Internet aplikacija za iznajmljivanje vozila

Franjkić, Filip

Undergraduate thesis / Završni rad

2017

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:200:634603

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-04**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

INTERNET APLIKACIJA ZA IZNAJMLJIVANJE VOZILA

Završni rad

Filip Franjkić

Osijek, 2017

SADRŽAJ:

1. UVOD 1
2. PREGLED TEHNOLOGIJA KORIŠTENIH U IZRADI APLIKACIJE 2
2.1. HTML jezik 2
2.2. CSS opisni jezik 2
2.3. Opis JavaScript skriptnog programskog jezika 3
2.4. PHP programski jezik 5
2.5. MySql baza podataka 6
2.6. Ngrok alat 9
3. RAZVOJ APLIKACIJE 10
3.1. Datotečna struktura 10
3.2. Baza podataka 10
3.2. Sustav za prijavu na aplikaciju 14
3.3. Naslovna stranica aplikacije 17
3.4. Ponuda i rezervacije vozila 20
3.5. Sustav naplate 22
4. ADMINISTRATORSKI DIO APLIKACIJE 24
5. ZAKLJUČAK 26
LITERATURA 27
SAŽETAK 28
ABSTRACT 28
ŽIVOTOPIS 29

1. UVOD

Razvojem tehnologije i sve većom pristupačnosti internetu javlja se potreba za razvitkom internet aplikacija. Kako bi se kompanije mogle što bolje promovirati, a korisnici na što brži i jednostavniji način informirati te pronaći željeni sadržaj, internet aplikacije se pokazuju kao vrlo pouzdano i jednostavno rješenje. Internet aplikacija za iznajmljivanje vozila će korisniku na vrlo jednostavan i pristupačan način omogućiti odabir, rezervaciju i plaćanje željenog vozila.

Prilikom izrade internet aplikacije korištene su razne tehnologije. Korišteni su opisni jezici, HTML (engl. *HyperText Markup Language*) i CSS (engl. *Cascading Style Sheets*), te JavaScript programski jezik kojim je omogućena funkcionalnost određenog broja elemenata korisničkog sučelja. Za izradu administratorskog dijela korišten je programski jezik PHP, dok je baza podataka izrađena pomoću SQL (engl. *Structured Query Language*) jezika za programiranje.

Prije samog početka izrade aplikacije potrebno je istražiti dostupne tehnologije koje su dostupne svima na korištenje (engl. *open source*). Uslijedila je izrada baze podataka koja je napravljena u sklopu četiri tablice. Nakon izrade baze podataka potpuna pozornost se mogla posvetiti kodiranju. Prvo se pristupilo rješenjima na strani servera. Bitno je omogućiti pristup bazi, plaćanje pomoću PayPal servisa te sigurnost svakom korisniku aplikacije. Nakon odrađenog administratorskog dijela aplikacije pristupilo se izradi rješenja na strani klijenta. Ovdje je bitno aplikaciju prilagoditi svim korisnicima, pružiti jednostavnost upravljanja te učiniti aplikaciju vizualno privlačnom korisnicima. Cilj je korisnicima u nekoliko koraka omogućiti odabir, rezervaciju i plaćanje željenog vozila.

U drugom poglavlju opisane su tehnologije korištene u izradi internet aplikacije. Treće poglavlje omogućuje pregled slijeda razvoja internet aplikacije te njenu svrhu i funkcionalnost. Četvrto poglavlje prikazuje opis administratorskog dijela aplikacije. Ovaj rad je zaključen petim poglavljem.

2. PREGLED TEHNOLOGIJA KORIŠTENIH U IZRADI APLIKACIJE

2.1. HTML jezik

HTML je osnovni, najčešće upotrebljavani jezik prilikom izrade internet stranice. HTML opisuje strukturu svake internet stranice, te je zbog toga opisni, a ne programski jezik [1]. Uporaba HTML jezika zahtjeva poznavanje sintakse, tj. određenih oznaka pomoću kojih internet preglednik prilikom interpretacije razlikuje oznake od teksta te točno zna kako treba izgledati ta internet stranica i što se na njoj treba nalaziti. HTML datoteke su tekstualne datoteke te je za njihovu izradu dovoljno imati uređivač teksta pomoću kojeg ćemo napraviti internet stranicu po vlastitoj želji te internet preglednik koji nam omogućava vizualni doživljaj kreirane stranice.

Kreiranu internet stranicu, napisanu HTML jezikom potrebno je testirati. Testiranje je moguće izvršiti pokretanjem internet stranice u jednom od internet preglednika. Preglednik dobivenu stranicu prevodi, interpretira oznake te prema HTML standardima prikazuje internet stranicu u cijelosti. Postoji nekolicina internet preglednika, a prema istraživanjima najpopularniji među njima su Mozilla Firefox i Google Chrome te se zbog toga svaka internet stranica prilagođava izvođenju na barem dva najpopularnija internet preglednika [2].

2.2. CSS opisni jezik

CSS je baš kao i HTML opisni jezik, smatramo ga skupom pravila koji nam omogućuje uređivanje i stiliziranje HTML dokumenta tj. njegovih elemenata. Iako nam HTML nudi mogućnosti uređivanja sadržaja, CSS je ipak mnogo praktičniji i popularniji jezik koji pruža nove standarde stiliziranja elemenata i moderniji dizajn. Osnovna zamisao prilikom stvaranja CSS-a bila je odvojiti sadržaj HTML dokumenta od dizajna tog dokumenta [3]. Upravo tim odvajanjem dokumenata postiže se bolja preglednost te je omogućeno povezivanje jednog CSS dokumenta sa svim datotekama sa sadržajem tj. sa svim HTML datotekama na kojima želimo imati jednako stilizirane elemente.

Kako bi CSS kod imao efekta vrlo važno ga je povezati s HTML dokumentom. Povezivanje se može izvršiti na tri načina [4]:

1. Povezivanje HTML dokumenta s vanjskim CSS dokumentom
2. Povezivanje internim skriptama
3. Umetanje CSS-a unutar odabranog HTML elementa.

Interno povezivanje se koristi samo kada želimo određeni CSS kod koristiti unutar jednog HTML dokumenta, a ne na cijelu internet stranicu, dok umetanje CSS-a unutar odabrani HTML element donosi vrlo nezgrapnu i nepreglednu internet stranicu. Stoga, povezivanje HTML dokumenta s vanjskim CSS dokumentom je opće prihvaćen i najčešće korišten način povezivanja jer nam omogućuje bolju preglednost koda.

Kada govorimo o sintaksi CSS-a, moramo naglasiti da se razlikuje od sintakse HTML-a te se sastoji od samo tri dijela [5]:

1. Selektor
2. Svojstvo
3. Vrijednost.

Selektorom se upućuje na pojedini HTML element kojem se želi mijenjati njegovo svojstvo određenim vrijednostima. Svojstvom smatramo unaprijed definiranu oznaku koja govori što se točno mijenja na tom HTML elementu, dok je vrijednost stil koji određuje izgled HTML elementa. Sintaksa CSS koda je prikazana slikom 2.1.

```
selektor {
    svojstvo: vrijednost;
}
```

Slika 2.1. *Sintaksa CSS koda*

2.3. Opis JavaScript skriptnog programskog jezika

Za razliku od HTML-a i CSS-a koji su opisni jezici, JavaScript je skriptni programski jezik koji se izvodi na strani korisnika tj. na klijentskom računalu (engl. *client side language*) [6]. Kada se kaže da je JavaScript skriptni jezik, pod tim se podrazumijeva da je njegovo izvođenje unaprijed omogućeno od strane internet preglednika te njegovo prevođenje nije potrebno (interpreter). Internet preglednik svaku liniju JavaScript jezika čita i izvršava zasebno (liniju po liniju). JavaScript je vrlo popularan jezik te se danas najviše koristi zbog svoje dinamičnosti i interaktivnosti koju pruža korisniku u doticaju s internet stranicom. JavaScript smanjuje komunikaciju i promet sa serverom te ima mogućnost provjere podataka na strani klijenta bez

doticaja sa serverom. Također, odlikuje ga velika brzina tj. korisniku istoga trenutka daje povratnu informaciju ukoliko neki od upisanih podataka nije ispravan, naknadno osvježavanje stranice nije potrebno.

JavaScript također moramo povezati s HTML dokumentom. Kako bi uspješno povezali ta dva dokumenta moramo upotrijebiti jedan od slijedećih načina [7]:

1. Dodavanjem koda unutar HTML elementa.
2. Povezivanje HTML dokumenta s vanjskom JavaScript datotekom.
3. Povezivanje internim skriptama.

