



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

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 |
| 1.1 | Besedilo naloge | 4 |
| 1.2 | Specifikacije in zahteve | 4 |
| 1.3 | Izbrano ogrodje in jeziki | 4 |
| 2. | Uporabljena orodja in rešitev problema | 5 |
| 2.1 | Laravel | 5 |
| 2.2 | Artisan | 5 |
| 2.3 | Laracasts | 6 |
| 3. | Podatkovna baza | 7 |
| 3.1 | Načrtovanje podatkovne baze | 7 |
| 3.2 | Ustvarjanje podatkovne baze | 9 |
| 4. | Uporabniški vmesnik | 10 |
| 4.1 | Materialize | 10 |
| 4.2 | Uporabniška izkušnja in vizualno načrtovanje | 10 |
| 5. | Opis komponent | 11 |
| 5.1 | Poti | 11 |
| 5.2 | Kontrolerji | 13 |
| 5.3 | Migracije | 14 |
| 5.4 | Blade | 15 |
| 5.5 | Prijava in registracija | 16 |
| 5.6 | E-pošta | 17 |
| 5.7 | Naročila, storitve in avtorizacija | 18 |
| 5.8 | Ocenjevanje in komentiranje | 19 |
| 6. | Varnost | 20 |
| 6.1 | Kodiranje gesel | 20 |
| 6.2 | Verifikacija | 20 |
| 6.3 | Nadzor dostopa | 21 |
| 7. | Testiranje | 22 |
| 7.1 | Registracija in vpis obrtnika | 22 |
| 7.2 | Iskanje po storitvah | 23 |
| 7.3 | Naročanje, ocenjevanje in komentiranje storitev | 25 |
| 7.4 | Ustvarjanje in urejanje storitev | 28 |
| 8. | Zaključek | 30 |

Kazalo slik :

| | |
|--|----|
| slika: 1 Primer dokumentacije | 5 |
| slika: 2 Laracasts..... | 6 |
| slika: 3 Osnutek | 7 |
| slika: 4 Prvoten ER model podatkovne baze | 8 |
| slika: 5 Končen ER model naše podatkovne baze | 8 |
| slika: 6 Ustvarjanje tabel v orodju MySQL | 9 |
| slika: 7 Ustvarjanje tabel z migracijami | 9 |
| slika: 8 Skripta za vstavljanje podatkov v podatkovno bazo | 9 |
| slika: 9 Primer oblikovanja z materialize | 10 |
| slika: 10 Domača stran | 10 |
| slika: 11 Poti | 11 |
| slika: 12 Navigacijska vrstica | 11 |
| slika: 13 Pot | 11 |
| slika: 14 Metoda v kontrolerju | 12 |
| slika: 15 Prikaz strani..... | 12 |
| slika: 16 Izbris profila..... | 13 |
| slika: 17 Pot za profil | 13 |
| slika: 18 Metoda za izbris uporabnika | 13 |
| slika: 19 Predefiniranje tabele za oceno | 14 |
| slika: 20 Tuji ključi za tabelo ocena | 14 |
| slika: 21 Env datoteka | 14 |
| slika: 22 Artisan ukaz..... | 14 |
| slika: 23 phpMyAdmin | 15 |
| slika: 24 Datoteka blade | 15 |
| slika: 25 Programska koda za registracijo | 16 |
| slika: 26 Naročilo | 17 |
| slika: 27 metoda za pošiljanje e-pošte | 17 |
| slika: 28 Primer e-pošte | 17 |
| slika: 29 Metoda za prikaz storitev..... | 18 |
| slika: 30 Metoda za prikaz naročil | 18 |
| slika: 31 Ocenjevanje in komentiranje | 19 |
| slika: 32 Geslo hesh-ano z bcrypt..... | 20 |
| slika: 33 Pogoji..... | 20 |
| slika: 34 E-poštni naslov že v uporabi | 22 |
| slika: 35 Gesli se ne ujemata | 22 |
| slika: 36 E-mail ali geslo sta napačna | 23 |
| slika: 37 Razvrščanje po izbrani kategoriji in regiji | 23 |
| slika: 38 Razvrščanje po najvišji oceni | 24 |
| slika: 39 Razvrščanje glede na priljubljenost (število vseh skupnih naročil) | 24 |
| slika: 40 Razvrščanje po najnovejšem/najstarejšem..... | 25 |
| slika: 41 Naročanje, ocenjevanje in komentiranje onemogočeno lastniku obrti (ni gumbov)..... | 25 |
| slika: 42 Preverjanje izpolnjenost obveznih vnosnih polj..... | 26 |
| slika: 43 Ustrezno izpolnjena vnosna polja | 26 |
| slika: 44 Sporočilo o uspešno oddanem naročilu | 27 |
| slika: 45 Sporočilo o novih naročilih | 27 |
| slika: 46 Prejeta naročila | 27 |
| slika: 47 Razvrščanje naročil | 28 |
| slika: 48 Ustvarjanje storitve | 28 |
| slika: 49 Urejanje storitve | 29 |
| slika: 50 Zavržen dostop do urejanja, saj storitev ni v lasti obrtnika | 29 |

1. Uvod

1.1 Besedilo naloge

Moj obrtnik

Načrtujte in izdelajte sistem, ki bo uporabnikov v pomoč pri iskanju obrtnikov ter pošiljanja poizvedb za želeno 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

1.2 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

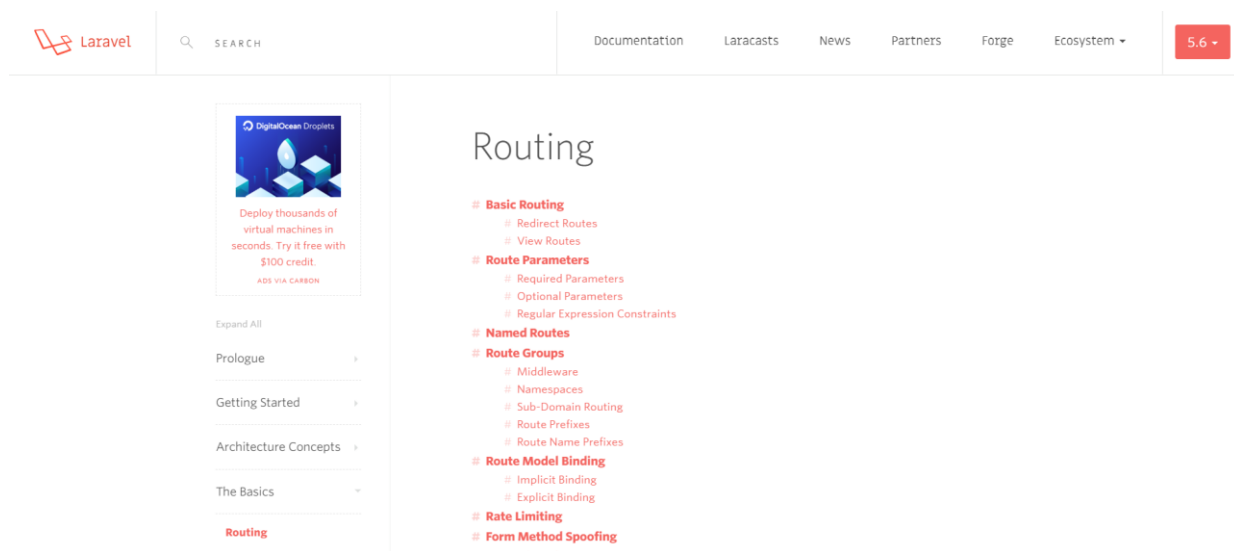
1.3 Izbrano ogrodje in jeziki

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 uporabljati. 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 uporabljali Microsoft Visual Studio Code. Za ustvarjanje spletnih strani smo uporabili označevalni jezik HTML, za stilsko oblikovanje in raznorazne efekte pa stilski jezik CSS.

