Izrada web aplikacije za najam vozila

Barić, Matija

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:751511

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Preddiplomski sveučilišni studij računarstvo

Izrada web aplikacije za najam vozila

Završni rad

Matija Barić

Osijek, 2022 godina.

SADRŽAJ

1. UVOD 1

1.1. Zadatak rada 1

2. PREGLED PODRUČJA RADA 2

2.1. Slične aplikacije 2

3. ALATI I TEHNOLOGIJE 3

3.1. PHP 3

3.2. Laravel 4

3.2.1. Usporedba razvojnih okruženja 5

3.2.2. MVC 6

3.3. Visual Studio Code 6

3.4. SQL 7

3.5. CMD 8

4. IZRADA WEB APLIKACIJE 9

4.1. Osnovni izgled web aplikacije 10

4.2. Registracija i prijava korisnika 14

4.3. Stvaranje novih oglasa 17

4.4. Spremanje korisnika i oglasa u bazu podataka 20

..... 23

4.5. Upravljanje, uređivanje i brisanje oglasa 23

4.6. Kreiranje, spremanje i prikazivanje rezervacija 28

5. ZAKLJUČAK 31

LITERATURA 32

SAŽETAK 33

ABSTRACT 34

1. UVOD

PHP je u današnjici poznat kao jedan od najpopularnijih programskih jezika diljem svijeta. Najnovija osma verzija PHP-a donosi mnoga poboljšanja kao što su imenovani argumenti, atributi, unapređenje svojsta konstruktora, nullsafe operatori, JIT i unaprijeđenje u error handling-u i dosljednosti. Razvojni programeri (developers) rado biraju PHP kada se radi o izradi web stranica ili web aplikacija koje koristi veliki broj ljudi. Prilikom izrade web stranice ili web aplikacije stvara se veliki broj redova koda. Kako bi se povećala preglednost, ubrzao razvoj projekta i poboljšalo održavanje koda koristi se razvojno okruženje. Laravel je jedno od mnogobrojnih razvojnih okruženja za PHP jezik koje se unazad nekoliko godina koristi sve više. Nastao je 2011. godine, a glavne prednosti su besplatno korištenje, jednostavnost, opširna dokumentacija i velika zajednica korisnika ovoga razvojnog okruženja. Zbog navedenih prednosti u završnome radu će se koristiti PHP i Laravel kao razvojno okruženje za izradu web aplikacije.

1.1. Zadatak rada

Glavni zadatak rada je napraviti potpuno funkcionalnu web aplikaciju koja korisnicima pruža mogućnost pregleda svih dostupnih automobila za najam uz uvjet da su korisnici registrirani u bazi podataka, kreiranje rezervacije za određeni automobil te kontakt putem elektroničke pošte s vlasnikom automobila.

2. PREGLED PODRUČJA RADA

Glavni zadatak ove aplikacije je omogućiti korisniku najam automobila putem web aplikacije. Kako bi to bilo moguće, nekoliko dijelova se mora povezati u funkcionalnu cjelinu. Kako bi web aplikacija bila uporabljiva, potreban je potpuno funkcionalan Laravel projekt. Uređujemo ga u nekome od uređivača koda, te nam uređivač koda pruža mogućnosti za lakšu izradu projekta. Nadalje, Laravel projekt se mora povezati s bazom podataka kako bi uvid bio omogućen u dostupne aute i njihove karakteristike. Xampp pomaže u pokretanju baze podataka i web poslužitelja na kojemu će raditi web aplikacija.

Sama problematika izrade ove web aplikacije je izraditi funkcionalan Laravel projekt koji neće imati greške u radu, a pružiti će sve ono što klijent očekuje od web aplikacije.

2.1. Slične aplikacije

Prva aplikacija slična ovome radu je aplikacija naziva Nova Rent a Car [1]. U aplikaciji korisnik ima mogućnost odabrati za koji grad traži automobil za najam, datum početka i datum kraja najma te vrijeme preuzimanja i povrata automobila. Također su predložene neke od zanimljivih destinacija turistima u Hrvatskoj.

Druga aplikacija slična ovome radu je aplikacija naziva Usluga Poslovnog Najma Enterprise [2]. Razlika ove i prve aplikacije je što u ovoj aplikaciji postoji mogućnost ugovaranja poslovnoga najma, kao i razni programi pogodnosti od sponzora. Također, postavljeni su savjeti što treba znati prije rezervacije automobila te postoji ponuda iznajmljivanja teretnih vozila.

Treća aplikacija slična ovome radu je aplikacija naziva Rent a Car Cheap [3]. Razlika ove aplikacije i ovoga rada je ta što u ovoj aplikaciji se korisnika traži da unese lokaciju preuzimanja, prije nego što vidi raspoložive automobile. Postavljena su i najčešća pitanja i odgovori na njih.

Četvrta aplikacija slična ovome radu je aplikacija naziva Rent a Car Hrvatska – Split – Zagreb – Dubrovnik [4]. Najveća razlika ove i svih ostalih spomenutih aplikacija je ta što korisnik automobil može preuzeti u jednome gradu, a vratiti ga vlasniku u drugome gradu. Dodana je i komunikacija sa korisničkom podrškom što ostale aplikacije nemaju.

3. ALATI I TEHNOLOGIJE

3.1. PHP

PHP (skraćenica od Hypertext Preprocessor) je raširen i vrlo popularan skriptni jezik, odnosno skriptni jezik otvorenog koda(engl. open source) te je prvobitno namjenjen za web development. PHP je lako kombinirati sa HTML-om te se kombiniranjem dobije dinamički kreirana web stranica. Razlikuje se od JavaScripta koji je klijentski skriptni jezik jer se PHP izvršava na poslužitelju. Rezultat toga je HTML kod kojega browser razumije te nema potrebe za nadogranjama. PHP je također vrlo pogodan za početnike koji se tek uče programirati, i novi su u tome svijetu, ali isto tako PHP nudi puno toga za iskusne programere [5].

PHP može biti korišten na 3 načina:

- Skriptiranje sa strane poslužitelja – njegova prvobitna namjena
- Naredbeno skriptiranje – u mogućnosti je izvoditi skripte iz naredbene linije (slično Unix shell-u).
- Skripte mogu biti korištene za administrativne poslove sustava. GUI aplikacije sa strane klijenta, na PHP-u može se napisati razvijena GUI (engl. graphical user interface) aplikacija.

PHP može biti korišten na nekoliko operacijskih sustava, od Unix-a do Windows-a i Mac OS X-a.

Početak koda u PHP-u koda je označen ovako:

<?php

Kraj koda u PHP-u je označen ovako:

?>

Postoje i neki drugi načini označavanja, ali u većini slučajeva će to biti na ovaj način. [6].

Ostale poslužiteljske tehnologije

Dugo godina veliki konkurent skriptnom jeziku PHP bio je ASP (Active Server Pages), Microsoftova poslužiteljska tehnologija čije su mogućnosti bile podjednake. No, ASP je zastario, a PHP se nastavio razvijati i dobivati bolju podršku za objektno-orijentirano programiranje.

Postoji i nekoliko razvojnih okvira (engl. framework) za PHP (Zend Framework, Symfony) čiji je cilj olakšati programiranje u njemu odvajanjem prezentacije (HTML) od logike (PHP kôd), te uvođenjem objekata koji predstavljaju HTML elemente i proširuju funkcionalnost web stranica, zatim objekata za rad s bazama podataka i drugih. Veliki konkurent je Microsoftov ASP.NET, donekle nasljednik tehnologije ASP, koji koristi .NET Framework za koji je moguće razvijati u više .NET jezika, od kojih su najrašireniji C# i Visual Basic.NET. Njegova prednost je veliki broj gotovih kontrola i objekata, i to što se njegove datoteke ne moraju prevoditi svaki put kod izvršavanja, već samo prvi put. Jaka konkurencija je i JSP (engl. JavaServer Pages). Za JSP danas postoji velik broj frameworka (Spring, Struts, JavaServer Faces...). Kao i ASP.NET, ne mora se prevoditi svaki put kod izvršavanja, i vrlo je raširen u svijetu poslovnih aplikacija.

3.2. Laravel

Laravel je vrlo poznat, besplatan za korištenje PHP web radni okvir otvorenog koda. Od početka, bio je namijenjen kao softwe za razvoj web aplikacija, a posebito aplikacija koje imaju MVC arhitekturu. Laravel ima velik broj korisnih opcija a neke su, sistem za modularno upakiravanje, koji se slaže sa Composer-om. Također, Laravel pruža pristup bazama podataka na nekoliko načina, i koristi osobine koje pridonose primjeni i održavanju aplikacije. Laravel je web radni okvir te pruža pomoć pri stvaranju novih aplikacija za web, podržavajući API ili web usluge. Laravel čini upravljanje i održavanje jednostavnijim. [6].

Laravel ima svojstva koja omogućuju vrlo brz razvoj aplikacije. Implementira prečace za česte i ponavljajuće dijelove koda poput keširanja, potvrde identiteta korisnika i slično. Prepoznatljiv je po veoma visokoj razini apstrakcije. Uz apstrakciju olakšava izradu aplikacija. Composer je alat za upravljanje zavisnostima u PHP-u. Brine o bibliotekama i njihovim nadogranjama. Ne upravlja paketima kao standardni upravitelj. Instalira pakete unutar direktorija projekta kada se stvara novi projekt. Međusloj (engl. Middleware) odnosno međusoftver, djeluje između mreže i aplikacije. Pruža mogućnost filtriranja HTTP zahtjeva koji dolaze aplikaciji. Laravel posjeduje vlastiti alat za predloške Blade. Alat za predloške smanjuje težinu posla izrade front-end koda te pruža bolju

funkcionalnost. Pogledi koji se stvaraju se spremaju u mapu app/resource/views. Način pisanja koda u PHP-u propisan Blade-om je prikazan u tablici 3.1. Iz tablice se vidi da je pisanje koda pomoću Blade-a puno jednostavnije i razumljivije.

view.php
<?php foreach($objekt as $objekt) ?> echo <?= $objekt->obiljezje ?>; <?php endforeach ?>
view.blade.php
@foreach($objekt as $objekt) echo {{ $objekt->obiljezje }}; @endforeach

Tablica 3.1 PHP propisan Blade-om

3.2.1. Usporedba razvojnih okruženja

Laravel i Node.js dvije su najutjecajnije i najučinkovitije tehnologije za web razvoj koje su se pojavile prije samo deset godina. Laravel je PHP okvir s većom snagom i stabilnošću. Budući da PHP pokreće 78% weba, Laravel, a time i Node.js, dobivaju značajan tržišni udio. Node.js konstruiran je s JavaScriptom, C i C++ i pruža sve alate koje razvojni programer treba za razvoj cijelog niza i brze performanse. Kada je Laravel iznimno moćan, Node je munjevito brz.