JavaScript kod dodajemo unutar HTML elementa samo u onom slučaju kada želimo da ima efekta na pojedini element. Interakcija se postiže u onom trenutku kada miš postavimo na HTML element. Pokretanje JavaScript-a unutar HTML dokumenta je prikazano programskim kodom 2.2.

```
<h1 onmouseover="alert('Mouseover pokreće JavaScript kôd!')">
    web dizajn
</h1>
```

Programski kod 2.2. *Pokretanje JavaScript-a unutar HTML dokumenta*

Povezivanjem s vanjskom JavaScript datotekom pruža nam se mogućnost korištenja skripte u svim HTML dokumentima. Još jedna od prednosti odvojenih sadržaja je lakše snalaženje u sadržaju i bolja preglednost, kako HTML dokumenta, tako i skripte. Kako bi postupak uspješno funkcionirao, potrebno je unutar HTML elementa dodati svojstvo „*src*“ koje pregledniku govori na koji se JavaScript dokument treba referencirati. Postupak je objašnjen na slici 2.3.

```
<script type="text/javascript" src="skripta.js">
</script>
```

Programski kod 2.3. *Povezivanje s vanjskom skriptom*

Povezivanje internim skriptama se odvija između oznaka HTML elementa. Važno je da se unutar željenog HTML elementa umetne <*script*> element kako bi preglednik mogao prepoznati

da se radi o skriptnom jeziku. Povezivanje internim skriptama je prikazano programskim kodom 2.4.

```
<!doctype html>
<html>
<head>
<script language="javascript" type="text/javascript" >

  /* JavaScript kôd */

</script>
</head>
```

Programski kod 2.4. *Povezivanje internim skriptama*

JavaScript je uvijek u upotrebi na internet preglednicima te bi njegovim isključivanjem funkcionalnost internet preglednika bila vrlo ograničena. Kroz razne web kolačiće (engl. *web cookies*), izbornike (engl. *menu*), kalendare i sl. vidimo prisutnost JavaScript programskog jezika u svakodnevnom radu na internetu.

2.4. PHP programski jezik

PHP (engl. *Hypertext preprocessor*) je programski, skriptni jezik koji se izvodi na strani servera (engl. *server side*). Otvorenog je kôda, a to znači da tko god poželi može pristupiti izvornim PHP kodovima te ih urediti prema vlastitim afinitetima [8]. Pomoću PHP-a programer može kreirati dinamične internet stranice na serveru te na taj način pretraživač prepoznaje kako će generirati stranicu i što će prikazati klijentu. Kôd kojim je internet stranica generirana klijentu ostaje nepoznat, može vidjeti konačni oblik HTML stranice bez PHP kôda.

PHP je kôd vrlo široke primjene. Može biti korišten pri izradi malih internet stranice, ali i vrlo zahtjevnih internet aplikacija. Moguće ga je koristiti na trenutno najpopularnijim operacijskim sustavima kao što su: Linux, Unix, Microsoft Windows i dr. Popularnosti PHP-a pridonosi i to što ima podršku za najpoznatije internet poslužitelje, a to su: Apache i IIS (engl. *Intenet Information Services*). Također, postoje još neki skriptni programski jezici srodni PHP-u

kao što su ASP i JSP, no ono što PHP stavlja ispred njih je podrška prema velikom broju baza podataka. Baze podataka koje PHP programski jezik podržava su: MySql, Oracle, Access, PostgreSQL i mnoge druge [9]. Funkcionalnost PHP programskog jezika nije samo ograničena na stvaranje HTML dokumenata. PHP omogućuje stvaranje teksta, PDF datoteka te slika raznih formata (.jpg .bnp). PHP može automatski generirati navedene datoteke i pohraniti ih u datotečni sustav na način da na strani poslužitelja kreira predmemoriju koja služi dinamičkom sadržaju.

Prilikom izrade internet aplikacije za iznajmljivanje vozila PHP programski jezik je imao vrlo široku primjenu. Korišten je prilikom izrade administratorskog dijela aplikacije. Služi za povezivanje s bazom podataka, pruža mogućnosti prijave na aplikaciju, odjavljivanje, registraciju i dr.

2.5. MySql baza podataka

Baza podataka je organizirani sustav podataka. SQL (engl. *Structured Query Language*) je programski jezik koji omogućava izradu, ažuriranje i brisanje baze podataka. SQL smatramo jednim od najpopularnijih poslužitelja baze podataka zbog njegove pouzdanosti i brzine [10]. To je baza podataka kojom se vrlo lako upravlja zbog toga što se lako uči, a razlog tomu je što se koriste riječi koje su standardni dio engleskog jezika. Otvorenog je koda, pokreće se na strani poslužitelja te može sadržavati veći broj zasebnih baza podataka. Naredbe SQL jezika možemo podijeliti u nekoliko grupa, točnije u njih četiri [11]:

1. Podatkovna manipulacija
2. Kontrola transakcije
3. Definicija podataka
4. Kontrola podataka.

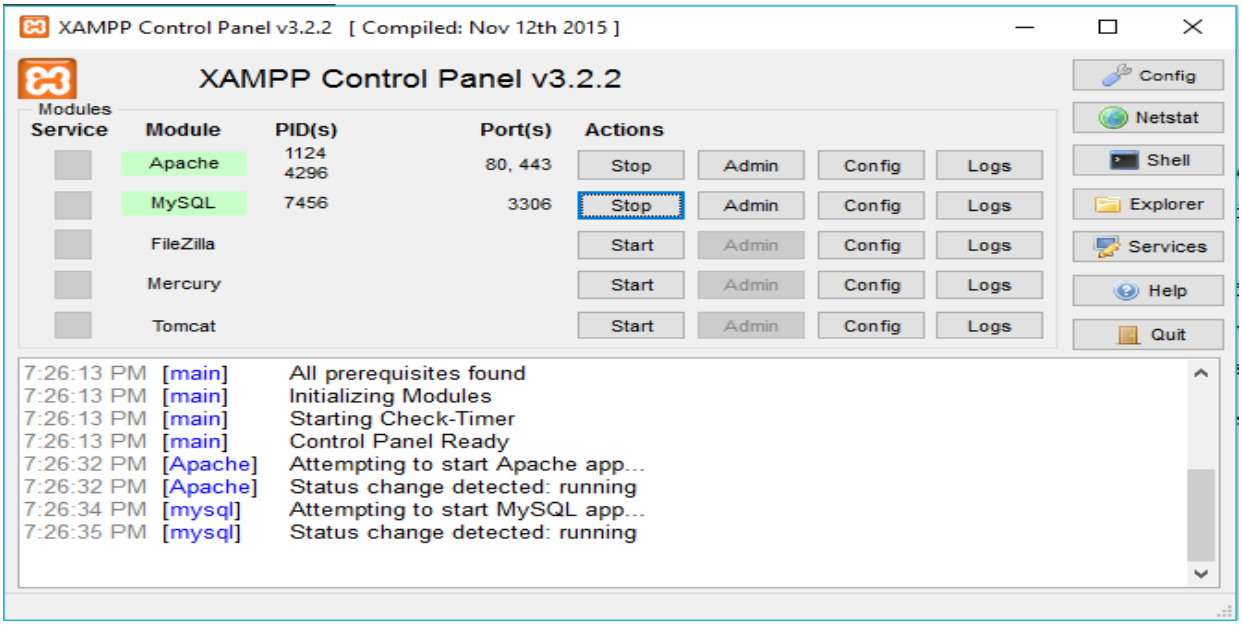
U podatkovnu manipulaciju spadaju naredbe poput: INSERT, UPDATE, DELETE i dr. Podatkovna manipulacija je najkorišteniji tip naredbe prilikom izrade baze podataka, a služi za ubacivanje, ažuriranje i brisanje podataka. Kontrola transakcije je drugi tip naredbi u SQL-u pomoću kojeg se upravlja transakcijama, a naredbe su: COMMIT, ROLLBACK i BEGIN TRANSACTION. U naredbe za definiciju podatka spadaju naredbe poput: CREATE TABLE i CREATE INDEKS. Takve naredbe definiraju strukturu podataka i omogućavaju kreiranje tablica, indeksa i dr. Naredbe kontrole podataka služe kako bi se dodijelila prava pristupa određenim

podacima u bazi podataka. U naredbe za kontrolu podataka spadaju naredbe poput: GRANT i REVOKE. GRANT služi za dodjeljivanje, a REVOKE za oduzimanje prava i privilegija nad bazom.