2. Uporabljena orodja in rešitev problema

2.1 Laravel

Laravel je eno izmed odprto kodnih orodij za izdelavo interaktivnih spletnih strani. Ustvarjeno je bilo specifično za programski jezik PHP. Tako kot Springboot, je tudi Laravel zasnovan na MVC arhitekturi. Ko smo se odločili 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

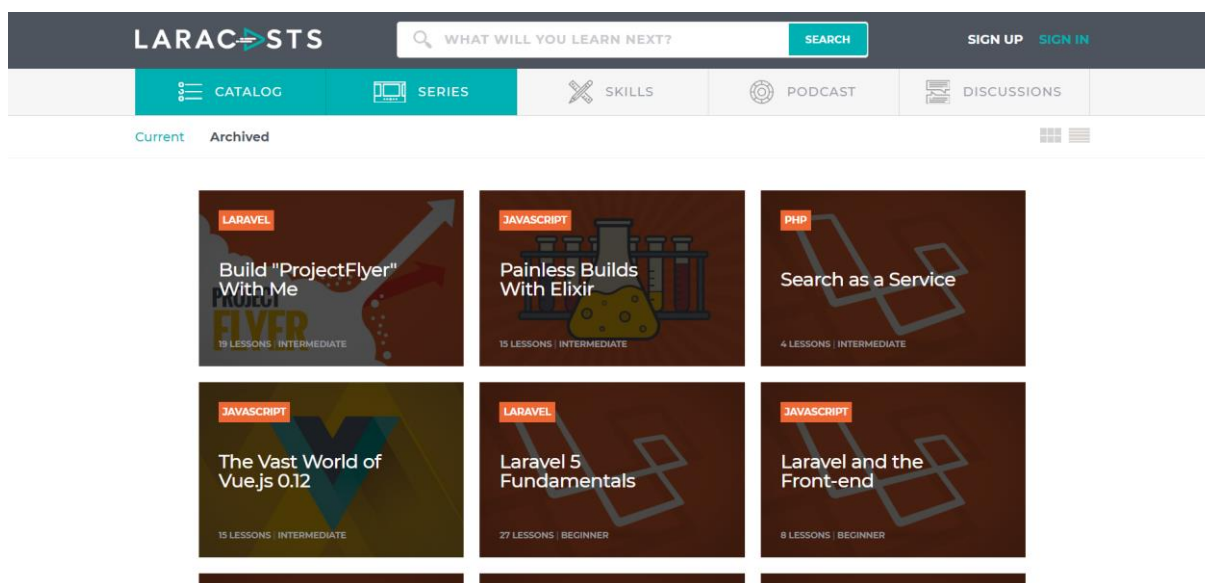
2.2 Artisan

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...)

2.3 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

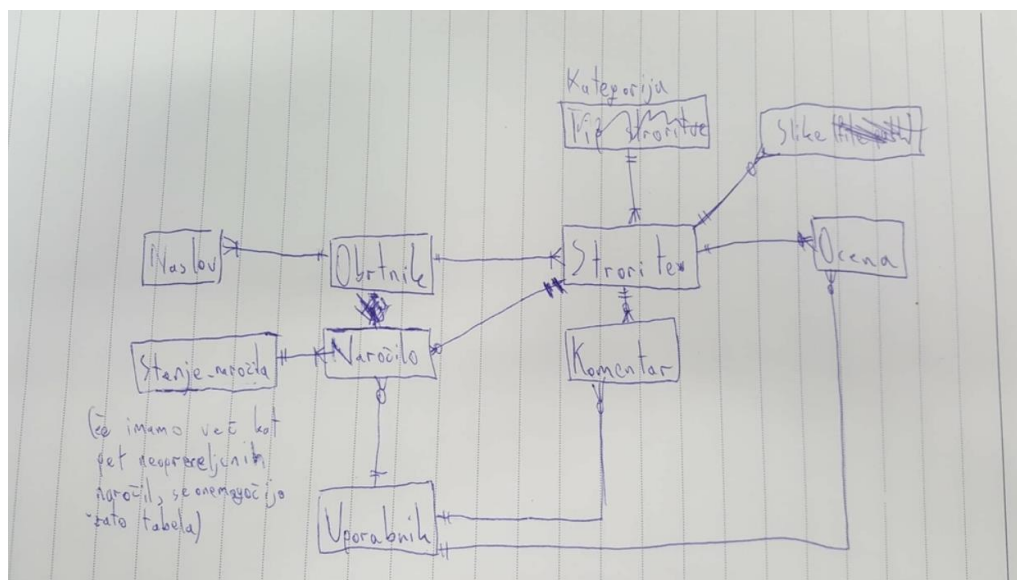
3. Podatkovna baza

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](#) je razložen in predstavljen v poglavju z programsko kodo.