3.2.2. MVC

Glavna zadaća je sve dijelove i odgovornosti aplikacije koji su u mogućnosti za odvajanje odvojiti u jedinstvene strukture/klase, ili kreiranje slojeva aplikacije.MVC je obrazac za arhitekturu aplikacije koji propisuje na koji će se način struktuirati aplikacija u jedinstvene komponente i njihove zadaće i međudjelovanje. Uobičajeno je podijeliti aplikaciju u tri sloja, a to su sloj poslovne logike, podatkovni sloj i prezentacijski. MVC je dobio ime po tri klase:

- Model
- Upravljač
- Pogled

3.3. Visual Studio Code

Visual studio Code je jednostavan, ali snažan uređivač izvornog koda. Radi na desktopu a dostupan je za nekoliko operacijskih sustava kao što su Windows, macOS i Linux. Ugrađena mu je podrška za JavaScript, TypeScript i Node.js i ima širok ekosustav proširenja za neke druge jezike i vremena izvođenja(kao što su C++, C#, Java, Python, PHP, Go, .NET). U samom srcu, VS Code je uređivač teksta. Kao mnogo ostalih uređivača teksta, VS Code usvaja uobičajeno korisničko sučelje i izgled pretraživača s lijeve strane, koji prikazuje sve datoteke i mape kojima imate pristup, i uređivača s desne strane, koji prikazuje sadržaj datoteka koje ste otvorili. VS Code dolazi sa jednostavnim i intuitivnim rasporedom koji maksimizira prostor predviđen za uređivača dok istovremeno ostavlja dovoljno prostora za pretraživanje i pristup punom kontekstu vaše mape ili projekta. Korisničko sučelje je podijeljeno u pet područja:

- Uređivač – glavno područje za uređivanje vaših datoteka. Možete otvoriti koliko god želite datoteka jednu do druge horizontalno ili vertikalno
- Bočna traka – sadrži različite poglede kao što je Explorer za pomaganje dok radite na projektu
- Statusna traka – informacije o otvorenim projektima i datotekama koje uređujete
- Aktivna traka – smještena na krajnjoj lijevoj strani, omogućava mijenjanje pogleda i daje dodatne informacije kao što su broj izlaznih promjena kada je Git omogućen
- Paneli – možete prikazati različite panele ispod područja uređivača za izlaze ili debug informacije, pogreške i upozorenja ili integrirane terminale.

Svaki puta kada se pokrene VS Code, on se otvori u istom stanju u kojemu je bio kada ga se prošli puta zatvorilo. Otvorene datoteke u svakome uređivaču su prikazane u tabovima na samom vrhu područja uređivača. VS Code spaja jednostavnost uređivača izvornog koda sa jakim alatom za razvojne programere, poput IntelliSense-a za dovršavanje koda i uklanjanje pogrešaka. Prvo i najvažnije, to je uređivač koji vam ne smeta dok radite. Vrlo jednostavan ciklus uređivanja-izgradnje-otklanjanja pogrešaka znači manje vremena koje morate provoditi zamarajući se sa radnim okruženjem, a više vremena za stvaranje novih ideja [9].

3.4. SQL

SQL je programski jezik za rad s bazama podataka. SQL je korišten za stvaranje, uređivanje te pretraživanje određenih podataka. Podaci su spremljeni u tablice. SQL možemo koristiti za pronalazak i uređivanje informacije u tablicama. Programi s relacijskim bazama podataka koriste SQL. Za razliku od nekih drugih računalnih jezika, SQL nije težak za čitati i razumjeti. Kao i drugi računalni jezici, SQL je međunarodni standard.

SQL se koristi kako bi opisali skupove podataka. Za korištenje SQL-a potrebna je ispravna sintaksa. Sintaksa je skup pravila, te se elementi jezika oblikuju prema njima. SQL sintaksa temelji se na Engleskoj sintaksi, te ima sličnosti saVisual Basic for Applications (VBA).

Za opis skupa podataka uporabom SQL-a, piše se naredba SELECT. Naredba SELECT sadrži potpuni opis nekoga skupa podataka koji su traženi iz baze podataka. To obuhvaća sljedeće:

- Tablice sa podacima.
- Odnose podataka iz različitih izvora.
- Koja od polja ili proračuna će proizvesti podatke.
- Podaci moraju odgovarati na određene kriterije da bi bili obuhvaćeni.
- Na koji način sortirati rezultate.

SQL se ne koristi za upravljanje podacima, već i za stvaranje i promjenu dizajna tablica. DDL se koristi za stvaranje i mijenjanje objekata baze [10].

3.5. CMD

CMD je tekstualno okruženje koje korisniku daje mogućnost zadavanja različitih naredbi koje operacijski sustav izvršava. Skoro svaki operacijski sustav ima CMD.

Pomoću komandne linije korisnik može proći kroz strukture direktorija, pokretati neke aplikacije, kopirati ili brisati, otvarati datoteke te provesti još neke druge operacije. Loše svojstvo je tekstualno okruženje, što zahtijeva poznavanje velikog broja komandi i njhovu sintaksu.

Često korištene naredbe su md i rd. Prva naredba će stvoriti folder, skraćenica od Make Directory. Direktorij će se stvoriti na ruti u kojoj se korisnik nalazi. Nasuprot naredbe Make Directory postoji naredba Remove Directory, koja briše neki direktorij. Sintaksa za te naredbe glasi: md ime_foldera, odnosno rd ime_foldera. Naredba cd omogućava prolazak kroz strukturu direktorija na računalu, kao i kroz mrežnu strukturu. Popis naredbi CMD-a može se vidjeti ako se upiše naredba help. CMD je dobar alat za popravak i održavanje MS Windows sustava [11].

4. IZRADA WEB APLIKACIJE

Kako bi se krenulo u izradu web aplikacije, potrebno je imati sve potrebne alate na računalu koji će biti za to potrebni. Ova aplikacija je temeljena na PHP-u, te za framework se koristi Laravel. Prema tome, potrebno je instalirati PHP na računalo. Poveznica za preuzimanje PHP-a:

https://www.php.net/downloads.php [12]

Sljedeći korak je instalacija frameworka, tj. Laravela. Jedan od načina na koji se Laravel može instalirati na računalo je putem Xampp-a. Poveznica za preuzimanje xampp-a je:

https://www.apachefriends.org/index.html [13]

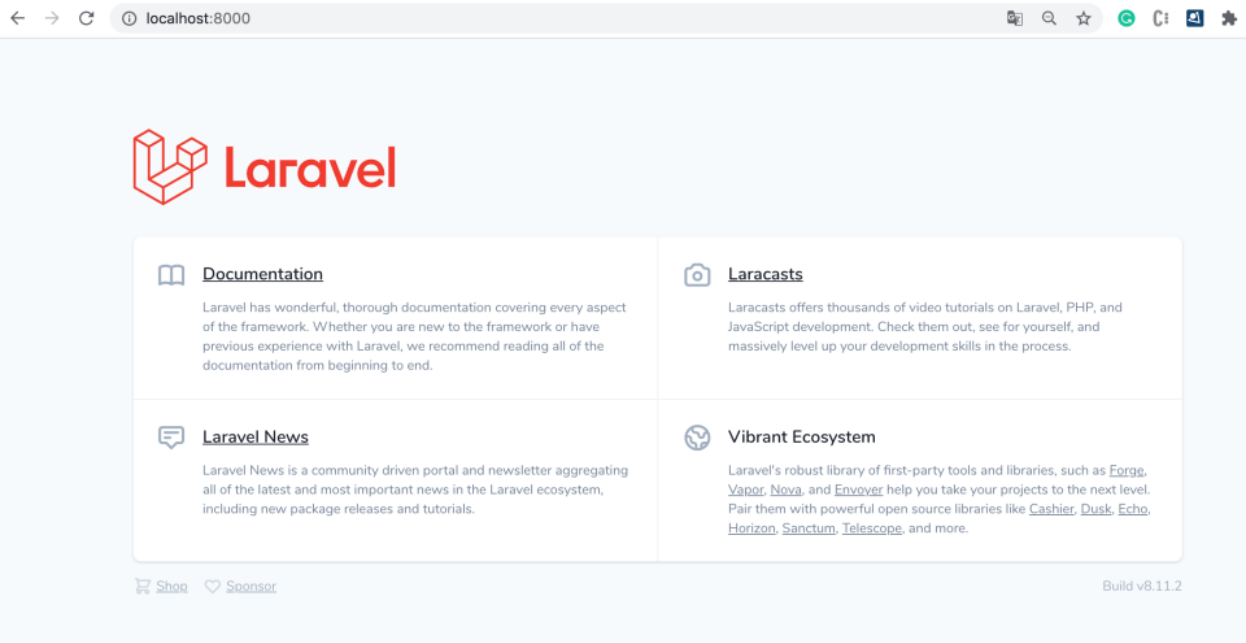
Nakon što je xampp instaliran na računalo, potrebno je preuzeti Composer i također ga instalirati. Poveznica za preuzimanje Composer-a je:

https://getcomposer.org/download [14]

Kada su ova tri koraka uspješno obavljena, možemo stvoriti Laravel projekt. Laravel projekt se stvara tako što pokrenemo CMD, premjesti se u htdocs mapu upotrebom naredbe cd, te upisivanjem ove naredbe:

composer create-project --prefer-dist laravel/laravel PROJECT_NAME

Konkretno za ovaj projekt, umjesto PROJECT_NAME upisano je carrentwebapp. Zadnji korak kako bi se pokrenula aplikacija je u Xampp-u poreknuti Appache modul te se vratiti u mapu c:\xampp\htdocs\carrentwebapp i u CMD-u pokrenuti naredbu php artisan serve. Dobiti će se poruka: Starting Laravel development server: http://127.0.0.1:8000. Kada se kopira web adresa i zalijepi u tražilicu dobije se početna web aplikacija koja je prikazana na slici 4.1.



Sl 4.1. Početni oblik Laravel projekta

4.1. Osnovni izgled web aplikacije

Kreiranjem Laravel projekta, automatski se dobiva unaprijed definirane mape sa potrebnim datotekama koje su potrebne za funkcioniranje aplikacije. U routes mapi, postoji datoteka po imenu web.php i u njoj se uređuje koji route će što prikazivati, odnosno koji će se kontroler koristiti. Nadalje će biti prikazan i objašnjen svaki route posebno, a objašnjenja kontrolera i ostalih funkcionalnosti povezanih s tim routeom će biti slijedno tome.

U sljedećem izlistanju koda je prikazana prva ruta (engl. Route), koji služi za prikaz svih oglasa na početnoj stranici.

```
1: //All Listings
2: Route::get('/', [ListingController::class, 'index']);
```

Primjer koda 4.1 Ruta (engl. Route) za sve oglase

URL na koji se ovaj route odnosi je '/', što znači da kada se upiše ta adresa u web tražilicu, proslijedit će se zahtjev ListingControlleru koji će pozvati funkciju index. Prije nego funkcije budu prikazane, treba reći nešto o ListingController-u. Listing Controller je kontroler koji se stvara putem CMD-a. Naredba za stvaranje je:

php artisan make::controller ListingController

U ListingControlleru su sadržane sve funkcije koje route može pozvati, a vezane su za taj kontroler. ListingController odvaja kod svake funkcije od same web.php datoteke te tako čini kod jednostavnijim i preglednijim. U sljedećem izlistanju koda će biti prikazana funkcija index u ListingControlleru.

```
1: public function index() {
2: return view('listings.index', [
3: 'listings' => Listing::latest()->filter(request(['tag', 'search']))-
   >paginate(4)
4: ]);
5: }
```

Primjer koda 4.2 Funkcija index

Funkcija index vraća pogled koji se nalazi u mapi listings, naziva index. Također se predaje parametar funkciji listings koja ima sortiranje da prikazuje zadnje dodane oglase te da filtrira oglase prema tag-ovima i search upitu. Nadalje će biti prikazan kod samog pogleda index stranice.