Također, SQL baza podataka podržava određene tipove podataka, a neki od osnovnih tipova podataka su:

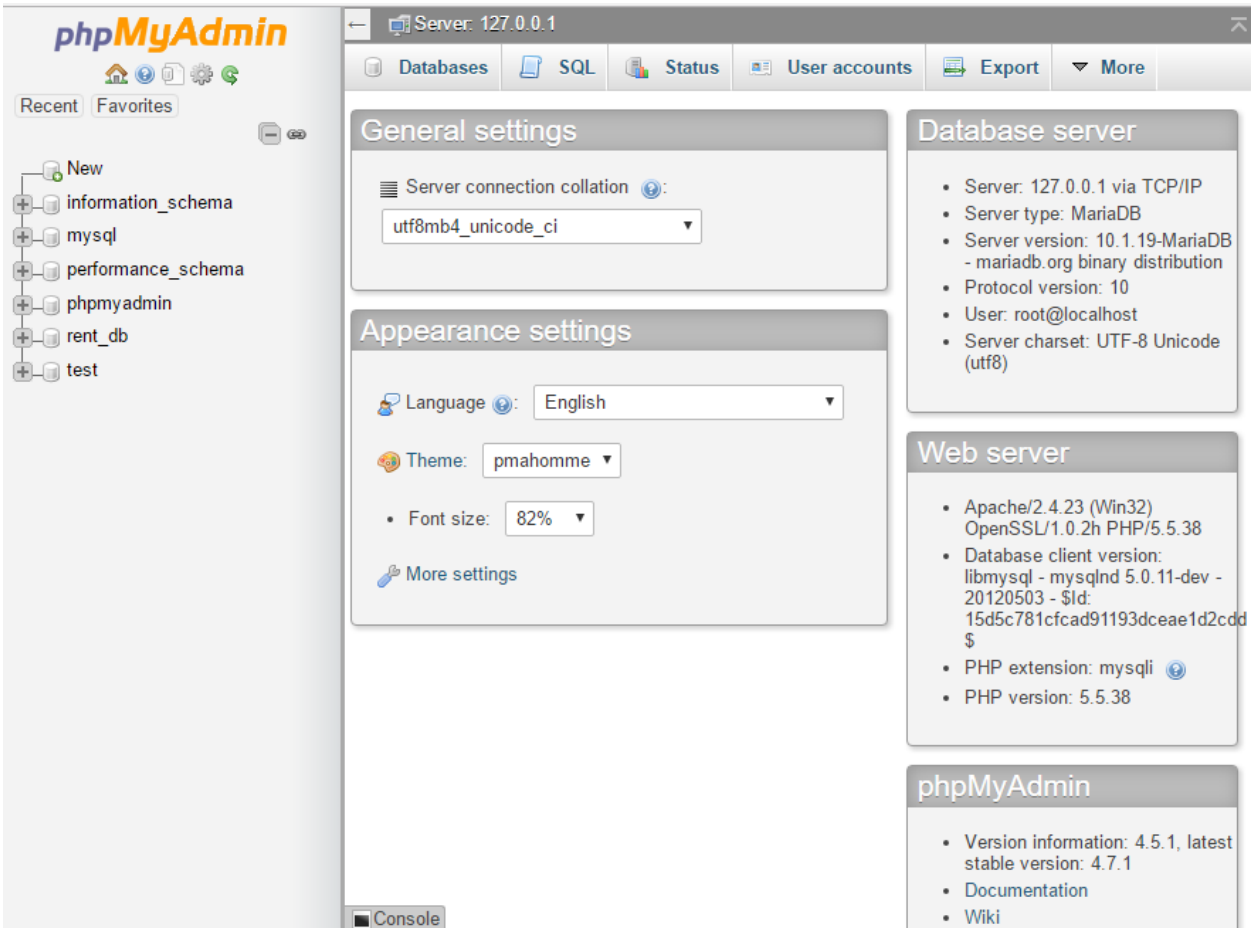
- INTEGER - koristi se za pohranu cijelih brojeva, koristeći 4 bajta memorije.
- CHAR - koristi se za pohranu znakova unutar jednostrukih navodnika.
- VARCHAR - varijabilna dužina zapisa znakova.
- DATE/TIME – za upisivanje datuma i vremena.

Kako bi aplikacija funkcionirala, potrebno je izraditi bazu podataka koja sadrži tablicu vozila koji su na raspolaganju za iznajmljivanje korisnicima. Baza podataka se izrađuje na samome početku izrade internet aplikacije, kako bi se kasnije pažnja mogla posvetiti isključivo kodiranju. Kako bi u svakom trenutku mogli kontrolirati, a naposljetku i testirati kreiranu bazu podataka, na računalo se instalira alat imena XAMPP. Pomoću XAMPP-a je omogućeno kreiranje lokalne baze podataka na vlastitom računalu. XAMPP je besplatan servis tj. platforma pomoću koje je moguće pokrenuti servise kao što su Apache i MySQL [12]. Izgled kontrolne ploče i sve mogućnosti XAMPP platforme prikazane su na slici 2.5.



Slika 2.5. *Kontrolna ploča XAMPP platforme*

Nakon što su pokrenuti servisi Apache (otvoreni internet server) i MySQL (poslužitelj baze podataka), otvara se nova mogućnost, a to je pristup kontrolno ploči baze podataka preko besplatnog alata za upravljanje MySQL-a iliti *phpmyadmin.* Phpmyadmin možemo definirati kao grafičko sučelje za upravljanje podacima MySQL baze podataka. Pristupanjem navedenom alatu moguće je uređivati bazu, umetati podatke, kreirati tablice i izvršavat ostale naredbe. Izgled alata je prikazan na slici 2.6.

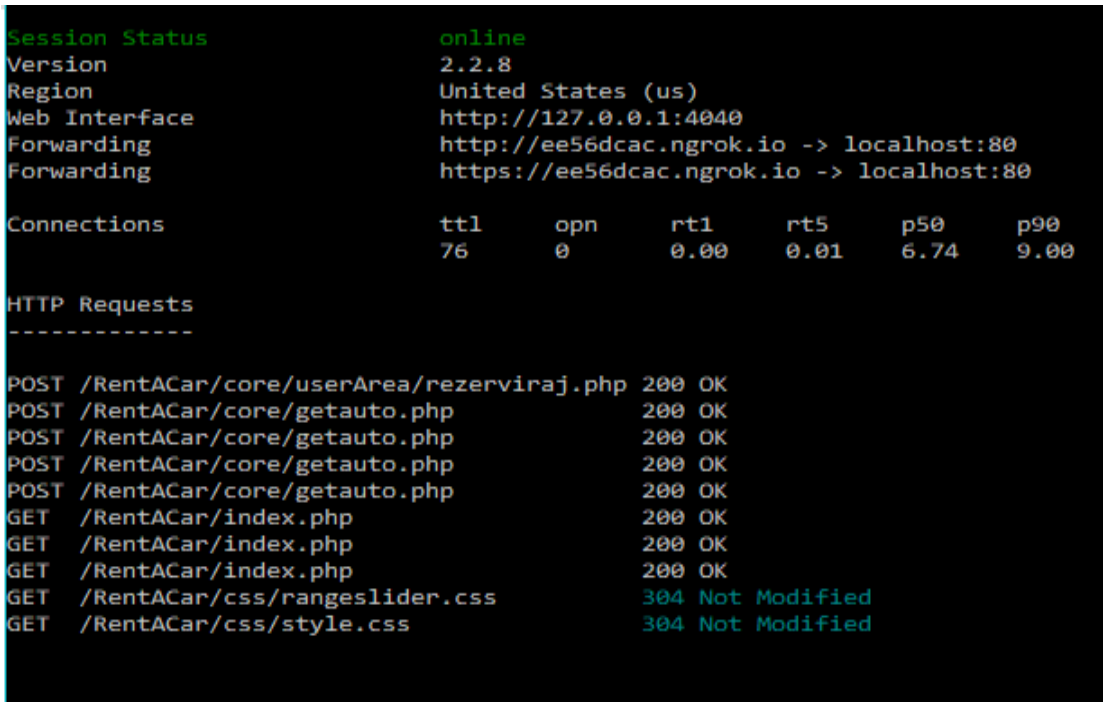


Slika 2.6. *Kontrolna ploča phpmyadmin alata*

Za pristup alatu phpmyadmin potrebno je u internet preglednik upisati *localhost/phpmyadmin.* Za rad u kontrolnoj ploči navedenog alata korisnik mora poznavati sintaksu MySQL programskog jezika

2.6. Ngrok alat

Ngrok je alat koji služi kao emulator, odnosno izlaže vlastiti lokalni server ostatku svijeta (internetu). U početku se sve radilo preko localhosta, dok nije omogućen sustav naplate uporabom PayPal-a koji je zahtijevao da aplikacija ima podignut server sa sigurnom domenom. Omogućava primanje stopostotnih potvrda o uplati novca za usluge. Alat se preuzima s interneta i instalira na računalo. Prilikom pokretanja, program zahtjeva od nas da upišemo naredbu „*ngrok.exe http 80*“. Nakon upisivanja naredbe, dobivamo adresu koji upisujemo u pretraživač umjesto dosadašnjeg *localhost*, a zatim kao i uobičajeno lokaciju i ime datoteke koju želimo pokrenuti [13]. Slika 2.7. nam prikazuje izgled Ngrok alata



Slika 2.7. *Ngrok alat*

3. RAZVOJ APLIKACIJE

3.1. Datotečna struktura

Prilikom izrade aplikacije, vrlo važno je aplikaciju podijeliti na nekoliko datoteka. Rastavljamo ju na nekoliko cjelina kako bi imali kvalitetan pregled napravljenog. To su datoteke koje razlažu skripte, stil, slike i ostale dokumente jedne od drugih kako bi jednostavnije upravljali datotečnom strukturom aplikacije i imali bolju preglednost. Aplikacija je podijeljena na četiri osnovna dijela, a to su:

1. Images – mapa *images* sadrži sve slike koje su usko vezane za aplikaciju, tj. sve slike koje aplikacija sadrži.
2. Style – u mapi *style* su pohranjeni svi CSS elementi te sve što je korišteno za uređivanje i stiliziranje aplikacije.
3. Scripts – mapa koja je vrlo važna za funkcionalnost internet aplikacije. Sadrži skripte koje su neophodne za povezivanje s bazom podataka, provjeru je li korisnik prijavljen/registriran te provjerava postoji li session, tj. postoji li prijava na aplikaciju, a da u međuvremenu nije ugašen pretraživač.
4. Sql- mapa Sql sadrži sve tablice baze podataka potrebne za funkcionalnost aplikacije.
5. Templates – sadrži sve predloške, kao što su forme za prijavu i odjavu, bočna traka, zaglavlje i dr.