3.1 Načrtovanje podatkovne baze

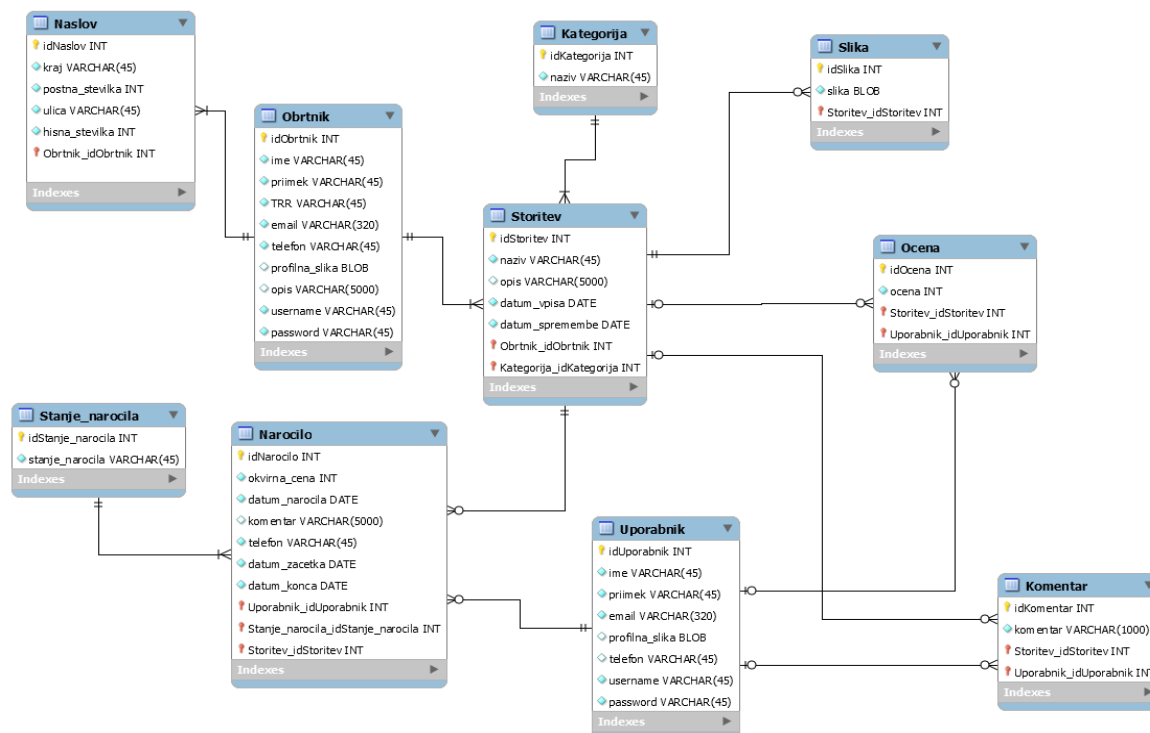
Razen manj pomembnih dodajanj in odstranjevanj atributov lahko razvoj naše podatkovne baze prikažemo v treh fazah. Pri vsaki izmed faz se je naša podatkovna baza spreminjala glede na potrebe in funkcionalnosti naše aplikacije.

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 nadaljnjem modeliranju in razvoju podatkovne baze.



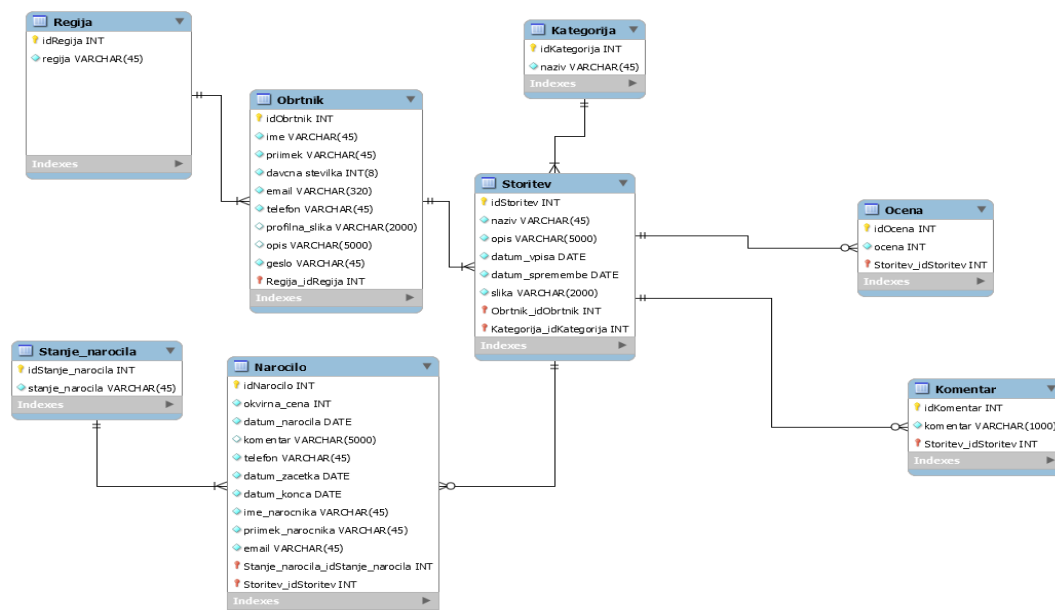
slika: 3 Osnutek

Faza 2: V drugi fazi načrtovanja smo na podlagi osnutka izdelali ER model, dodali attribute 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 primanjčovanja časa in zahtevnosti projekta smo bili primorani odstraniti ta tip uporabnika. Vse entitete ki so bile povezane z uporabnikom smo zaradi tega morali dopolniti z ustreznimi atributi, s katerimi smo lahko delno kompenzirali izločitev uporabnika.



slika: 5 Končen ER model naše podatkovne baze

3.2 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:

```
126 CREATE TABLE IF NOT EXISTS `mydb`.`storitev` (  
127   `idStoritev` INT NOT NULL,  
128   `naziv` VARCHAR(45) NOT NULL,  
129   `opis` VARCHAR(45) NULL,  
130   `vpisano` DATETIME NOT NULL,  
131   `spremenjeno` DATETIME NOT NULL,  
132   `Obrtnik_idObrtnik` INT NOT NULL,  
133   `Kategorija_idKategorija` INT NOT NULL,  
134   PRIMARY KEY (`idStoritev`),  
135   INDEX `fk_Storitev_Obrtnik1_idx` (`Obrtnik_idObrtnik` ASC),  
136   INDEX `fk_Storitev_Kategorija1_idx` (`Kategorija_idKategorija` ASC),  
137   CONSTRAINT `fk_Storitev_Obrtnik1`  
138     FOREIGN KEY (`Obrtnik_idObrtnik`)  
139       REFERENCES `mydb`.`Obrtnik` (`idObrtnik`)  
140     ON DELETE NO ACTION  
141     ON UPDATE NO ACTION,  
142   CONSTRAINT `fk_Storitev_Kategorija1`  
143     FOREIGN KEY (`Kategorija_idKategorija`)  
144       REFERENCES `mydb`.`Kategorija` (`idKategorija`)  
145     ON DELETE NO ACTION  
146     ON UPDATE NO ACTION  
147   ENGINE = InnoDB;  
148
```

```
public function up()  
{  
    Schema::create('storitev', function (Blueprint $table) {  
        $table->increments('id');  
        $table->string('naziv',45);  
        $table->string('opis',5000);  
        $table->string('slika')->nullable(true);  
        $table->timestamps();  
  
        //foreign keys  
        $table->integer('user_id')->unsigned();  
        $table->integer('kategorija_id')->unsigned();  
    });  
}
```