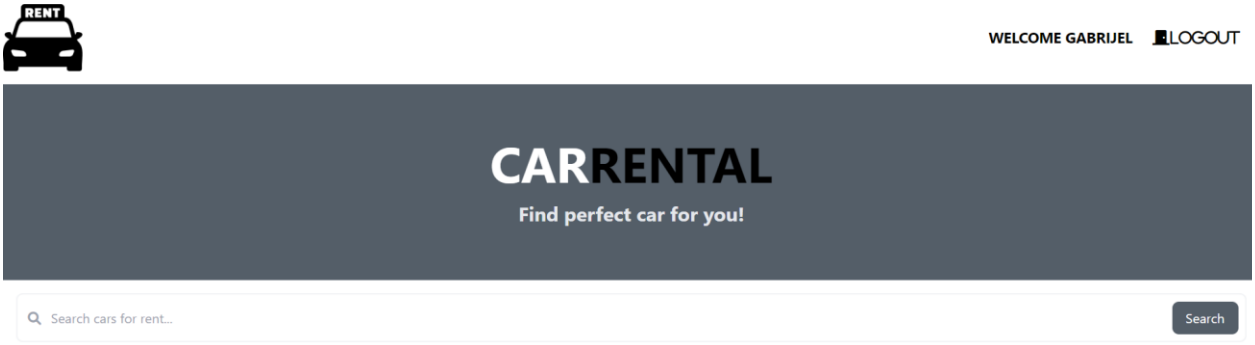
```
1: <x-layout>
2: @include('partials._hero')
3: @include('partials._search')
4:
5:
6: <div class="lg:grid lg:grid-cols-2 gap-4 space-y-4 md:space-y-0 mx-4">
7: @unless (count($listings)==0)
8:
9: @foreach($listings as $listing)
10:  <x-listing-card :listing="$listing" />
11:  @endforeach
12:
13:  @else
14:  <p> No listings</p>
15:  @endunless
16:
17:  </div>
18:
19:  <div class="mt-6 p-4">
20:  {{$listings->links()}}
21:  </div>
22:
23:  </x-layout>
```

Primjer koda 4.3 index.blade.php

Kako se par dijelova koda ponavlja u više pogleda kao što je ovaj, create, edit, manage i show, sada će biti objašnjeni. Prva linija koda uključuje layout stranice, tj. kako će stranica izgledati. Layout.blade.php se nalazi također u mapi pogleda, te datoteka sadrži navigation dio, gdje su smješteni home button, login button te zavisno da li je admin prijavljen, manage listings button i logout button. U podnožju (engl. footer) se prikazuje button za dodavanje novog oglasa samo ako je administrator prijavljen.

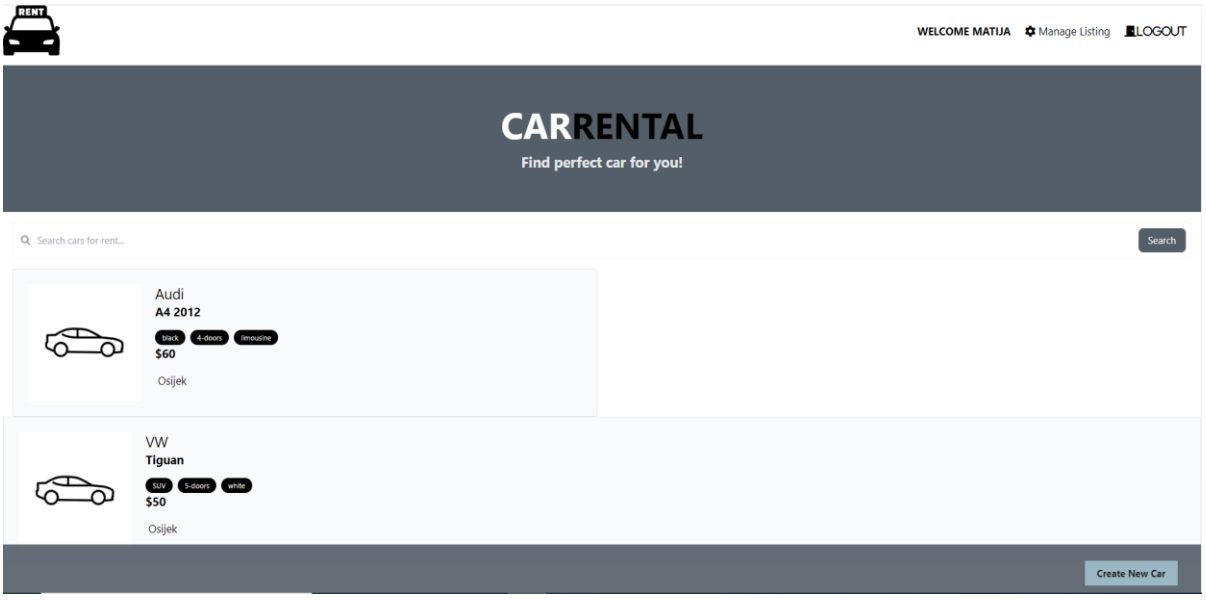
Sljedeće što je uključeno su partials datoteke. Datoteka partials._hero je uključena samo u index pogledu i s njom se prikazuje naslov web stranice i poruka korisniku što je uokvireno.

Datoteka partials._search je uključena u index i show pogledu te služi korisniku kako bi pretražio ključne riječi.Primjer kako izgleda kada su datoteke uključene je prikazan na slici 4.2.



Sl. 4.2. Izgled partials._search i partials._hero na početnoj stranici

Nakon što je objašnjen izgled početne stranice, tj. layout, sljedeća je funkcionalnost, odnosno kako se oglasi prikazuju. U index funkciji u ListingController-u, nakon što je definiran pogled koji se vraća listings.index, dodano je koji se podaci prosljeđuju tome pogledu. U ovom slučaju prosljeđuju se oglasi('listings' => Listing::latest()->filter(request(['tag', 'search']))->paginate(4)). Način na koji se dohvaćaju oglasi je da se poveže sa modelom Listing koji je stvoren ranije, zatraže se zadnji dodani oglasi i filtriraju se ako korisnik u tražilicu upisuje određen pojam ili klikne na određeni 'tag' u oglasu. Ti podaci u pogledu index se ispisuju pomoću listing-card-a, koji podatke uređuje pomoću CSS-a. Primjer kako izgleda početna stranice web aplikacije prikazano je na slici 4.3.



Sl. 4.3. Izgled početne stranice

Sljedeće što korisnik može napraviti je pogledati svaki oglas posebno. Korisnik ulazi u određeni oglas tako što mišem klikne na ime auta, što ga vodi na detaljniji prikaz toga oglasa. Detaljniji prikaz sadrži više informacija o autu, kao što je opis auta, rezervacije za taj auto koje je napravio korisnik i gumb za kontakt vlasnika automobila. Izlistanje koda za detaljniji prikaz oglasa u web.php datoteci:

```
1: //Single Listing
2: Route::get('/listings/{listing}', [ListingController::class, 'show']);
```

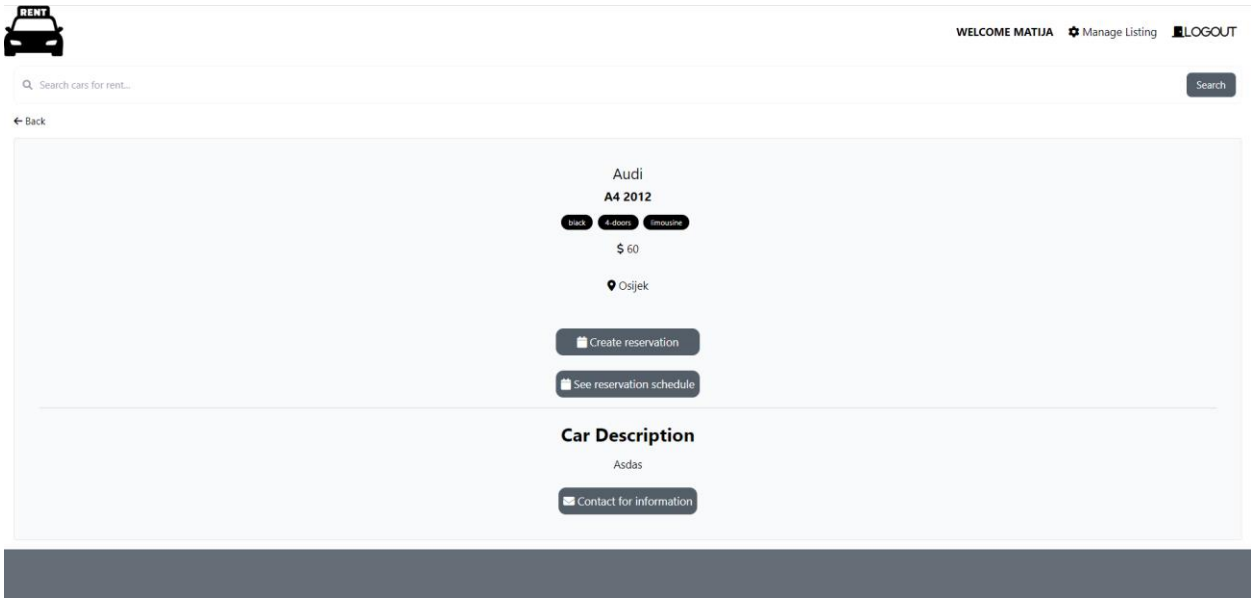
Primjer koda 4.4 Ruta (engl. Route) za detaljniji prikaz

Iz koda se vidi da se poziva funkcija show te slijedi izlistanje za funkciju show u ListingController-u:

```
1: public function show(Listing $listing, Reservation $reservation) {
2: return view('listings.show', ['listing'=> $listing, 'reservation' =>
   $reservation]);
3: }
```

Primjer koda 4.5 Funkcija show

Funkcija vraća istoimeni pogled listings.show te se prosljeđuju podaci za taj određeni oglas i za njegove rezervacije. Primjer detaljnijeg prikaza oglasa prikazan je na slici 4.4.



Sl. 4.4. Izgled detaljnijeg prikaza oglasa

Do sada je prikazano osnovni izgled stranice te funkcionalnost prikaza pojedinih oglasa. U nastavku će biti objašnjeno kako se stvaraju novi korisnici te prijavljuju postojeći, kako se stvaraju novi oglasi i koji je uvjet za njihovo stvaranje, spremanje korisnika i oglasa u bazu podataka, upravljanje oglasima od strane admina, uređivanje postojećih oglasa, brisanje oglasa te kreiranje rezervacija, prikazivanje i spremanje istih u bazu podataka.

4.2. Registracija i prijava korisnika

Na početnoj stranici web aplikacije nalaze se dva gumba, od kojih je jedan za registraciju korisnika, a drugi za prijavu, odnosno login. Kada gost(neprijavljeni korisnik) klikne na gumb za registraciju otvorit će mu se formular za registraciju. To je omogućeno putem route-a koji poziva funkciju create u UserController-u. Izlistanje koda za pozivanje create funkcije:

```
1: //Show Register/Create Form
2: Route::get('/register', [UserController::class, 'create'])-
   >middleware('guest');
```

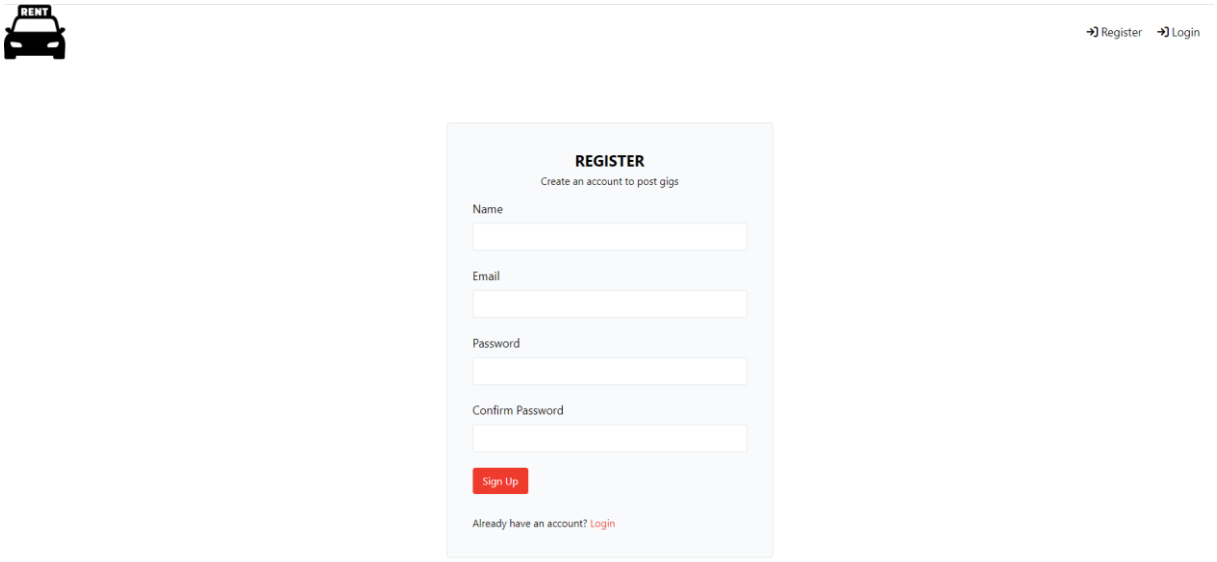
Primjer koda 4.6 Ruta (engl. Route) za kreiranje novog oglasa

Izlistanje koda za funkciju create:

```
1: public function create(){
       return view('users.register');
2: }
```

Primjer koda 4.7 Funkcija create

Funkcija create vraća pogled users.register. Primjer pogleda register prikazan je na slici 4.5.