Osim gore navedenih, postoji i *root* mapa, koja sadrži glavnu; *index.php* datoteku. *Root* mapom smatramo glavnu mapu, koja sadrži sve datoteke koje nisu smještene u jednu od prethodno navedenih mapa.

3.2. Baza podataka

Nakon organiziranja datotečne strukture i prikupljanja potrebnih dokumenata za izradu aplikacije, sljedeća stvar je izrada baze podataka. Vrlo važno je organizirati podatke koji će se koristiti prilikom izrade aplikacije. Baza podataka je stvorena kroz četiri tablice. Tablice su nazvane: korisnici, plaćanja, proizvodi i rezervacije.

Prva tablica je nazvana imenom „korisnici“. Navedena tablica sadrži podatke o korisnicima, a to nam prikazuje slika 3.1. U tablici su sadržani ime, prezime, adresa, korisnička imena i lozinke korisnika, email te kontakt broj korisnika. Navedeni stupci su tipa VARCHAR.

Stupac „id“ označava identifikaciju korisnika(zbog toga je postavljen primarni ključ upravo na to polje) te je uvijek tip podatka INT. Stupac „id“ je napravljen na način da mu se svakim unosom broj automatski povećava. Nadalje, stupac „razina“ određuje vrstu korisnika te je tip podatka ENUM. Razina korisnika može biti: „admin“, „*user*“ i „ne_jos_user“,

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**ime**	varchar(20)			Yes	*NULL*	
3	**prezime**	varchar(32)			Yes	*NULL*	
4	**username**	varchar(8)			No	*None*	
5	**password**	varchar(12)			No	*None*	
6	**kontakt**	varchar(255)			Yes	*NULL*	
7	**adresa**	varchar(255)			Yes	*NULL*	
8	**razina**	enum('admin', 'user', 'ne_jos_user')			No	*None*	
9	**status**	tinyint(1)			No	0	
10	**email_reg**	varchar(255)			Yes	*NULL*	
11	**email**	varchar(255)			No	*None*	

Slika 3.1. *Tablica korisnici*

Druga tablica je nazvana imenom „plaćanja“. Ova tablica sadrži podatke o naplati ponuđenih proizvoda. Tablica „plaćanja“ sadrži podatke o plaćanju proizvoda, o porezima koje administrator odredi na cijenu iznajmljivanja vozila, broju artikla, cijeni plaćanja, valuti u kojoj korisnik plaća. Stupac „*payment_id*“ označava broj transakcije te je tip podataka INT. Započinje s nulom, te se svakom transakcijom njegov broj automatski povećava. Primarni ključ je postavljen upravo na to polje. Nadalje, dolazi stupac „*item_number*“ broj artikla tj. vozila koji se iznajmljuje te mu je tip podataka VARCHAR. Stupac „*txn_id*“ označava porez koji korisnik plaća na cijenu auta te mu je tip podataka također VARCHAR. „*Payment_gross*“ daje povratnu informaciju o iznosu koji je klijent uplatio za vozilo, tip podataka je FLOAT. Float tip podataka zahtjeva postavljanje dva parametra, prvi označava ukupni broj znamenki koji se može zapisati, a drugi parametar označava koliko od tih znamenki je sačuvano za decimalne brojeve (u ovom slučaju to je 10 i 2) . Naposljetku podaci tipa VARCHAR opisuju stupce „*currency_code*“ i

„*payment_status*“. Navedeni stupci označavaju valutu kojom korisnik plaća, odnosno status uplate novca korisnik. Taj status nam služi kao povratna informacija servisa kao što je PayPal te govori je li transakcija obavljena, odbijena, na čekanju, itd. Tablica plaćanja je prikazana na slici 3.2.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
☐	1	**payment_id** 🔑	int(11)			No	*None*	AUTO_INCREMENT
☐	2	**item_number**	varchar(255)			No	*None*	
☐	3	**txn_id**	varchar(255)			No	*None*	
☐	4	**payment_gross**	float(10,2)			No	*None*	
☐	5	**currency_code**	varchar(5)			No	*None*	
☐	6	**payment_status**	varchar(255)			No	*None*	

Slika 3.2. *Tablica plaćanja*

Nakon tablice „plaćanja“, na red dolazi tablica „proizvodi“. Ona sadrži id proizvoda, ime, sliku, cijenu, snagu motora, status tog vozila (je li raspoloživ ili ne), potrošnju goriva, opis vozila te je li vozilo trenutno raspoloživo za iznajmljivanje. Stupac „id“ je označen primarnim ključem te se povećanjem proizvoda u bazi automatski povećava id za jedan, za svaki proizvod. Stupac „name“ je tipa VARCHAR i u to polje stane 255 znakova isto kao i u polje „*image*“. Stupac „*price*“ označava cijenu vozila, tip podataka FLOAT (10,2). „Status“ označava je li proizvod dostupan ili nije, stoga je tip podataka TINYINT (može biti 0 ili 1). Stupci „*horse_power*“ i „*fuel_economy*“ koriste INT tip podataka te označavaju snagu motora vozila i njegovu potrošnju goriva. „*Description*“ sadrži kratak opis vozila i dodatne karakteristike te je tip podataka VARCHAR. Posljednji stupac u ovoj tablici je „raspoloživost“ koji govori koliko je vozila pojedine marke/modela raspoloživo na lageru, za iznajmljivanje klijentima. Tip podataka je INT. Tablica proizvodi je prikazana slikom 3.3.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**name**	varchar(255)			No	*None*	
3	**image**	varchar(255)			No	*None*	
4	**price**	float(10,2)			No	*None*	
5	**status**	tinyint(1)			No	1	
6	**horsepower**	int(11)			Yes	*NULL*	
7	**fuel_economy**	int(11)			Yes	*NULL*	
8	**description**	varchar(500)			Yes	*NULL*	
9	**raspolozivost**	int(11)			No	5	

Slika 3.3. *Tablica proizvodi*

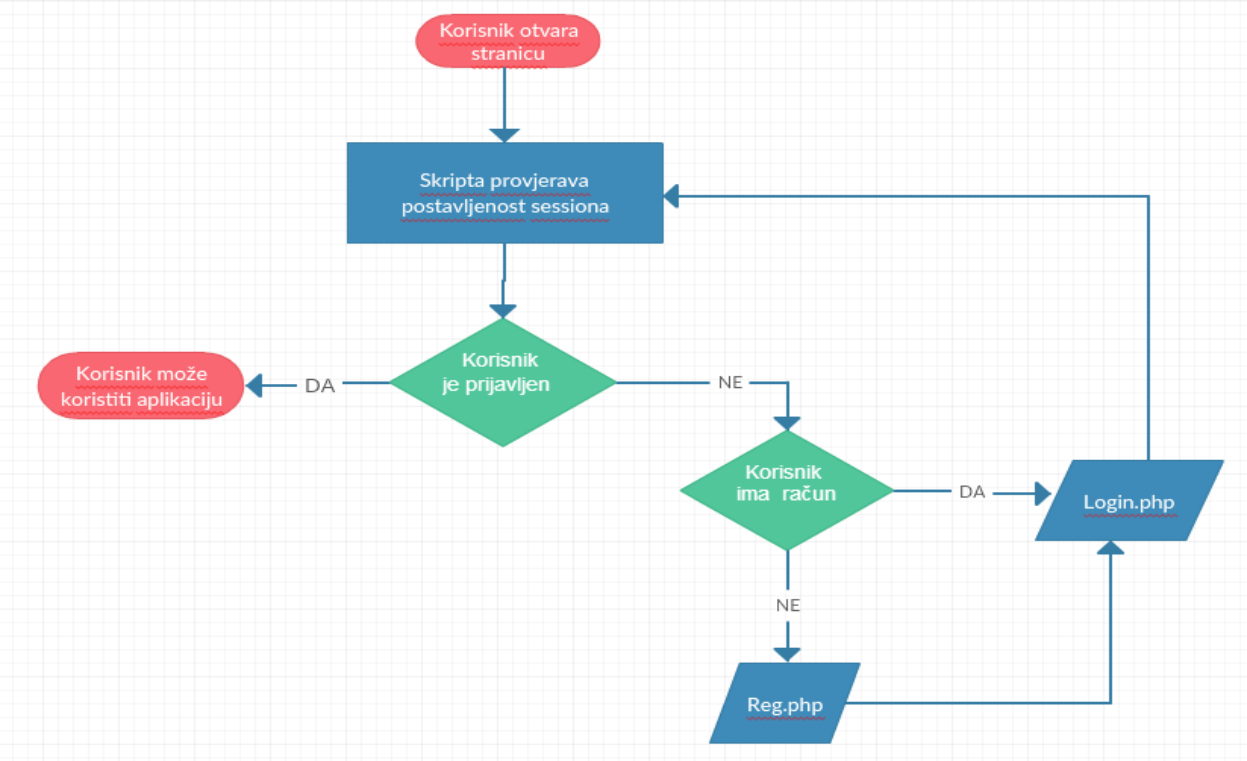
Posljednja tablica koja čini ovu bazu podataka zove se rezervacije. Ova tablica sadrži id rezervacije, datum rezervacije, id kupca, id vozila, te potvrdu rezervacije. „Id“ je označen primarnim ključem te se za svaku rezervaciju njegova vrijednost automatski povećava. Tip podataka za stupac „id“ je INT. Stupac „vrijeme_rezervacije“ označava datum pojedine rezervacije, stoga je tip podataka DATE. Stupac „potvrđeno“ označava je li administrator korisniku potvrdio rezervaciju ili ju je odbio te je zato tip podataka TINYINT (vrijednost mu je 0 ili 1). Stupci „kupac_id“ i „vozilo_id“ su označeni stranim ključevima. Stupac „kupac_id“ veže tablicu za id korisnika te omogućava da jedan kupac može imati više rezervacije, ukoliko je dostupno toliko vozila. Ovaj način je opisan relacijom „*one to many relationship*“. Također, jedan tip vozila može biti rezerviran više puta, sve dok na lageru postoji dovoljan broj traženih vozila. Tablica rezervacije je opisana slikom 3.4.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
	1	id	int(11)			No	None	AUTO_INCREMENT
	2	vrijeme_rezervacije	date			No	None	
	3	kupac_id	int(11)			No	None	
	4	vozilo_id	int(11)			No	None	
	5	potvrdeno	tinyint(1)			No	None	