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.

```
1 use mojObrtnik;  
2  
3 insert into stanje_narocila values(null, 'V čakanju');  
4 insert into stanje_narocila values(null, 'Sprejeto');  
5 insert into stanje_narocila values(null, 'Zavrnjeno');  
6  
7 insert into regija values(null, 'Osrednjeslovenska');  
8 insert into regija values(null, 'Podravska');  
9 insert into regija values(null, 'Savinjska');  
10 insert into regija values(null, 'Pomurska');  
11 insert into regija values(null, 'Gorenjska');  
12 insert into regija values(null, 'Koroška');  
13 insert into regija values(null, 'Goriška');  
14 insert into regija values(null, 'Jugo-vzhodna Slovenija');  
15 insert into regija values(null, 'Primorsko-Notranjska');  
16 insert into regija values(null, 'Obalno-kraška');  
17 insert into regija values(null, 'Zasavska');  
18 insert into regija values(null, 'Posavska');  
19  
20 insert into kategorija values(null, 'Avtoličarstvo');  
21 insert into kategorija values(null, 'Čiščenje');  
22 insert into kategorija values(null, 'Dimnikarstvo');  
23 insert into kategorija values(null, 'Elektro inštalacije');  
24 insert into kategorija values(null, 'Fasaderstvo');  
25 insert into kategorija values(null, 'Fotografija');  
26 insert into kategorija values(null, 'Frizerstvo');
```

slika: 8 Skripto za vstavljanje podatkov v podatkovno bazo

4. Uporabniški vmesnik

4.1 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 manj 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.

```
<div class="parallax-container center valign-wrapper" style="height:30em;border-bottom:1px solid green;">
  <div class="container">
    <div class="row">
      <div class="col s12 white-text">
        <h1 class="center-align green-text text-darken-2" id="indexPictureText">Vaša spletna obrtniška
      </div>
    </div>
  </div>

  <div class="parallax">
    
  </div>
</div>
```

slika: 9 Primer oblikovanja z materialize

4.2 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 čim bolj 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

5. 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
- Kontrolerji
- Migracije
- Blade
- Prijava in registracija
- E-pošta
- Naročila, storitve in autorizacija
- Prijava in registracija

5.1 Poti

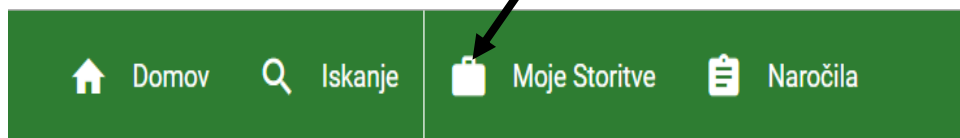
Poti oziroma "Routes", so pomemben in potreben element kode saj se uporabljajo za koordinacijo med kontrolerji in usmerjajo aplikacijo po poteh ki so definirane.

```
/*kontroler za menjavanje strani*/
Route::get('/', 'PagesController@Pindex');
/*kontrolerji za prijavo*/
Route::get('/prijava', 'PagesController@Plog');
Route::post('/prijava', 'LoginController@login');
/*kontrolerji za registracijo*/
Route::get('/registracija', 'PagesController@Preg');
Route::post('/registracija', 'RegController@register');
/*kontrolerji za narocila*/
Route::get('/narocila', 'NarocilaController@index')->name('n_index');
Route::get('/narocila/{id}', 'NarocilaController@show');
Route::post('/narocila/{id}/o', 'NarocilaController@zavrni')->name('zavrni');
Route::post('/narocila/{id}/z', 'NarocilaController@odobri')->name('odobri');
Route::post('/narocila', 'NarocilaController@razvrsti')->name('razvrsti');
```

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.

```
Route::resource('/iskanje', 'IskanjeController');
```

slika: 13 Pot

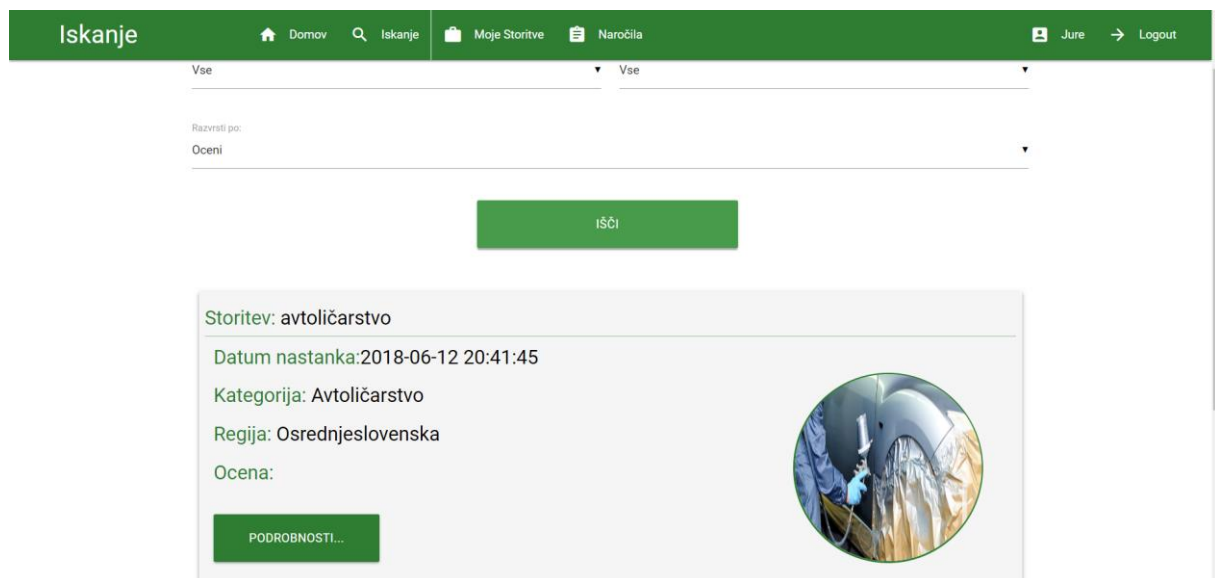
Korak 3: prožijo se metode v navedenem kontrolerju.

```
class IskanjeController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $kategorije = Kategorija::all();
        $regije = Regija::all();

        $storitve = DB::select("SELECT s.id, s.naziv, s.opis, s.created_at, s.slika, k.naziv as k_naziv, r.regija as r_naziv,
        AVG(o.ocena) as avg_ocena FROM users u JOIN storitev s ON u.id = s.user_id JOIN kategorija k ON k.id = s.kategorija_id
        JOIN regija r ON r.id = u.regija_id LEFT JOIN ocena o ON s.id = o.storitev_id GROUP BY s.id ORDER BY case when
        avg(o.ocena) is null then 1 else 0 end, avg(o.ocena) DESC, s.created_at DESC");
        return view('pages.iskanje')->with(array('storitve'=> $storitve,'kategorije'=>$kategorije,'regije'=>$regije));
    }
}
```