Sl. 4.5. Izgled register pogleda

Pravilnim popunjavanjem formulara za registraciju te klikom na gumb *Sign Up* stvara se novi korisnik u bazi podataka. To se ostvaruje putem sljedeće rute(engl. Route):

```
1: //Create New User
2: Route::post('/users', [UserController::class, 'store']);
```

Primjer koda 4.8 Ruta (engl. Route) za funkciju store

Poziva se funkcija store u UserController-u koja je zadužena za spremanje podataka o novome korisniku. Izlistanje koda za store funkciju:

```
1: //Create New User
2: public function store(Request $request){
3: $formFields = $request->validate([
4: 'email' =>['required', 'email', Rule::unique('email')],
5: 'name' => ['required', 'min:5'],
6: 'password' => 'required|confirmed|min:6'
7: ]);

8: //Hash Password
9: $formFields['password'] = bcrypt($formFields['password']);

10: //Create User
11: $user = User::create($formFields);

12: //Login
13: auth()->login($user);

14: return redirect('/')->with('message', 'User created and logged in');
15: }
```

Primjer koda 4.9 Funkcija store

Uspješnim spremanjem podataka o novom korisniku preusmjerava se korisnika na početnu stranicu sa porukom *User created and logged in*. Druga funkcionalnost je prijava korisnika. Klikom na gumb *Login* otvara se formular za prijavu korisnika. To je omogućeno putem route-a koji poziva funkciju login. Izlistanje koda za pozivanje funkcije login:

```
1: //Show Login Form
2: Route::get('/login', [UserController::class, 'login'])->name('login')-
   >middleware('guest');
```

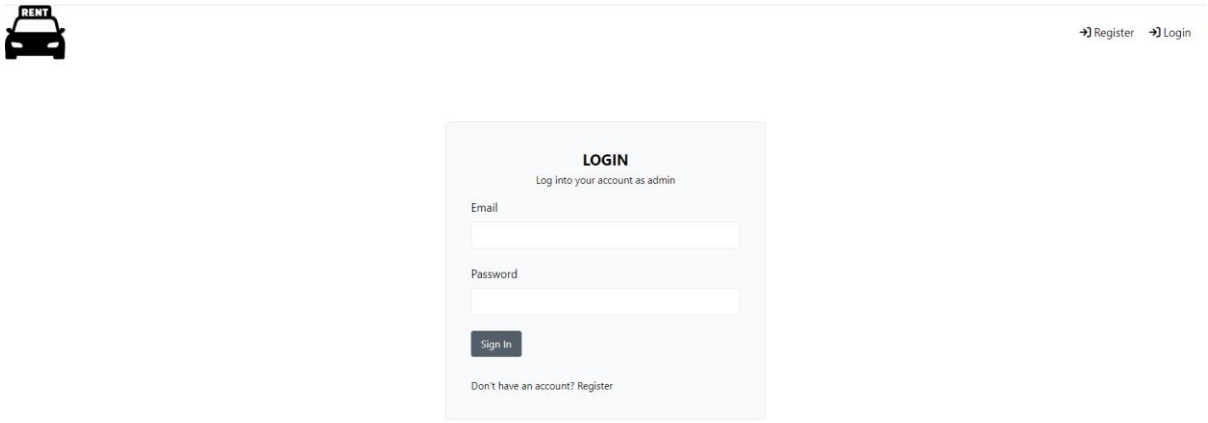
Primjer koda 4.10 Ruta (engl. Route) za funkciju login

Izlistanje koda funkcije login:

```
1: //Show Login Fomr
2: public function login(){
3: return view('users.login');
4: }
```

Primjer koda 4.11 Funkcija login

Funkcija login vraća pogled users.login. Izgled pogleda prikazan je na slici 4.6.



Sl. 4.6. Izgled login pogleda

Kada korisnik unese pravilne podatke za prijavu i klikne na dugme *Sign In* poziva se sljedeća ruta (engl. Route):

```
1: //Log in User
2: Route::post('/users/authenticate', [UserController::class,
'authenticate']);
```

Primjer koda 4.12 Ruta (engl. Route) za funkciju authenticate

Poziva se funkcija authenticate koja je zadužena za provjeru ispravnosti unesenih podataka i logiranje korisnika u sustav. Izlistanje funkcije authenticate:

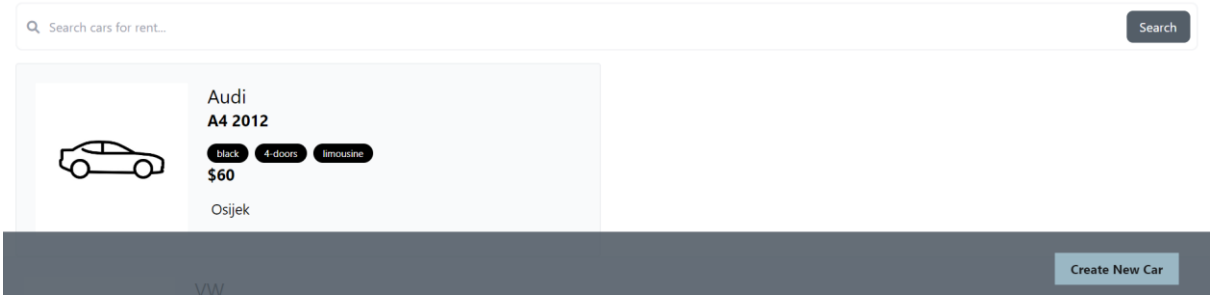
```
1: public function authenticate(Request $request){
2: $formFields = $request->validate([
3: 'email' =>['required', 'email'],
4: 'password' => 'required'
5: ]);
6:
7: if(auth()->attempt($formFields)){
8: $request->session()->regenerate();
9:
10:  return redirect('/')->with('message', 'You are now logged in');
11:  }
12:
13:  return back()->withErrors(['email' => 'Invalid Credentials']);
14:  }
```

Primjer koda 4.13 Funkcija authenticate

Ako je prijava bila uspješna korisnika se vraća na početnu stranicu aplikacije s porukom *You are now logged in.*

4.3. Stvaranje novih oglasa

Mogućnost stvaranja novih oglasa ima samo vlasnik aplikacije, odnosno admin. Nakon prijave kao vlasnik, pojavljuje se u donjem desnom kutu stranice gumb za stvaranje novoga oglasa. Primjer kako izgleda gumb kada se vlasnik prijavi prikazan je na slici 4.7.



Sl. 4.7. Gumb za stvaranje novog oglasa

1: Nakon pritiska na gumb *Create New Car* poziva se sljedeća ruta (engl. Route):

```
2: //Show Create Form
3: Route::get('/listings/create', [ListingController::class, 'create'])-
   >middleware('auth');
```

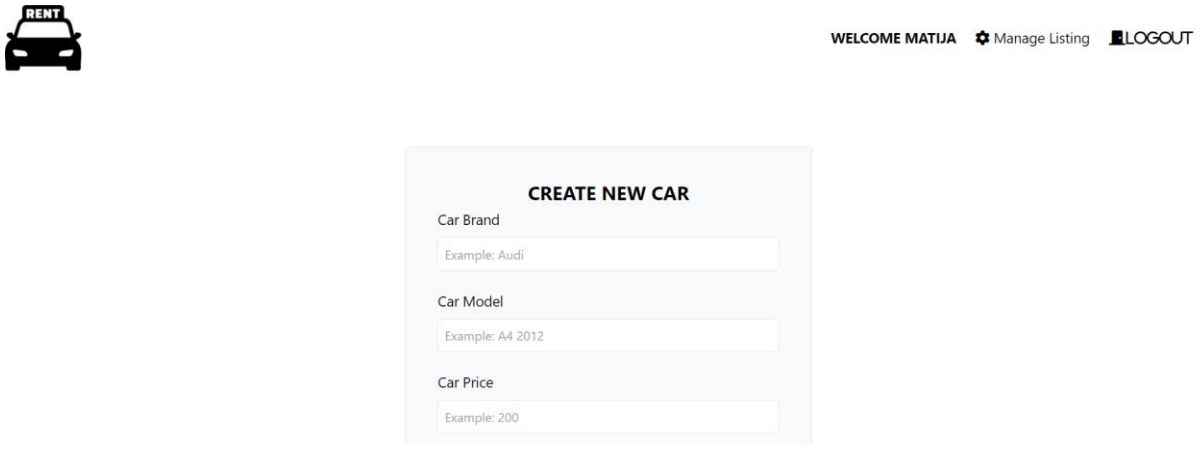
Primjer koda 4.14 Ruta (engl. Route) za create funkciju

Ruta (engl. Route) poziva funkciju create koja se nalazi u ListingController-u. Izlistanje funkcije create:

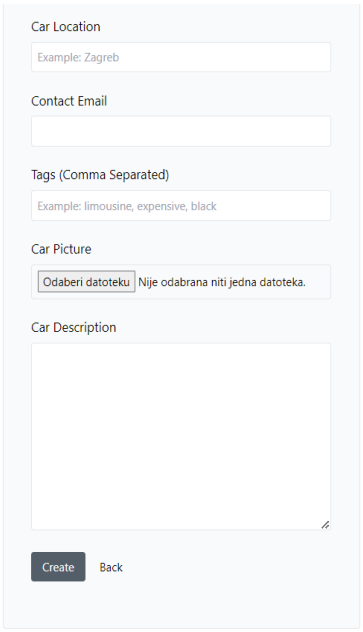
```
1: //Show create Form
2: public function create(){
       return view('listings.create');
3: }
```

Primjer koda 4.15 Funkcija create

Funkcija vraća pogled listings.create. Primjer kako izgleda taj pogled, odnosno formular za stvaranje novoga oglasa prikazan je na slikama 4.8 i 4.9 .



Sl. 4.8. Formular za stvaranje novog oglasa



P

Sl. 4.9. Formular za stvaranje novog oglasa

Pravilnim popunjavanjem formulara i pritiskom na gumb *Create* poziva se sljedeća ruta (engl. Route):

```
1: //Store Listing Data
2: Route::post('/listings', [ListingController::class, 'store'])-
   >middleware('auth');
```

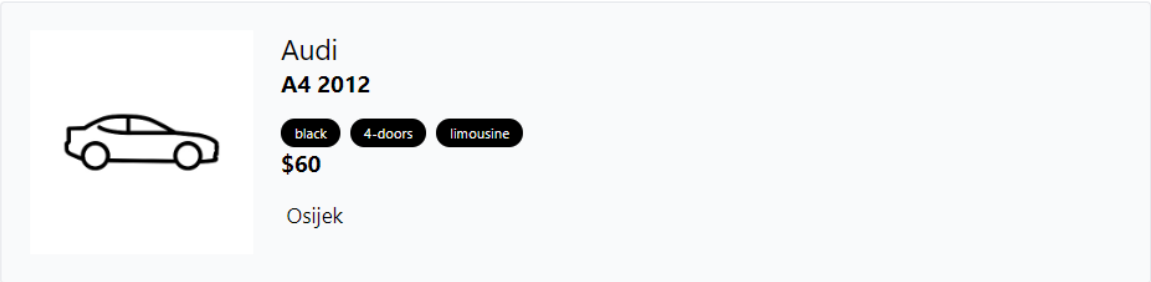
Primjer koda 4.16 Ruta (engl. Route) za store funkciju

Ruta (engl. Route) poziva funkciju store koja se nalazi u ListingController-u. Izlistanje koda za funkciju store:

```
1: //Store Listing Data
2: public function store(Request $request){
3: $formFields = $request->validate([
4: 'brand'=> 'required',
5: 'model' => 'required',
6: 'location' => 'required',
7: 'email' => ['required', 'email'],
8: 'tags' => 'required',
9: 'price' => 'required',
10:  'description' => 'required'
11:  ]);

12:  if($request->hasFile('logo')){
13:  $formFields['logo'] = $request->file('logo')->store('logos','public');
14:  }

15:  $formFields['user_id'] = auth()->id();

16:  Listing::create($formFields);

17:  return redirect('/')->with('message', 'Car created!');
18:  }
```

Primjer koda 4.17 Funkcija store

Ako je oglas uspješno kreiran, vlasnika se vraća na početnu stranicu uz poruku *Car created!* Unos datoteke za sliku je neobvezan, ako se ne unese, na oglasu će se prikazati unaprijed određena slika za sve oglase bez unesene slike. Primjer kako izgleda oglas bez unesene slike prikazan je na slici 4.10.