Slika 3.4. *Tablica rezervacije*

3.2. Sustav za prijavu na aplikaciju

Pokretanje internet aplikacije za iznajmljivanje vozila zahtjeva prijavu korisnika na stranicu. Aplikacija je napravljena na takav način, da ukoliko korisnik želi koristiti aplikaciju mora izvršiti prijavu, odnosno registraciju ukoliko nema korisnički račun. Najjednostavniji način za prikazati, kako funkcionira sustav za prijavu na aplikaciju je preko dijagrama toka. Dijagram toka sustava za prijavu na aplikaciju je prikazan na slici 3.5.



Slika 3.5. *Dijagram toka sustava za prijavu na aplikaciju*

Iz priloženog možemo vidjeti da prilikom korisnikovog pokretanja aplikacije, dolazi do provjere postoji li prijava na aplikaciju, a da u međuvremenu nije ugašen pretraživač. Postoji li inicijalizirana sesija, znači da je korisnik prijavljen i može koristiti aplikaciju. Ukoliko inicijalizirana sesija ne postoji, korisnika se prebacuje na stranicu za prijavu. Na stranici za prijavu korisnik mora unijeti svoje podatke kako bi pristupio aplikaciji. Traženi podaci su pohranjeni u tablici „korisnici“, a postojanost korisnika se provjerava na način da skripta za prijavu šalje upit bazi podataka, postoji li navedeni korisnik u bazi. Slika 3.6. prikazuje izgled obrasca za prijavu na aplikaciju.

Slika 3.6. *Obrazac za prijavu na aplikaciju*

Ukoliko korisnik nije registriran, postoji mogućnost da se registrira. Registracija korisnika se obavlja na način da korisnik ispuni formu u kojoj mora upisati svoje podatke kao što su: ime, prezime, korisničko ime, mail i lozinku. Pritiskom na dugme „Potvrdi“ korisnik na mail dobiva link kojim potvrđuje svoje mail adresu. Pritiskom na taj link korisnik je preusmjeren na internet stranicu te na taj način dovršava proces registracije. Slika 3.7. prikazuje programski kod skripte za registraciju na aplikaciju.

```php
<?php
date_default_timezone_set('Etc/UTC'); //trenutno vrijeme - zbog gmaila
include("connect.php");
require '../vendor/phpmailer/phpmailer/PHPMailerAutoload.php';

if(isset($_POST['username']) && isset($_POST['password']) &&
isset($_POST['email']) && isset($_POST['address']) && isset($_POST['phone'])
&& isset($_POST['name']) && isset($_POST['surname'])){
    $user =mysqli_real_escape_string($db_conx, $_POST['username']);
    $pass =mysqli_real_escape_string($db_conx, $_POST['password']);
    $email=mysqli_real_escape_string($db_conx, $_POST['email']);
    $address=mysqli_real_escape_string($db_conx, $_POST['address']);
    $name=mysqli_real_escape_string($db_conx, $_POST['name']);
    $surname=mysqli_real_escape_string($db_conx, $_POST['surname']);

    $secret = $user . $pass;
    $secret = md5($secret);

    $sql = "INSERT INTO korisnici(username, password, razina, email_reg, email, kontakt, adresa, ime, prezime)
    VALUES ('$user', '$pass', 'ne_jos_user', '$secret', '$email', '$phone', '$address', '$name', '$surname') ";

    $query = mysqli_query($db_conx, $sql);

    /************************************************************************
    **********     POTVRDA REGISTRACIJE PUTEM E-MAILA    **************
    ************************************************************************/
    $mail = new PHPMailer;
    $mail->isSMTP();
    $mail->SMTPDebug = 0;
    $mail->Debugoutput = 'html';
    //postavke g-maila
    $mail->Host = 'smtp.gmail.com';
    $mail->Port = 587;
    $mail->SMTPSecure = 'tls'; //gmail odbija suradivati pod ssl-om
    $mail->SMTPAuth = true;
    $mail->Username = $mail_username;
    $mail->Password = $mail_password;

    $mail->SMTPOptions = array(
            'ssl' => array(
                'verify_peer' => false,
                'verify_peer_name' => false,
                'allow_self_signed' => true
            ));

    //headeri
    $mail->setFrom('info@rent-a-car.com', 'Autosend');
    $mail->addAddress($email, $user);
    $mail->Subject = 'Rent-a-car.com registracija';
    //html mail
    $mail->Body = "
                    <h3>$user,</h3>
                    <p>drago nam je sto ste odlučili postati članom naše zajednice. Kako biste dovršili registraciju kliknite na link ispod.</p>
                    <p><a href='$site_address/core/register.php?token=$secret'>$site_address/core/register.php?token=$secret</a></p>
                ";
```

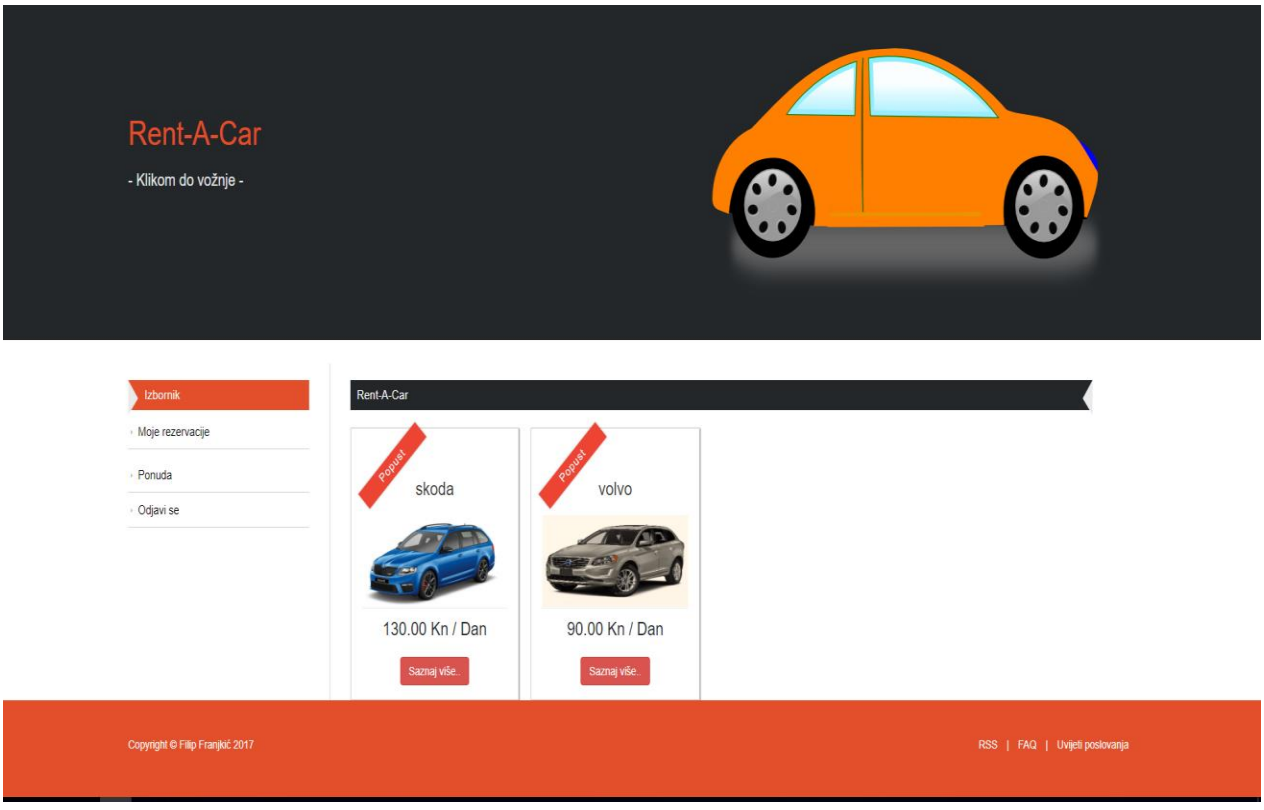
Programski kod 3.7. *Skripta za registraciju*

Nakon što se korisnik prijavio, pristupa aplikaciji te može koristiti sve njene pogodnosti. Omogućen mu je pregled ponuđenih vozila, rezervacija i plaćanje istih.