slika: 14 Metoda v kontrolerju

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



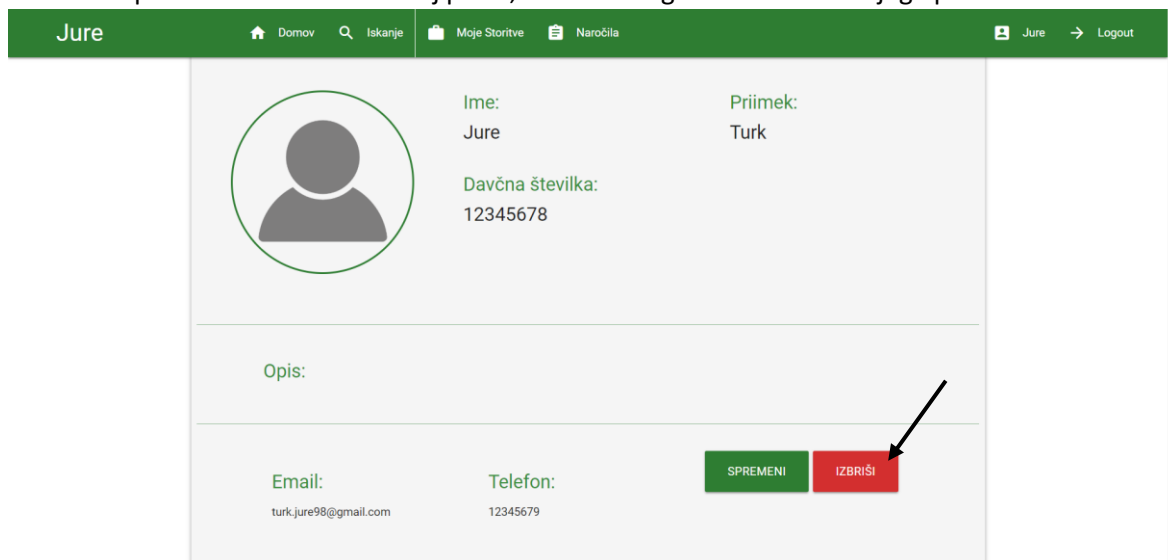
slika: 15 Prikaz strani

5.2 Kontrolerji

Kot je že prikazano v zgornjem primeru, lahko kontrolerji vračajo različen "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.

```
Route::post('/profile/d','ProfileController@destroy')->name('u_destroy');
```

slika: 17 Pot za profil

Korak3: Metoda izbriše uporabnika iz podatkovne baze.

```
public function destroy(){
    $user_id = auth()->user()->id;
    $user = User::find($user_id);
    $slika = $user->slika;
    if($slika != 'noprofile.png'){
        Storage::delete('public/cover_images/' . $slika);
    }
    $tests = DB::select("SELECT s.slika as s_slika FROM users u JOIN storitev s ON u.id = s.user_id");
    foreach($tests as $test){
        $slika = $test->s_slika;
        if($slika != 'noimage.png'){
            Storage::delete('public/cover_images/' . $slika);
        }
    }
    $user->delete();
    return redirect('/');
}
```

slika: 18 Metoda za izbris uporabnika

5.3 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 attribute, 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.

```
public function up()
{
    Schema::create('ocena', function (Blueprint $table) {
        $table->increments('id');
        $table->integer('ocena');
        $table->timestamps();

        //foreign keys
        $table->integer('storitev_id')->unsigned();
    });
}
```

slika: 19 Predefiniranje tabele za oceno

```
Schema::table('ocena', function (Blueprint $table) {
    $table->foreign('storitev_id')
        ->references('id')
        ->on('storitev')
        ->onDelete('cascade');
});
```

slika: 20 Tuji ključi za tabelo ocena

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

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=mojobrtnik
DB_USERNAME=root
DB_PASSWORD=
```

slika: 21 Env datoteka

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

```
Asus@ASUS MINGW64 /c/xampp/htdocs/Obrtnik
$ php artisan migrate
```

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.

phpMyAdmin interface showing a database table named 'narocila' with columns: kategorija, komentar, migrations, narocilo, ocena, password_resets, regija, stanje_narocila, storitev, storitev, users. The table has 10 labels and a total size of 288 KiB.

| Tabela | Dejanje | Vrstic | Vrsta | Pravilo za razvrščanje znakov | Velikost | Presežek |
|-----------------|-----------|-----------|---------|-------------------------------|----------|----------|
| kategorija | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| komentar | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| migrations | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| narocilo | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| ocena | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| password_resets | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| regija | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| stanje_narocila | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| storitev | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| storitev | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| users | Prebrskaj | Struktura | Iskanje | Vstavi | Izprazni | Zavrzi |
| 10 tabel | Vsota | | | | | |

slika: 23 phpMyAdmin

5.4 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

```

1 @extends('layouts.footer')
2 @extends('layouts.main')
3 @section('title')
4 <title>Moj Obrtnik | Iskanje</title>
5 @endsection
6 @section('Logo')
7 Iskanje
8 @endsection
9
10 @section('Content')
11 <div style="height:5em;">
12
13 </div>
14 <h5 class="center-align">Da se naročiš na storitev <b>{{ $storitev->naziv }}</b> vpiši svoje podatke:</h5><br>
15 <form action="{{ route('naroci', $storitev->id) }}" method="POST" type="hidden" name="_token" enctype="multipart/form-data">
16 {{ csrf_field() }}
17
18 <div class="row">
19 <div class="col s10 offset-s1">
20 <div class="row">
21 <div class="input-field col s12 l6">
22 <input id="naroc" type="text" class="validate" name="ime">
23 <label for="naroc">Ime</label>
24 </div>
25 <div class="input-field col s12 l6">
26 <input id="naroc" type="text" class="validate" name="priimek">
27 <label for="naroc">Priimek</label>
28 </div>
29 </div>
30 </div>
31 <div class="row">
32 <div class="input-field col s12 l12">
33 <input id="naroc" type="text" class="validate" name="email">
34 <label for="naroc">E-mail</label>
35 </div>
36 </div>
37
38 </div>

```

slika: 24 Datoteka blade

5.5 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.