Sl. 4.10. Oglas bez unesene slike

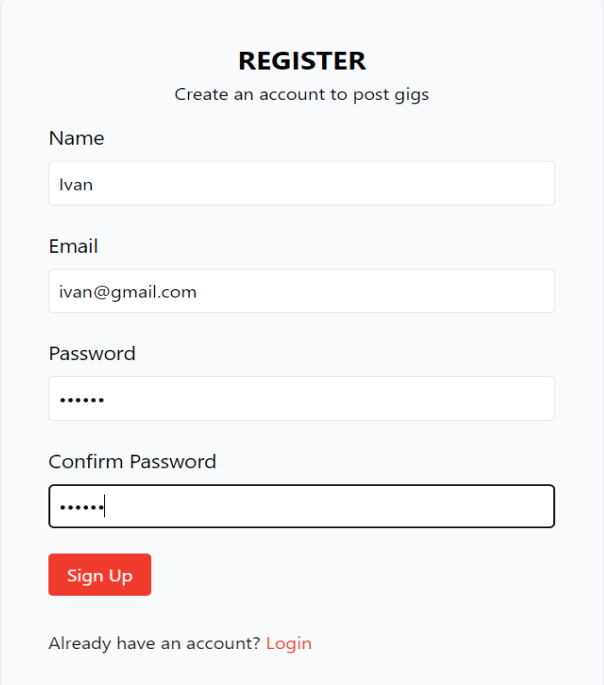
4.4. Spremanje korisnika i oglasa u bazu podataka

Do sada u poglavlju 4.2 i 4.3 su prikazane funkcionalnosti kako se stvaraju novi, odnosno prijavljuju postojeći korisnici te kako se stvaraju novi oglasi za aute. U ovome poglavlju će biti objašnjeno kako se baza podataka povezuje s tim funkcionalnostima. Naredba koja stvara novog korisnika je:

```
1: //Create User
2: $user = User::create($formFields);
```

Primjer koda 4.18 Naredba create, stvaranje korisnika

Modelu User se predaju podaci o korisniku koji su spremljeni u varijabli $formFields. U nastavku će biti prikazan popunjen formular za stvaranje novog korisnika te kako to izgleda u bazi podataka. Primjer popunjenog formulara je prikazan na slici 4.11.



Sl. 4.11. Popunjen formular za registraciju korisnika

Kada je korisnik stvoren, on se pojavljuje u bazi podataka. Primjer kako izgleda dodani korisnik u bazi podataka prikazan je na slici 4.12.

id	name	email	email_verified_at	password	remember_token	created_at
1	Admin	matija.baric1@gmail.com	2022-09-08 17:38:45	10ksH4kDKC/XHmcmkUf8/Ahumf0oQqjaSSxs...	MDQ7AEcaoE	2022-09-08 17:38
2	Matija	admin@gmail.com	NULL	$2y$10$1DatjY5RFutTNnKEa2U.WOk3T2.0486...	NULL	2022-09-08 17:39
3	Gabrijel	gabrije@gmail.com	NULL	$2y$10$ZA9XvEu6aXIlR3GjjwpwebxQO75YiMjj...	NULL	2022-09-08 20:02
4	Ivan	ivan@gmail.com	NULL	$2y$10$ksH4kDKC/XHmcmkUf8/Ahumf0oQqjaS...	NULL	2022-09-11 14:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

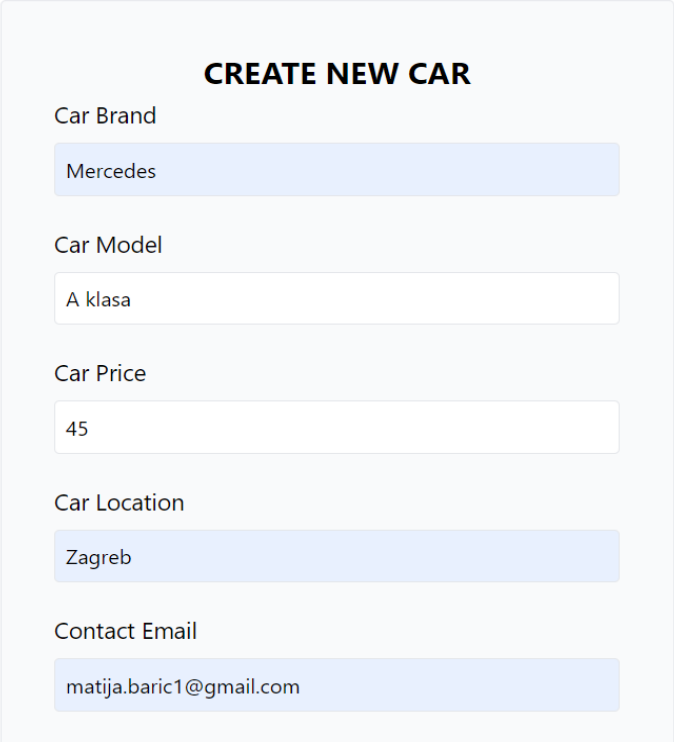
Sl. 4.12. Novi korisnik u bazi podataka

Stvaranje novog oglasa omogućeno je sljedećom naredbom:

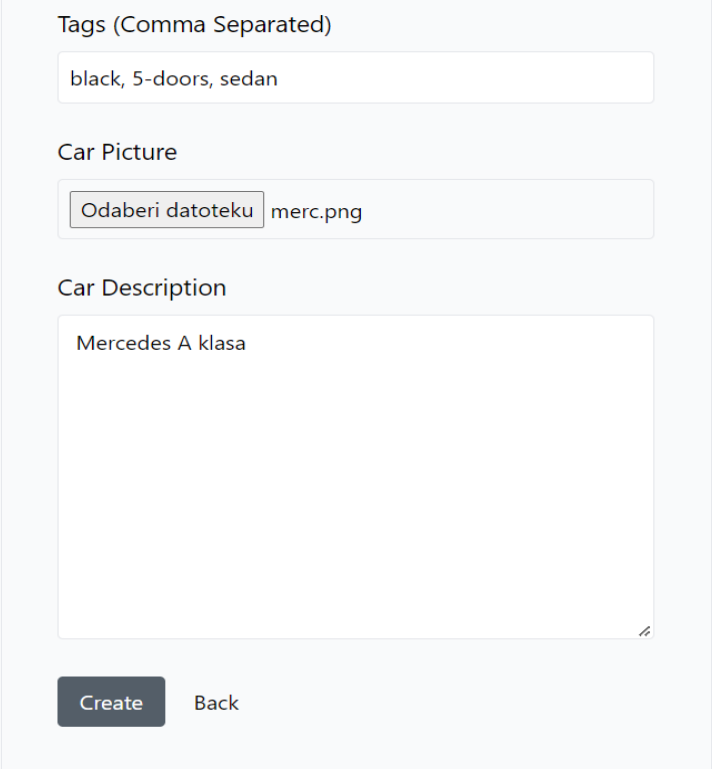
```
1: //Create listing
2: Listing::create($formFields);
```

Primjer koda 4.19 Naredba create, stvaranje oglasa

Modelu Listing se predaju podaci spremljeni u varijabli $formfields. U nastavku će biti prikazan popunjen popunjen formular za stvaranje novog auta te kako on izgleda u bazi podataka. Primjer popunjenog formulara prikazan je na slikam 4.13 i 4.14.

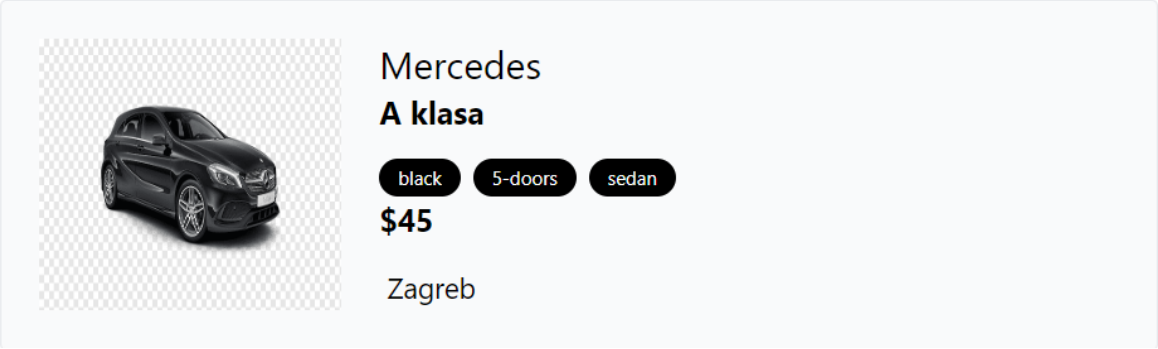


Sl. 4.13. Popunjen formular za kreiranje novog oglasa za auto



Sl. 4.14. Popunjen formular za kreiranje novog oglasa za auto

Kada je auto stvoren, spremljen je u bazu podatak. Primjer kako izgleda oglas na početnoj stranici prikazan je na slici 4.15, a kako izgleda u bazi podataka na slici 4.16.



Sl. 4.15. Novi oglas za auto na početnoj stranici

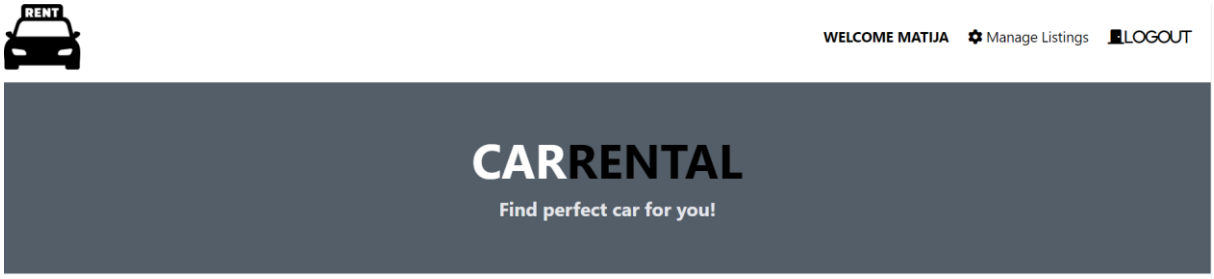
id	user_id	brand	logo	tags
1	1	VW	NULL	SUV, 5-doors, white
2	2	Audi	NULL	black, 4-doors, limousine
3	2	Mercedes	logos/eOOugvxItNOD76eoOB7jxM1Imqj0W0OL...	black, 5-doors, sedan
NULL	NULL	NULL	NULL	NULL

model	location	price	email	description	created_at	updated_at
Tiguan	Osijek	50	matija.baric1@gmail.com	VW.....	2022-09-08 17:38:45	2022-09-08 17:38:45
A4 2012	Osijek	60	matija.baric1@gmail.com	Asdas	2022-09-08 17:39:25	2022-09-10 11:04:10
A klasa	Zagreb	45	matija.baric1@gmail.com	Mercedes A klasa	2022-09-11 14:32:57	2022-09-11 14:32:57
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Sl. 4.16. Novi oglas u bazi podataka

4.5. Upravljanje, uređivanje i brisanje oglasa

Pristup upravljanju, uređivanju i birsanju oglasa ima vlasnik, odnosno administrator aplikacije. Prilikom prijave kao administrator u gornjem desnom kutu, administratoru se pojavljuje gumb *Manage Listings*. Primjer gumba prikazan je na slici 4.17.



