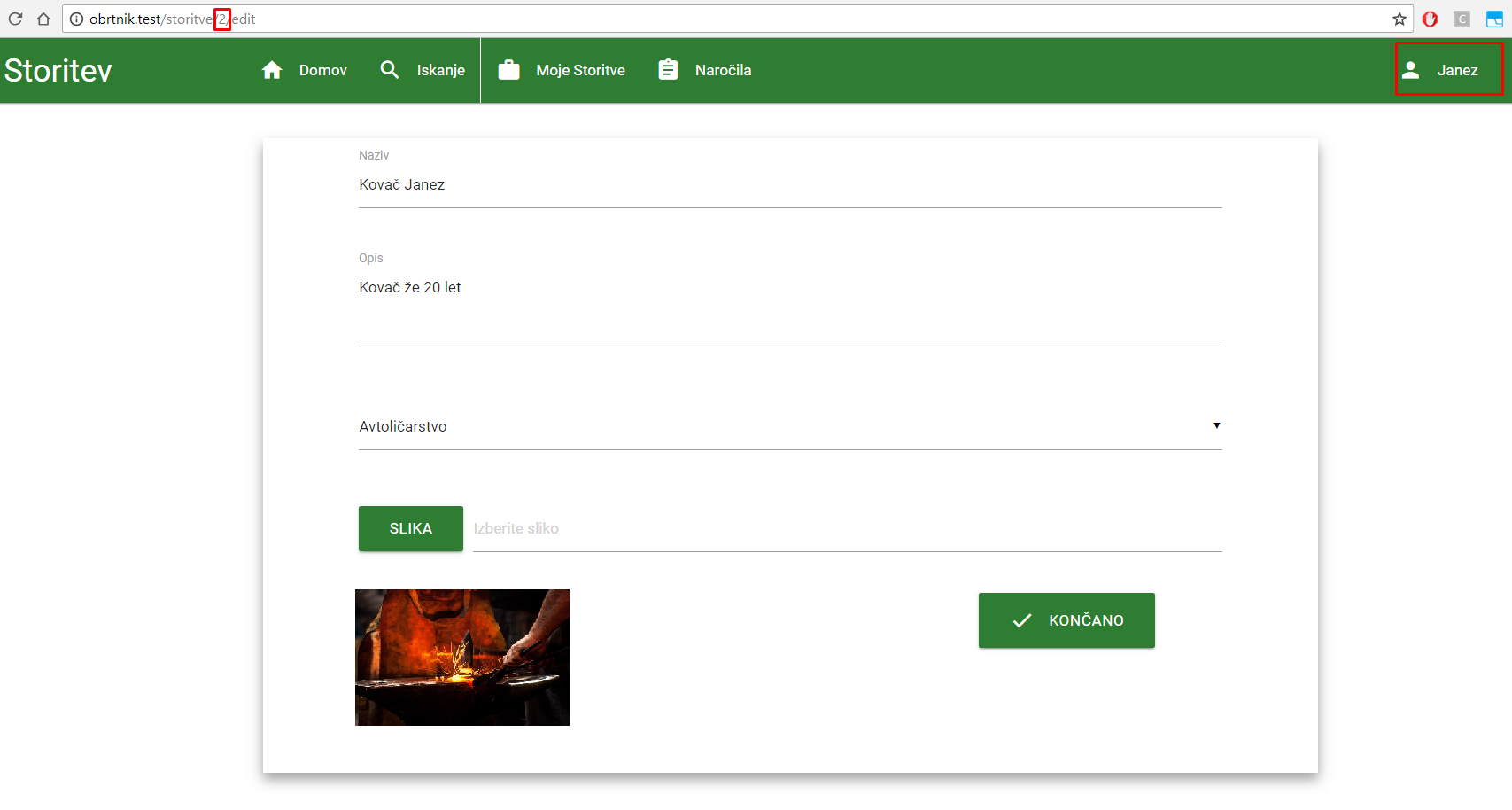
# 7.4. Ustvarjanje in urejanje storitev

Registriran obrtnik takoj po registraciji nima nobenih storitev, zato jo mora ustvariti. Na zavihku »Moje storitve« so mu na voljo orodja za ustvarjanje storitev.