3.3. Naslovna stranica aplikacije

Nakon što se korisnik prijavi na aplikaciju, prvo što ugleda i zajedničko je svim korisnicima je naslovna stranica. Na naslovnoj stranici aplikacije se nalaze sve mogućnosti koje pruža aplikacija. Kao što je i uobičajeno, naslovna stranica je po svojoj prirodi vrlo jednostavna, kako bi se svi korisnici mogli vrlo lako služiti njome. U samom središtu stranice, kako bi korisnicima bilo što uočljivije sadrži vozila koja su korisnicima ponuđena za iznajmljivanje. S lijeve strane se nalazi navigacijska traka, tzv. *sidebar menu*. Njezina uloga je da omogući korisniku navigaciju kroz internet stranicu. Naslovna stranica je prikazana na slici 3.8. Sadrži tri kartice, a to su:

- *Moje rezervacije* – stranica koja korisnicima omogućava pregled vlastitih rezervacija vozila. Također pruža uvid u njihov status, jesu li prihvaćene ili odbijene.
- *Ponuda* – omogućava pregled ponuđenih vozila na stranici.
- *Odjavi se* – pruža korisniku završetak rada na stranici i odjavu s iste



Slika 3.8. *Izgled naslovne stranice aplikacije*

Naslovnica je praktički podijeljena na tri dijela, a to su poglavlje, tijelo i podnožje stranice. Poglavlje stranice (engl. *header*) sadrži važne informacije o poduzeću koje iznajmljuje vozila, kao

što su: ime poduzeća, logo ili nekakav moto, ili poruku klijentima. U tijelu stranice (engl. *body*) se sadrži ono najbitnije što korisniku omogućava pregled usluga i izvršavanje transakcija. Podnožje stranice (engl. *footer*), kao što je to i uobičajeno sadrži autorska prava, u ovom slučaju i autora stranice te poveznice prema određenim medijima. Programski kod 3.9 prikazuje kod poglavlja i tijela stranice, dok programski kod 3.10. prikazuje kod podnožja stranice.

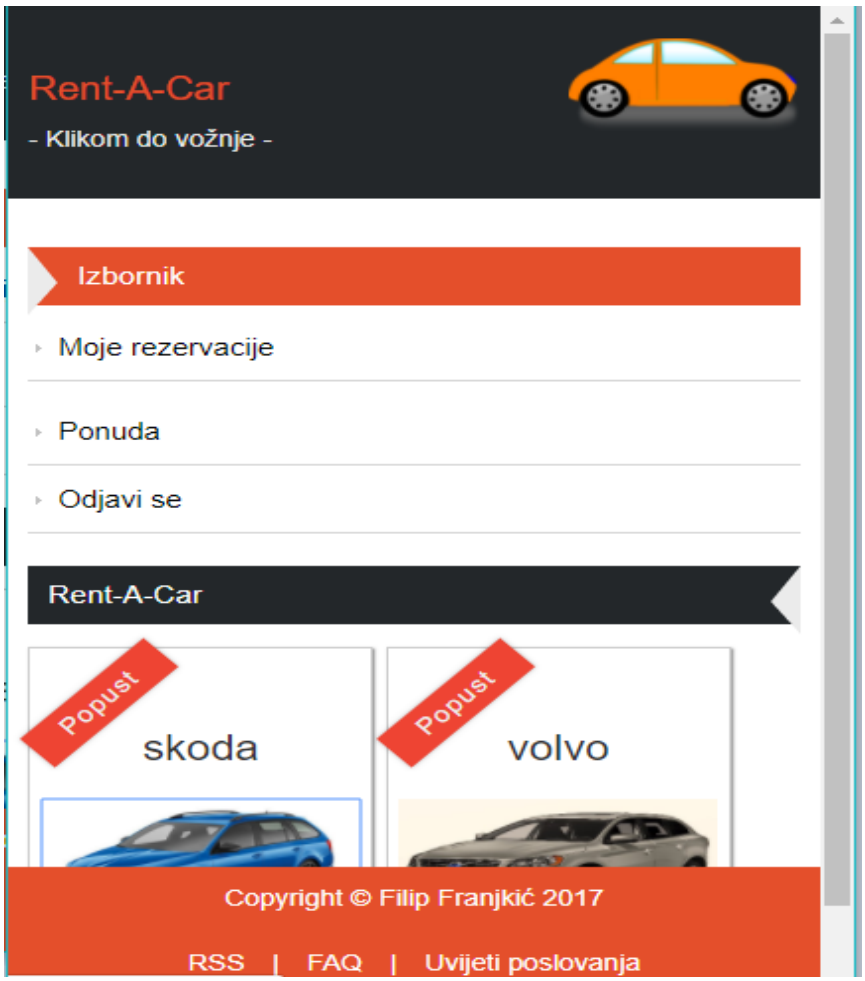
```php
<?php
include_once('core/status.php');
include_once('core/render.php');
include_once('core/getauto.php');

 ?>
<!DOCTYPE HTML>
<head>
<title>Rent-A-Car</title>
<?php include_once('templates/includes.php'); ?>
</head>
<body>
    <?php include_once('templates/header.php');  ?>
    <div class="main">
      <div class="content">
          <div class="content_bottom">
            <div class="wrap">
                <?php include_once('templates/sidebar.php'); ?>
                <div class="content-bottom-right">
                <h3>Rent-A-Car</h3>
                <div class="section group results">
                  <?php auti();  ?>
                </div>
              </div>
              <div class="clear"></div>
           </div>
        </div>
      </div>
    </div>
    <?php include_once('templates/autoDetailModal.php'); ?>
    <?php include_once('templates/loginForm.php'); ?>
    <?php include_once('templates/registerForm.php'); ?>
    <?php include_once('templates/footer.php'); ?>

    <script>
        $(document).ready(function() {
            $().UItoTop({ easingType: 'easeOutQuart' });
        });
    </script>
    <a href="#" id="toTop"> </a>
    <script type="text/javascript" src="js/navigation.js"></script>

</body>
</html>
```

Programski kod 3.9. *Prikaz koda poglavlja i tijela stranice*

```
<div class="footer navbar-fixed-bottom">
  <div class="wrap">
         <div class="copy_right">
            <p>Copyright &copy; Filip Franjkić 2017</p>
        </div>
        <div class="footer-nav">
         <ul>
            <li><a href="#" class="fa fa-facebook"></a> | </li>
            <li><a href="#" class="fa fa-google"></a></li>
         </ul>
        </div>
```

Programski kod 3.10. *Podnožje stranice*

Vrijedi napomenuti kako je ova internet aplikacija responzivnog dizajna (engl. *responsive design*). Drugim riječima, ova internet aplikacija mijenja svoj dizajn ovisno o rezoluciji i o uređaju na kojem se prikazuje. Slika 3.11. prikazuje izgled naslovne stranice na mobitelu.