Sl. 4.17. Gumb za upravljanje oglasima

Pritiskom na taj gumb poziva se sljedeća ruta (engl. Route):

```
1: //Manage Listings
2: Route::get('/listings/manage', [ListingController::class, 'manage'])-
   >middleware('auth');
```

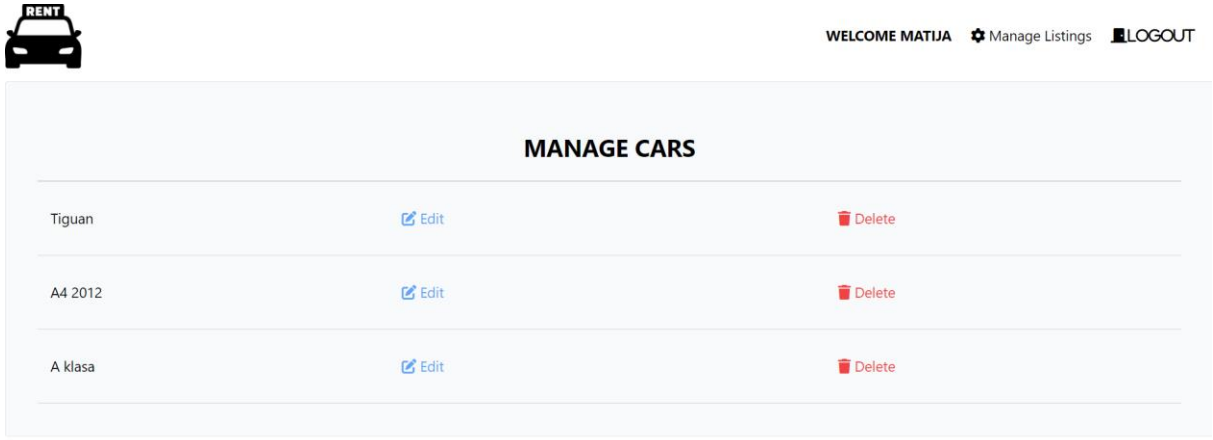
Primjer koda 4.20 Ruta (engl. Route) za manage funkciju

Ruta (engl. Route) poziva funkciju manage koja se nalazi u ListingController-u. Izlistanje koda za funkciju manage:

```
1: //Manage Listings
2: public function manage(){
3: return view('listings.manage',['listings' => auth()->user()-
   >listings()->get()]);
4: }
```

Primjer koda 4.21 Funkcija manage

Funkcija vraća pogled listings.manage. Primjer listings.manage pogleda prikazan je na slici 4.18.



Sl. 4.18. Gumb za upravljanje oglasima

U ovome pogledu imamo dvije funkcionalnosti. Prva je da vlasnik može uređivati oglas pritiskom na gumb *Edit*. Pritiskom na taj gumb poziva se sljedeća ruta (engl. Route):

```
1: //Show EDIT Form
2: Route::get('/listings/{listing}/edit',[ListingController::class,
   'edit'])->middleware('auth');
```

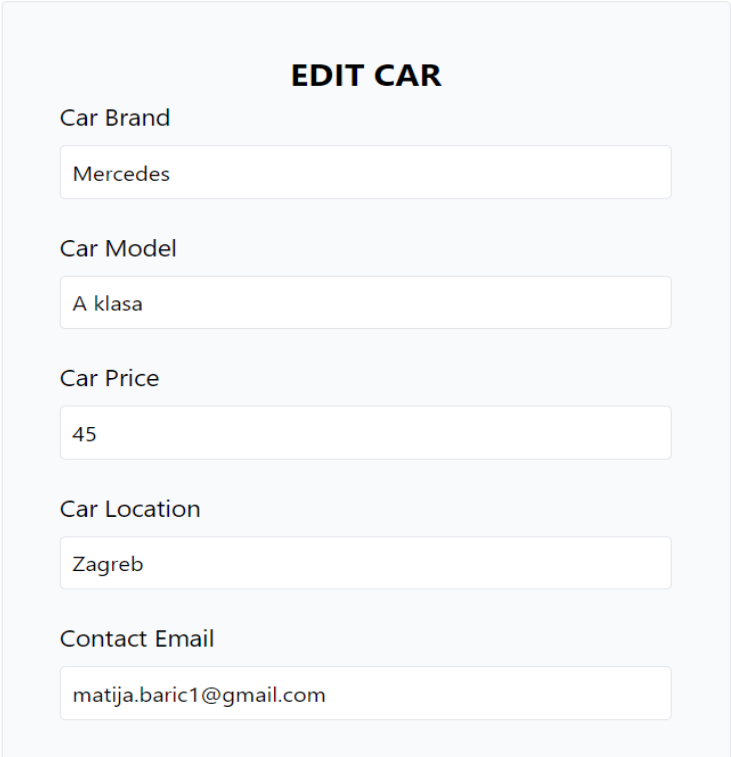
Primjer koda 4.22 Ruta (engl. Route) za edit funkciju

Ruta (engl. Route) poziva funkciju edit koja se nalazi u ListingController-u. Izlistanje koda funkcije edit:

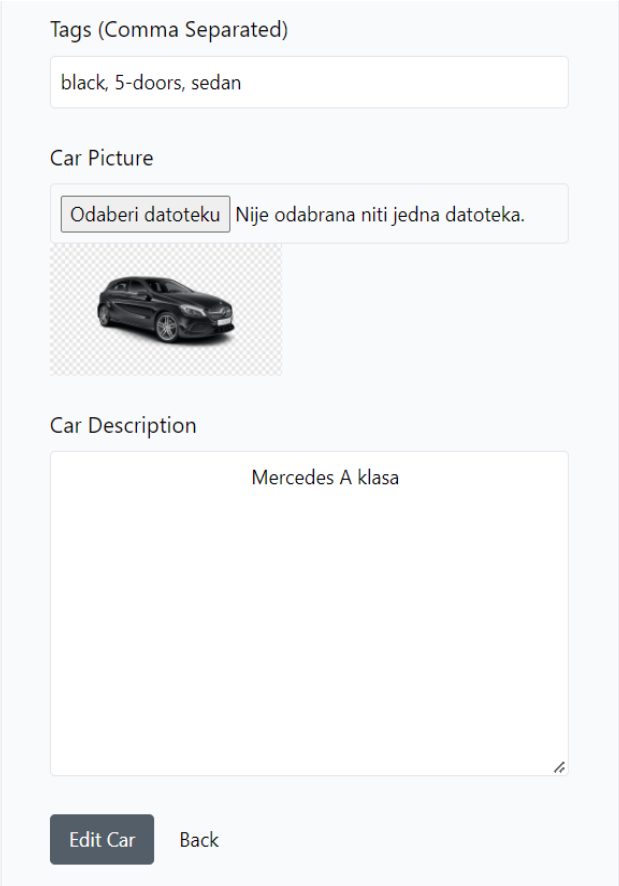
```
1: public function edit(Listing $listing){
        return view('listings.edit', ['listing' => $listing]);
2: }
```

Primjer koda 4.23 Funkcija edit

Funkcija vraća pogled listings.edit za određeni oglas. Primjer kako izgleda listings.edit pogled za auto koji je prethodno dodan prikazan je na slikama 4.19 i 4.20.



Sl. 4.19. Formular za uređivanje postojećg oglasa

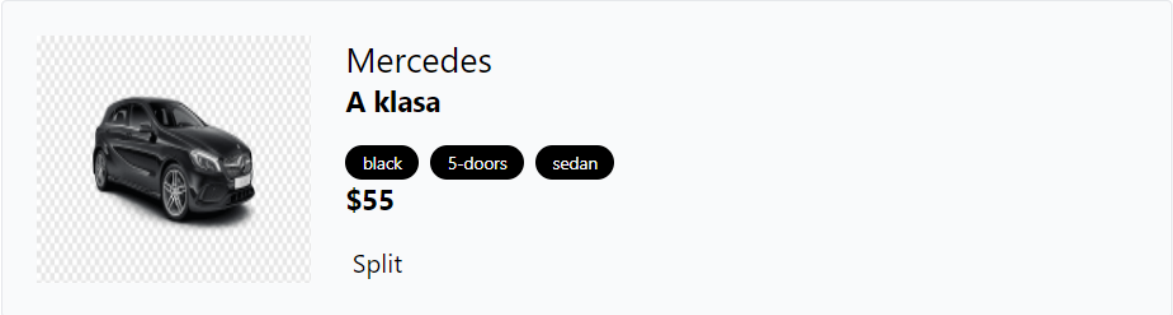


Sl. 4.20. Formular za uređivanje postojećg oglasa

Kako bi se vidjela funkcionalnost uređivanja oglasa, prethodno dodanom oglasu, promjenit će se cijena i lokacija auta. Trenutna cijena je 45, nova će biti 55. Trenutna lokacija je Zagreb, nova će biti Split. Primjer izmjene oglasa prikazan je na slici 4.21.

Sl. 4.21. Formular za uređivanje postojećg oglasa

Klikom na gumb *Edit* oglas će se izmijeniti. Izgled izmijenjenoga oglasa prikazan je na slici 4.22.



Sl. 4.22. Izmjenjeni oglas

Druga funkcionalnost je brisanje oglasa. Klikom na gumb *Delete* poziva se sljedeća ruta (engl. Route):

```
1: //Delete Listing
2:Route::delete('/listings/{listing}',[ListingController::class,'destroy']
)->middleware('auth');
```

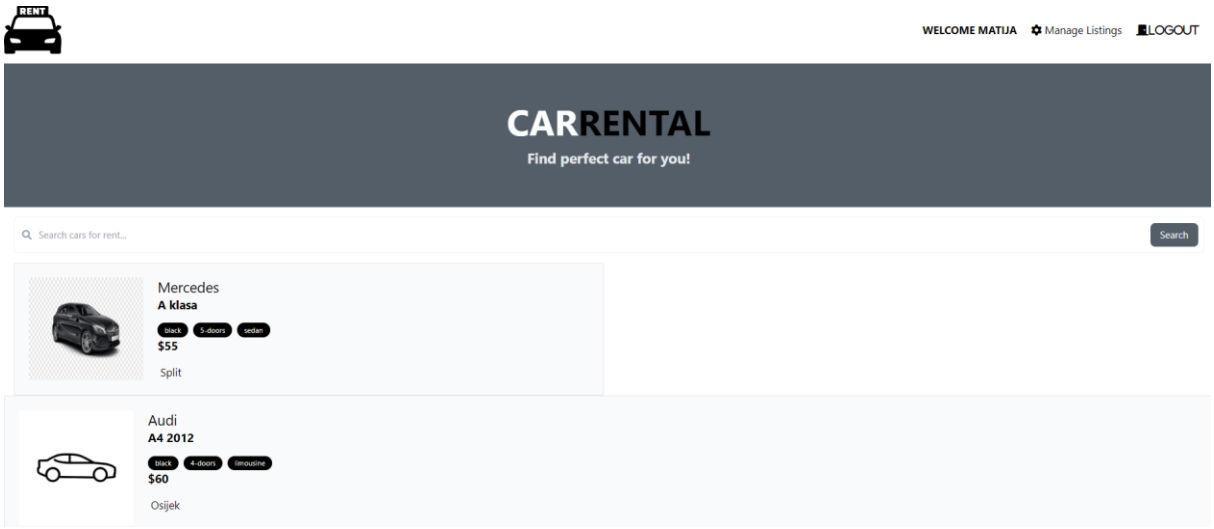
Primjer koda 4.24 Ruta (engl. Route) za destroy funkciju