```
<div class="input-field">
  <input id="name" type="text" class="{{ $errors->has('name') ? ' is-invalid' : '' }}" name="name" value="{{ old('name') }}" required autofocus>
  <label for="name">{{ __('Ime') }}{{ __('Priimek') }}{{ __('E-Mail Naslov') }}
```

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](#).

5.6 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.

| | |
|-------------------------------------|---------------------------------|
| Naziv storitve: avtoličarstvo | Številka naročila: 1 |
| Naročnik: Jure Turk | Email: turk.jure98@gmail.com |
| Datum naročila: 2018-06-13 11:02:48 | Datum pričetka dela: 2018-01-01 |
| Datum konca dela: 2018-12-31 | |
| Telefon: 12345679 | Okvirna cena: 450€ |
| Komentar: test | |
| ZAVRNI | ODOBRI |

slika: 26 Naročilo

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

```
public function zavrne($id)
{
    $narcilo = Narcilo::find($id);
    $narcilo->stanje_narocila_id = 3;
    $narcilo->save();


    Mail::send(['text'=>'mail2'], ['name','test'], function ($message) use ($narcilo){
        $message->to($narcilo->email,'Test')->subject('Narocilo zavrnjeno');
        $message->from('mojobrtnikinfo@gmail.com','MojObrtnik');
    });

    return redirect('/narocila/'.$id);
}
```

slika: 27 metoda za pošiljanje e-pošte

Korak 3: Uporabnik prejme ustrezno e-pošto.

Narocilo zavrnjeno ☐ Prejeto x

 **MojObrtnik** <mojobrtnikinfo@gmail.com>
Za meni ▾
Pozdravljeni!
Sporočamo vam da je vaše naročilo bilo Zavrnjeno. Za več informacij se obrnite na izbranega obrtnika.
Ekipa Moj Obrtnik

Ta e-pošta je bila pregledana z Avast protivirusnim programom.
<https://www.avast.com/antivirus>

slika: 28 Primer e-pošte

5.7 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 verifikacija.

Storitve

Metoda ki iz podatkovne baze prikaže vse storitve uporabniku. Da

```
public function show($id)
{
    $storitev = DB::select("SELECT s.id, u.opis as u_opis, s.naziv as s_naziv, s.opis, s.created_at, s.slika as s_slika, u.slika as u_slika, k.naziv as k_naziv, r.regija as r_naziv, u.name, u.surname, AVG(o.ocena) as avg_ocena FROM users u JOIN storitev s ON u.id = s.user_id JOIN kategorija k ON k.id = s.kategorija_id JOIN regija r ON r.id = u.regija_id JOIN ocena o ON s.id = o.storitev_id WHERE s.id='$id'");
    return view('storitev.show')->with(array('storitev'=>$storitev));
}
```

slika: 29 Metoda za prikaz storitev

Naročila

Spodaj navedena metoda prikaže naročila, ki pripadajo določenemu uporabniku.

```
public function show($id)
{
    $user_id = auth()->user()->id;

    $narocila = DB::select("SELECT n.id as n_id, u.id as user_id, s.naziv as s_naziv, n.ime, n.priimek, n.email, n.telefon, n.created_at, n.datum_zacetka, n.datum_konca, n.komentar, n.okvirna_cena, n.stanje_narocila_id as stanje FROM narocilo n JOIN storitev s ON s.id = n.storitev_id JOIN users u ON u.id = s.user_id WHERE n.id = '$id'");
    foreach($narocila as $narocilo){
        if($user_id != $narocilo->user_id){
            return 'Probu sil!';
        }
    }

    return view('user.narocilo')->with('narocila',$narocila);
}
```


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.

5.8 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.

 Opis Storitve


urejanje pričesk


Kategorija:
Frizerstvo


Datum nastanka:
14.06.2018

Ocena:

Regija:
Osrednjeslovenska

 Avtor

 Komentarji

 OCENI

slika: 31 Ocenjevanje in komentiranje

6. Varnost

6.1 Kodiranje gesel

Pri shranjevanju gesel smo poskrbeli za njihovo enkripcijo, 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

6.2 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.