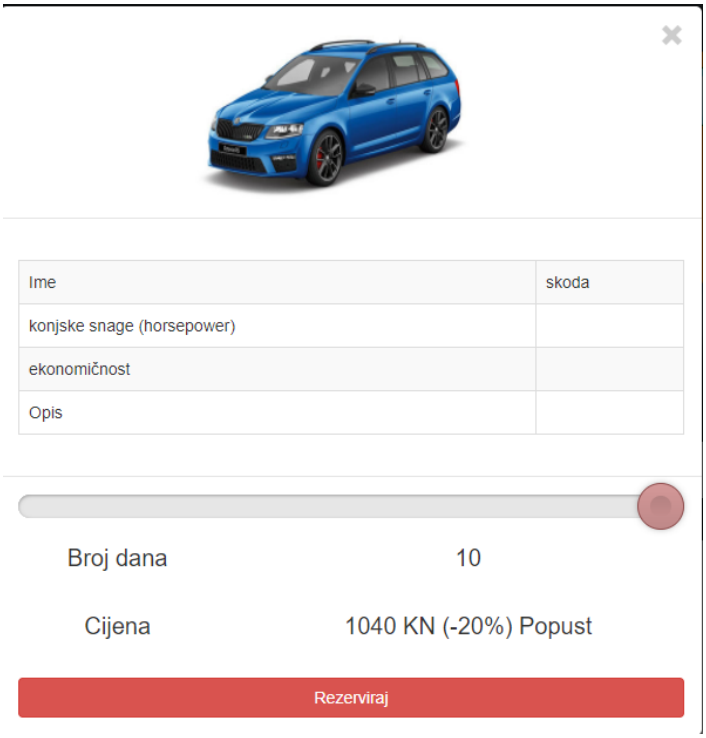
Slika 3. 11. *Responzivni dizajn aplikacije*

3.4. Ponuda i rezervacije vozila

Pritiskom na dugme „Ponuda“ korisnik se vraća na naslovnu stranicu gdje ima uvid u sva ponuđena vozila. Na toj stranici se nalaze sva ponuđena/raspoloživa vozila te je moguće saznati više detalja o njima. Pritiskom na dugme „saznaj više“, otvara se novi dijaloški okvir koji pruža više detalja o vozilu. Moguće je odabrati na koliko dana korisnik želi rezervirati vozilo te se automatski „klizanjem po traci“ uz broj dana izračunava cijena usluge. Odabirom šest ili više dana korisniku se automatski obračunava 20% popusta. Nakon što je odabrao željenu uslugu, korisnik pritiskom na tipku „Rezerviraj“ ima mogućnost plaćanja rezerviranog vozila. Aplikacija ga preusmjeri na PayPal stranicu gdje može dovršiti proces iznajmljivanja vozila. Programski kod 3.12. prikazuje kod, a slika 3.13. izgled dijaloškog okvira.

```
<input type="range" min="1" max="10" step="1"
value="100" data-orientation="horizontal">
<br>
<div class="row"   style="font-size: 20px;">
    <div class="col-sm-4 text-center">
        Broj dana
    </div>
    <div id="range-output" class="col-sm-8 text-center">
    </div>
</div>
<div class="row" style="margin-top: 30px; margin-bottom: 30px; font-size: 20px;">
    <div class="col-sm-4 text-center">
        Cijena
    </div>
    <div  class='col-sm-8 text-center' style="font-size: 20px;">
    <span  id="range-output-cijena" style="margin-bottom: 20px;">

    </span>
    <span id ="valuta" style="margin-bottom: 20px">
        KN
    </span>
    </span>
    <span id ="popust" style="margin-bottom: 20px">
    </span>
    </div>
</div>
</div>
<!-- Display the payment button. -->
                                <button type="submit" id="rezervacija" class="btn btn-danger btn-md btn-block">Rezerviraj</button>

                            </form>
```

Programski kod 3.12. *Kod dijaloškog okvira*



Slika 3.13. *Izgled dijaloškog okvira*

Kartica „Moje rezervacije“ u izborniku korisniku omogućava pogled na zatražene, odobrene i odbijene rezervacije. Prikazuje koje je vozilo rezervirano, vrijeme rezervacije te status pojedine rezervacije. Ukoliko je rezervacija odobrena, taj red će biti obojen zelenom bojom, a ako je odbijena, upit za rezervacijom će biti obrisan. Rezervacije na čekanju su obojene crvenom bojom. Nakon što korisnik izvrši transakciju, na e-mail mu dolazi potvrda rezervacije i poruka da može pokupiti vozilo ispred salona. Programski kod 3.14. prikazuje funkciju „moje rezervacije“ kojom korisnik može pratiti vlastite rezervacije vozila na aplikaciji.

```php
function mojeRezervacije()
{
    global $db_conx;
    $output = "";
    $statement = $db_conx->prepare("SELECT * FROM korisnici JOIN rezervacije ON
    rezervacije.kupac_id=korisnici.id JOIN products ON products.id=rezervacije.vozilo_id
    WHERE korisnici.id=?");
    $statement->bind_param('i', $_SESSION['userid']);
    $statement->execute();
    $result = $statement->get_result();

    $output .=
    '
    <div class="table-responsive">
        <table class="table table-inverse table-responsive">
            <thead>
                <tr>
                <th>Vozilo</th>
                <th>Vrijeme rezervacije</th>
                <th>Status</th>
                </tr>
            </thead>
            <tbody>
    ';

    while ($rezervacija = $result->fetch_array(MYSQLI_ASSOC)) {
         if ($rezervacija['potvrdeno'])
        {
            $output .= '
                <tr style="background-color: #caf7d2">
                    <td>' . $rezervacija['name'] . '</td>
                    <td>' . $rezervacija['vrijeme_rezervacije'] . '</td>
                    <td>Potvrđeno</td>
                </tr>
            ';
        }
        else {
            $output .= '
            <tr style="background-color: #f7c3c3">
                <td>' . $rezervacija['name'] . '</td>
                <td>' . $rezervacija['vrijeme_rezervacije'] . '</td>
                <td>Nije potvrđeno</td>
            </tr>
        ';
        }
    }
    $output .=
    '
    </tbody>
```

Programski kod 3.14. *Praćenje vlastitih rezervacija*

3.5. Sustav naplate

Naplata vozila se izvršava preko PayPal servisa, odnosno preko Sandboxa koji predstavlja virtualno plaćanje za testiranje internet aplikacija. Korisnik klikom na dugme pokrene proces rezervacije na način da ga se preusmjeri na PayPal server, šalju se parametri kao što je broj vozila koji je rezerviran, cijena i valuta u kojoj korisnik plaća. Sve navedeno ide na kreirani test račun. Korisnik dobiva opciju da potvrdi rezervaciju, a na testnom računu se odobrava uplata. Što se tiče

same baze, tamo se uplata ne kreira dok PayPal ne odobri (PayPal šalje potvrdu). PayPal šalje potvrdu na naš server, na skriptu *success.php* sa parametrima kao što su: broj uplate, ime onoga što je kupljeno, cijena, valuta u kojoj je korisnik platio. Kada se skripta s parametrima izvrši, onda se sprema na bazu (tablica *payments*). Važno je napomenuti kako PayPal ne šalje potvrdu na localhost, nego je potrebno imati „podignut“ vlastiti server sa sigurnom domenom. U ovom slučaju to je omogućio alat *Ngrok*. Programski kod 3.15. prikazuje skriptu koja izvršava transakciju.

```php
<?php
include("core/connect.php");
include("core/posaljiEmail.php");


$item_name = $_POST['item_name'];
$item_number = $_POST['item_number'];
$payment_status = $_POST['payment_status'];
$payment_amount = $_POST['mc_gross'];
$payment_currency = $_POST['mc_currency'];
$txn_id = $_POST['txn_id'];
$receiver_email = $_POST['receiver_email'];
$payer_email = $_POST['payer_email'];

$productResult = $db_conx->query("SELECT price FROM products WHERE id = ".$item_number);
$productRow = $productResult->fetch_assoc();
$productPrice = $productRow['price'];


if($payment_status == "Completed"){
    $prevPaymentResult = $db_conx->query("SELECT payment_id FROM payments WHERE txn_id = '".$txn_id."'");

    if($prevPaymentResult->num_rows > 0){
        $paymentRow = $prevPaymentResult->fetch_assoc();
        $last_insert_id = $paymentRow['payment_id'];
    }else{
        $insert = $db_conx->query("INSERT INTO payments(item_number,txn_id,payment_gross,currency_code,payment_status) VALUES
        ('".$item_number."','".$txn_id."','".$payment_amount."','".$payment_currency."','".$payment_status."')");
        $last_insert_id = $db_conx->insert_id;
        $update = $db_conx->query("UPDATE products SET status=1 where id=$item_number");
    }

    ?>
    <h1>Uplaceno.</h1>
    <h1>ID uplate - <?php echo $last_insert_id; ?></h1>
    <h1>Potvrdu o rezervaciji cete uskoro dobiti na email</h1>
    <a href="/">nazad</a>
<?php } else if($payment_status == "Pending"){
    $prevPaymentResult = $db_conx->query("SELECT payment_id FROM payments WHERE txn_id = '".$txn_id."'");

    if($prevPaymentResult->num_rows > 0){
        $paymentRow = $prevPaymentResult->fetch_assoc();
        $last_insert_id = $paymentRow['payment_id'];
    }else{
        $insert = $db_conx->query("INSERT INTO payments(item_number,txn_id,payment_gross,currency_code,payment_status) VALUES
        ('".$item_number."','".$txn_id."','".$payment_amount."','".$payment_currency."','".$payment_status."')");
        $last_insert_id = $db_conx->insert_id;
        $update = $db_conx->query("UPDATE products SET status=1 where id=$item_number");
    }
    ?>

    <h1>Uplaceno.</h1>
    <h1>ID uplate - <?php echo $last_insert_id; ?></h1>
    <h1>Potvrdu o rezervaciji cete uskoro dobiti na email</h1>
    <a href="/">nazad</a>
<?php } else{ ?>
    <h1>Uplata nije prošla..</h1>
    <a href="/">nazad</a>
<?php } ?>
```

Programski kod 3.15. *Skripta izvršenja transakcije*

4. ADMINISTRATORSKI DIO APLIKACIJE

Sve do sada opisano i prikazano u ovom radu je pogled iz perspektive korisnika, ono što svaki oglašivač kao administratorski dio mora imat je CRUD. CRUD je skraćenica za *create, read, undo, delete*. Administrator aplikacije može stvoriti novi oglas, promijeniti već postojeći ili obrisati oglas.