Ruta (engl. Route) poziva funkciju destroy koja se nalazi u ListingController-u. Izlistanje koda za funkciju destroy:

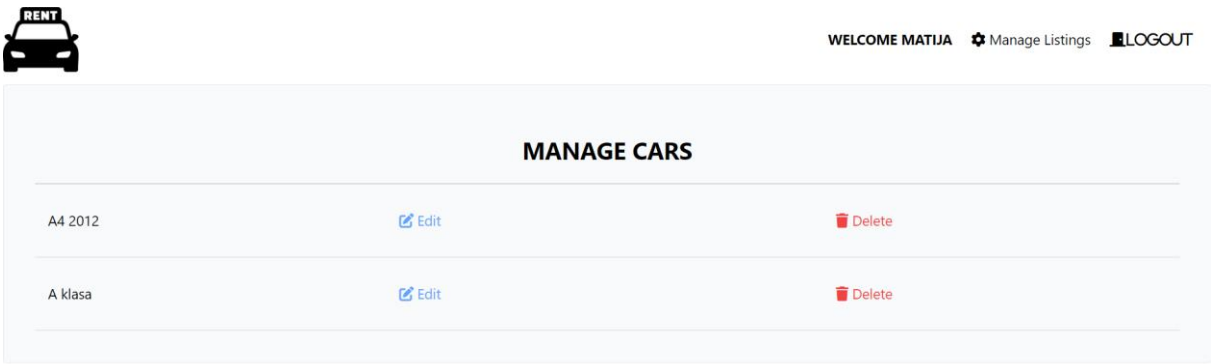
```
1: //Delete Listing
2:     public function destroy(Listing $listing){
3:         if($listing->user_id != auth()->id()){
4:             abort(403, 'Unauthorized Action');
5:         }
6:         $listing->delete();
7:         return redirect('/')->with('message', 'Car Deleted
   Succesfully');
8:     }
```

Primjer koda 4.25 Funkcija destroy

Nakon pozivanja funkcije, oglas se briše te nije više vidljiv na početnoj stranici aplikacije. Izgled početne stranice te manage pogleda kada je obrisan oglas za VW Tiguan-a prikazan je na slikama 4.23 i 4.24.



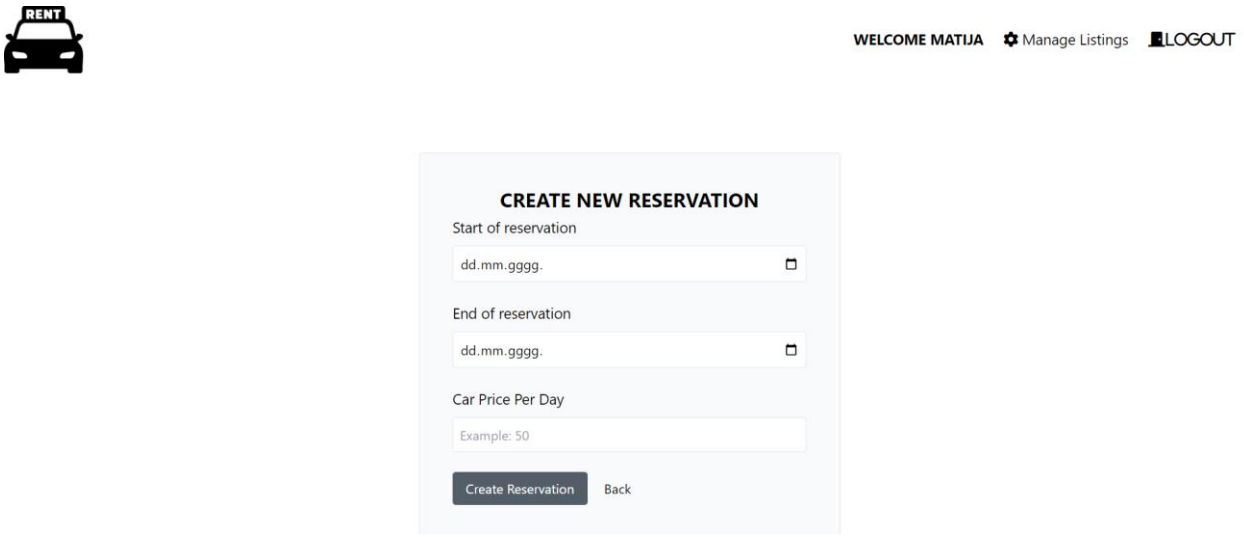
Sl. 4.23. Izgled početne stranice nakon obrisanog oglasa



Sl. 4.24. Izgled manage pogleda nakon obrisanog oglasa

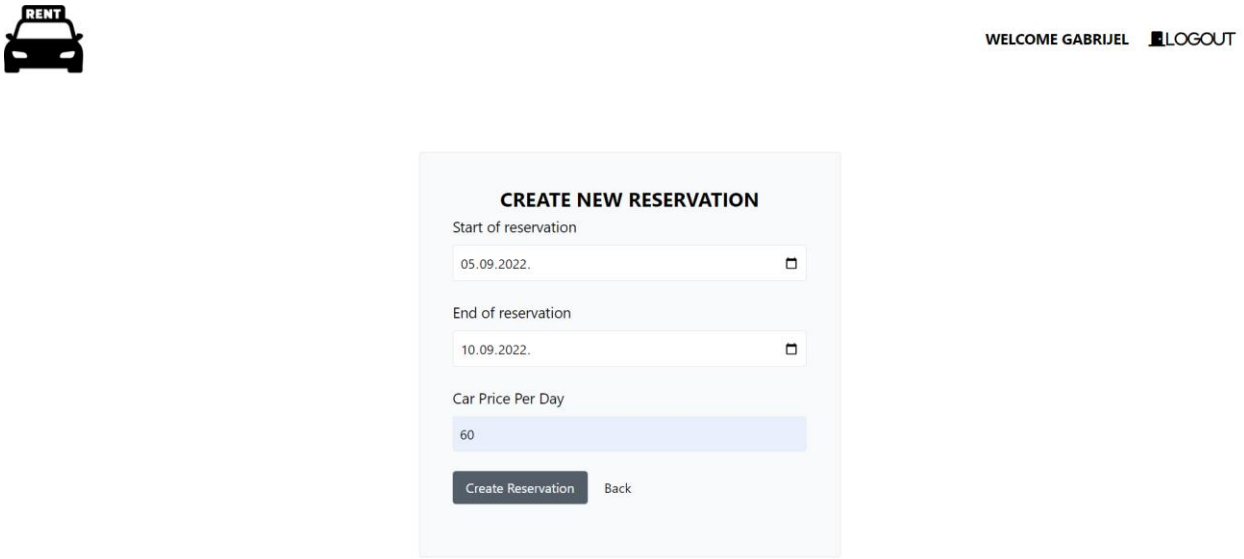
4.6. Kreiranje, spremanje i prikazivanje rezervacija

Svaki prijavljeni korisnik može kreirati rezervaciju za svaki dostupni auto. Kreiranju rezervacije i pregledu svih ostalih koje je taj korisnik napravio se pristupa iz detaljnijeg prikaza oglasa za određeni automobil. Korisnik ne može vidjeti rezervacije koje je napravio neki drugi korisnik dok vlasnik odnosno administrator može vidjeti sve rezervacije. Primjer kako izgleda formular za kreiranje nove rezervacije prikazan je na slici 4.25.

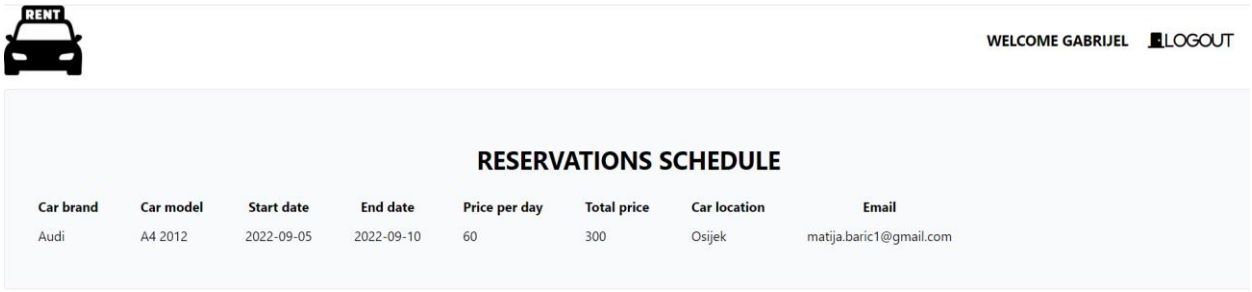


Sl. 4.25. Formular za stvaranje rezervacije

Pravilnim popunjavanjem formulara stvaramo rezervaciju. Primjer popunjenog formulara prikazan je na slici 4.26, a primjer rezervacije u rasporedu rezervacija je prikazan na slici 4.27.

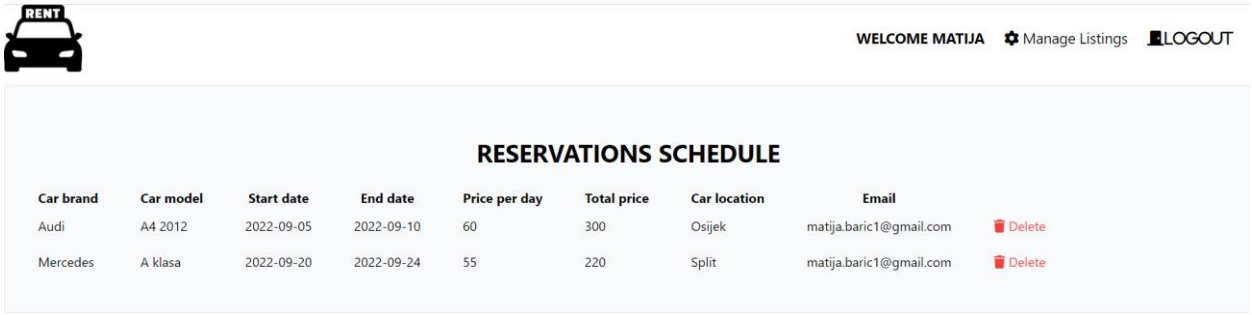


Sl. 4.26. Popunjeni formular za stvaranje rezervacije



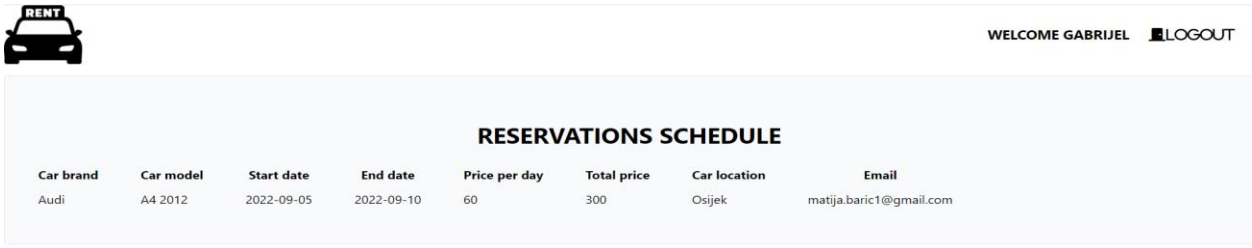
Sl. 4.27. Prikaz prethodno kreirane rezervacije u rasporedu rezervacija

Vlasnik aplikacije, odnosno administrator, također može kreirati rezervacije na isti način. Za sljedeći primjer biti će dodana još jedna rezervacija za Mercedes A klasu, ali od strane administratora, te će raspored biti prikazan na slici 4.28.