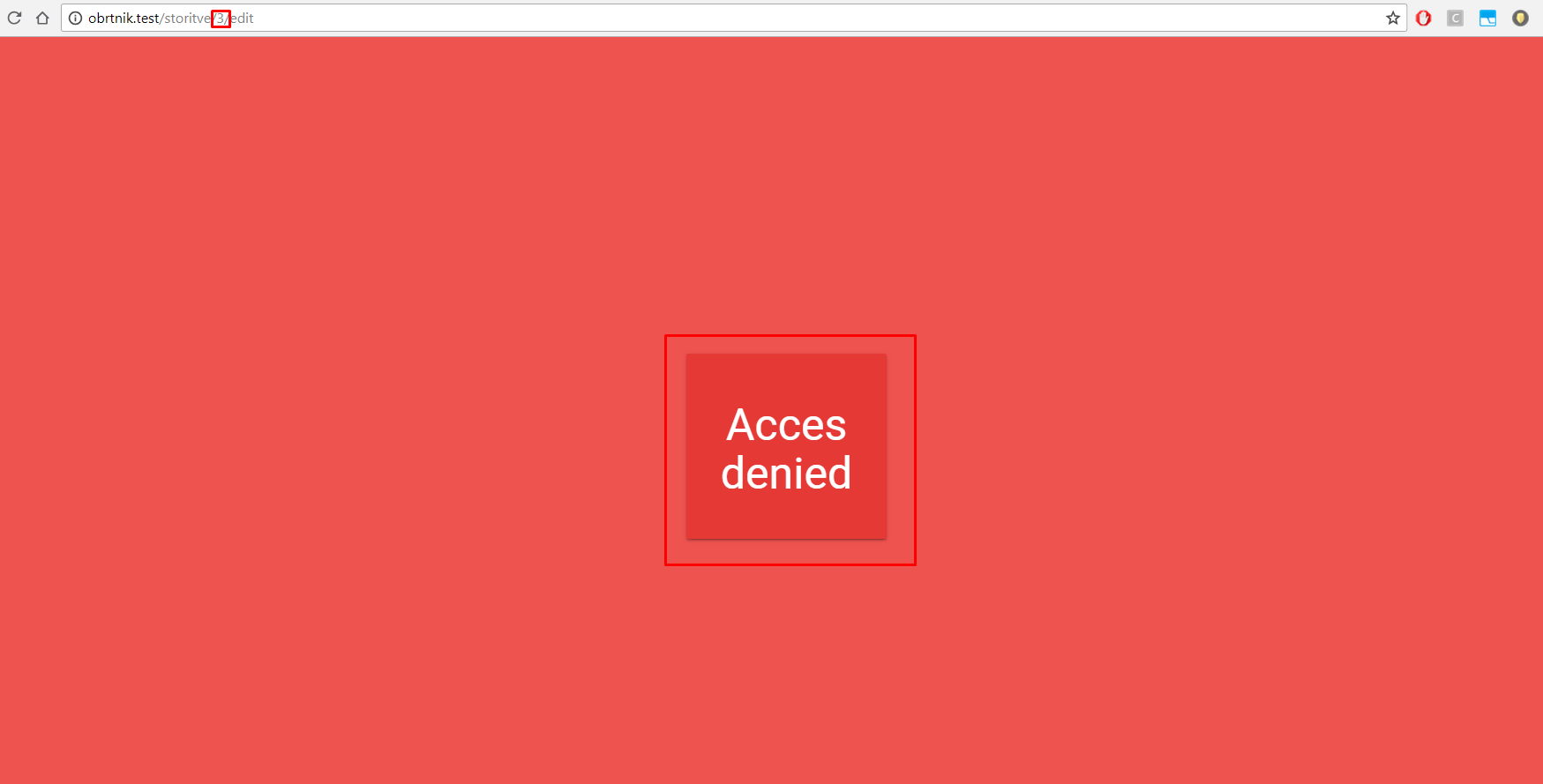


slika: 48 Ustvarjanje storitve

Ko ima obrtnik že objavljeno storitev, to lahko ureja s pomočjo orodja za urejanje storitve. Obrtnik, ki hoče urejati storitev, mora biti tudi lastnik le-te, saj mu je dostop do urejanja v nasprotnem primeru zavrnjen.



slika: 49 Urejanje storitve



slika: 50 Zavrnjen dostop do urejanja, saj storitev ni v lasti obrtnika

1. Zaključek

Sam projekt nam je predstavljal velik izziv, saj smo se odločili uporabiti nove tehnologije, ki jih do sedaj še nismo uporabljali. Zadali smo si relativno zahtevno nalogo, saj smo si izbrali dve novi orodji katerih še nobeden izmed članov skupine ni uporabljal. Namesto ogrodja spring, smo uporabili L Laravel, kot ogrodje ki skrbi za odzivnost in izgled spletne strani pa materilaize. Kljub izzivom pa smo dosegli izpolnitev vseh funkcionalnosti, ki jih navodila naloge zahtevajo. Velik del časa smo namenili testiranju naše aplikacije, saj smo želeli da je naša aplikacija tehnično dovršena in izpopolnjena. Po temeljitem testiranju smo odpravili vse napake, ki so nastale med razvojem aplikacije. Naša aplikacija tako dosega tehnično funkcionalnost, ima delujočo podatkovno bazo ter je vizualno privlačna za uporabnika. Dosegli smo vse cilje ki smo si jih na začetku zastavili, spoznali smo nova orodja ter osvojili nova znanja glede skupinskega dela. Ne glede na zaplete, ki so nas neprestano spremljali med razvojem, je projekt za nas bil prijetna in uporabna izkušnja saj smo dosegli prenekatera nova znanja.