U ovom slučaju, administratorski dio je baziran na dodavanje novog, izmjenu i brisanje postojećeg vozila. Prilikom dodavanja vozila administrator mora upisati osnovne podatke tog vozila. Osnovni podaci su: naziv, opis, cijena, raspoloživost, ekonomičnost, konjske snage i slika odabranog vozila. JQuery ajax (*front-end*) POST metodom šalje podatke na PHP skriptu (*back-end*) koja potom sprema podatke u bazu podataka. Programski kod 4.1. prikazuje skriptu gdje Ajax POST metodom šalje podatke. Nadalje, programski kod 4.2. prikazuje skriptu koja pomoću prikupljenih podataka dodaje novo vozilo na internet aplikaciju.

```
$.ajax({
    url: "core/adminArea/dodajVozilo.php",
    type: "POST",
    data: {
        "vozilo_naziv": vozilo_naziv,
        "vozilo_opis": vozilo_opis,
        "vozilo_cijena": vozilo_cijena,
        "vozilo_raspolozivost": vozilo_raspolozivost,
        "vozilo_slika": vozilo_slika,
        "vozilo_snage" : vozilo_snage,
        "vozilo_ekonomicnost": vozilo_ekonomicnost
    },
    dataType: "html"
})
.done(function(data) {

})
.fail(function() {
    $('#dynamic-content').html('<i class="glyphicon glyphicon-info-sign"></i> Nešto je pošlo krivo.. pokušajte ponovno.. ');
})
.always(function() {
    $('#opcije-vozilo-dodaj').modal('hide');
    $('#opcije-vozilo-modal').modal('show');
})
```

Programski kod 4.1. *Ajax POST metodom šalje podatke*

```php
<?php

include "../connect.php";

if (isset($_POST["vozilo_naziv"]) && isset($_POST["vozilo_opis"]) &&
isset($_POST["vozilo_cijena"]) && isset($_POST["vozilo_raspolozivost"]) && isset($_POST["vozilo_slika"]) &&
isset($_POST["vozilo_snage"]) && isset($_POST["vozilo_ekonomicnost"]))
{
    $vozilo_naziv = $_POST["vozilo_naziv"];
    $vozilo_opis = $_POST["vozilo_opis"];
    $vozilo_cijena = $_POST["vozilo_cijena"];
    $vozilo_raspolozivost = $_POST["vozilo_raspolozivost"];
    $vozilo_slika = $_POST["vozilo_slika"];
    $vozilo_snage = $_POST["vozilo_slika"];
    $vozilo_ekonomicnost = $_POST["vozilo_slika"];

    $statement = $db_conx->prepare("INSERT INTO products (name, image, price, status, horsepower,
    fuel_economy, description, raspolozivost) VALUES (?, ?, ?, 0, ?, ?, ?, ?)");
    $statement->bind_param('ssiiisi', $vozilo_naziv, $vozilo_slika, $vozilo_cijena, $vozilo_snage,
                                      $vozilo_ekonomicnost, $vozilo_opis, $vozilo_raspolozivost);
    $statement->execute();
    $statement->close();
}

else {
    echo "Nešto je pošlo po zlu..";
}
```

Programski kod 4.2. *Skripta dodavanja vozila*

Slika 4.3. prikazuje dizajn forme za dodavanje novog vozila na aplikaciju od strane administratora. Nakon što administrator ispuni sva ponuđena polja, pritiskom na dugme „spremi“ na klijentskoj strani se dinamički, pomoću JavaScript skripte odnosno Ajax *POST* metode (postoji i *GET* metoda, koja se koristi za dinamično dohvaćanje podataka) šalje na PHP skriptu koja je odgovorna za uspješno pohranjivanje vozila odnosno oglasa u bazu podataka.

Slika 4.3. *Dodavanje novog vozila na aplikaciju*

5. ZAKLJUČAK

Nakon što je internet aplikacija testirana, imali smo priliku uvidjeti da sve radi onako kako je zamišljeno, stoga možemo zaključiti je ovaj projekt uspješno završen. Svakodnevnim napredovanjem tehnologije sve se više poslovnih događaja odvija upravo preko interneta. Internet aplikacija za iznajmljivanje vozila svakom korisniku omogućava na vrlo jednostavan način registraciju, a potom i korištenje ove aplikacije. Korisnici aplikacije imaju mogućnost pregleda, odabira, a zatim i plaćanja željenog vozila. Administratoru je omogućeno upravljanje cjelokupnom aplikacijom; potvrđivanje/odbijanje rezervacije, dodavanje novih vozila u ponudu, odobravanje transakcije.

Prilikom izrade internet aplikacije za iznajmljivanje vozila bilo je potrebno poznavati nekolicinu tehnologija, kao što su: PHP, CSS, HTML, JavaScript i MySQL baze podataka. Izradu ove aplikaciju možemo podijeliti na dva dijela: administratorski i korisnički. U administratorskom dijelu je omogućena registracija i prijava korisnika, validacija podataka, pristup bazi i plaćanje PayPal servisom. Korisnički dio služi za interakciju između korisnika i aplikacije, u ovom dijelu je uređen dizajn kako bi aplikacija bila organizirana i izgledala privlačno korisnicima. Ovaj projekt je završen uspješnim testiranjem te je tada utvrđeno da je svakom korisniku omogućeno u nekoliko jednostavnih koraka doći do željene usluge.

LITERATURA

[1] W3Schools, *What is HTML*, https://www.w3schools.com/html/, svibanj 2017.

[2] Wikipedia, *HTML*, https://hr.wikipedia.org/wiki/HTML, svibanj 2017.

[3] W3Schools, *HTML&CSS*, https://www.w3.org/standards/webdesign/htmlcss, lipanj 2017.

[4] Wikipedia, *Što je* CSS, https://hr.wikipedia.org/wiki/CSS, lipanj 2017.

[5] Lukić, Ivica; Köhler, Mirko. Osnove Internet programiranja, 2011. kolovoz 2017.

[6] W3Schools, *JavaScript*, https://www.w3schools.com/js/, srpanj 2017.

[7] R.W. Sebesta, Programming the World Wide Web (2nd Ed.), Addison-Wesley, Boston, MA, 2004. srpanj 2017

[8] Wikipedia, *PHP* https://hr.wikipedia.org/wiki/PHP, srpanj 2017.

[9] php.com.hr, *Uvod u PHP* http://php.com.hr/77, srpanj 2017.

[10] W3Schools, *SQL intro*, https://www.w3schools.com/sql/sql_intro.asp, lipanj 2017

[11] Wikipedia, SQL, https://hr.wikipedia.org/wiki/SQL , srpanj 2017

[12] Apache friends, *What is xampp*, https://www.apachefriends.org/index.html, srpanj 2017.

[13] Ngrok, *What is* Ngrok, https://ngrok.com/, kolovoz 2017.

SAŽETAK

U ovome završnom radu cilj je bio stvoriti internet aplikaciju za iznajmljivanje vozila. Aplikacija omogućava pregled ponuđenih vozila za iznajmljivanje, njihovih karakteristika, cijene te pruža mogućnost rezervacije i plaćanja. Teorijski dio rada opisuje tehnologije korištene u izradi aplikacije te detaljno opisuje slijed razvoja aplikacije. Praktični dio rada sastoji se od istraživanja dostupnih tehnologija koje su otvorenog koda i razvoja internet aplikacije. Prilikom izrade praktičnog dijela aplikacije korišteni su PHP i JavaScript skriptni jezici te HTML i CSS opisni jezici. Za izradu baze podataka korišten je MySql programski jezik.

Ključne riječi: internet aplikacija, iznajmljivanje vozila, skriptni jezik, opisni jezik, baza podataka.

ABSTRACT

Title: Internet application for vehicle rental

In this paper, the goal was to create an Internet application for vehicle rental. This application allows overview of the offered cars, their characteristics, price and also it provides the possibility of booking and payment. Theoretical part describes technologies used in the process of developing an application and it also describes the sequence of application development in detail. Practical part of the work consists of researching available technologies which are open source and application development. Scripting languages as JavaScript and PHP and markup languages as HTML and CSS were used in the process of an application development. The MySql programming language was used for database creation.

Keywords: internet application, vehicle rental, scripting language, markup language, database.

ŽIVOTOPIS

Filip Franjkić rođen je 15. svibnja 1995.godine u Osijeku. 2002.godine započinje svoje školovanje u Osnovnoj školi Tenja, u Tenji. Nakon završetka osnovne škole, 2010.godine upisuje srednju Elektrotehničku i prometnu školu Osijek, u Osijeku. Srednja školi završava 2014 godine te položenom državnom maturom, 2014.godine upisuje Elektrotehnički fakultet u Osijeku, stručni studij elektrotehnike, smjer informatika.

Filip Franjkić

_____