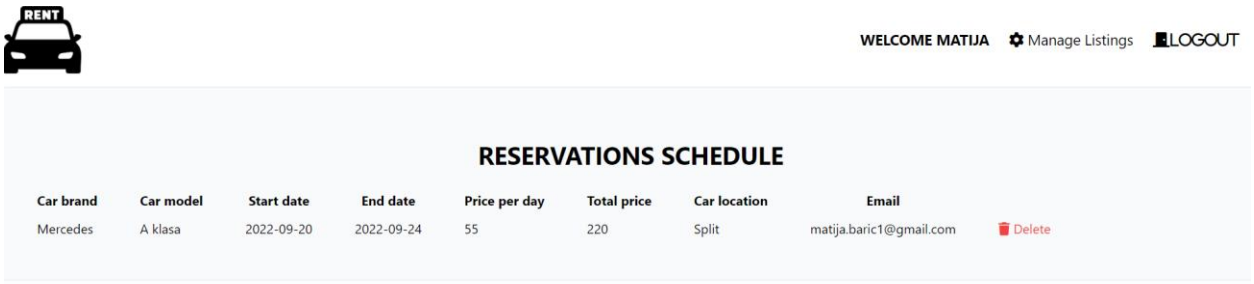
Sl. 4.28. Prikaz kreirane rezervacije od strane administratora u rasporedu rezervacija

Razlika između ova dva rasporeda je prvenstveno u korisniku koji je prijavljen. Gabrijel je obični korisnik, dok je Matija registriran kao administrator. Administrator u desnoj strani ima gumb za brisanje pojedine rezervacije, ali prije prikaza funkcionalnosti brisanja, biti će prikazan trenutni raspored rezervacija kada se prijavi korisnik Gabrijel. Primjer sadašnjeg rasporeda rezervacija prikazan je na slici 4.29.

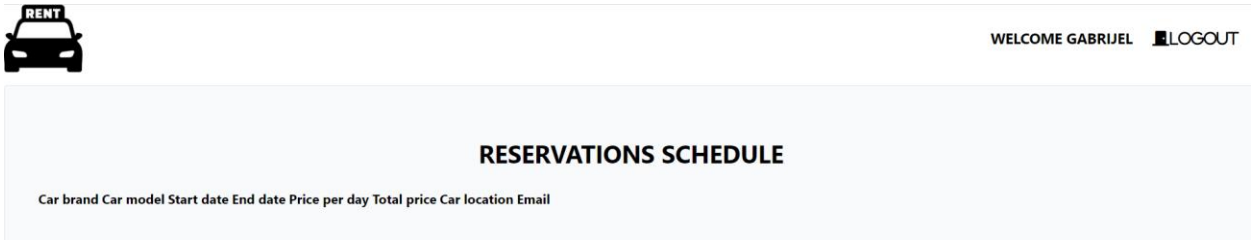


Sl. 4.29. Prikaz rasporeda rezervacija nakon kreirane administratorove rezervacije

Iz primjera na slici 4.28 se vidi da obični korisnik ne vidi rezervaciju koju je kreirao administrator. Sljedeća funkcionalnost je brisanje rezervacija. Rezervacija koju je kreirao obični korisnik, dakle rezervacija za Audi, biti će obrisana. Razlika u prikazu rasporeda kada je prijavljen obični korisnik i administrator biti će prikazana na slikama 4.30 i 4.31.



Sl. 4.30. Prikaz rasporeda rezervacija kada je administrator logiran nakon izbrisane korisnikove rezervacije



Sl. 4.31. Prikaz rasporeda rezervacija kada je korisnik logiran nakon izbrisane korisnikove rezervacije

Razlika u primjerima na slikama 4.30 i 4.31 je ta što korisnik više ne vidi svoju rezervaciju koja je obrisana, ali isto tako ne vidi rezervaciju koju je kreirao administrator.

5. ZAKLJUČAK

Korištenjem PHP-a i Laravel-a kao razvojnog okruženja stvaraju se kompleksne web aplikacije koje svakodnevno koriste milijuni ljudi za profesionalne i privatne potrebe. Takve aplikacije nude korisnicima velik broj mogućnosti, te mogu biti personalizirane prema određenim zahtjevima korisnika. Developeri se odlučuju na korištenje Laravel-a zbog njegove opsežne dokumentacije i podrške od strane zajednice koja koristi Laravel. Iz tog razloga se i ovaj rad bazira na korištenju Laravel-a i PHP-a. Kao što je već spomenuto, jedan od razloga za odabir Laravel-a je detaljnja i opsežna dokumentacija te velika zajednica korisnika. Puno toga se u kratkome vremenu može naučiti iz različitih izvora i odmah primijeniti. Pri izradi rada nisu korištene samo ove 2 tehnologije. Korištene su i tehnologije poput HTML-a, CSS-a, MySql-a i ostalih. Produkt korištenja svih tih tehnologija je potpuno funkcionalna web aplikacija. Studenti koji žele naučiti nešto više o izradi web aplikacija svakako bi trebali proučiti ove tehnologije i krenuti izrađivati svoje aplikacije.

LITERATURA

[1] “Nova Rent a Car - Car Rental in Croatia - Nova Rent a Car Croatia.” https://novarentacar.hr/?gclid=CjwKCAjw1ICZBhAzEiwAFfvFhGHnJFJqKLUP1fAf5dvPlr9oNI9TrRCleMyf1jXRSjimDmmB8RJ9wRoCnZcQAvD_BwE (pristupljeno rujan. 13, 2022).

[2] “Enterprise Rent-A-Car Hrvatska – jeftin najam automobila | Enterprise Rent-A-Car.” https://www.enterprise.hr/hr/pocetna.html (pristupljeno rujan. 13, 2022).

[3] “Rent a Car Cheap, Usporedite Cijene - Rentalcars.com.” https://www.rentalcars.com/?affiliateCode=google&preflang=hr&label=generic-sfEnBl*PeHtYXzzABLr13wS110279708315&ws=&ppc_placement=&ppc_target=&ppc_param1=&ppc_param2=&aceid=&adposition=&ppc_network=g&feeditemid=&ppc_targetid=kwd-28173230&loc_physical_ms=1028861&loc_interest_ms=&ppc_device=c&ppc_devicemodel=&gclid=Cj0KCQjw94WZBhDtARIsAKxWG-_CiYDD30fY8PibTx4d1YnMZui6YPYdiccNcYYzrQIJT76rOURhNFQaAlfNEALw_wcB (pristupljeno rujan. 14, 2022).

[4] “Rent a car Hrvatska - Split - Zagreb - Dubrovnik - Iznajmljivanje vozila | LAST MINUTE Rent a car.” https://www.rentacarlastminute.hr/hr/index.aspx (pristupljeno rujan. 14, 2022).

[5] “Što je PHP?,” NetAkademija. http://www.netakademija.hr/sto-je-php/ (pristupljeno lipanj. 28, 2022).

[6] “Uvod u PHP programski jezik.” https://www.ucionica.net/programiranje/uvod-u-php-programski-jezik-104/ (pristupljeno kolovoz. 30, 2022).

[7] “Šta je Laravel?,” PopArt Studio. https://www.popwebdesign.net/sta-je-laravel.html (pristupljeno lipanj. 28, 2022).

[8] V. Dumbovic, “RAZVOJ APLIKACIJE U LARAVEL OKRUŽENJU,” p. 66.w

[9] “Documentation for Visual Studio Code.” https://code.visualstudio.com/docs (pristupljeno kolovoz. 24, 2022).

[10] “Access SQL: osnovni koncepti, vokabular i sintaksa.” https://support.microsoft.com/hr-hr/office/access-sql-osnovni-koncepti-vokabular-i-sintaksa-444d0303-cde1-424e-9a74-e8dc3e460671 (pristupljeno kolovoz. 30, 2022).

[11] “Što je Windows cmd - komandna linija? - dir.hr.” https://dir.hr/sto-je-windows-cmd-komandna-linija/ (pristupljeno kolovoz. 30, 2022).

[12] “PHP: Downloads.” https://www.php.net/downloads.php (pristupljeno rujan. 10, 2022).

[13] “XAMPP Installers and Downloads for Apache Friends.” https://www.apachefriends.org/index.html (pristupljeno rujan. 10, 2022).

[14] “Composer.” https://getcomposer.org/download/ (pristupljeno rujan. 10, 2022).

SAŽETAK

U uvodu su izdvojene glavne značajke PHP-a kao i njegova kratka povijest. Opisan je Laravel koji se koristi kao razvojno okruženje. Visual Studio Code je uređivač koda, MySQL Workbench je program kojim se upravlja s bazom podataka. Za stvaranje aplikacije koristile su se unaprijed kreirane datoteke koje je stvorio Laravel pri kreiranju projekta, kao i naknadno kreirane datoteke. Neke od naknadno kreiranih datoteka su bili modeli podatak koji su se koristili u projektu, te pogledi koji se prikazuju u samoj web aplikaciji. U radu je prvo kreiran početni izgled aplikacije, odnosno *layout*. Nakon uključivanja dodatnih datoteka, kao što je tražilica, kreirana je baza podataka za oglase. Sukladno tome kreirana je baza podataka za korisnike te je omogućena registracija i prijava korisnika u sustav. Kako bi se stvorili oglasi, korisnicima i administratoru je omogućeno popunjavanje formulara za kreiranje oglasa. Administratoru je dodatno omogućeno uređivanje svakoga oglasa kao i njegovo brisanje što administrator odrađuje iz manage pogleda. Svaki oglas ima svoj detaljni pregled oglasa, što se postiže klikom miša na sami oglas. U detaljnome pregledu korisnici mogu vidjeti više informacija o samome automobilu te po potrebi kontaktirati vlasnika automobila. Sljedeća funkcionalnost koja je omogućena je kreiranje rezervacija za pojedini automobil. Korisniku je omogućeno da vidi samo njegove rezervacije dok administrator ima uvid u sve kreirane rezervacije i mogućnost brisanja svake od njih. Izmjena između pogleda se vrši pozivanjem rute koja prosljeđuje zahtjev odgovarajućem kontroleru, odnosno poziva funkciju koja pripada tom kontroleru. Izvršavanje neke funkcionalnosti radi na vrlo sličan način kao i izmjena pogleda. Nakon klika na gumb, zahtjev se prosljeđuje ruti koja poziva funkciju za određenu funkcionalnost.

Ključne riječi: administrator, automobil, baza podataka, datoteke, funkcionalnost, Laravel, kontroler, korisnik, oglas, PHP, pogled, projekt, rezervacije.

ABSTRACT

Title: Web application for car rent.

The introduction highlights the main features of PHP as well as its brief history. Laravel, which is used as a development environment, is described. Visual Studio Code is a code editor, MySQL Workbench is a database management program. To create the application, pre-created files created by Laravel when creating the project, as well as subsequently created files, were used. Some of the subsequently created files were data models that were used in the project, and views that are displayed in the web application itself. In this work, the initial appearance of the application, or 'layout', was first created. After including additional files, such as a search engine, the ad database is created. Accordingly, a database for users was created and user registration and login into the system was enabled. In order to create ads, users and administrators are enabled to fill in the form for creating ads. The administrator is additionally enabled to edit each ad as well as delete it, which the administrator does from the manage view. Each ad has its own detailed overview of the ad, which can be achieved by clicking the mouse on the ad itself. In the detailed overview, users can see more information about the car itself and, if necessary, contact the owner of the car. The next functionality that is enabled is the creation of reservations for an individual car. The user is allowed to see only his reservations, while the administrator has insight into all created reservations and the possibility of deleting each of them. Changing between views is done by calling Route, which forwards the request to the corresponding controller, that is, calls the function that belongs to that controller. Executing some functionality works in a very similar way to modifying a view. After the button is clicked, the request is forwarded to the Route which calls the function for the specific functionality.

Keywords: admin, car, database, files, functionality, Laravel, controller, user, ad, PHP, view, project, reservations.