```
$this->validate($request,[
    'naziv'=>'required|max:50',
    'opis'=>'required|max:5000',
    'slika'=>'image|nullable|max:1999'
],[
    'naziv.max'=>'Všišite naziv pod 50 črkami.',
    'opis.max'=>'Vpišite opis pod 5000 znaki.',
    'slika.ax'=>'Slika je prevelika',
    'naziv.required'=>'Prosim vpišite naziv.',
    'opis.required'=>'Prosim vpišite opis.',
]);
```

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

6.3 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

7. 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.

Ime
Janez

Priimek
Novak

E-Mail Naslov
janez.novak@gmail.com

The email has already been taken.

Geslo

Ponovi Geslo

Davčna številka
12345678

Telefon
031 222 453

slika: 34 E-poštni naslov že v uporabi

Ime
Franc

Priimek
Kopito

E-Mail Naslov
franc.kopito@mail.com

Geslo

The password confirmation does not match.

Ponovi Geslo

Davčna številka
12345678

Telefon
031 644 213

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.

E-Mail Address
janez.movak@mail.com

Password

E-Mail ali Geslo sta napačna.

✓ PRIJAVI

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

Kategorija
Frizerstvo

Regija
Osrednjeslovenska

Razvrsti po:
Oceni

IŠČI

Frizerstvo Tina

Kategorija:
Frizerstvo

Regija:
Osrednjeslovenska

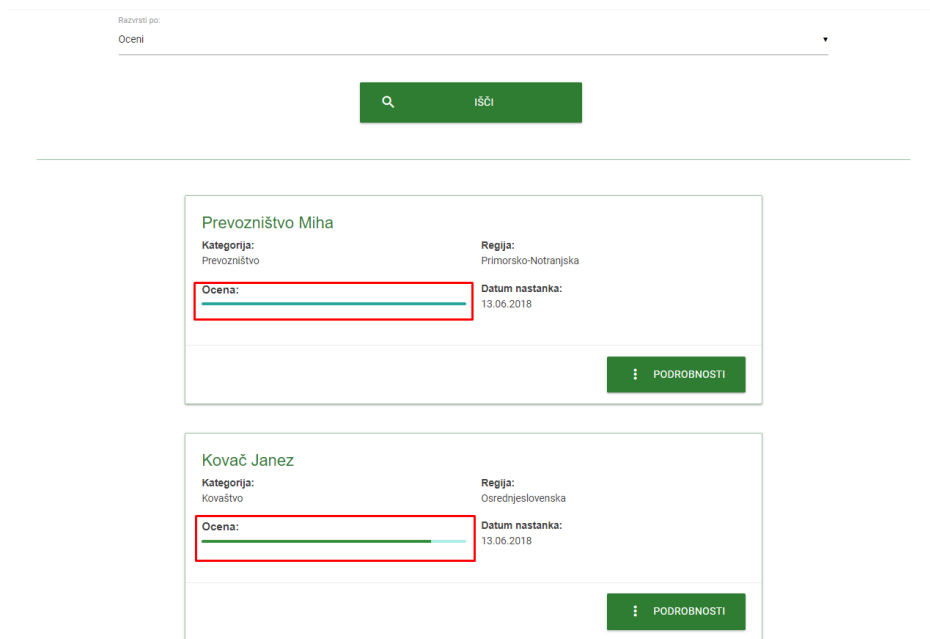
Ocena:

Datum nastanka:
13.06.2018

PODROBNOSTI

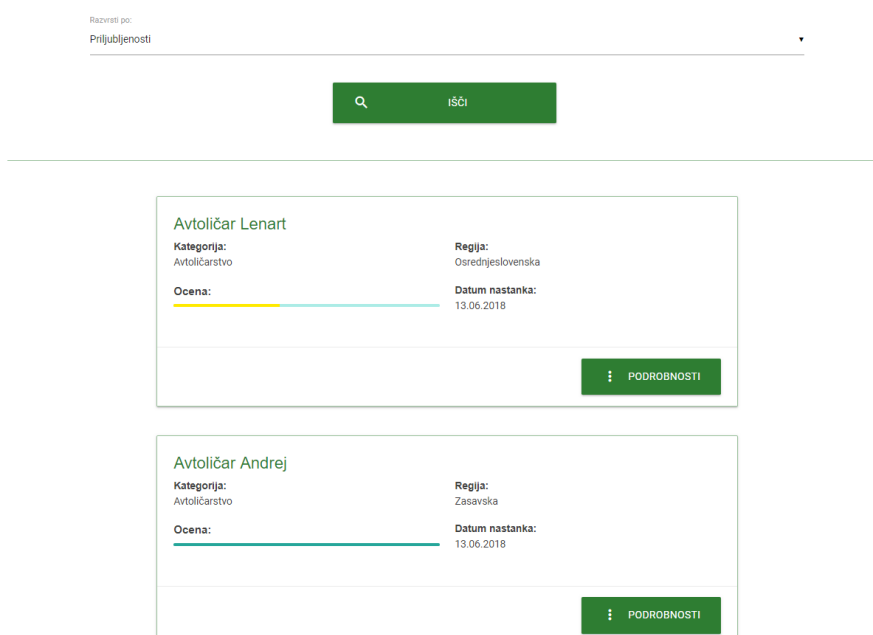
slika: 37 Razvrščanje po izbrani kategoriji in regiji

- 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



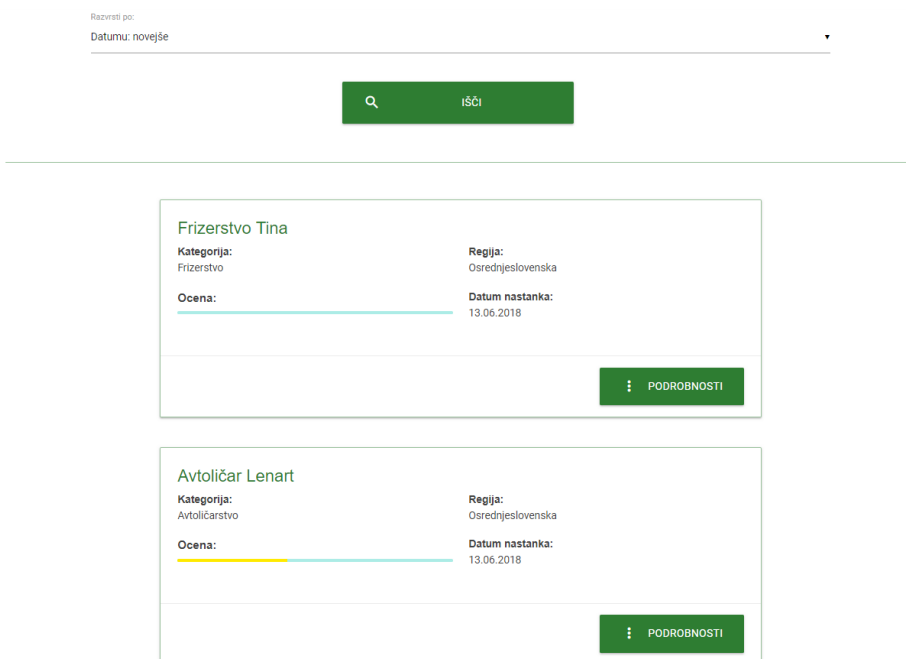
slika: 38 Razvrščanje po najvišji oceni

- 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)

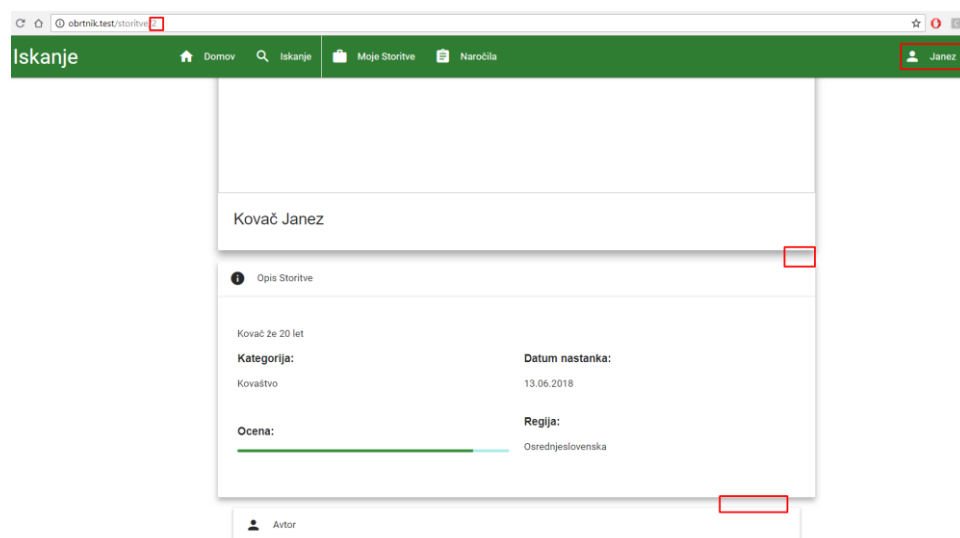
- razvrščanje glede na datum (najstarejši in najnovejši), pri katerem se storitve razvrstijo po datumu objave storitve



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)

Da se naročiš na storitev **Kovač Janez** vpiši svoje podatke:

Ime

Preimek

E-mail

Prosim vnesite E-mail naslov.

Podatki o naročilu:

Začetek:

01. 01. 2018

Konec:

31. 12. 2018

Okvirna cena (€)

Telefon

Komentar



NAROČI

slika: 42 Preverjanje izpolnjenost obveznih vnosnih polj

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

Da se naročiš na storitev **Kovač Janez** vpiši svoje podatke:

Ime

Janez

Preimek

Movak

E-mail

janez.movak@mail.com

Podatki o naročilu:

Začetek:

11. 07. 2018

Konec:

12. 07. 2018

Okvirna cena (€)

200

Telefon

031 234 567

Komentar

Potrebujem izdelavo kopit za konje.



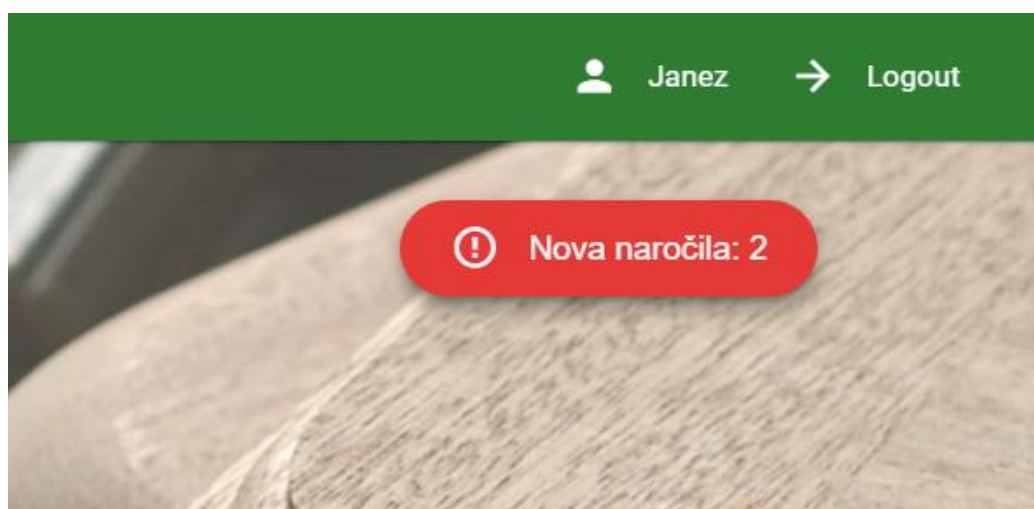
NAROČI

slika: 43 Ustrezno izpolnjena vnosna polja



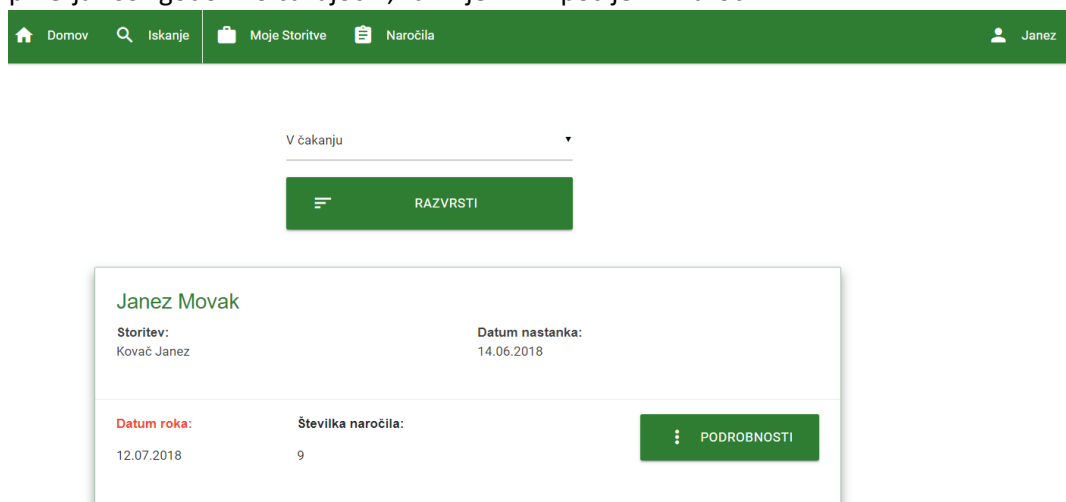
slika: 44 Sporočilo o uspešno oddanem naročilu

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 preverja vso zgodovino čakajočih, zavrnenih in potrjenih naročil.



slika: 46 Prejeta naročila

Sprejeta ▼

RAZVRSTI

Janez Movak

Storitev: Kovač Janez Datum nastanka: 14.06.2018

Datum roka: 12.07.2018 Številka naročila: 9

PODROBNOSTI

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.

Naziv

Opis

Avtoličarstvo ▼

SLIKA Izberite sliko

DODAJ

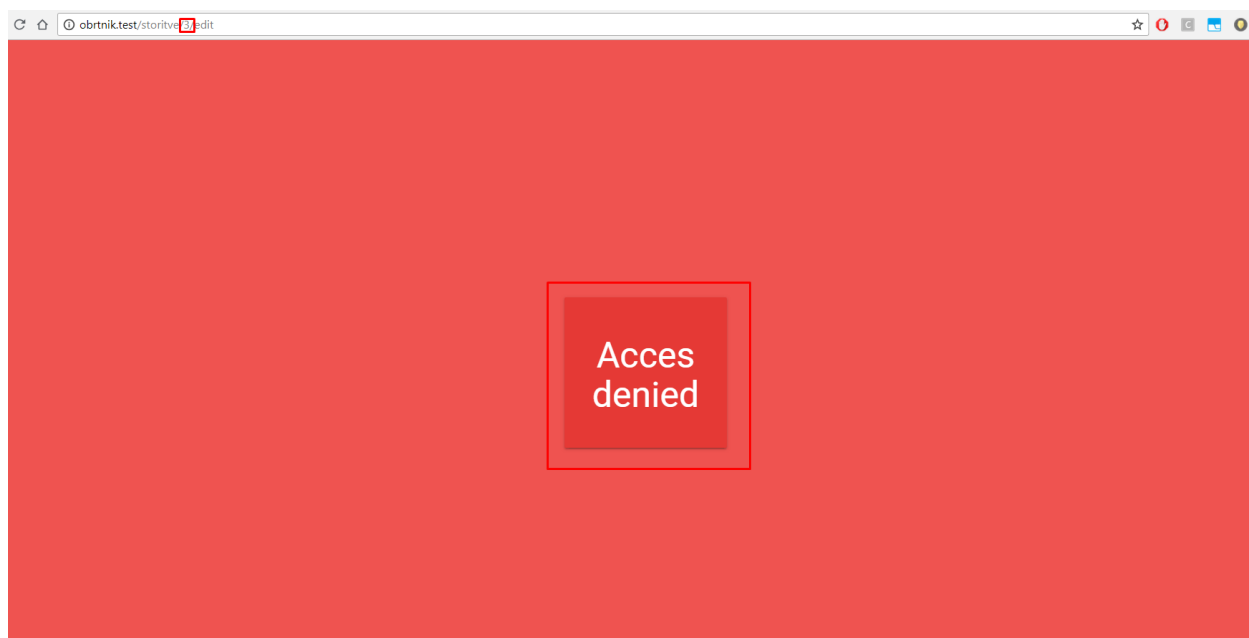
slika: 48 Ustvarjanje storitve

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 zavrjen.

The screenshot shows a web browser window with the address bar displaying 'obrtnik.test/storitev/2/edit'. The application has a green header bar with the title 'Storitev' and navigation links: 'Domov', 'Iskanje', 'Moje Storitve', and 'Naročila'. A user profile icon labeled 'Janez' is in the top right corner. The main form contains the following fields:

- Naziv:** Kovač Janez
- Opis:** Kovač že 20 let
- Avtoličarstvo:** A dropdown menu.
- SLIKA:** A section with the text 'Izberite sliko' and a placeholder image of a blacksmith working.
- KONČANO:** A green button with a checkmark icon.

slika: 49 Urejanje storitve



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

8. 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.